Factorized Smoothing with Weighted Hyperbolic Householder Reflectors

Felipe Giraldo-Grueso*
The University of Texas at Austin, Austin, TX 78712

James S. McCabe† and Christopher D'Souza‡
NASA Johnson Space Center, Houston, TX 77058

Renato Zanetti§
The University of Texas at Austin, Austin, TX 78712

The goal of filtering is to estimate the current state using past and present information. In contrast, smoothing refers to estimating past states using current and future information. Under certain assumptions, the Kalman filter (KF) provides a closed-form solution to the filtering problem, from which fixed-interval smoothers such as the Fraser-Potter (FP) and Rauch-Tung-Striebel (RTS) algorithms can be derived. To address numerical instabilities in the KF and associated smoothers, covariance factorization techniques, such as UDU factorization, are used. In this work, different versions of the FP and RTS smoothers are derived and analyzed. Specifically, a FP UDU smoother and a RTS UDU smoother are proposed. For the derivation of the RTS smoother, the weighted hyperbolic Householder reflector (WHHR) is introduced as a generalization of both the Householder reflector (HR) and the hyperbolic Householder reflector (HHR). The UDU smoothers are compared with traditional and stable formulations in a numerical example, demonstrating their numerical equivalence and validating their implementation.

I. Introduction

In state estimation, quantities of interest, often referred to as states, are inferred from dynamics models and measurements collected over time. The state estimation problem is typically divided into three categories: prediction, filtering, and smoothing. Given all available information, current and past, prediction refers to the estimation of a future state, filtering of the current state, while smoothing estimates one or many past states. Therefore, smoothing algorithms can incorporate data collected after the time for which the state was estimated. This work focuses on filtering and smoothing, with particular emphasis on the latter.

*Ph.D. Candidate, Department of Aerospace Engineering and Engineering Mechanics; fgiraldo@utexas.edu (corresponding author)
†Spacecraft Navigation Technical Discipline Expert, Aeroscience and Flight Mechanics Division (EG6); james.s.mccabe@nasa.gov
‡NASA Technical Fellow for Guidance, Navigation and Control, AIAA Fellow; chris.dsouza-1@nasa.gov
§Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Associate Fellow; renato@utexas.edu

Under the assumptions of additive zero-mean Gaussian noise, linear models, and an initial Gaussian probability density function (PDF), the Kalman filter (KF) provides the closed-form solution to the filtering problem [1]. Since the Gaussian distribution maximizes entropy among all distributions with a given covariance [2, 3], the KF has been used in many filtering problems beyond its original assumptions. However, the KF is known to suffer from numerical instability, mainly due to the effects of finite-precision arithmetic [4]. Due to the practical value of the KF, significant effort has gone into addressing these numerical issues. Some solutions include avoiding matrix subtraction and using factorized covariance representations.

Various factorized covariance representations have been proposed to address the numerical instability in the KF. One of the earliest is square-root filtering, where the filter carries the square root of the covariance matrix instead of the full matrix [5]. This approach improves numerical precision, effectively doubling it, while still allowing the full covariance to be reconstructed when needed. Nevertheless, due to limitations in flight software, other factorizations have also been proposed. A commonly used alternative is UDU factorization, where the covariance matrix is factored into an upper triangular matrix (with ones on its diagonal) and a positive diagonal matrix [4, 6]. A crucial requirement for UDU-based filters is maintaining the UDU form of the covariance matrix, which has led to the development of various techniques to preserve this structure. For example, rank-1 [7–9] and rank-2 [10] update algorithms have been proposed to efficiently obtain the UDU factors after a low-rank update. Additionally, orthogonalization methods, such as the modified weighted Gram-Schmidt (MWGS) algorithm, have been used to maintain the UDU form when the covariance matrix is pre- and post-multiplied by arbitrary matrices [6].

For fixed-interval smoothing, two main algorithms have been proposed: the Fraser-Potter (FP) smoother [11, 12] and the Rauch-Tung-Striebel (RTS) smoother [13]. Although these smoothers are mathematically equivalent in the linear and Gaussian cases [14], they differ in their formulations. As with the KF, factorized versions, primarily of the RTS smoother, also exist. Bierman [15] presented a UDU-factorized RTS smoother by performing process noise updates in a rank-1 fashion, resulting in a rank-2 update for the smoothed covariance. McReynolds [16] then proposed a more computationally efficient variant of Bierman's solution. Shortly after, Watanabe and Tzafestas [17] derived a new UDU formulation for the RTS smoother by expressing the smoothing equations in a Joseph-like form and performing rank-1 updates to obtain the UDU factors of the smoothed covariance. All of these approaches build upon the RTS smoother equations, using well-known techniques such as rank-1 and rank-2 updates, as well as the MWGS algorithm.

This work proposes two new UDU-factorized algorithms for both the FP and RTS smoothers. For the FP smoother, the backward filter is used in an information UDU formulation (inspired from [18]), allowing the UDU factors of the smoothed covariance matrix to be obtained using a rank-1 update. For the RTS smoother, the smoothed covariance equation is reformulated such that it is simultaneously updated and downdated by a weighted matrix product. To maintain the UDU structure after the simultaneous update and downdate, the weighted hyperbolic Householder reflector

(WHHR) is introduced. The derivation of the WHHR is proposed as a generalization to the Householder reflector (HR) [19] and the hyperbolic Householder reflector (HHR) [20]. In addition, a numerically stable form of the RTS smoother, similar to that presented in [17], is derived.

In summary, the key contributions of this work are:

- Derivation of the WHHR and analysis of its applicability to UDU filtering and smoothing.
- Derivation of a FP UDU smoother.
- Derivation of a RTS UDU smoother.
- Derivation of a stable form of the RTS smoother.

In addition to these contributions, this work provides a unified reference that brings together the different smoothing approaches, serving as a guide for selecting and implementing the most suitable strategy for a given application.

The rest of this paper is organized as follows. First, Section II introduces the key concepts that are used throughout the work. This is followed by Section III, which presents the derivation and application of the WHHR. In Section IV, three formulations of the FP smoother are presented, including the UDU version. Section V discusses three versions of the RTS smoother, including both the stable and UDU implementations. A numerical example validating the equivalence of the various smoothers is provided in Section VI. Final remarks on the practical use of the WHHR and the proposed smoothers are given in Section VII, and Section VIII draws conclusions.

II. Preliminaries

This section provides an overview of the key concepts used in the following sections. First, a brief description of filtering and smoothing is provided. Following that, the UDU factorization is introduced. Finally, key algorithms are presented for performing the UDU factorization when modifying UDU-factorized covariance matrices.

A. Filtering

In filtering, the main task is to estimate the current state based on dynamic knowledge and measurements collected through time. Within a Bayesian framework, the filtering problem is solved with the Bayesian recursive relations (BRRs) [1]. To introduce the BRRs, consider the following *linear* and *discrete* system

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{q}_k, \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{\eta}_k, \tag{2}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state at time step $k \in \mathbb{R}$, $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) \in \mathbb{R}^{n\times n}$ is the state transition matrix, $\boldsymbol{q}_k \in \mathbb{R}^p$ is the process noise, $\boldsymbol{G}_k \in \mathbb{R}^{n\times p}$ is the process noise shaping matrix, $\boldsymbol{y}_k \in \mathbb{R}^m$ is the measurement, $\boldsymbol{H}_k \in \mathbb{R}^{m\times n}$ is the measurement mapping matrix, and $\boldsymbol{\eta}_k \in \mathbb{R}^m$ is the measurement noise. The measurement noise shaping matrix is assumed to be the identity to unclutter notation in the following derivations.

The first step in the BRRs is to propagate the state PDF using the Chapman-Kolmogorov (CK) equation [1]. Given an initial posterior state PDF, $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, where $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k\}$, the CK equation reads

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k}\right) = \int\limits_{\mathcal{S}(\boldsymbol{x}_k)} p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}\right)\, d\boldsymbol{x}_k, \tag{3}$$

where $\mathcal{S}\left(\boldsymbol{x}_k\right)$ denotes the support of $\boldsymbol{x}_k$, $p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k\right)$ is the transitional PDF, and $p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k}\right)$ is referred to as the prior state PDF. Once a measurement becomes available, $(k \leftarrow k+1)$, and the state PDF has been propagated, the second step is to update the PDF using Bayes' rule [1], where*

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}\right) = \frac{p\left(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1}\right) p\left(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}\right)}{p\left(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}\right)}. \tag{4}$$

If the system is *linear*, the initial state PDF is Gaussian, and both $\boldsymbol{q}_k$ and $\boldsymbol{\eta}_k$ are zero-mean Gaussian white noise with covariance $\boldsymbol{Q}_k \in \mathbb{R}^{p\times p}$ and $\boldsymbol{R}_k \in \mathbb{R}^{m\times m}$, respectively, the KF is the closed form solution to the BRRs [1].

