Chandrasekhar-based Maximum Correntropy Kalman Filtering with the Adaptive Kernel Size Selection

Maria V. Kulikova

Abstract-This technical note is aimed to derive the Chandrasekhartype recursion for the maximum correntropy criterion (MCC) Kalman filtering (KF). For the classical KF, the first Chandrasekhar difference equation was proposed at the beginning of 1970s. This is the alternative to the traditionally used Riccati recursion and it yields the so-called fast implementations known as the Morf-Sidhu-Kailath-Saved KF algorithms. They are proved to be computationally cheap because of propagating the matrices of a smaller size than $n \times n$ error covariance matrix in the Riccati recursion. The problem of deriving the Chandrasekhar-type recursion within the MCC estimation methodology has never been raised yet in engineering literature. In this technical note, we do the first step and derive the Chandrasekhar MCC-KF estimators for the case of adaptive kernel size selection strategy, which implies a constant scalar adjusting weight. Numerical examples substantiate a practical feasibility of the newly suggested MCC-KF implementations and correctness of the presented theoretical derivations.

Index Terms—Maximum correntropy, Kalman filtering, Chandrasekhar recursion, fast implementations.

I. INTRODUCTION

The maximum correntropy criterion (MCC) filtering/smoothing has become an important topic for an analysis in the past few years, both for linear [1]-[9] and nonlinear systems [10]-[15]. In engineering literature, the MCC Kalman-like estimators were proved to be robust with respect to outliers/impulsive noises and to outperform the classical Kalman filtering (KF) for estimation accuracy in case of non-Gaussian uncertainties in state-space models. The problem of designing the "distributionally robust" filtering/smoothing methods has a long history [16]. The related problem of constructing the statistically valid uncertainty bounds has been studied in [17]-[19]. Apart from the MCC-KF methodology examined in this paper, we mention a few other strategies for detecting the outliers. These are the Huberbased and M-estimator-based KF algorithms suggested in [20]-[22], the unknown input filtering (UIF) methodology proposed to model the unknown external excitations as unknown inputs and to derive the robust observer in [23], and many other estimation strategies. We may also note the most recent and comprehensive survey of the existed Kalman-like smoothing methods developed for the non-Gaussian state-space models in [24]. In this paper, we focus on the Kalman filtering under the MCC methodology.

Previous research on the MCC-KF implementation methods has produced the Riccati recursion-based algorithms, only. However, it is worth noting here that the classical KF allows for an alternative filter mechanization suggested at the beginning of 1970s in [25]. It implies the so-called Chandrasekhar recursion and yields the class of the fast KF implementations known as the Morf-Sidhu-Kailath-Sayed algorithms [26]–[28]. To the best of author's knowledge, the question about possibility to derive the Chandrasekhar-type recursion under the MCC filtering strategy has never been raised before. This is the challenge to be addressed in this technical note.

The essential starting point for further discussion is to review the Chandrasekhar-type algorithms existed for the classical KF in engineering literature. The first fast KF implementations were derived for linear time-invariant systems in [25]. The key idea behind the Chandrasekhar-based filtering is to propagate the difference between the error covariance matrices $\Delta_k = P_k - P_{k-1}$ instead of updating the matrix $P_k \in \mathbb{R}^{n imes n}$ through the classical Riccati recursion. It is motivated by the fact that the rank of matrix Δ_k is often less than the rank of P_k , especially for time-invariant state-space models, i.e. for the systems with constant (over time) matrices characterized the model. As a result, a low-rank approximation of the difference matrix Δ_k can be utilized and the obtained filleting algorithms update the matrices of a smaller size than the number of states n to be estimated. Thus, the Chandrasekhar-type algorithms are, in general, computationally cheaper than the Riccati-based KF implementations. To achieve our goal, the Riccati recursion required for propagating the covariance matrix P_k should be mathematically re-formulated in terms of the difference matrix Δ_k propagation. Having done this step, one receives the Chandrasekhar-type recursion. Recall, at the first time, this problem was solved for the case of constant continuoustime systems in [25]. The derived differential equations turned out to be similar to a certain equation introduced by the astrophysicist S. Chandrasekhar in 1948 for solving finite-interval Wiener-Hopf equation, and the obtained KF recursion was called of Chandrasekhar type. Almost at the same time, the problem was solved for constant discrete-time systems as well in [26]. However, the related solution for time-variant systems is rather complicated and it was derived twenty years later in [28]. Nowadays, there exists a wide variety of the Chandrasekhar-based KF implementations. For instance, we may mention the robust factored-form (square-root) filtering/smoothing methods in [27], [29] as well as the H_{∞} filtering algorithms in [30], [31]. Meanwhile, the Chandrasekhar-type recursion under the MCC filtering methodology has never been derived, so far. In other words, all existed MCC-KF implementation methods involve the Riccati-type recursion for propagating the error covariance matrix.

For the stated problem, a number of questions arise: (i) is it possible to derive Chandrasekhar-type recursion for the Kalman-like filtering under the examined MCC methodology? (ii) If the recursion exists, does it allow for designing the related low-rank implementations for the MCC-KF estimator? In this paper, we answer positively for both questions and suggest a solution for the case of the MCC-KF adaptive kernel size selection strategy with a constant scalar adjusting weight. The results of numerical experiments substantiate the theoretical derivations presented in this paper and prove a practical feasibility of the suggested MCC-KF Chandrasekhar-type implementations.

II. RICCATI RECURSION-BASED MAXIMUM CORRENTROPY KALMAN FILTERING

Consider a linear discrete-time stochastic system

$$x_{k+1} = Fx_k + Gw_k,\tag{1}$$

$$y_k = Hx_k + v_k, \quad k \ge 0 \tag{2}$$

Manuscript received ??; revised ??. The author acknowledges the financial support of the Portuguese FCT — *Fundação para a Ciência e a Tecnologia*, through the project UID/Multi/04621/2019 of CEMAT/IST-ID, Center for Computational and Stochastic Mathematics, Instituto Superior Técnico, University of Lisbon.

The author is with CEMAT (Center for Computational and Stochastic Mathematics), Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 LISBOA, Portugal; Email: maria.kulikova@ist.utl.pt

where the system matrices $F \in \mathbb{R}^{n \times n}$, $G \in \mathbb{R}^{n \times q}$ and $H \in \mathbb{R}^{m \times n}$ are known and constant over time. The vectors $x_k \in \mathbb{R}^n$ and $y_k \in \mathbb{R}^m$ are the unknown dynamic state and available measurements, respectively. The random variables x_0 , w_k and v_k satisfy

$$\mathbf{E} \{x_0\} = \bar{x}_0, \qquad \mathbf{E} \left\{ (x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^\top \right\} = \Pi_0, \\ \mathbf{E} \{w_k\} = \mathbf{E} \{v_k\} = 0, \qquad \mathbf{E} \left\{ w_k x_0^\top \right\} = \mathbf{E} \left\{ v_k x_0^\top \right\} = 0, \\ \mathbf{E} \left\{ w_k v_k^\top \right\} = 0, \qquad \mathbf{E} \left\{ w_k w_j^\top \right\} = Q \delta_{kj}, \mathbf{E} \left\{ v_k v_j^\top \right\} = R \delta_k$$

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

The classical KF produces the minimum *linear* expected mean square error (MSE) estimate $\hat{x}_{k|k}$ of the state vector x_k , given the measurements $\mathcal{Y}_0^k = \{y_0, \ldots, y_k\}$. The estimator can be formulated in the *a priori* form as proposed in [32, Theorem 9.2.1], i.e. when the one-step ahead predicted estimate $\hat{x}_{k+1|k}$ (*a priori* estimate) is propagated as follows:

$$\hat{x}_{k+1|k} = F\hat{x}_{k|k-1} + K_{p,k}e_k, \qquad e_k = y_k - H\hat{x}_{k|k-1}, \qquad (3)$$

$$K_{p,k} = FP_{k|k-1}H^{\top}R_{e,k}^{-1}, \quad R_{e,k} = R + HP_{k|k-1}H^{\top}$$
 (4)

where the innovations are defined as $e_k = y_k - H\hat{x}_{k|k-1}$ with the covariance matrix $R_{e,k} = \mathbf{E} \{e_k e_k^\top\}$, and $K_k = \mathbf{E} \{\hat{x}_{k+1|k} e_k^\top\} = FP_{k|k-1}H^T$. The matrix $P_{k|k-1}$ is the one-step ahead predicted error covariance $P_{k|k-1} = \mathbf{E} \{(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^\top\}$ propagated through the Riccati difference recursion:

$$P_{k+1|k} = FP_{k|k-1}F^{\top} + GQG^{\top} - K_{p,k}R_{e,k}K_{p,k}^{\top}$$
(5)

with the initial values $P_{0|-1} = \Pi_0 > 0$ and $\hat{x}_{0|-1} = \bar{x}_0$.