The KF, like the BRRs, is divided in two steps, propagate and update. Starting from the previous posterior state estimate $\hat{\boldsymbol{x}}_k^+$ and covariance $\boldsymbol{P}_k^+$ (where the $\hat{}$ represents an estimated quantity and the $(+)$ represents a posterior state), both are first propagated to time step $k+1$, following

$$\hat{\boldsymbol{x}}_{k+1}^- = \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_k^+, \tag{5}$$

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\Phi}_k \boldsymbol{P}_k^+ \boldsymbol{\Phi}_k^T + \boldsymbol{G}_k \boldsymbol{Q}_k \boldsymbol{G}_k^T, \tag{6}$$

where the $(-)$ represents a prior state. Note that the KF only keeps track of the first two moments of the state PDF. Once a measurement is obtained, $(k \leftarrow k+1)$, the second step is to perform a Kalman update on both the prior state and covariance, where

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-\right), \tag{7}$$

$$\boldsymbol{P}_k^+ = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\right) \boldsymbol{P}_k^-, \tag{8}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_{\boldsymbol{k}}^T \left(\boldsymbol{H}_{\boldsymbol{k}} \boldsymbol{P}_k^- \boldsymbol{H}_{\boldsymbol{k}}^T + \boldsymbol{R}_k\right)^{-1}. \tag{9}$$

B. Smoothing

Once all the measurements have been processed, the state estimate and its covariance can be *smoothed* by enforcing the dynamics in a backward manner. There are several forms of smoothing, but in this work, the focus lies on

*If the measurement noise is assumed to be a white sequence uncorrelated from other error sources, the likelihood function reduces to $p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{x}_k)$.

fixed-interval smoothing, where the entire batch of measurements is used to provide a best possible estimate over a fixed interval [14]. Two main algorithms for fixed-interval smoothing exist: the FP smoother [11, 12] and the RTS smoother [13]. These algorithms are equivalent in the linear case (please see the proof in the appendix for details). This work studies both versions, as discussed in their respective sections.

Please note that a discussion on smoothing is postponed to the next sections, as it is the main focus of the work.

C. UDU Factorization

Many covariance factorization methods have been implemented to improve the numerical stability of the KF [4]. The UDU factorization in the KF has recently been used in various aerospace engineering applications, such as the flight software in Orion [21, 22]. The UDU-factorized KF has three main benefits compared to the full-covariance KF: less accumulation of floating-point errors, trivial evaluation of the positive definiteness of the covariance matrix (avoiding expensive eigenvalue or singular value decompositions), and trivial preservation of the covariance matrix's symmetry [23].

In the UDU factorization, the covariance matrix $\boldsymbol{P}$ is factorized as

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T, \tag{10}$$

where $\boldsymbol{U}$ is an upper triangular matrix with ones on its diagonal, and $\boldsymbol{D}$ is a diagonal matrix. In this form, the positive definiteness of $\boldsymbol{P}$ can be easily assessed by looking at the diagonal entries of $\boldsymbol{D}$. In addition, if storage limitations are a concern, instead of saving an $n \times n$ matrix, $\boldsymbol{D}$ can be stored as a vector with n entries, and $\boldsymbol{U}$ can be stored as a vector with $n(n-1)/2$ entries, where the ones on its diagonal can be ignored [6]. This can be particularly useful when n is large and only requires bookkeeping to map between vector and matrix indices.

Various implementations of the UDU-factorized KF exist, such as the information formulation [18]. In many of these implementations, two common details tend to surface. The first is finding the UDU factors of a matrix after a rank-1 update

$$\tilde{\boldsymbol{P}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \pm c\boldsymbol{\xi}\boldsymbol{\xi}^T, \tag{11}$$

such that

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T, \tag{12}$$

where $c \in \mathbb{R}$, and $\boldsymbol{\xi} \in \mathbb{R}^n$ is a vector, making $\boldsymbol{\xi}\boldsymbol{\xi}^T$ a rank-1 matrix. The second is finding the UDU factors after a matrix multiplication of the form

$$\tilde{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{U}\boldsymbol{D}\left(\boldsymbol{A}\boldsymbol{U}\right)^T, \tag{13}$$

where $\boldsymbol{A}$ is not necessarily upper triangular, and, so, $\boldsymbol{AU}$ is not necessarily upper triangular. To address these two implementation considerations, various strategies have been proposed. The following sections build on these by reviewing established methods and developing new approaches.

D. Rank-1 Updates

As previously mentioned, some of the UDU-factorized KF implementations often have to deal with rank-1 updates, as detailed in Eq. (11). Notably, there are two different flavors for a rank-1 update, depending on whether the UDU factorization is being updated (plus sign) or downdated (minus sign). In the case that the UDU factorization is being updated, that is

$$\tilde{\boldsymbol{P}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T + c\boldsymbol{\xi}\boldsymbol{\xi}^T, \tag{14}$$

the Agee-Turner update algorithm can be used to obtain the UDU factors of $\tilde{\boldsymbol{P}}$ [7]. However, downdates of the form

$$\tilde{\boldsymbol{P}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T - c\boldsymbol{\xi}\boldsymbol{\xi}^T, \tag{15}$$

can be more common in the UDU-factorized KF, where the Agee-Turner update algorithm is known to be highly numerically unstable.

To understand where such downdates may appear, recall the covariance update in the KF as described in Eq. (8). Without loss of generality, consider the processing of vector measurements as individual scalars, such that $\boldsymbol{H}_k \in \mathbb{R}^{1\times n}$, $R_k \in \mathbb{R}$, and $\boldsymbol{K}_k \in \mathbb{R}^{n\times 1}$. Note that if the measurements are correlated, they can be uncorrelated by performing noise pre-whitening [6]. If the update is rewritten in UDU form, such that

$$\boldsymbol{P}_k^+ = \boldsymbol{U}_k^-\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{U}_k^-\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T, \tag{16}$$

and the Kalman gain is expanded in terms of the UDU factorization

$$\boldsymbol{K}_k = \frac{\boldsymbol{U}_k^-\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T\boldsymbol{H}_k^T}{\boldsymbol{H}_k\boldsymbol{U}_k^-\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T\boldsymbol{H}_k^T + R_k} = c\left(\boldsymbol{U}_k^-\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T\boldsymbol{H}_k^T\right), \tag{17}$$

with

$$c = \frac{1}{\boldsymbol{H}_k\boldsymbol{U}_k^-\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T\boldsymbol{H}_k^T + R_k}, \tag{18}$$

then the covariance update becomes

$$\boldsymbol{P}_k^+ = \boldsymbol{U}_k^-\left[\boldsymbol{D}_k^- - c\left(\boldsymbol{D}_k^-\left(\boldsymbol{U}_k^-\right)^T\boldsymbol{H}_k^T\right)\boldsymbol{H}_k\boldsymbol{U}_k^-\boldsymbol{D}_k^-\right]\left(\boldsymbol{U}_k^-\right)^T = \boldsymbol{U}_k^-\left(\boldsymbol{D}_k^- - c\boldsymbol{\xi}\boldsymbol{\xi}^T\right)\left(\boldsymbol{U}_k^-\right)^T, \tag{19}$$

with

$$\boldsymbol{\xi} = \boldsymbol{D}_k^- \left(\boldsymbol{U}_k^-\right)^T \boldsymbol{H}_k^T. \tag{20}$$

In this case, $\boldsymbol{D}_k^-$ is being downdated by a rank-1 matrix. This downdate has a slightly different form than Eq. (15). However, the UDU factors of the quantity in brackets can be found by setting

$$\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T - c\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \left(\boldsymbol{U}_k^-\right)^T \boldsymbol{H}_k^T\right) \boldsymbol{H}_k \boldsymbol{U}_k^- \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T, \tag{21}$$

where $\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{D} = \boldsymbol{D}_k^-$ for the first downdate. By plugging in the UDU factors in Eq. (19),

$$\boldsymbol{P}_k^+ = \boldsymbol{U}_k^- \left(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T\right) \left(\boldsymbol{U}_k^-\right)^T, \tag{22}$$

therefore

$$\boldsymbol{U}_k^+ = \boldsymbol{U}_k^- \tilde{\boldsymbol{U}}, \quad \boldsymbol{D}_k^+ = \tilde{\boldsymbol{D}}, \quad \boldsymbol{P}_k^+ = \boldsymbol{U}_k^+ \boldsymbol{D}_k^+ \left(\boldsymbol{U}_k^+\right)^T, \tag{23}$$

since the product of two upper triangular matrices with ones in the diagonal is an upper triangular matrix with ones in the diagonal.

In theory, the Agee-Turner update algorithm could be used to perform the downdate. However, the minus sign can result in numerical issues [6]. In this case, the so-called Carlson update can be used to perform the downdate, as it obtains a more numerically stable and efficient UDU factorization [8]. However, this algorithm is derived for the optimal Kalman gain,However, this algorithm is derived for the optimal Kalman gain,[†] as described in Eq. (9). Modifying the Kalman gain, such as when using measurement underweighting or general suboptimal gains, can result in a suboptimal UDU factorization if the Carlson update is performed [23]. For general rank-1 updates as given in Eq. (11), the Fletcher-Powell rank-1 algorithm has been proposed [9]. When $c > 0$, the Fletcher-Powell algorithm is numerically identical to the Agee-Turner. In the case where $c < 0$, the Fletcher-Powell algorithm avoids the numerical issues that the Agee-Turner update has, requiring only a modest increase in computational complexity [23]. Therefore, the Fletcher-Powell algorithm provides a more robust solution to the rank-1 updates without restriction on the sign of c.

E. Modified Weighted Gram-Schmidt

In some cases, the UDU factorization in the KF may be multiplied by an arbitrary matrix, as in Eq. (13). The most common step where this happens is when the UDU-factorized covariance is being propagated. Consider Eq. (6), written

[†] Optimal in a mean-square error sense.

in UDU form and factorizing terms in block matrices[‡]

$$\boldsymbol{P}_{k+1}^{-}=\begin{bmatrix}\boldsymbol{\Phi}_k\boldsymbol{U}_k^{+} & \boldsymbol{G}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{D}_k^{+} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_k\boldsymbol{U}_k^{+} & \boldsymbol{G}_k\end{bmatrix}^T. \tag{24}$$

In this case, to find the UDU factors of $\boldsymbol{P}_{k+1}^{-}$, namely $\boldsymbol{U}_{k+1}^{-}$ and $\boldsymbol{D}_{k+1}^{-}$, a matrix $\boldsymbol{\Gamma}_k$ needs to be found, such that

$$\boldsymbol{P}_{k+1}^{-}=\begin{bmatrix}\boldsymbol{\Phi}_k\boldsymbol{U}_k^{+} & \boldsymbol{G}_k\end{bmatrix}\boldsymbol{\Gamma}_k^{-T}\boldsymbol{\Gamma}_k^{T}\begin{bmatrix}\boldsymbol{D}_k^{+} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_k\end{bmatrix}\boldsymbol{\Gamma}_k\boldsymbol{\Gamma}_k^{-1}\begin{bmatrix}\boldsymbol{\Phi}_k\boldsymbol{U}_k^{+} & \boldsymbol{G}_k\end{bmatrix}^T. \tag{25}$$

Where the UDU factorization is given by

$$\begin{bmatrix}\boldsymbol{U}_{k+1}^{-} & \boldsymbol{0}\end{bmatrix}=\begin{bmatrix}\boldsymbol{\Phi}_k\boldsymbol{U}_k^{+} & \boldsymbol{G}_k\end{bmatrix}\boldsymbol{\Gamma}_k^{-T},\quad \boldsymbol{D}_{k+1}^{-}=\boldsymbol{\Gamma}_k^{T}\begin{bmatrix}\boldsymbol{D}_k^{+} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_k\end{bmatrix}\boldsymbol{\Gamma}_k. \tag{26}$$

Finding $\boldsymbol{\Gamma}_k$ is equivalent to a change of basis, or an orthogonalization problem, which, in the UDU-factorized KF, is typically found within literature using the MWGS algorithm [6].

III. Weighted Hyperbolic Householder Reflectors

While rank-1 updates and downdates are common in the UDU-factorized version of the KF, there is another form of update and downdate that is worth discussing. Suppose that the following simultaneous update and downdate is being performed on the UDU-factorized covariance

$$\tilde{\boldsymbol{P}}=\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T-\boldsymbol{A}\boldsymbol{E}\boldsymbol{A}^T+\boldsymbol{B}\boldsymbol{F}\boldsymbol{B}^T, \tag{27}$$

where both $\boldsymbol{E}$ and $\boldsymbol{F}$ are positive diagonal matrices, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices of congruent sizes. In this case, the task of finding the UDU factors of $\tilde{\boldsymbol{P}}$, namely $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{D}}$, is not straightforward. To this end, WHHRs are proposed and derived in this work. The derivation starts with the introduction of HRs, followed by the HHRs. Finally, the WHHRs are derived as a generalization of the previous.

The well-known HR is a matrix that defines the reflection of a vector $\boldsymbol{v}\in\mathbb{R}^n$ across its orthogonal plane [24]. In [19] the HR, $\boldsymbol{Q}$, is given by

$$\boldsymbol{Q}=\boldsymbol{I}-\frac{2\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}, \tag{28}$$

where $\boldsymbol{Q}$ is both Hermitian and orthonormal [20], and $\boldsymbol{v}$ is referred to as the Householder vector. Due to the

[‡]For the purposes of this example, assume $\boldsymbol{Q}_k$ a diagonal matrix.

orthonormality of the HR, when pre-multiplying a vector, the norm of the vector is preserved. Therefore, the HR can be used to force all the energy (norm) of a vector[§] $\boldsymbol{u} \in \mathbb{C}^n$ into a single entry, such that

$$\boldsymbol{Q}\boldsymbol{u} = \sigma \boldsymbol{e}_j, \tag{29}$$

where $\sigma \in \mathbb{C}$ is a yet-to-be-defined scalar, and $\boldsymbol{e}_j$ is the j-th canonical basis of $\mathbb{R}^n$. This makes the HR particularly useful for triangularization. By letting the Householder vector be defined as

$$\boldsymbol{v} = \boldsymbol{u} + \sigma \boldsymbol{e}_j, \tag{30}$$

the scalar σ can be solved for using the orthonormality of $\boldsymbol{Q}$, where (to see the full proof, please refer to [20, 25])

$$\sigma = \pm \frac{u^{(j)}}{|u^{(j)}|} \sqrt{\boldsymbol{u}^\dagger \boldsymbol{u}}, \tag{31}$$

with $\boldsymbol{u}^\dagger$ representing the conjugate transpose of $\boldsymbol{u}$ and $u^{(j)}$ the j-th component of the vector $\boldsymbol{u}$. As a consequence, an arbitrary matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, with $n \leq m$, can be triangularized by performing a succession of Householder reflections, such that

$$\boldsymbol{A}\boldsymbol{Q}_1\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_n = \boldsymbol{A}\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{0}_{n \times m-n} & \hat{\boldsymbol{U}} \end{bmatrix} \tag{32}$$

where $\hat{\boldsymbol{U}} \in \mathbb{R}^{n \times n}$ is an upper triangular matrix (not to be confused with $\boldsymbol{U}$ which is also upper triangular but, in addition, has ones on its diagonal).

Remark III.1. *Consider the following update:*

$$\tilde{\boldsymbol{P}} = \boldsymbol{S}\boldsymbol{S}^T + \boldsymbol{A}\boldsymbol{A}^T, \tag{33}$$

where the matrix $\boldsymbol{S}\boldsymbol{S}^T$ is being updated by $\boldsymbol{A}\boldsymbol{A}^T$. The HR can be used to find the square root decomposition of $\tilde{\boldsymbol{P}}$, such that [24]

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{S}}^T. \tag{34}$$

Since $\boldsymbol{Q}$ is orthonormal,

$$\tilde{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{A} \end{bmatrix} \boldsymbol{Q}\boldsymbol{Q}^T \begin{bmatrix} \boldsymbol{S} & \boldsymbol{A} \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{0} & \hat{\boldsymbol{U}} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \hat{\boldsymbol{U}} \end{bmatrix}^T = \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^T, \tag{35}$$

where, $\boldsymbol{Q} = \boldsymbol{Q}_1\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_n$, making $\tilde{\boldsymbol{S}} = \hat{\boldsymbol{U}}$.

A generalization of the HR has been developed for least-squares applications and is referred to as the HHR [20].

[§]General engineering applications only require consideration of real values, but the potential for complex-domain reflections are maintained for generality.

The HHR, $\mathcal{H}$, is defined as [20, 24]

$$\mathcal{H} = \boldsymbol{Y} - \frac{2\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{Y}\boldsymbol{v}}, \tag{36}$$

where $\boldsymbol{Y} = \text{blkdiag}\{\pm\boldsymbol{I}, \pm\boldsymbol{I}, \ldots, \pm\boldsymbol{I}\}$ is the signature matrix of $\mathcal{H}$ and $\boldsymbol{v}$ is now the hyperbolic Householder vector. The matrix $\boldsymbol{Y}$ is called the signature matrix of $\mathcal{H}$ since $\mathcal{H}$ is $\boldsymbol{Y}$-hypernormal, that is, $\mathcal{H}\boldsymbol{Y}\mathcal{H}^T = \mathcal{H}^T\boldsymbol{Y}\mathcal{H} = \boldsymbol{Y}$ [20]. The term *hypernormal* comes from the fact that a hypernormal matrix will conserve the hyperbolic norm of a vector. For a vector $\boldsymbol{u}$, the hyperbolic norm is defined as [20]

$$\boldsymbol{u}^T\boldsymbol{Y}\boldsymbol{u} = \sum_{i=1}^{n} |u^{(i)}|^2\, Y^{(i,i)}. \tag{37}$$

It is called a hyperbolic norm since hyperbolic functions can be characterized as differences of sums of squares. Therefore, if $\boldsymbol{v}^T = \boldsymbol{u}^T\mathcal{H}$, then $\boldsymbol{v}^T\boldsymbol{Y}\boldsymbol{v} = \boldsymbol{u}^T\boldsymbol{Y}\boldsymbol{u}$, since $\mathcal{H}$ is $\boldsymbol{Y}$-hypernormal.

Similar to the HR, the HHR can also be used to force all of the energy of $\boldsymbol{u}$ into a single entry. In this case, the hyperbolic Householder vector, and the scalar σ are defined as (to see the full proof, please refer to [20]):

$$\boldsymbol{v} = \boldsymbol{Y}\boldsymbol{u} + \sigma\boldsymbol{e}_j, \quad \sigma = \pm\frac{u^{(j)}}{|u^{(j)}|}\sqrt{\boldsymbol{u}^\dagger\boldsymbol{Y}\boldsymbol{u}}. \tag{38}$$

Therefore, the HHR can also be used for triangularization.

Remark III.2. *Consider the following simultaneous downdate and update:*

$$\tilde{\boldsymbol{P}} = \boldsymbol{S}\boldsymbol{S}^T - \boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T, \tag{39}$$

where the matrix $\boldsymbol{S}\boldsymbol{S}^T$ is being downdated by $\boldsymbol{A}\boldsymbol{A}^T$ and updated by $\boldsymbol{B}\boldsymbol{B}^T$. In this case, the HHR can be used to find the square root decomposition of $\tilde{\boldsymbol{P}}$ [24]. Considering the $\boldsymbol{Y}$-hypernormality of $\mathcal{H}$, and letting $\boldsymbol{Y} = \text{blkdiag}\{\boldsymbol{I}, -\boldsymbol{I}, \boldsymbol{I}\}$

$$\tilde{\boldsymbol{P}} = \begin{bmatrix}\boldsymbol{S} & \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\mathcal{H}\boldsymbol{Y}\mathcal{H}^T\begin{bmatrix}\boldsymbol{S} & \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}^T = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{U}}\end{bmatrix}\boldsymbol{Y}\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{U}}\end{bmatrix}^T = \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^T, \tag{40}$$

where $\mathcal{H} = \mathcal{H}_1\mathcal{H}_2\cdots\mathcal{H}_n$, following a similar construction to $\boldsymbol{Q} = \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_n$, then $\tilde{\boldsymbol{S}} = \hat{\boldsymbol{U}}$.