Being a *linear* estimator, the classical KF exhibits only sub-optimal behavior in non-Gaussian settings. To enhance its estimation quality and robustness with respect to outliers (impulsive noise), the KF *linear* expected MSE estimation criterion has been combined with the maximum correntropy approach in [5]–[7]. More precisely, the concept of *correntropy* represents a similarity measure of two random variables [1]. It can be used as an optimization cost in the related estimation problem as discussed in [33, Chapter 5]: an estimator of unknown state $X \in \mathbb{R}$ can be defined as a function of observations $Y \in \mathbb{R}^m$, i.e. $\hat{X} = g(Y)$ where g is solved by maximizing the correntropy between X and \hat{X} , which is defined as follows [34]:

$$g_{MCC} = \arg\max_{g \in G} V(X, \hat{X}) = \arg\max_{g \in G} \mathbf{E} \left\{ k_{\sigma} \left(X - g(Y) \right) \right\}$$
(6)

where G stands for the collection of all measurable functions of Y, $k_{\sigma}(\cdot)$ is a kernel function and $\sigma > 0$ is the kernel size (bandwidth). One of the most popular kernel function utilized in practice is the Gaussian kernel given as follows:

$$k_{\sigma}(X - \hat{X}) = \exp\left\{-(X - \hat{X})^2/(2\sigma^2)\right\}.$$
 (7)

It is not difficult to see that the MCC cost (6) with Gaussian kernel (7) reaches its maximum if and only if $X = \hat{X}$.

In [6], the MCC-KF is developed by solving the following estimation problem with the Gaussian kernel:

$$\hat{x}_{k|k} = \arg\max J(k),$$
 (8)

$$J(k) = k_{\sigma}(\|\ddot{x}_{k|k} - F\ddot{x}_{k-1|k-1}\|) + k_{\sigma}(\|y_{k} - H\ddot{x}_{k|k}\|).$$
(9)

Next, a fixed point rule (with one iterate, only) has been used for solving (with respect to $\hat{x}_{k|k}$) the resulted nonlinear equation arisen in the optimization problem above. It results to the following filtering recursion for the state [6, p. 503]:

$$\hat{x}_{k|k} = F\hat{x}_{k-1|k-1} + K_k^{\lambda}(y_k - H\hat{x}_{k|k-1}) \tag{10}$$

2

where the gain matrix is proved to be computed as follows [6]: $K_k^{\lambda} = \lambda_k \left(P_{k|k-1}^{-1} + \lambda_k H^{\top} R^{-1} H \right)^{-1} H^{\top} R^{-1}$ and the scalar adjusting weight λ_k is given by

$$\lambda_k = \frac{k_{\sigma}(\|y_k - H\hat{x}_{k|k-1}\|_{R^{-1}})}{k_{\sigma}(\|\hat{x}_{k|k-1} - F\hat{x}_{k-1|k-1}\|_{P_{k|k-1}^{-1}})}.$$
(11)

Finally, the recursion for the state estimate in (10) is utilized with a symmetric Joseph stabilized equation existed for the classical KF for the error covariance matrix $P_{k|k}$ calculation; see [35], [36]. The resulted computational method was called the MCC-KF estimator and it is summarized in [6, p. 503]. Further, the estimation quality of the original MCC-KF method has been boosted by deriving the mathematically equivalent formulas for the gain matrix and error covariance, which are similar to the classical KF equations presented in [35, pp. 128-129]. This approach yields the so-called improved MCC-KF (IMCC-KF) estimator suggested in [9]. It consists of the following steps.

TIME UPDATE (k = 1, ..., K). At this stage, the one-step ahead predicted (*a priori*) estimate, $\hat{x}_{k|k-1}$, is computed together with the corresponding error covariance matrix $P_{k|k-1}$ as follows:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1},\tag{12}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^{+} + GQG^{+}.$$
 (13)

MEASUREMENT UPDATE (k = 1, ..., K). The correction step is called the measurement update where the *a posteriori* estimate $\hat{x}_{k|k}$ is calculated together with the corresponding error covariance matrix $P_{k|k}$ as follows:

$$K_{k}^{\lambda} = \lambda_{k} P_{k|k-1} H^{\top} [R_{e,k}^{\lambda}]^{-1}, \ R_{e,k}^{\lambda} = \lambda_{k} H P_{k|k-1} H^{\top} + R, \ (14)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{\wedge} e_k, \qquad e_k = y_k - H \hat{x}_{k|k-1}, \qquad (15)$$

$$P_{k|k} = (I - K_k^{A}H)P_{k|k-1}$$
(16)

where the scaling (inflation) parameter λ_k is computed by (11).

The IMCC-KF estimator can be re-formulated in the *a priori* form similar to the classical KF equations (3) – (5). Having substituted (15) into (12), we get the recursion for the one-step ahead predicted (*a priori*) estimate $\hat{x}_{k+1|k}$ as follows:

$$\hat{x}_{k+1|k} = F\hat{x}_{k|k-1} + K_{p,k}^{\lambda}e_k, \tag{17}$$

$$K_{p,k}^{\lambda} = \lambda_k F P_{k|k-1} H^{\top} [R_{e,k}^{\lambda}]^{-1}$$
(18)

where $R_{e,k}^{\lambda}$ is defined by (14) and the *a priori* error covariance $P_{k+1|k}$ is computed via the following Riccati-type recursion:

$$P_{k+1|k} = FP_{k|k-1}F^{\top} + GQG^{\top} - \frac{1}{\lambda_k}K_{p,k}^{\lambda}R_{e,k}^{\lambda}[K_{p,k}^{\lambda}]^{\top}.$$
 (19)

Indeed, having substituted formula (16) into (13) and taking into account the symmetric form of any covariance matrix and the fact that $K_{p,k}^{\lambda} = F K_k^{\lambda}$, we prove (19) as follows:

$$\begin{split} P_{k+1|k} &= F(I - K_k^{\lambda} H) P_{k|k-1} F^{\top} + GQG^{\top} \\ &= FP_{k|k-1} F^{\top} + GQG^{\top} - FK_k^{\lambda} HP_{k|k-1} F^{\top} \\ &= FP_{k|k-1} F^{\top} + GQG^{\top} - K_{p,k}^{\lambda} (FP_{k|k-1} H^{\top})^{\top} \\ &= FP_{k|k-1} F^{\top} + GQG^{\top} - \frac{1}{\lambda_k} K_{p,k}^{\lambda} R_{e,k}^{\lambda} [K_{p,k}^{\lambda}]^{\top} \end{split}$$

Although λ_k is a scalar value, it is preferable to avoid the division. Thus, equations (17) – (19) can be written as follows:

$$\hat{x}_{k+1|k} = F\hat{x}_{k|k-1} + \lambda_k K_{p,k} e_k,$$
(20)

$$K_{p,k} = F P_{k|k-1} H^{\top} [R_{e,k}^{\lambda}]^{-1}$$
(21)

$$P_{k+1|k} = FP_{k|k-1}F^{\top} + GQG^{\top} - \lambda_k K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top}.$$
 (22)

III. CHANDRASEKHAR-BASED MAXIMUM CORRENTROPY KALMAN FILTERING

For a better presentation of a new material, we introduce the backward difference operator $\Delta[\cdot]$ that means $\Delta[A_k] = A_k A_{k-1}$ for any matrix A. The Riccati-based filtering implies the error covariance matrix $P_{k+1|k} \in \mathbb{R}^{n \times n}$ propagation through the classical KF recursion (5) as well as the IMCC-KF equation (22). The key idea of the Chandrasekhar recursion-based filtering is to propagate the difference $\Delta \left[P_{k+1|k} \right] = P_{k+1|k} - P_{k|k-1}$ instead of $P_{k+1|k}$. It is motivated by the fact that the rank of matrix $\Delta \left[P_{k+1|k} \right] \in \mathbb{R}^{n \times n}$ is often less than n, which is a number of states to be estimated by the filter, especially for time-invariant state-space models (i.e. with constant system matrices). Hence, the difference matrix $\Delta |P_{k+1|k}|$ implies a low-rank approximation and, thus, the related filtering algorithms can be derived for propagating the resulted lower rank factors. More precisely, the fast Morf-Sidhu-Kailath-Sayed algorithms derived for the classical KF are based on a factorization (non-uniquely defined) of symmetric indefinite matrix $\Delta [P_{1|0}]$ in the form $\Delta \left[P_{1|0} \right] = L_0 M_0 L_0^{\top}$ where $L_0 \in \mathbb{R}^{n \times \alpha}$, $M_0 \in \mathbb{R}^{\alpha \times \alpha}$, and α is called the displacement rank. Similar, for the IMCC-KF recursion in (22), we have

$$\begin{aligned} \alpha &= \operatorname{rank} \Delta \left[P_{1|0} \right] = \operatorname{rank}(P_{1|0} - P_{0|-1}) \\ &= \operatorname{rank}(F \Pi_0 F^\top + G Q G^\top - \lambda_0 K_{p,0} R_{e,0}^\lambda K_{p,0}^\top - \Pi_0). \end{aligned}$$

It is important to acknowledge that the mentioned factorization is performed only once, i.e. at the initial step of any Chandrasekharbased filtering method. When the resulted factors $L_0 \in \mathbb{R}^{n \times \alpha}$ and $M_0 \in \mathbb{R}^{\alpha \times \alpha}$ are defined, they are propagated instead of entire matrices $P_{k+1|k}$ and/or $\Delta [P_{k+1|k}]$. In general, the displacement rank $\alpha \leq 1$ n and, hence, the Chandrasekhar-type algorithms for propagating $M_k \in \mathbb{R}^{\alpha \times \alpha}$ and $L_k \in \mathbb{R}^{n \times \alpha}$ are computationally cheaper than the Riccati-based implementations that update a full rank error covariance matrix $P_{k+1|k}$ of size *n*. The computational complexity is shown to be reduced from $O(n^3)$ related to the Riccati recursion to $O(n^2\alpha)$ related to the Chandrasekhar recursion per iteration [26], [28]. The methods for implementing the underlying low-rank approximation of $\Delta \left[P_{1|0} \right]$ will be discussed at the end of this section. The first aim of this section is to derive the Chandrasekhar recursion for the IMCC-KF filtering. In other words, the corresponding IMCC-KF Riccati-type recursion in (22) should be re-formulated in terms of the matrices $\Delta [P_{k+1|k}]$, i.e. equation (22) and the newly derived Chandrasekharbased formulas should be mathematically equivalent. We prove the following theoretical result.

Lemma 1: The IMCC-KF Riccati-based filtering formulas (20) – (22) with a constant adjusting parameter λ are equivalent to the following Chandrasekhar-type recursions:

$$\Delta \left[P_{k+1|k} \right] = \left(F - \lambda K_{p,k} H \right) \left(\Delta \left[P_{k|k-1} \right] + \lambda \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k-1}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] \right) \times \left(F - \lambda K_{p,k} H \right)^{\top}, \qquad (23)$$
$$\Delta \left[P_{k+1|k} \right] = \left(F - \lambda K_{p,k-1} H \right) \left(\Delta \left[P_{k|k-1} \right] - \lambda \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] \right) \times \left(F - \lambda K_{p,k-1} H \right)^{\top}. \qquad (24)$$
Proof: First from expression (14) for R^{λ} , we derive

and, hence

$$R_{e,k-1}^{\lambda} = R_{e,k}^{\lambda} - \Delta \left[R_{e,k}^{\lambda} \right] = R_{e,k}^{\lambda} - \lambda H \Delta \left[P_{k|k-1} \right] H^{\top}.$$
 (25)

 $\Delta \left[R_{e,k}^{\lambda} \right] = \lambda H \Delta \left[P_{k|k-1} \right] H^{\top}$

Next, for the gain matrix $K_{p,k}$ in (21), we get

$$\Delta \left[K_{p,k} R_{e,k}^{\lambda} \right] = K_{p,k} R_{e,k}^{\lambda} - K_{p,k-1} R_{e,k-1}^{\lambda}$$
$$= F \Delta \left[P_{k|k-1} \right] H^{\top}$$
(26)

and, hence

$$K_{p,k} = \left[K_{p,k-1} R_{e,k-1}^{\lambda} + F \Delta \left[P_{k|k-1} \right] H^{\top} \right] \left[R_{e,k}^{\lambda} \right]^{-1}.$$
 (27)

From the IMCC-based Riccati-type recursion in (22), we obtain

$$\Delta \left[P_{k+1|k} \right] = P_{k+1|k} - P_{k|k-1} = F \Delta \left[P_{k|k-1} \right] F^{\top} - \lambda K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top} + \lambda K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top} = F \Delta \left[P_{k|k-1} \right] F^{\top} - \lambda \Delta \left[K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top} \right].$$
(28)

To derive the expression for the last term in equation (28), i.e. for $\Delta \left[K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top} \right]$, one multiplies formula (26) by $K_{p,k}^{\top}$ and $K_{p,k-1}^{\top}$ values, respectively. Thus, we have

$$\Delta \begin{bmatrix} K_{p,k} R_{e,k}^{\lambda} \end{bmatrix} K_{p,k}^{\top} = F \Delta \begin{bmatrix} P_{k|k-1} \end{bmatrix} H^{\top} K_{p,k}^{\top}$$
$$= K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top} - K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k}^{\top},$$
$$\Delta \begin{bmatrix} K_{p,k} R_{e,k}^{\lambda} \end{bmatrix} K_{p,k-1}^{\top} = F \Delta \begin{bmatrix} P_{k|k-1} \end{bmatrix} H^{\top} K_{p,k-1}^{\top}$$
$$= K_{p,k} R_{e,k}^{\lambda} K_{p,k-1}^{\top} - K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}$$

and, next, we summarize

$$\Delta \left[K_{p,k} R_{e,k}^{\lambda} \right] K_{p,k}^{\top} + \Delta \left[K_{p,k} R_{e,k}^{\lambda} \right] K_{p,k-1}^{\top}$$

= $\Delta \left[K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top} \right] + K_{p,k} R_{e,k}^{\lambda} K_{p,k-1}^{\top} - K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k}^{\top}$
= $F \Delta \left[P_{k|k-1} \right] H^{\top} K_{p,k}^{\top} + F \Delta \left[P_{k|k-1} \right] H^{\top} K_{p,k-1}^{\top}.$

Finally,

$$\Delta \begin{bmatrix} K_{p,k} R_{e,k}^{\lambda} K_{p,k}^{\top} \end{bmatrix} = F \Delta \begin{bmatrix} P_{k|k-1} \end{bmatrix} H^{\top} K_{p,k}^{\top} + F \Delta \begin{bmatrix} P_{k|k-1} \end{bmatrix} H^{\top} K_{p,k-1}^{\top} + K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k}^{\top} - K_{p,k} R_{e,k}^{\lambda} K_{p,k-1}^{\top}.$$
(29)