Remark III.2 closely aligns with the objective of updating and downdating a UDU-factorized covariance, as outlined in Eq. (27). However, the weighting matrices are missing, which is the motivation for the WHHRs.

Consider a new matrix, $\boldsymbol{\Omega}$, which will be referred to as the WHHR. Motivated by the construction of the HR and the HHR, let the WHHR be defined as:

$$\boldsymbol{\Omega} = \boldsymbol{Y} - \frac{2\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{W}\boldsymbol{Y}\boldsymbol{v}}, \tag{41}$$

where $\boldsymbol{Y}$ is the signature matrix, and $\boldsymbol{W} = \text{diag}\{w_1, w_2, \ldots, w_n\}$ with $w_i > 0\,\forall\, i \in \{1, 2, \ldots, n\}$. With this definition, the HHR can be recovered by setting $\boldsymbol{W} = \boldsymbol{I}$, and further setting $\boldsymbol{Y} = \boldsymbol{I}$ recovers the HR. Consequently, the WHHR is a generalization of both the HHR and the HR.

Lemma III.1. *The WHHR is* $(\boldsymbol{WY})$*-hypernormal, that is*

$$\boldsymbol{\Omega}^T \boldsymbol{WY\Omega} = \boldsymbol{WY}. \tag{42}$$

Proof. To prove $(\boldsymbol{WY})$-hypernomality, consider the following

$$\boldsymbol{\Omega}^T \boldsymbol{WY\Omega} = \left(\boldsymbol{Y} - \frac{2\boldsymbol{vv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}}\right)^T \boldsymbol{WY}\left(\boldsymbol{Y} - \frac{2\boldsymbol{vv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}}\right) = \left(\boldsymbol{Y} - \frac{2\boldsymbol{Wvv}^T}{\boldsymbol{v}^T\boldsymbol{WYv}}\right)\boldsymbol{WY}\left(\boldsymbol{Y} - \frac{2\boldsymbol{vv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}}\right), \tag{43}$$

$$= \boldsymbol{WY}^3 - \frac{2\boldsymbol{Wvv}^T\boldsymbol{WY}^2}{\boldsymbol{v}^T\boldsymbol{WYv}} - \frac{2\boldsymbol{WY}^2\boldsymbol{vv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}} + \frac{4\boldsymbol{Wvv}^T\boldsymbol{WYvv}^T\boldsymbol{W}}{\left(\boldsymbol{v}^T\boldsymbol{WYv}\right)^2}. \tag{44}$$

Since $\boldsymbol{Y}$ is the signature matrix, $\boldsymbol{Y}^3 = \boldsymbol{Y}$ and $\boldsymbol{Y}^2 = \boldsymbol{I}$, therefore

$$\boldsymbol{\Omega}^T \boldsymbol{WY\Omega} = \boldsymbol{WY} - \frac{2\boldsymbol{Wvv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}} - \frac{2\boldsymbol{Wvv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}} + \frac{4\boldsymbol{Wv}\left(\boldsymbol{v}^T\boldsymbol{WYv}\right)\boldsymbol{v}^T\boldsymbol{W}}{\left(\boldsymbol{v}^T\boldsymbol{WYv}\right)^2} \tag{45}$$

$$= \boldsymbol{WY} - \frac{4\boldsymbol{Wvv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}} + \frac{4\boldsymbol{Wvv}^T\boldsymbol{W}}{\boldsymbol{v}^T\boldsymbol{WYv}} \tag{46}$$

$$= \boldsymbol{WY}. \tag{47}$$

However, it is important to acknowledge that $\boldsymbol{\Omega}^T\boldsymbol{WY\Omega} \neq \boldsymbol{\Omega WY\Omega}^T$ and $\boldsymbol{\Omega}^T\boldsymbol{Y\Omega} = \boldsymbol{Y}$. The proofs of these properties are omitted for brevity, but they follow from the same technique used to prove $(\boldsymbol{WY})$-hypernormality. □

Lemma III.2. *The WHHR can also be used to compress all the energy of a vector into a single entry, such that*

$$\boldsymbol{\Omega u} = \sigma \boldsymbol{e}_j, \tag{48}$$

for some $\sigma \in \mathbb{C}$*, where it must hold that*

$$\boldsymbol{u}^\dagger \boldsymbol{WYu} = |\sigma|^2 W^{(j,j)} Y^{(j,j)}, \tag{49}$$

with $W^{(j,j)}$ *and* $Y^{(j,j)}$ *denoting the* j*-th diagonal elements of* $\boldsymbol{W}$ *and* $\boldsymbol{Y}$*, respectively.*

Proof. Using the $(\boldsymbol{WY})$-hypernormality of the WHHR,

$$\boldsymbol{u}^{\dagger}\boldsymbol{WYu} = \boldsymbol{u}^{\dagger}\boldsymbol{\Omega}^{T}\boldsymbol{WY\Omega u}, \tag{50}$$

$$= \left(\sigma\boldsymbol{e}_j\right)^{T}\boldsymbol{WY}\left(\sigma\boldsymbol{e}_j\right), \tag{51}$$

$$= \sigma^{\dagger}\sigma\boldsymbol{e}_j^{T}\boldsymbol{WYe}_j, \tag{52}$$

$$= |\sigma|^{2}W^{(j,j)}Y^{(j,j)}, \tag{53}$$

where $\sigma^{\dagger}$ is the complex conjugate of σ. $\square$

To find σ, consider the same hyperbolic Householder vector used for the HHR, such that

$$\boldsymbol{\Omega} = \boldsymbol{Y} - \frac{2\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)^{T}\boldsymbol{W}}{\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)^{T}\boldsymbol{WY}\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)} = \boldsymbol{Y} - \frac{2\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)\left(\boldsymbol{u}^{\dagger}\boldsymbol{Y}+\sigma^{\dagger}\boldsymbol{e}_j^{T}\right)\boldsymbol{W}}{\left(\boldsymbol{u}^{\dagger}\boldsymbol{Y}+\sigma^{\dagger}\boldsymbol{e}_j^{T}\right)\boldsymbol{WY}\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)}. \tag{54}$$

Right-multiplying $\boldsymbol{\Omega}$ by $\boldsymbol{u}$ and expanding the denominator yields

$$\boldsymbol{\Omega u} = \boldsymbol{Yu} - \frac{2\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)\left(\boldsymbol{u}^{\dagger}\boldsymbol{WYu}+\sigma^{\dagger}\boldsymbol{e}_j^{T}\boldsymbol{Wu}\right)}{\boldsymbol{u}^{\dagger}\boldsymbol{WY}^{3}\boldsymbol{u}+\sigma\boldsymbol{u}^{\dagger}\boldsymbol{WY}^{2}\boldsymbol{e}_j+\sigma^{\dagger}\boldsymbol{e}_j^{T}\boldsymbol{WY}^{2}\boldsymbol{u}+\sigma^{\dagger}\sigma\boldsymbol{e}_j^{T}\boldsymbol{WYe}_j}. \tag{55}$$

Simplifying $\boldsymbol{Y}^{3} = \boldsymbol{Y}$, $\boldsymbol{Y}^{2} = \boldsymbol{I}$, and noting that $\sigma^{\dagger}\sigma\boldsymbol{e}_j^{T}\boldsymbol{WYe}_j = \boldsymbol{u}^{\dagger}\boldsymbol{WYu}$, $\sigma^{\dagger}\boldsymbol{e}_j^{T}\boldsymbol{Wu} = \sigma^{\dagger}W^{(j,j)}u^{(j)}$,

$$\boldsymbol{\Omega u} = \boldsymbol{Yu} - \frac{2\left(\boldsymbol{Yu}+\sigma\boldsymbol{e}_j\right)\left(\boldsymbol{u}^{\dagger}\boldsymbol{WYu}+\sigma^{\dagger}W^{(j,j)}u^{(j)}\right)}{2\boldsymbol{u}^{\dagger}\boldsymbol{WYu}+\sigma W^{(j,j)}u^{(j)}+\sigma^{\dagger}W^{(j,j)}u^{(j)}}. \tag{56}$$

At this point, σ can be chosen in any convenient form, as long as Eq. (49) is satisfied. Therefore, note that the denominator can be guaranteed to be real with the choice

$$\sigma = \frac{u^{(j)}}{|u^{(j)}|}\sqrt{\frac{\boldsymbol{u}^{\dagger}\boldsymbol{WYu}}{\boldsymbol{Y}^{(j,j)}\boldsymbol{W}^{(j,j)}}}. \tag{57}$$