Having substituted (29) into (28), we obtain

$$\Delta \left[P_{k+1|k} \right] = F \Delta \left[P_{k|k-1} \right] F^{\top} - \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} K_{p,k}^{\top} - \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} K_{p,k-1}^{\top} - \lambda K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k}^{\top} + \lambda K_{p,k} R_{e,k}^{\lambda} K_{p,k-1}^{\top}.$$
(30)

The required formulas (23), (24) are both derived from equation (30) by substituting the related recursions for $K_{p,k}$ and $R_{e,k}$ and, then, by expanding the terms in the resulted equation. Indeed, we prove (24) by taking into account that $R_{e,k}$ is a symmetric matrix and by substituting (26), (27) into (30) as follows:

$$\Delta \left[P_{k+1|k} \right] = F \Delta \left[P_{k|k-1} \right] F'$$

$$- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}$$

$$- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] F^{\top}$$

$$- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} K_{p,k-1}^{\top} - \lambda K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}$$

$$+ \lambda \left[K_{p,k-1} R_{e,k-1}^{\lambda} + F \Delta \left[P_{k|k-1} \right] H^{\top} \right] K_{p,k-1}^{\top}$$

$$= F \Delta \left[P_{k|k-1} \right] F^{\top}$$

$$- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}$$

$$- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] F^{\top}$$

$$- \lambda K_{p,k-1} R_{e,k-1}^{\lambda} \left[R_{e,k}^{\lambda} \right]^{-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}$$

$$-\lambda K_{p,k-1} R_{e,k-1}^{\lambda} \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] F^{\top} +\lambda K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}.$$
(31)

Recall, the matrices $R_{e,k}$ and $R_{e,k-1}$ are symmetric as well as the matrix $\Delta \left[P_{k|k-1} \right]$. From equation (25) we have

$$\left[R_{e,k}^{\lambda}\right]^{-1}R_{e,k-1}^{\lambda} = \left(I - \lambda \left[R_{e,k}^{\lambda}\right]^{-1} H \Delta \left[P_{k|k-1}\right] H^{\top}\right). \quad (32)$$

Having substituted (32) into (31), we obtain

$$\begin{split} \Delta \left[P_{k+1|k} \right] &= F \Delta \left[P_{k|k-1} \right] F^{\top} \\ &- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} \left(I - \lambda \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] H^{\top} \right) K_{p,k-1}^{\top} \\ &- \lambda F \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] F^{\top} \\ &- \lambda K_{p,k-1} R_{e,k-1}^{\lambda} \left(I - \lambda \left[R_{e,k}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] H^{\top} \right) K_{p,k-1}^{\top} \\ &- \lambda K_{p,k-1} \left(I - \lambda H \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k}^{\lambda} \right]^{-1} \right) H \Delta \left[P_{k|k-1} \right] F^{\top} \\ &+ \lambda K_{p,k-1} R_{e,k-1}^{\lambda} K_{p,k-1}^{\top}. \end{split}$$

Finally, having substituted (32) one more time into equation above and, next, having collected the similar terms, we arrive at (24). The same approach is used for deriving (23). The difference with the derivation above is in the replacement of the term $K_{p,k-1}$ by $K_{p,k}$ via the corresponding recursion instead of avoiding the terms $K_{p,k}$ in equation (30) and replacing them by $K_{p,k-1}$.

Remark 1: As can be seen, Lemma 1 and, hence, the first Chandrasekhar MCC-KF-type recursions are proved for a constant scalar adjusting parameter λ . We stress that there exist some adaptive kernel size selection strategies for σ_k that yield a constant adjusting weight λ . A particular example of such adaptive selection rules can be found in [6] and the resulted estimator is successfully applied for solving practical application in [37], [38]. Thus, the new theoretical result in Lemma 1 has a practical interest, although it is restricted to a constant adjusting weight case. The derivation of the Chandrasekhar recursion for a general case of time-varying λ_k is complicated for the MCC-KF estimators. This is an open question for a future research.

Now, we are ready to propose the first Chandrasekhar-based IMCC-KF implementations. Let's consider the first recursion in Lemma 1, i.e. equation (23). Recall, the goal is to propagate the low rank factors $L_k \in \mathbb{R}^{n \times \alpha}$ and $M_k \in \mathbb{R}^{\alpha \times \alpha}$ of the difference matrix $\Delta \left[P_{k+1|k} \right] = P_{k+1|k} - P_{k|k-1} \text{ where } \Delta \left[P_{k+1|k} \right] = L_k M_k L_k^{\dagger}$ instead of the full matrix $P_{k+1|k} \in \mathbb{R}^{n \times n}$. Taking into account the required factorization, one may express equation (23) as follows:

$$\Delta \left[P_{k+1|k} \right] = L_k M_k L_k^{\top} = \left(F - \lambda K_{p,k} H \right) \left(\Delta \left[P_{k|k-1} \right] \right. \\ \left. + \lambda \Delta \left[P_{k|k-1} \right] H^{\top} \left[R_{e,k-1}^{\lambda} \right]^{-1} H \Delta \left[P_{k|k-1} \right] \right) \right] \\ \times \left(F - \lambda K_{p,k} H \right)^{\top} = \left(F - \lambda K_{p,k} H \right) \left(L_{k-1} M_{k-1} L_{k-1}^{\top} \right. \\ \left. + \lambda L_{k-1} M_{k-1} L_{k-1}^{\top} H^{\top} \left[R_{e,k-1}^{\lambda} \right]^{-1} H L_{k-1} M_{k-1} L_{k-1}^{\top} \right) \right] \\ \times \left(F - \lambda K_{p,k} H \right)^{\top} = \left(F - \lambda K_{p,k} H \right) L_{k-1} \\ \times \left(M_{k-1} + \lambda M_{k-1} L_{k-1}^{\top} H^{\top} \left[R_{e,k-1}^{\lambda} \right]^{-1} H L_{k-1} M_{k-1} \right) \\ \times L_{k-1}^{\top} \left(F - \lambda K_{p,k} H \right)^{\top}.$$

Having compared both sides of the resulted equality, we conclude

$$L_k := \left(F - \lambda K_{p,k}H\right)L_{k-1},\tag{33}$$

$$M_k := M_{k-1} + \lambda M_{k-1} L_{k-1}^{\top} H^{\top} [R_{e,k-1}^{\lambda}]^{-1} H L_{k-1} M_{k-1}.$$
 (34)

The last step is to express $K_{p,k}$ and $R_{e,k}$ in terms of the factors L_k and M_k . From formula (25), we have

$$R_{e,k}^{\lambda} = R_{e,k-1}^{\lambda} + \lambda H \Delta \left[P_{k|k-1} \right] H^{\top}$$
$$= R_{e,k-1}^{\lambda} + \lambda H L_{k-1} M_{k-1} L_{k-1}^{\top} H^{\top}.$$
(35)

At the same way, from equation (27) we obtain

$$K_{p,k} = \left[K_{p,k-1} R_{e,k-1}^{\lambda} + F \Delta \left[P_{k|k-1} \right] H^{\top} \right] \left[R_{e,k}^{\lambda} \right]^{-1} \\ \left[K_{p,k-1} R_{e,k-1}^{\lambda} + F L_{k-1} M_{k-1} L_{k-1}^{\top} H^{\top} \right] \left[R_{e,k}^{\lambda} \right]^{-1}$$
(36)

Having collected equations (33) - (36) and formula (20) used for computing the state estimate, the first Chandrasekhar recursion-based IMCC-KF implementation is designed. For readers' convenience, it is summarized in the form of pseudo-code in Algorithm 1.

Algorithm 1. CHANDRASEKHAR IMCC-KF based on recursion (23) INITIALIZATION: (k = 0)

Set $x_{0|-1} = \bar{x}_0$, $P_{0|-1} = \Pi_0$; Compute $R_{e,0}^{\lambda} = R + \lambda H \Pi_0 H^{\top}, K_{p,0} = F \Pi_0 H^{\top} [R_{e,0}^{\lambda}]^{-1};$ 2 Find $\Delta \left[P_{1|0} \right] = F \Pi_0 F^\top + G Q G^\top - \lambda K_{p,0} R_{e,0}^\lambda K_{p,0}^\top - \Pi_0;$ Factorize $\Delta \left[P_{1|0} \right] = L_0 M_0 L_0^{\top}, L_0 \in \mathbb{R}^{n \times \alpha}, M_0 \in \mathbb{R}^{\alpha \times \alpha};$ FILTER RECURSION: $(k = \overline{0, N})$ $R_{e,k+1}^{\lambda} = R_{e,k} + \lambda H L_k M_k L_k^{\top} H^{\top};$ 6 $K_{p,k+1} = \left[K_{p,k} R_{e,k}^{\lambda} + F L_k M_k L_k^{\top} H^{\top} \right] \left[R_{e,k+1}^{\lambda} \right]^{-1};$ $L_{k+1} = (F - \lambda K_{p,k+1}H)L_k;$ $M_{k+1} = M_k + \lambda M_k L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1} H L_k M_k;$ $\hat{x}_{k+1|k} = F \hat{x}_{k|k-1} + \lambda K_{p,k} (y_k - H \hat{x}_{k|k-1}).$ 8 9

It is worth noting here that the error covariance matrix $P_{k+1|k}$ is simply recovered from the propagated factors $L_k \in \mathbb{R}^{n \times \alpha}$ and $M_k \in \mathbb{R}^{\alpha \times \alpha}$ of the matrix $\Delta \left[P_{k+1|k} \right] = P_{k+1|k} - P_{k|k-1}$ at any time instance, if necessary:

$$P_{k+1|k} = P_{k|k-1} + L_k M_k L_k^{\top} = \Pi_0 + \sum_{j=0}^k L_j M_j L_j^{\top}.$$

Alternatively, the Chandrasekhar recursion in (24) might be utilized for designing the related IMCC-KF implementation as follows:

$$\Delta \left[P_{k+1|k} \right] = L_k M_k L_k^\top = \left(F - \lambda K_{p,k-1} H \right) L_{k-1}$$
$$\times \left(M_{k-1} - \lambda M_{k-1} L_{k-1}^\top H^\top [R_{e,k}^\lambda]^{-1} H L_{k-1} M_{k-1} \right)$$
$$\times L_{k-1}^\top \left(F - \lambda K_{p,k-1} H \right)^\top,$$

i.e. we conclude

1

5

6

7

8 9

1

3

4

5

7

$$L_k := \left(F - \lambda K_{p,k-1}H\right)L_{k-1},\tag{37}$$

$$M_k := M_{k-1} - \lambda M_{k-1} L_{k-1}^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1} H L_{k-1} M_{k-1}.$$
(38)

Having summarized equations (35) - (38) with (20), a new Chandrasekhar-based IMCC-KF implementation is designed.

Algorithm 2. CHANDRASEKHAR IMCC-KF based on recursion (24) INITIALIZATION: (k = 0)

- Set $x_{0|-1} = \bar{x}_0$, $P_{0|-1} = \Pi_0$;
- 2
- 3
- Compute $R_{e,0}^{\lambda} = R + \lambda H \Pi_0 H^{\top}, K_{p,0} = F \Pi_0 H^{\top} [R_{e,0}^{\lambda}]^{-1};$ Find $\Delta [P_{1|0}] = F \Pi_0 F^{\top} + G Q G^{\top} \lambda K_{p,0} R_{e,0}^{\lambda} K_{p,0}^{\top} \Pi_0;$ Factorize $\Delta [P_{1|0}] = L_0 M_0 L_0^{\top}, L_0 \in \mathbb{R}^{n \times \alpha}, M_0 \in \mathbb{R}^{\alpha \times \alpha};$ 4 FILTER RECURSION: $(k = \overline{0, N})$
 - $R_{e,k+1}^{\lambda} = R_{e,k} + \lambda H L_k M_k L_k^{\top} H^{\top};$ $K_{p,k+1} = \left[K_{p,k} R_{e,k}^{\lambda} + F L_k M_k L_k^{\top} H^{\top} \right] \left[R_{e,k+1}^{\lambda} \right]^{-1};$ $L_{k+1} = (F - \lambda K_{p,k}H)L_k;$
 - $M_{k+1} = M_k \lambda M_k L_k^{\dagger} H^{\dagger} [R_{e,k+1}^{\lambda}]^{-1} H L_k M_k;$
 - $\hat{x}_{k+1|k} = F\hat{x}_{k|k-1} + \lambda K_{p,k}(y_k H\hat{x}_{k|k-1}).$

Having compared equations (33) and (37) for computing L_k factor, we observe a difference between Algorithms 1 and 2. In fact, equation (33) of Algorithm 1 implies calculation of L_k through $K_{p,k}$ value obtained at the same filtering step t_k . Meanwhile equation (37) of Algorithm 2 computes L_k by using $K_{p,k-1}$, i.e. by using the value from the previous filtering step t_{k-1} . Besides, a similar difference in calculating M_k factor between Algorithms 1 and 2 is observed from equations (34) and (38). Indeed, Algorithm 1 implies calculation of M_k through the previous step value $R_{e,k-1}$, meanwhile equation (37) of Algorithm 2 finds M_k by using the current value $R_{e,k}$. Finally, we note that each filtering step of Algorithm 1 require two $m \times m$ matrix inversions, i.e. the terms $[R_{e,k}^{\lambda}]^{-1}$ and $[R_{e,k+1}^{\lambda}]^{-1}$. In contrast, Algorithm 2 involves only $[R_{e,k+1}^{\lambda}]^{-1}$. However, a careful implementation of Algorithm 1 suggests to save the previously obtained $[R_{e,k}^{\lambda}]^{-1}$ value and, hence, to reduce the number of matrix inversions up to one operation. In this case, the computational complexity of Algorithms 1 and 2 are the same, i.e. the same number of flops are required. Indeed, Algorithms 1 and 2 differ by the way of calculating L_k and M_k factors, only. Besides, they are computed at a similar ways. Hence, a careful implementation of Algorithm 1 yields the same number of flops.

Next, we consider Algorithm 2 and derive a few more implementations based on recursion (24). The same can be done with respect to recursion (23). First, we suggest a variant of Algorithm 2 that requires the inversion of α -by- α matrix instead of $R_{e,k+1}$ of size m involved in Algorithm 2. Such implementation might be preferable for practical use, especially when $\alpha \ll m$, where m is the size of measurement vector y_k in the state-space model. Having applied the matrix inversion lemma (Sherman-Morrison-Woodbury formula) to the recursion in (35), we get

$$[R_{e,k+1}^{\lambda}]^{-1} = [R_{e,k}^{\lambda} + \lambda H L_k M_k L_k^{\top} H^{\top}]^{-1} = [R_{e,k}^{\lambda}]^{-1} - \lambda [R_{e,k}^{\lambda}]^{-1} H L_k \left(M_k^{-1} + \lambda L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1} H L_k \right)^{-1} \times L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1}.$$

Furthermore, taking into account that

$$\begin{pmatrix} M_k^{-1} + \lambda L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1} H L_k \end{pmatrix}^{-1}$$

= $M_k - \lambda M_k L_k^{\top} H^{\top} (R_{e,k}^{\lambda} + \lambda H L_k M_k L_k^{\top} H^{\top})^{-1} H L_k M_k$
= $M_k - \lambda M_k L_k^{\top} H^{\top} (R_{e,k+1}^{\lambda})^{-1} H L_k M_k = M_{k+1},$

we conclude that if M_k is updated through (38), then

$$[R_{e,k+1}^{\lambda}]^{-1} = [R_{e,k}^{\lambda}]^{-1} - \lambda [R_{e,k}^{\lambda}]^{-1} H L_k M_{k+1} L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1}.$$

Having summarized the formulas above, we formulate Algorithm 3, which is mathematically equivalent to Algorithm 2 and through Lemma 1 it is equivalent to Algorithm 1 as well.