Remark III.3. *The WHHR can be used to find the UDU factors, $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{D}}$, of $\tilde{\boldsymbol{P}}$ in Eq.* (27)*. Let Eq.* (27) *be rewritten as*

$$\tilde{\boldsymbol{P}} = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{A} & \boldsymbol{U}\end{bmatrix}\boldsymbol{WY}\begin{bmatrix}\boldsymbol{B} & \boldsymbol{A} & \boldsymbol{U}\end{bmatrix}^{T}, \tag{58}$$

where $\boldsymbol{W} = \mathrm{blkdiag}\{\boldsymbol{F},\boldsymbol{E},\boldsymbol{D}\}$ and $\boldsymbol{Y} = \mathrm{blkdiag}\{\boldsymbol{I},-\boldsymbol{I},\boldsymbol{I}\}$. Using the $(\boldsymbol{WY})$-hypernormality of $\boldsymbol{\Omega}$,

$$\tilde{\boldsymbol{P}} = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{A} & \boldsymbol{U}\end{bmatrix}\boldsymbol{\Omega}^{T}\boldsymbol{WY\Omega}\begin{bmatrix}\boldsymbol{B} & \boldsymbol{A} & \boldsymbol{U}\end{bmatrix}^{T} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{U}}\end{bmatrix}\boldsymbol{WY}\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{U}}\end{bmatrix}^{T}, \tag{59}$$

where $\mathbf{\Omega} = \mathbf{\Omega}_1\mathbf{\Omega}_2\cdots\mathbf{\Omega}_n$ *follows a similar construction to* $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{H}}_1\boldsymbol{\mathcal{H}}_2\cdots\boldsymbol{\mathcal{H}}_n$ *and* $\boldsymbol{Q} = \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_n$*. Note, however, that even though* $\hat{\boldsymbol{U}}$ *is upper triangular, it does not have ones on its diagonal, therefore*

$$\tilde{\boldsymbol{U}} = \hat{\boldsymbol{U}}\boldsymbol{\mathcal{D}}^{-1}, \quad \tilde{\boldsymbol{D}} = \boldsymbol{\mathcal{D}}\boldsymbol{D}\boldsymbol{\mathcal{D}}^T, \tag{60}$$

where $\boldsymbol{\mathcal{D}}$ *is a diagonal matrix whose diagonal elements are the diagonal elements of* $\hat{\boldsymbol{U}}$*, and* $\boldsymbol{\mathcal{D}}^{-1}$ *can easily be obtained.*

Remark III.4. *The WHHR algorithm can also be used as an alternative to the MWGS algorithm for finding the UDU factors of* $\tilde{\boldsymbol{P}}$ *in Eq.* (13)*. Let* $\boldsymbol{W} = \boldsymbol{D}$ *and* $\boldsymbol{Y} = \boldsymbol{I}$

$$\tilde{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{U}\boldsymbol{W}\boldsymbol{Y}\boldsymbol{U}^T\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{U}\mathbf{\Omega}^T\boldsymbol{W}\boldsymbol{Y}\mathbf{\Omega}\left(\boldsymbol{A}\boldsymbol{U}\right)^T = \hat{\boldsymbol{U}}\boldsymbol{W}\boldsymbol{Y}\hat{\boldsymbol{U}}^T, \tag{61}$$

such that

$$\tilde{\boldsymbol{U}} = \hat{\boldsymbol{U}}\boldsymbol{\mathcal{D}}^{-1}, \quad \tilde{\boldsymbol{D}} = \boldsymbol{\mathcal{D}}\boldsymbol{D}\boldsymbol{\mathcal{D}}^T. \tag{62}$$

Algorithm 1 provides pseudocode for updating and downdating the UDU factors of a covariance matrix using the WHHR. Remark III.4 can be solved using Algorithm 1 by setting $\boldsymbol{A} = \boldsymbol{E} = \boldsymbol{B} = \boldsymbol{F} = [\,]$ (empty vector), and $\boldsymbol{U} = \boldsymbol{A}\boldsymbol{U}$.

Algorithm 1 WHHR for updating and downdating $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$

Input: $\boldsymbol{U}, \boldsymbol{D}, \boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{F}$
Output: $\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{D}}$
▷ $\boldsymbol{U}$ *is* $n \times n$, $\boldsymbol{A}$ *is* $n \times p$, $\boldsymbol{B}$ *is* $n \times q$ ◁
▷ $\boldsymbol{D}$ *is* $n \times n$, $\boldsymbol{E}$ *is* $p \times p$, $\boldsymbol{F}$ *is* $q \times q$ ◁
$\boldsymbol{X} = [\boldsymbol{B} \mid \boldsymbol{A} \mid \boldsymbol{U}]$ ▷ $\boldsymbol{X}$ *is* $n \times m$
$\boldsymbol{W} = \text{blkdiag}\left(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{D}\right)$
$\boldsymbol{Y} = \text{blkdiag}\left(\boldsymbol{I}_{q\times q}, -\boldsymbol{I}_{p\times p}, \boldsymbol{I}_{(m-p-q)\times(m-p-q)}\right)$
for $k = 1 : n$ **do**
  $i = n - k + 1$
  $j = m - k + 1$
  $\boldsymbol{u} = \text{zeros}(m, 1)$
  $\boldsymbol{u}(1 : j) \leftarrow \boldsymbol{X}(i, 1 : j)^T$ ▷ *MATLAB notation*
  $\boldsymbol{e}_j = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ ▷ *j-th canonical basis of* $\mathbb{R}^n$
  if $|\boldsymbol{X}(i, j)| = 0$ **then**
    $\sigma = \sqrt{\boldsymbol{u}^T\boldsymbol{W}\boldsymbol{Y}\boldsymbol{u}/(\boldsymbol{Y}(j, j)\boldsymbol{W}(j, j)}$
  else
    $\sigma = (\boldsymbol{X}(i, j)/|\boldsymbol{X}(i, j)|)\sqrt{\boldsymbol{u}^T\boldsymbol{W}\boldsymbol{Y}\boldsymbol{u}/(\boldsymbol{Y}(j, j)\boldsymbol{W}(j, j)}$
  $\boldsymbol{v} = \boldsymbol{Y}\boldsymbol{u} + \sigma\boldsymbol{e}_j$
  $\mathbf{\Omega} = \boldsymbol{Y} - 2\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{W}/\left(\boldsymbol{v}^T\boldsymbol{W}\boldsymbol{Y}\boldsymbol{v}\right)$
  $\boldsymbol{X} \leftarrow \boldsymbol{X}\mathbf{\Omega}^T$
$\hat{\boldsymbol{U}} = \boldsymbol{X}(1 : n, m - n + 1 : m)$, $\hat{\boldsymbol{D}} = \boldsymbol{D}$
$\boldsymbol{\mathcal{D}} = \text{diag}(\text{diag}(\hat{\boldsymbol{U}}))$
$\tilde{\boldsymbol{U}} = \hat{\boldsymbol{U}}\boldsymbol{\mathcal{D}}^{-1}, \tilde{\boldsymbol{D}} = \boldsymbol{\mathcal{D}}\hat{\boldsymbol{D}}\boldsymbol{\mathcal{D}}^T$

IV. Fraser-Potter

As mentioned before, two main smoothing algorithms have been developed for the fixed-interval smoothing problem. This section provides three different formulations for the FP smoother. The first formulation, found in literature [14], uses the backward covariance in information form such that the amount of matrix inverses, needed to obtain the smoothed state and covariance, is reduced. The second formulation, also found in literature [14], provides a stable version to the FP smoother, by expressing the smoothed covariance as a sum of two positive definite matrices, resembling the Joseph form in the KF. The third formulation introduces a new variant of the FP smoother in a UDU-factorized framework. Note that, since the FP formulation optimally combines the outputs of a forward and a backward filter, the subscripts $_f$ and $_b$ are used to denote each, respectively.

A. Fraser-Potter Smoother

The smoothing equations for the FP smoother are given by [11, 12]

$$\boldsymbol{P}_k^\star = \left[\left(\boldsymbol{P}_{f,k}^+\right)^{-1} + \left(\boldsymbol{P}_{b,k}^-\right)^{-1}\right]^{-1}, \tag{63}$$

$$\hat{\boldsymbol{x}}_k^\star = \boldsymbol{P}_k^\star \left[\left(\boldsymbol{P}_{f,k}^+\right)^{-1} \hat{\boldsymbol{x}}_{f,k}^+ + \left(\boldsymbol{P}_{b,k}^-\right)^{-1} \hat{\boldsymbol{x}}_{b,k}^-\right]. \tag{64}$$

where $\boldsymbol{P}_k^\star$ is the smoothed covariance and $\hat{\boldsymbol{x}}_k^\star$ is the smoothed state. In this formulation, both a forward filter and a backward filter have to be used, and a total of three matrix inverses have to be performed at each time step. However, if the information formulation for the backward filter is used, i.e., instead of using the full-covariance, the information matrix is used ($\boldsymbol{\mathcal{I}}_k = (\boldsymbol{P}_k)^{-1}$), the equations can be reduced to [14]

$$\boldsymbol{P}_k^\star = (\boldsymbol{I} - \boldsymbol{\mathcal{A}}_k)\, \boldsymbol{P}_{f,k}^+, \tag{65}$$

$$\hat{\boldsymbol{x}}_k^\star = (\boldsymbol{I} - \boldsymbol{\mathcal{A}}_k)\, \hat{\boldsymbol{x}}_{f,k}^+ + \boldsymbol{P}_k^\star \boldsymbol{\mathcal{I}}_{b,k}^- \hat{\boldsymbol{x}}_{b,k}^-, \tag{66}$$

with

$$\boldsymbol{\mathcal{A}}_k = \boldsymbol{P}_{f,k}^+ \boldsymbol{\mathcal{I}}_{b,k}^- \left(\boldsymbol{I} + \boldsymbol{P}_{f,k}^+ \boldsymbol{\mathcal{I}}_{b,k}^-\right)^{-1}. \tag{67}$$

Here, only one matrix inverse has to be performed. To obtain this reduced formulation, the matrix inversion lemma is invoked twice. To see the full derivation, please refer to the appendix.

B. Fraser-Potter Stable Form Smoother

Equation (65) can be expanded to a symmetric form by adding and subtracting $\boldsymbol{\mathcal{W}}_k \boldsymbol{\mathcal{I}}_{b,k}^- \boldsymbol{P}_{f,k}^+$ [14], with

$$\boldsymbol{\mathcal{W}}_k = \boldsymbol{P}_{f,k}^+ \left(\boldsymbol{I} + \boldsymbol{P}_{f,k}^+ \boldsymbol{\mathcal{I}}_{b,k}^-\right)^{-T}, \tag{68}$$

such that

$$\boldsymbol{P}_k^{\star} = \left(\boldsymbol{I} - \boldsymbol{\mathcal{W}}_k \boldsymbol{\mathcal{I}}_{b,k}^{-}\right) \boldsymbol{P}_{f,k}^{+} \left(\boldsymbol{I} - \boldsymbol{\mathcal{W}}_k \boldsymbol{\mathcal{I}}_{b,k}^{-}\right)^T + \boldsymbol{\mathcal{W}}_k \boldsymbol{\mathcal{I}}_{b,k}^{-} \boldsymbol{\mathcal{W}}_k. \tag{69}$$

Equation (69) is the sum of two positive definite matrices, resembling the Joseph form used in the KF. Using this stable version can provide a more robust result in terms of numerical stability. In this formulation, the smoothed state is calculated as in Eq. (66).