Algorithm 3. CHANDRASEKHAR IMCC-KF based on recursion (24) INITIALIZATION: (k = 0)

1

- 2
- 3
- Set $x_{0|-1} = \bar{x}_0$, $P_{0|-1} = \Pi_0$; Compute $R_{e,0}^{\lambda} = R + \lambda H \Pi_0 H^{\top}$, $K_0 = F \Pi_0 H^{\top}$; Find $\Delta [P_{1|0}] = F \Pi_0 F^{\top} + GQG^{\top} \lambda K_0 [R_{e,0}^{\lambda}]^{-1} K_0^{\top} \Pi_0$; Factorize $\Delta [P_{1|0}] = L_0 M_0 L_0^{\top}$, $L_0 \in \mathbb{R}^{n \times \alpha}$, $M_0 \in \mathbb{R}^{\alpha \times \alpha}$; FILTER RECURSION: $(k = \overline{0, N})$ 4
- $L_{k+1} = \left(F \lambda K_k [R_{e,k}^{\lambda}]^{-1} H\right) L_k;$ 5

$$\begin{array}{ccc} 6 & M_{k+1}^{-1} = M_{k}^{-1} + \lambda L_{k}^{\top} H^{\top} [R_{e,k}^{\lambda'}]^{-1} H L_{k}; \\ & T_{k+1}^{-1} = L_{k}^{-1} [R_{k}^{\lambda'}]^{-1} H L_{k}; \end{array}$$

7
$$[R_{e,k+1}^{\lambda}]^{-1} = [R_{e,k}^{\lambda}]^{-1} - \lambda [R_{e,k}^{\lambda}]^{-1} H L_k M_{k+1} L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1};$$

8
$$K_{k+1} = K_k + FL_k M_k L_k^\top H^\top;$$

9
$$\hat{x}_{k+1|k} = F\hat{x}_{k|k-1} + \lambda K_k [R_{e,k}^{\lambda}]^{-1} (y_k - H\hat{x}_{k|k-1}).$$

Having analyzed the implementation above, we conclude that Algorithm 3 propagates the inverse matrices M_k^{-1} and $[R_{e,k}^{\lambda}]^{-1}$. However, the value M_k is required in line 7 of Algorithm 3, i.e. it still demands the matrix inversion operation, but the matrix to be inverted is of size α . In other words, if $\alpha \ll m$, then Algorithm 3 is preferable for practical use from the computational complexity point of view. Meanwhile, the numerical stability issues of Algorithms 1 - 3 should be also taken into account when one decides between Algorithms 1, 2 and 3. The numerical behaviour depends on properties of the matrices to be inverted, i.e. on the condition numbers of $R_{e,k}$ in Algorithms 1, 2 and M_k in Algorithm 3.

Finally, we may suggest a symmetric implementation based on Chandrasekhar recursion (24) by re-formulating Algorithm 3.

Algorithm 4. CHANDRASEKHAR IMCC-KF based on recursion (24) INITIALIZATION: (k = 0)

- Set $x_{0|-1} = \bar{x}_0$, $P_{0|-1} = \Pi_0$;
- 1 2
- 3
- Compute $R_{e,0}^{\lambda} = R + \lambda H \Pi_0 H^{\top}, K_{p,0} = F \Pi_0 H^{\top} [R_{e,0}^{\lambda}]^{-1};$ Find $\Delta [P_{1|0}] = F \Pi_0 F^{\top} + G Q G^{\top} \lambda K_{p,0} R_{e,0}^{\lambda} K_{p,0}^{\top} \Pi_0;$ Factorize $\Delta [P_{1|0}] = L_0 M_0 L_0^{\top}, L_0 \in \mathbb{R}^{n \times \alpha}, M_0 \in \mathbb{R}^{\alpha \times \alpha};$ 4
- FILTER RECURSION: $(k = \overline{0, N})$
- $$\begin{split} R_{e,k+1}^{\lambda} &= R_{e,k} + \lambda H L_k M_k L_k^{\top} H^{\top}; \\ M_{k+1}^{-1} &= M_k^{-1} + \lambda L_k^{\top} H^{\top} [R_{e,k}^{\lambda}]^{-1} H L_k; \end{split}$$
 5 6 $K_{p,k+1} = \left[K_{p,k} R_{e,k}^{\lambda} + F L_k M_k L_k^{\top} H^{\top} \right] \left[R_{e,k+1}^{\lambda} \right]^{-1};$ 7

$$L_{k+1} = (F - \lambda K_{p,k}H)L_k;$$

8

9
$$\hat{x}_{k+1|k} = F\hat{x}_{k|k-1} + \lambda K_{p,k}(y_k - H\hat{x}_{k|k-1}).$$

From the numerical stability and computational complexity reasons, Algorithm 4 seems to be the worst implementation, although the formulas in lines 5, 6 have a symmetric form. Indeed, Algorithm 4 involves both the $m \times m$ and $\alpha \times \alpha$ matrix inversions in each iterate.

Finally, the required low-rank approximation for $\Delta [P_{1|0}]$ and the displacement rank α are discussed. In contrast to a symmetric positive (semi-) definite error covariance matrix $P_{k|k-1}$ involved in the Riccati recursion, the matrix $\Delta \left[P_{k|k-1} \right]$ is a symmetric indefinite matrix. Following [32, Chapter 13], the required factorization $\Delta \left[P_{1|0} \right] = L_0 M_0 L_0^{\top}, L_0 \in \mathbb{R}^{n \times \alpha}, M_0 \in \mathbb{R}^{\alpha \times \alpha}$ can be performed in various ways; e.g. by using Bunch-Kaufman algorithm [39], [40]. More precisely, the accurate Bunch-Kaufman method from [41] with corresponding MATLAB routine ldl is utilized for implementing Algorithms 1–4. Given a symmetric indefinite matrix $A \in \mathbb{R}^{n \times n}$, it performs factorization $PAP^{\top} = LDL^{\top}$ with a permutation $P \in \mathbb{R}^{n \times n}$, a unit lower triangular $L \in \mathbb{R}^{n \times n}$, and a real block diagonal $D \in \mathbb{R}^{n \times n}$. Thus, the displacement rank α equals the rank of the resulted matrix D. We stress that it is defined automatically and it heavily depends on initial value P_0 and the problem statement. For instance, if $\Pi_0 = 0$, then $K_{p,0} = 0$, $R_{e,0}^{\lambda} = R$ and $\Delta [P_{1|0}] = GQG^{\top}$, i.e. one may set $L_0 := G \in \mathbb{R}^{n \times q}$ and $M_0 := Q \in \mathbb{R}^{q \times q}$, i.e. $\alpha := q$. In practice, the process covariance matrix is often a diagonal matrix with a few non-zero diagonal elements (that is q) and, hence, in this case $\alpha \ll n$. In general, the displacement rank satisfies $\alpha \leq n$. For any initial value Π_0 , the value of α is defined and, next, the resulted block diagonal matrix D of size n is approximated by its part of size α with corresponding non-zero (block) diagonal elements, i.e. we set $M_0 := [D]_{\alpha} \in \mathbb{R}^{\alpha \times \alpha}$. Finally, the product $P^{+}L$ is computed and, next, the columns that correspond to nonzero elements in D are pulled out to get an approximation $L_0 :=$ $[P^{\top}L] \in \mathbb{R}^{n \times \alpha}$. It should be stressed that the required factorization is performed only once, i.e. at the initial filtering step and, next, all fast Morf-Sidhu-Kailath-Sayed implementations propagate the resulted factors $L_k \in \mathbb{R}^{n \times \alpha}$ and $M_k \in \mathbb{R}^{\alpha \times \alpha}$ according to the underlying Chandrasekhar recursions.

		$n = 4, \Pi_0 = diag\{[1, 1, 1, 10^{-2}]\}, \alpha = 4$				$n = 4, \Pi_0 = 0, \alpha = 1$							
Ratio	KF and IMCC-KF	Root mean square errors $RMSE_{x_i}$				Root mean square errors $RMSE_{x_i}$					CPU time		
q_4/R	implementations	x_1	x_2	x_3	x_4	$\ \cdot\ _2$	x_1	x_2	x_3	x_4	$\ \cdot \ _{2}$	(s.)	(%)
$0.63 \cdot 10^{-2}$	Classical Riccati KF	80.95	1.77	0.42	0.92	80.97	78.37	2.07	0.00	0.91	78.40	0.0159	NA
	IMCC-KF Riccati (20)-(22)	80.90	1.95	0.42	0.93	80.93	77.69	2.34	0.00	0.91	77.73	0.0212	-
	IMCC-KF Algorithm 1	80.90	1.95	0.42	0.93	80.93	77.69	2.34	0.00	0.91	77.73	0.0203	4.4
	IMCC-KF Algorithm 2	80.90	1.95	0.42	0.93	80.93	77.69	2.34	0.00	0.91	77.73	0.0202	5.2
	IMCC-KF Algorithm 3	80.90	1.95	0.42	0.93	80.93	77.69	2.34	0.00	0.91	77.73	0.0205	3.7
	IMCC-KF Algorithm 4	80.90	1.95	0.42	0.93	80.93	77.69	2.34	0.00	0.91	77.73	0.0206	3.2
$0.63 \cdot 10^{-4}$	Classical Riccati KF	81.77	4.26	0.44	0.93	81.89	60.75	5.93	0.00	0.92	61.05	0.0167	NA
	IMCC-KF Riccati (20)-(22)	81.32	3.95	0.44	0.93	81.42	56.54	6.61	0.00	0.92	56.93	0.0218	-
	IMCC-KF Algorithm 1	81.32	3.95	0.44	0.93	81.42	56.54	6.61	0.00	0.92	56.93	0.0210	3.6
	IMCC-KF Algorithm 2	81.32	3.95	0.44	0.93	81.42	56.54	6.61	0.00	0.92	56.93	0.0206	6.0
	IMCC-KF Algorithm 3	81.32	3.95	0.44	0.93	81.42	56.54	6.61	0.00	0.92	56.93	0.0211	3.5
	IMCC KE Algorithm 4	81.22	2.05	0.44	0.03	81 42	56 54	6.61	0.00	0.02	56.02	0.0214	1.0

 TABLE I

 The RMSE, the average CPU time (s) and runtime benefit of the Chandrasekhar methods over the Riccati filtering in Example 1.

IV. NUMERICAL EXPERIMENTS

The goal is to justify the theoretical derivation of the newly suggested Chandrasekhar-based MCC-KF estimators in Section III. *Example 1:* The dynamic of the in-track motion of a satellite

traveling in a circular orbit is given as follows [42, p. 1448]:

	[1	1	0.5	0.5		0	0	0	0]	
	0	1	1	1		0	0	0	0	,
$x_{k+1} \equiv$	0	0	1	0	$x_k + w_k, Q =$	0	0	0	0	
	0	0	0	0.606		0	0	0	q_4	
$y_k =$	1	0	0 0	$\left[x_{k} + v \right]$	$v_k, R=1,$					

with zero-mean initial state and $\Pi_0 = diag\{[1, 1, 1, 10^{-2}]\}$. In [42], two cases are examined: (i) $q_4 = 0.63 \cdot 10^{-2}$, and (ii) $q_4 = 0.63 \cdot 10^{-4}$.

The original MCC-KF estimator has been derived for dealing with the impulsive noise case in [6]. Here, we follow the same experimental conditions as in the cited paper, i.e.

$$w_k \sim \mathcal{N}(0, Q) + \text{Shot noise},$$

 $v_k \sim \mathcal{N}(0, R) + \text{Shot noise}$

where the short noise is generated as follows: (i) only 10% of samples are corrupted by the outliers; (ii) the discrete time instants t_k corrupted are selected randomly from the uniform discrete distribution in [21, N - 1] where the system is simulated for N = 300 discrete time points; (iii) the magnitude of each impulse is chosen randomly from the uniform discrete distribution in the interval [0, 3].

When the stochastic model is simulated for N = 300, the inverse problem (i.e. the state estimation from the observed signal) is solved by various filtering methods under examination. The root mean square error (RMSE) is calculated over M = 500 Monte Carlo runs to justify the estimation accuracy. Additionally, the average CPU time (s) is collected for each implementation in Table I. The runtime benefit is also computed for Algorithms 1-4 compared to the Riccatibased counterpart as follows: $(CPU_{Chandrasekhar}/CPU_{Riccati}-1)$ and it is expressed in percent. It is worth noting here that the CPU time benefit is remarkable when $\alpha \ll n$. It is not difficult to see that if $\Pi_0 = 0$ in Example 1, then $\Delta [P_{1|0}] = GQG^{\top}$ where $G = I_4$ and, hence, $\alpha = 1$ while n = 4. For this case we present the CPU time and the computed computational benefit in Table I. In general case of Π_0 utilized in Example 1, we do not know in advance the exact value of α , because it is defined automatically from the related factorization. However, for readers' convenience, we force Algorithms 1-4 to return this value and summarize the results for various Π_0 together with the outcomes of the numerical experiments in Table I.

Having analyzed the results summarized in Table I, we make a few conclusions. First, we observe that the MCC-KF implementations outperform the classical KF for estimation quality in case of impulsive noise examined in this paper. This result was anticipated, because the MCC KF-like estimators are shown to be more robust with respect to outliers than the classical KF methodology in many recent papers. In our experiments, the difference in estimation accuracies between the classical KF and the MCC KF-like methods is modest, because we use the adaptive kernel size selection approach suggested in [6] that yields the constant adjusting parameter λ . This allows for testing the suggested Chandrasekhar implementations. For other adaptive kernel size selection strategies the difference in estimation quality between the classical KF and the MCC KF-like estimators might be more impressive. However, this paper is rather focused on the existence of the Chandrasekhar-type recursion under the MCC methodology.

Having compared the Riccati- and Chandrasekhar-based MCC KF-like implementations, we conclude that they produce the same estimates of the state vector, i.e. their resulted accuracies are the same for any initial value Π_0 and various signal-to-noise-ratio values under examination. This substantiates the correctness of the theoretical derivation presented in Section III. In other words, the mathematical equivalence between the classical Riccati- and the newly suggested Chandrasekhar-based filtering under the MCC-KF methodology is validated in practice.

Finally, having compared the CPU time averaged over Monte Carlo runs, we conclude that the Chandrasekhar-based implementations are faster than their algebraically equivalent Riccati-based counterparts when $\alpha < n$. This is in line with the previous research focused on the Chandrasekhar-based KF implementations in [26], [28]. More precisely, from the last column in Table I we observe that the Chandrasekhar-based MCC-KF implementations work approximately on 5% faster than the Riccati-based MCC-KF algorithm in case of low-dimensional problem in Example 1, i.e. when $\alpha = 1$ and n = 4. Thus, we conclude that the newly-developed Chandrasekhar-based MCC-KF implementation quality to that of the original Riccati-based IMCC-KF algorithm, but at a reduced CPU time. This runtime difference is expected to be significant in case of estimating the large-scale dynamical systems.

V. CONCLUDING REMARKS

In this technical note, the first Chandrasekhar recursion is derived for the maximum correntropy Kalman filtering. The theory is inferred for a case of the adaptive kernel size selection mechanism that yields a constant adjusting parameter. Several Chandrasekhar-based MCC KF-like implementations have been proposed, and the numerical experiments substantiate their practical feasibility and efficiency. The derivation of the Chandrasekhar-type recursion for a general case of a time-variant adjusting weight in the MCC-KF estimators is rather complicated. This is an area for a future research. Another open question is a design of the Kalman-like nonlinear filtering methods under the MCC methodology; e.g. the MCC-EKF is planned for a future work based on the results in [43]–[45].