C. Fraser-Potter UDU Smoother

Before presenting the derivation to this new UDU-factorized smoother, it is important to note that the UDU-factorized KF [4] and the information UDU-factorized KF [18] have been proposed before and are considered as resources to use in this derivation. With this in mind, the following UDU factorizations are available

$$\boldsymbol{P}_{f,k}^{+} = \boldsymbol{U}_{f,k}^{+} \boldsymbol{D}_{f,k}^{+} \boldsymbol{U}_{f,k}^{+T}, \quad \boldsymbol{\mathcal{I}}_{f,k}^{+} = \boldsymbol{\mathcal{U}}_{f,k}^{+} \boldsymbol{\mathcal{D}}_{f,k}^{+} \boldsymbol{\mathcal{U}}_{f,k}^{+T}. \tag{70}$$

In addition, assume the backward formulation of the information UDU-factorized KF is also available, that is

$$\boldsymbol{\mathcal{I}}_{b,k}^{-} = \boldsymbol{\mathcal{U}}_{b,k}^{-} \boldsymbol{\mathcal{D}}_{b,k}^{-} \left(\boldsymbol{\mathcal{U}}_{b,k}^{-}\right)^T. \tag{71}$$

The specifics on how to obtain the backward information UDU-factorized KF will be presented in the next section.

Therefore, starting with Eq. (63), the UDU formulation of the forward filter and the information UDU formulation of the backward filter can be used, such that

$$\boldsymbol{P}_k^{\star} = \left[\left(\boldsymbol{U}_{f,k}^{+} \boldsymbol{D}_{f,k}^{+} \boldsymbol{U}_{f,k}^{+T}\right)^{-1} + \boldsymbol{\mathcal{U}}_{b,k}^{-} \boldsymbol{\mathcal{D}}_{b,k}^{-} \left(\boldsymbol{\mathcal{U}}_{b,k}^{-}\right)^T\right]^{-1}. \tag{72}$$

Just as with the previous formulations of the FP smoother, the matrix inversion lemma can be used to simplify this expression. The matrix inversion lemma reads

$$\boldsymbol{F} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{C}\boldsymbol{D})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{C}^{-1}\right)^{-1} \boldsymbol{D}\boldsymbol{A}^{-1}. \tag{73}$$

Then, by letting

$$\boldsymbol{A} = \left(\boldsymbol{U}_{f,k}^{+} \boldsymbol{D}_{f,k}^{+} \boldsymbol{U}_{f,k}^{+T}\right)^{-1}, \quad \boldsymbol{B} = \boldsymbol{\mathcal{U}}_{b,k}^{-}, \quad \boldsymbol{C} = \boldsymbol{\mathcal{D}}_{b,k}^{-}, \quad \boldsymbol{D} = \left(\boldsymbol{\mathcal{U}}_{b,k}^{-}\right)^T, \tag{74}$$

Eq. (72) can be rewritten as

$$\begin{aligned}\boldsymbol{P}_k^\star &= \boldsymbol{U}_{f,k}^+\boldsymbol{D}_{f,k}^+\boldsymbol{U}_{f,k}^{+T}\\ &\quad - \boldsymbol{U}_{f,k}^+\boldsymbol{D}_{f,k}^+\boldsymbol{U}_{f,k}^{+T}\boldsymbol{\mathcal{U}}_{b,k}^-\left(\left(\boldsymbol{\mathcal{U}}_{b,k}^-\right)^T\boldsymbol{U}_{f,k}^+\boldsymbol{D}_{f,k}^+\boldsymbol{U}_{f,k}^{+T}\boldsymbol{\mathcal{U}}_{b,k}^- + \left(\boldsymbol{\mathcal{D}}_{b,k}^-\right)^{-1}\right)^{-1}\left(\boldsymbol{\mathcal{U}}_{b,k}^-\right)^T\boldsymbol{U}_{f,k}^+\boldsymbol{D}_{f,k}^+\boldsymbol{U}_{f,k}^{+T}. \qquad (75)\end{aligned}$$

To unclutter notation, let $\boldsymbol{\mathcal{B}}_k = \boldsymbol{U}_{f,k}^{+T}\boldsymbol{\mathcal{U}}_{b,k}^-$. Factorizing and re-arranging terms results in

$$\boldsymbol{P}_k^\star = \boldsymbol{U}_{f,k}^+\boldsymbol{C}_k\boldsymbol{U}_{f,k}^{+T}, \quad \boldsymbol{C}_k = \left[\boldsymbol{D}_{f,k}^+ - \boldsymbol{D}_{f,k}^+\boldsymbol{\mathcal{B}}_k\left(\boldsymbol{\mathcal{B}}_k^T\boldsymbol{D}_{f,k}^+\boldsymbol{\mathcal{B}}_k + \left(\boldsymbol{\mathcal{D}}_{b,k}^-\right)^{-1}\right)^{-1}\boldsymbol{\mathcal{B}}_k^T\boldsymbol{D}_{f,k}^+\right]. \qquad (76)$$

Taking a closer look at $\boldsymbol{C}_k$, note that it takes a familiar form, resembling a full-covariance Kalman update with

$$\boldsymbol{C}_k = \left(\boldsymbol{I} - \boldsymbol{\mathcal{E}}_k\boldsymbol{\mathcal{B}}_k^T\right)\boldsymbol{D}_{f,k}^+, \quad \boldsymbol{\mathcal{E}}_k = \boldsymbol{D}_{f,k}^+\boldsymbol{\mathcal{B}}_k\left(\boldsymbol{\mathcal{B}}_k^T\boldsymbol{D}_{f,k}^+\boldsymbol{\mathcal{B}}_k + \left(\boldsymbol{\mathcal{D}}_{b,k}^-\right)^{-1}\right)^{-1}. \qquad (77)$$

Considering this resemblance, one might be tempted to find the UDU factors of $\boldsymbol{C}_k$ using a Carlson update. However, as mentioned previously, the Carlson update is only analytically valid for the optimal Kalman gain, and certainly $\boldsymbol{\mathcal{E}}_k$ is not the optimal Kalman gain. Instead, since $\boldsymbol{\Delta}_{b,k} = (\boldsymbol{\mathcal{D}}_{b,k}^-)^{-1}$ is a diagonal matrix, the UDU factors of $\boldsymbol{C}_k$ can be found by using the Fletcher-Powell rank-1 modification. Let $\boldsymbol{\mathcal{B}}_k^{(i)} \in \mathbb{R}^n$ be the i-th column of $\boldsymbol{\mathcal{B}}_k$, and $\Delta_{b,k}^{(i,i)} \in \mathbb{R}$ be the i-th diagonal entry of $\boldsymbol{\Delta}_{b,k}$, such that the i-th column of $\boldsymbol{\mathcal{E}}_k$ becomes

$$\boldsymbol{\mathcal{E}}_k^{(i)} = c\left(\boldsymbol{D}_{f,k}^+\boldsymbol{\mathcal{B}}_k^{(i)}\right), \qquad (78)$$

with

$$c = \frac{1}{\boldsymbol{\mathcal{B}}_k^{(i)T}\boldsymbol{D}_{f,k}^+\boldsymbol{\mathcal{B}}_k^{(i)} + \Delta_{b,k}^{(i,i)}}. \qquad (79)$$

Thus, the UDU factors of $\boldsymbol{C}_k$ can be obtained by processing n Fletcher-Powell rank-1 updates sequentially, $i \in \{1, \ldots, n\}$, with

$$\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T - c\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{B}}_k^{(i)}\right)\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{B}}_k^{(i)}\right)^T, \qquad (80)$$

where $\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{D} = \boldsymbol{D}_{f,k}^+$ for the first downdate, and $\boldsymbol{\xi} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{B}}_k^{(i)}$.

Finally, by plugging in the UDU factorization of $\boldsymbol{C}_k$ in Eq. (76),

$$\boldsymbol{P}_k^\star = \boldsymbol{U}_{f,k}^+\left(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T\right)\boldsymbol{U}_{f,k}^{+T}, \qquad (81)$$

and since the product of two upper triangular matrices with ones on their diagonal is also an upper triangular matrix with ones on its diagonal,

$$\boldsymbol{U}_k^\star = \boldsymbol{U}_{f,k}^+ \tilde{\boldsymbol{U}}, \tag{82}$$

$$\boldsymbol{D}_k^\star = \tilde{\boldsymbol{D}}. \tag{83}$$

For this formulation, the smoothed state is also calculated as in Eq. (66), where $\boldsymbol{\mathcal{A}}_k$ can be easily obtained, and $\hat{\boldsymbol{\mathcal{X}}}_{b,k}^- = \boldsymbol{\mathcal{I}}_{b,k}^- \hat{\boldsymbol{x}}_{b,k}^-$ comes from the backward information UDU-factorized KF. Pseudocode for this strategy is presented in Algorithm 2. The pseudocode for the Fletcher-Powell update can be found in [23].