REFERENCES

- W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: properties and applications in non-Gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.
- [2] B. Chen, L. Xing, J. Liang, N. Zheng, and J. C. Príncipe, "Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 880–884, 2014.
- [3] B. Chen, J. Wang, H. Zhao, N. Zheng, and J. C. Príncipe, "Convergence of a fixed-point algorithm under maximum correntropy criterion," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1723–1727, 2015.
- [4] G. T. Cinar and J. C. Príncipe, "Adaptive background estimation using an information theoretic cost for hidden state estimation," in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, July 2011, pp. 489–494.
- [5] —, "Hidden state estimation using the Correntropy filter with fixed point update and adaptive kernel size," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, June 2012, pp. 1–6.
- [6] R. Izanloo, S. A. Fakoorian, H. S. Yazdi, and D. Simon, "Kalman filtering based on the maximum correntropy criterion in the presence of non-Gaussian noise," in 2016 Annual Conference on Information Science and Systems (CISS), 2016, pp. 500–505.
- [7] B. Chen, X. Liu, H. Zhao, and J. C. Príncipe, "Maximum Correntropy Kalman Filter," *Automatica*, vol. 76, pp. 70–77, 2017.
- [8] X. Liu, H. Qu, J. Zhao, and B. Chen, "State space maximum correntropy filter," Signal Processing, vol. 130, pp. 152–158, 2017.
- [9] M. V. Kulikova, "Square-root algorithms for maximum correntropy estimation of linear discrete-time systems in presence of non-Gaussian noise," *Systems & Control Letters*, vol. 108, pp. 8–15, 2017.
- [10] X. Liu, H. Qu, J. Zhao, and B. Chen, "Extended Kalman filter under maximum correntropy criterion," in *International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1733–1737.
- [11] G. Yu. Kulikov and M. V. Kulikova, "Estimation of maneuvering target in the presence of non-Gaussian noise: A coordinated turn case study," *Signal Processing*, vol. 145, pp. 241–257, 2018.
- [12] X. Liu, B. Chen, B. Xu, Z. Wu, and P. Honeine, "Maximum correntropy unscented filter," *International Journal of Systems Science*, vol. 48, no. 8, pp. 1607–1615, 2017.
- [13] W. Qin, X. Wang, and N. Cui, "Maximum correntropy sparse Gauss-Hermite quadrature filter and its application in tracking ballistic missile," *IET Radar, Sonar & Navigation*, vol. 11, no. 9, pp. 1388–1396, 2017.
- [14] G. Wang, N. Li, and Y. Zhang, "Maximum correntropy unscented Kalman and information filters for non-Gaussian measurement noise," *Journal of the Franklin Institute*, vol. 354, no. 18, pp. 8659–8677, 2017.
- [15] Y. Wang, W. Zheng, S. Sun, and L. Li, "Robust information filter based on maximum correntropy criterion," *Journal of Guidance, Control, and Dynamics*, pp. 1126–1131, 2016.
- [16] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.
- [17] J. C. Spall and K. D. Wall, "Asymptotic distribution theory for the Kalman filter state estimator," *Communications in Statistics-Theory and Methods*, vol. 13, no. 16, pp. 1981–2003, 1984.
- [18] J. C. Spall, "The Kantorovich inequality for error analysis of the Kalman filter with unknown noise distributions," *Automatica*, vol. 31, no. 10, pp. 1513–1517, 1995.
- [19] J. L. Maryak, J. C. Spall, and B. D. Heydon, "Use of the Kalman filter for inference in state-space models with unknown noise distributions," *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 87–90, 2004.
- [20] C. Masreliez and R. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, pp. 361–371, 1977.
- [21] C. Hajiyev and H. E. Soken, "Robust estimation of UAV dynamics in the presence of measurement faults," *Journal of Aerospace Engineering*, vol. 25, no. 1, pp. 80–89, 2010.

- [22] L. Chang, B. Hu, G. Chang, and A. Li, "Robust derivative-free Kalman filter based on Huber's M-estimation methodology," *Journal of Process Control*, vol. 23, no. 10, pp. 1555–1561, 2013.
- [23] B. A. Charandabi and H. J. Marquez, "A novel approach to unknown input filter design for discrete-time linear systems," *Automatica*, vol. 50, no. 11, pp. 2835–2839, 2014.
- [24] A. Aravkin, J. V. Burke, L. Ljung, A. Lozano, and G. Pillonetto, "Generalized Kalman smoothing: Modeling and algorithms," *Automatica*, vol. 86, pp. 63–86, 2017.
- [25] T. Kailath, "Some new algorithms for recursive estimation in constant linear systems," *IEEE Transactions on Information Theory*, vol. 19, no. 6, pp. 750–760, Nov. 1973.
- [26] M. Morf, G. Sidhu, and T. Kailath, "Some new algorithms for recursive estimation in constant, linear, discrete-time systems," *IEEE Transactions* on Automatic Control, vol. 19, no. 4, pp. 315–323, Aug. 1974.
- [27] M. Morf and T. Kailath, "Square-root algorithms for least-squares estimation," *IEEE Transactions on Automatic Control*, vol. 20, no. 4, pp. 487–497, Aug. 1975.
- [28] A. H. Sayed and T. Kailath, "Extended Chandrasekhar recursion," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 619–622, Mar. 1994.
- [29] P. Park and T. Kailath, "Square-root Bryson-Frazier smoothing algorithms," *IEEE Transactions on Automatic Control*, vol. 40, no. 4, pp. 761–766, Apr. 1995.
- [30] B. Hassibi, A. H. Sayed, and T. Kailath, "Square-root arrays and Chandrasekhar recursions for H_{∞} problems," in *The 33rd IEEE Conference* on Decision and Control, vol. 3, Lake Buena Vista, FL, Dec. 1994, pp. 2237–2242.
- [31] B. Hassibi, T. Kailath, and A. H. Sayed, "Array algorithms for H_{∞} estimation," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 702–706, Apr. 2000.
- [32] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. New Jersey: Prentice Hall, 2000.
- [33] D. Comminiello and J. C. Príncipe, Adaptive Learning Methods for Nonlinear System Modeling, 1st ed. Elsevier, 2018.
- [34] B. Chen and J. C. Príncipe, "Maximum correntropy estimation is a smoothed MAP estimation," *IEEE Signal Processing Letters*, vol. 19, no. 8, pp. 491–494, 2012.
- [35] D. Simon, Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches. John Wiley & Sons, 2006.
- [36] M. S. Grewal and A. P. Andrews, Kalman Filtering: Theory and Practice using MATLAB, 4th ed. New Jersey: John Wiley & Sons, 2015.
- [37] Y. Yang and G. Huang, "Map-based localization under adversarial attacks," in *International Symposium on Robotics Research (ISRR)*, *Puerto Varas, Chile*, Dec. 2017.
- [38] —, "Attack-resilient Map-based localization," in RSS Workshop: Adversarial Robotics, Carnegie Mellon University, Pittsburgh, USA, June 2018.
- [39] J. R. Bunch and L. Kaufman, "Some stable methods for calculating inertia and solving symmetric linear systems," *Mathematics of computation*, vol. 31, no. 137, pp. 163–179, 1977.
- [40] C. Ashcraft, R. G. Grimes, and J. G. Lewis, "Accurate symmetric indefinite linear equation solvers," *SIAM Journal on Matrix Analysis* and Applications, vol. 20, no. 2, pp. 513–561, 1998.
- [41] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed. Philadelphia: SIAM, 2002.
- [42] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.
- [43] G. Yu. Kulikov and M. V. Kulikova, "Estimating the state in stiff continuous-time stochastic systems within extended Kalman filtering," *SIAM Journal on Scientific Computing*, vol. 38, no. 6, pp. A3565–A3588, 2016.
- [44] —, "Accurate cubature and extended Kalman filtering methods for estimating continuous-time nonlinear stochastic systems with discrete measurements," *Applied Numerical Mathematics*, vol. 111, pp. 260–275, 2017.
- [45] —, "Accurate continuous-discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking," *Signal Processing*, vol. 139, pp. 25–35, 2017.