Algorithm 2 Fraser-Potter UDU Smoother

Input: $\hat{\boldsymbol{x}}_{f,k}^+$, $\boldsymbol{U}_{f,k}^+$, $\boldsymbol{D}_{f,k}^+$, $\hat{\boldsymbol{\mathcal{X}}}_{b,k}^-$, $\boldsymbol{\mathcal{U}}_{b,k}^-$, $\boldsymbol{\mathcal{D}}_{b,k}^-$

Output: $\hat{\boldsymbol{x}}_k^\star$, $\boldsymbol{U}_k^\star$, $\boldsymbol{D}_k^\star$

▷ *$\hat{\boldsymbol{x}}_{f,k}^+$, $\hat{\boldsymbol{\mathcal{X}}}_{b,k}^-$ are $n \times 1$* ◁

▷ *$\boldsymbol{U}_{f,k}^+$, $\boldsymbol{\mathcal{U}}_{b,k}^-$, $\boldsymbol{D}_{f,k}^+$, $\boldsymbol{\mathcal{D}}_{b,k}^-$ are $n \times n$* ◁

$\boldsymbol{\mathcal{B}}_k = (\boldsymbol{U}_{f,k}^+)^T \boldsymbol{\mathcal{U}}_{b,k}^-$

$\boldsymbol{\Delta}_{b,k} = \mathrm{diag}(1/\mathrm{diag}(\boldsymbol{D}_{b,k}))$

▷ *Smoothed covariance* ◁

$\boldsymbol{U} = \boldsymbol{I}_{n\times n}$, $\boldsymbol{D} = \boldsymbol{D}_{f,k}^+$

for $i = 1$ to n **do**

    $c = -1/\left(\boldsymbol{\mathcal{B}}_k(:,i)^T \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \boldsymbol{\mathcal{B}}_k(:,i) + \boldsymbol{\Delta}_{b,k}(i,i)\right)$ ▷ *MATLAB notation*

    $\boldsymbol{\xi} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \boldsymbol{B}_k(:,i)$

    $[\boldsymbol{U}, \boldsymbol{D}] \leftarrow \textsc{Rank1FletcherPowell}(\boldsymbol{U}, \boldsymbol{D}, c, \boldsymbol{\xi})$ ▷ *Pseudocode in [23]*

$\boldsymbol{U}_k^\star = \boldsymbol{U}_{f,k}^+ \boldsymbol{U}$, $\boldsymbol{D}_k^\star = \boldsymbol{D}$

▷ *Smoothed state* ◁

$\boldsymbol{P}_{f,k}^+ = \boldsymbol{U}_{f,k}^+ \boldsymbol{D}_{f,k}^+ (\boldsymbol{U}_{f,k}^+)^T$

$\boldsymbol{\mathcal{I}}_{b,k}^- = \boldsymbol{\mathcal{U}}_{b,k}^- \boldsymbol{\mathcal{D}}_{b,k}^- (\boldsymbol{\mathcal{U}}_{b,k}^-)^T$

$\hat{\boldsymbol{x}}_k^\star = (\boldsymbol{P}_{f,k}^+ \boldsymbol{\mathcal{I}}_{b,k}^-)(\boldsymbol{I}_n + \boldsymbol{P}_{f,k}^+ \boldsymbol{\mathcal{I}}_{b,k}^-)^{-1} \hat{\boldsymbol{x}}_{f,k} + \boldsymbol{U}_k^\star \boldsymbol{D}_k^\star (\boldsymbol{U}_k^\star)^T \hat{\boldsymbol{\mathcal{X}}}_{b,k}^-$

D. Backward Information UDU Kalman Filter

To derive the backward information KF in UDU form, note that only the backpropagation step needs to be formulated. The measurement update remains unchanged from the forward information UDU-factorized KF [18]. The derivation of the backpropagation starts from (the derivation for this equation is presented in the preliminaries section of the appendix, and follows from [14])

$$\boldsymbol{P}_{b,k}^- = \boldsymbol{\Phi}_k^{-1} \left(\boldsymbol{P}_{b,k+1}^+ + \boldsymbol{G}_k \boldsymbol{U}_{\boldsymbol{Q}} \boldsymbol{D}_{\boldsymbol{Q}} \boldsymbol{U}_{\boldsymbol{Q}}^T \boldsymbol{G}_k^T \right) \boldsymbol{\Phi}_k^{-T}, \tag{84}$$

where $\boldsymbol{U}_{\boldsymbol{Q}} \boldsymbol{D}_{\boldsymbol{Q}} \boldsymbol{U}_{\boldsymbol{Q}}^T$ are the UDU factors of $\boldsymbol{Q}_k$, such that

$$\boldsymbol{U}_{\boldsymbol{Q}} \boldsymbol{D}_{\boldsymbol{Q}} \boldsymbol{U}_{\boldsymbol{Q}}^T = \boldsymbol{Q}_k. \tag{85}$$

Expanding Eq. (84), the following form is obtained

$$\boldsymbol{P}_{b,k}^{-} = \boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{b,k+1}^{+}\boldsymbol{\Phi}_k^{-T} + \boldsymbol{\Phi}_k^{-1}\boldsymbol{G}_k\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{D}_{\boldsymbol{Q}}\boldsymbol{U}_{\boldsymbol{Q}}^T\boldsymbol{G}_k^T\boldsymbol{\Phi}_k^{-T}, \tag{86}$$

$$= \boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{b,k+1}^{+}\boldsymbol{\Phi}_k^{-T} + \overline{\boldsymbol{G}}_k\boldsymbol{D}_{\boldsymbol{Q}}\overline{\boldsymbol{G}}_k^T, \tag{87}$$

with

$$\overline{\boldsymbol{G}}_k = \boldsymbol{\Phi}_k^{-1}\boldsymbol{G}_k\boldsymbol{U}_{\boldsymbol{Q}}. \tag{88}$$

By invoking the matrix inversion lemma (Eq. (73)), with

$$\boldsymbol{A} = \boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{b,k+1}^{+}\boldsymbol{\Phi}_k^{-T}, \quad \boldsymbol{B} = \overline{\boldsymbol{G}}_k, \quad \boldsymbol{C} = \boldsymbol{D}_{\boldsymbol{Q}}, \quad \boldsymbol{D} = \overline{\boldsymbol{G}}_k^T, \tag{89}$$

Eq. (87) can be rewritten as

$$\boldsymbol{\mathcal{I}}_{b,k}^{-} = \boldsymbol{\Phi}_k^T\left[\left(\boldsymbol{I} - \boldsymbol{\mathcal{F}}_k\boldsymbol{\mathcal{G}}_k^T\right)\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\right]\boldsymbol{\Phi}_k = \boldsymbol{\Phi}_k^T\boldsymbol{\mathcal{J}}_k\boldsymbol{\Phi}_k, \tag{90}$$

with

$$\boldsymbol{\mathcal{J}}_k = \left(\boldsymbol{I} - \boldsymbol{\mathcal{F}}_k\boldsymbol{\mathcal{G}}_k^T\right)\boldsymbol{\mathcal{I}}_{b,k+1}^{+}, \quad \boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{I}}_{b,k+1}^{+}\boldsymbol{\mathcal{G}}_k\left(\boldsymbol{\mathcal{G}}_k^T\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\boldsymbol{\mathcal{G}}_k + \boldsymbol{\Delta}_{\boldsymbol{Q}}\right)^{-1}, \quad \boldsymbol{\mathcal{G}}_k = \boldsymbol{\Phi}_k\overline{\boldsymbol{G}}_k, \quad \boldsymbol{\Delta}_{\boldsymbol{Q}} = \boldsymbol{D}_{\boldsymbol{Q}}^{-1}. \tag{91}$$

Note again that $\boldsymbol{\mathcal{J}}_k$ resembles a covariance update of the form in Eq. (8). However, since $\boldsymbol{\mathcal{F}}_k$ is not the optimal Kalman gain, a Fletcher-Powell rank-1 update is performed to find its UDU factors. Following a similar procedure as in the previous section, let $\boldsymbol{\mathcal{G}}_k^{(i)} \in \mathbb{R}^n$ be the i-th column of $\boldsymbol{\mathcal{G}}_k$, and $\Delta_{\boldsymbol{Q}}^{(i,i)} \in \mathbb{R}$ be the i-th diagonal entry of $\boldsymbol{\Delta}_{\boldsymbol{Q}}$, such that the i-th column of $\boldsymbol{\mathcal{F}}_k$ becomes

$$\boldsymbol{\mathcal{F}}_k^{(i)} = c\left(\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\boldsymbol{\mathcal{G}}_k^{(i)}\right), \tag{92}$$

with

$$c = \frac{1}{\boldsymbol{\mathcal{G}}_k^{(i)T}\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\boldsymbol{\mathcal{G}}_k^{(i)} + \Delta_{\boldsymbol{Q}}^{(i,i)}}. \tag{93}$$

The UDU factors of $\boldsymbol{\mathcal{J}}_k$ are obtained by processing n Fletcher-Powell rank-1 updates sequentially, $i \in \{1, \dots, n\}$, such that

$$\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T - c\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{G}}_k^{(i)}\right)\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{G}}_k^{(i)}\right)^T, \tag{94}$$

where $\boldsymbol{U} = \boldsymbol{\mathcal{U}}_{b,k+1}^{+}$, $\boldsymbol{D} = \boldsymbol{\mathcal{D}}_{b,k+1}^{+}$ for the first downdate, and $\boldsymbol{\xi} = \left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{G}}_k^{(i)}\right)$. Finally, either the MWGS or the WHHR algorithm (Remark III.4) can be used to find a factorization such that

$$\boldsymbol{\mathcal{U}}_{b,k}^{-}\boldsymbol{\mathcal{D}}_{b,k}^{-}\left(\boldsymbol{\mathcal{U}}_{b,k}^{-}\right)^T = \boldsymbol{\Phi}_k^T\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T\boldsymbol{\Phi}_k. \tag{95}$$

For the backpropagation of the state, note that [14]

$$\hat{\boldsymbol{x}}_{b,k}^{-} = \boldsymbol{\Phi}_k^{-1}\hat{\boldsymbol{x}}_{b,k+1}^{+}. \tag{96}$$

Using the information formulation ($\hat{\boldsymbol{\mathcal{X}}}_{b,k}^{-} = \boldsymbol{\mathcal{I}}_{b,k}^{-}\hat{\boldsymbol{x}}_{b,k}^{-}$),

$$\hat{\boldsymbol{\mathcal{X}}}_{b,k}^{-} = \boldsymbol{\mathcal{I}}_{b,k}^{-}\boldsymbol{\Phi}_k^{-1}\left(\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\right)^{-1}\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{+}, \tag{97}$$

$$= \boldsymbol{\Phi}_k^{T}\left[\left(\boldsymbol{I} - \boldsymbol{\mathcal{F}}_k\boldsymbol{\mathcal{G}}_k^{T}\right)\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\right]\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{-1}\left(\boldsymbol{\mathcal{I}}_{b,k+1}^{+}\right)^{-1}\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{+}, \tag{98}$$

$$= \boldsymbol{\Phi}_k^{T}\left(\boldsymbol{I} - \boldsymbol{\mathcal{F}}_k\boldsymbol{\mathcal{G}}_k^{T}\right)\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{+}. \tag{99}$$

Pseudocode for the the backward information UDU KF is presented in Algorithm 3.

Algorithm 3 Backward Information UDU KF

Input: $\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{-}$, $\boldsymbol{\mathcal{U}}_{b,k+1}^{-}$, $\boldsymbol{\mathcal{D}}_{b,k+1}^{-}$, $\boldsymbol{\Phi}_k$, $\boldsymbol{y}_{k+1}$, $\boldsymbol{H}_{k+1}$, $\boldsymbol{G}_k$, $\boldsymbol{\mathcal{U}}_{\boldsymbol{R}}$, $\boldsymbol{\mathcal{D}}_{\boldsymbol{R}}$, $\boldsymbol{U}_{\boldsymbol{Q}}$, $\boldsymbol{D}_{\boldsymbol{Q}}$
Output: $\hat{\boldsymbol{\mathcal{X}}}_{b,k}^{-}$, $\boldsymbol{\mathcal{U}}_{b,k}^{-}$, $\boldsymbol{\mathcal{D}}_{b,k}^{-}$

▷ *$\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{-}$ is $n \times 1$, $\boldsymbol{y}_{k+1}$ is $m \times 1$* ◁
▷ *$\boldsymbol{\mathcal{U}}_{b,k+1}^{-}$, $\boldsymbol{\mathcal{D}}_{b,k+1}^{-}$ are $n \times n$, $\boldsymbol{\Phi}_k$ is $n \times n$, $\boldsymbol{H}_k$ is $m \times n$, $\boldsymbol{G}_k$ is $n \times p$* ◁
▷ *$\boldsymbol{\mathcal{U}}_{\boldsymbol{R}}\boldsymbol{\mathcal{D}}_{\boldsymbol{R}}\boldsymbol{\mathcal{U}}_{\boldsymbol{R}}^{T} = \boldsymbol{R}_{k+1}^{-1} \in \mathbb{R}^{m\times m}$, $\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{D}_{\boldsymbol{Q}}\boldsymbol{U}_{\boldsymbol{Q}}^{T} = \boldsymbol{Q}_k \in \mathbb{R}^{p\times p}$* ◁
▷ *Update* ◁
$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_{b,k+1}^{-}$, $\boldsymbol{D} = \boldsymbol{\mathcal{D}}_{b,k+1}^{-}$
$\boldsymbol{U}_{\boldsymbol{H}} = \boldsymbol{U}_{\boldsymbol{R}}^{T}\boldsymbol{H}_{k+1}$
for $i = 1$ to m **do**
  $c = \boldsymbol{\mathcal{D}}_{\boldsymbol{R}}(i,i)$ ▷ *MATLAB notation*
  $\boldsymbol{\xi} = \boldsymbol{U}_{\boldsymbol{H}}(i,:)^{T}$
  $[\boldsymbol{U},\boldsymbol{D}] \leftarrow$ RANK1FLETCHERPOWELL$(\boldsymbol{U},\boldsymbol{D},c,\boldsymbol{\xi})$ ▷ *Pseudocode in [23]*
$\boldsymbol{\mathcal{U}}_{b,k+1}^{+} = \boldsymbol{U}$, $\boldsymbol{\mathcal{D}}_{b,k+1}^{+} = \boldsymbol{D}$
$\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{+} = \hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{-} + \boldsymbol{H}_{k+1}^{T}(\boldsymbol{\mathcal{U}}_{\boldsymbol{R}}\boldsymbol{\mathcal{D}}_{\boldsymbol{R}}\boldsymbol{\mathcal{U}}_{\boldsymbol{R}}^{T})\boldsymbol{y}_{k+1}$
▷ *Propagate* ◁
$\boldsymbol{\Delta}_{\boldsymbol{Q}} = \mathrm{diag}(1/\mathrm{diag}(\boldsymbol{D}_{\boldsymbol{Q}}))$
$\boldsymbol{\mathcal{G}}_k = \boldsymbol{G}_k\boldsymbol{U}_{\boldsymbol{Q}}$
$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_{b,k+1}^{+}$, $\boldsymbol{D} = \boldsymbol{\mathcal{D}}_{b,k+1}^{+}$
for $i = 1$ to n **do**
  $c = -1/(\boldsymbol{\mathcal{G}}_k(:,i)^{T}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{T}\boldsymbol{\mathcal{G}}_k(:,i)) + \boldsymbol{\Delta}_{\boldsymbol{Q}}(i,i)$
  $\boldsymbol{\xi} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{T}\boldsymbol{\mathcal{G}}_k(:,i)$
  $[\boldsymbol{U},\boldsymbol{D}] \leftarrow$ RANK1FLETCHERPOWELL$(\boldsymbol{U},\boldsymbol{D},c,\boldsymbol{\xi})$ ▷ *Pseudocode in [23]*
$[\boldsymbol{U},\boldsymbol{D}] \leftarrow \mathrm{WHHR}(\boldsymbol{\Phi}_k^{T}\boldsymbol{U},\boldsymbol{D},[\,],[\,],[\,],[\,])$ ▷ *Algorithm 1*
$\boldsymbol{\mathcal{U}}_{b,k}^{-} = \boldsymbol{U}$, $\boldsymbol{\mathcal{D}}_{b,k}^{-} = \boldsymbol{D}$
$\boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{U}}_{b,k+1}^{+}\boldsymbol{\mathcal{D}}_{b,k+1}^{+}\boldsymbol{\mathcal{U}}_{b,k+1}^{+}\boldsymbol{\mathcal{G}}_k/(\boldsymbol{\mathcal{G}}_k^{T}\boldsymbol{\mathcal{U}}_{b,k+1}^{+}\boldsymbol{\mathcal{D}}_{b,k+1}^{+}\boldsymbol{\mathcal{U}}_{b,k+1}^{+}\boldsymbol{\mathcal{G}}_k + \boldsymbol{\Delta}_{\boldsymbol{Q}})$
$\hat{\boldsymbol{\mathcal{X}}}_{b,k}^{-} = \boldsymbol{\Phi}_k^{T}(\boldsymbol{I}_{n\times n} - \boldsymbol{\mathcal{F}}_k\boldsymbol{\mathcal{G}}_k)\hat{\boldsymbol{\mathcal{X}}}_{b,k+1}^{+}$

V. Rauch-Tung-Striebel

This section provides three different formulations for the second smoother considered in this work, the RTS smoother. The first formulation follows the traditional equations, obtained by expressing the backward covariance of the FP smoother in terms of the forward and smoothed covariances [14] (please see the proof in the appendix for details). Analogous to the stable FP smoother, the second formulation in this section introduces a new stable version of the RTS smoother, by expressing the smoothed covariance as a sum of two positive definite matrices (similar to the proposed RTS smoother in [17]). Finally, the third formulation derives a new version of the RTS smoother in a UDU-factorized framework. For this section, the forward and backward subscripts are dropped, as only the forward filter is used.

A. Rauch-Tung-Striebel Smoother

The smoothing equations for the RTS smoother are given by [13]

$$\boldsymbol{P}_k^\star = \boldsymbol{P}_k^+ - \boldsymbol{P}_k^+ \boldsymbol{\Phi}_k^T \left(\boldsymbol{P}_{k+1}^-\right)^{-1} \left(\boldsymbol{P}_{k+1}^- - \boldsymbol{P}_{k+1}^\star\right) \left(\boldsymbol{P}_{k+1}^-\right)^{-1} \boldsymbol{\Phi}_k \boldsymbol{P}_k^+, \tag{100}$$

$$\hat{\boldsymbol{x}}_k^\star = \hat{\boldsymbol{x}}_k^+ + \boldsymbol{P}_k^+ \boldsymbol{\Phi}_k^T \left(\boldsymbol{P}_{k+1}^-\right)^{-1} \left(\hat{\boldsymbol{x}}_{k+1}^\star - \hat{\boldsymbol{x}}_{k+1}^-\right). \tag{101}$$

For ease of notation, let

$$\boldsymbol{\mathcal{K}}_k = \boldsymbol{P}_k^+ \boldsymbol{\Phi}_k^T \left(\boldsymbol{P}_{k+1}^-\right)^{-1}, \tag{102}$$

such that

$$\boldsymbol{P}_k^\star = \boldsymbol{P}_k^+ - \boldsymbol{\mathcal{K}}_k \left(\boldsymbol{P}_{k+1}^- - \boldsymbol{P}_{k+1}^\star\right) \boldsymbol{\mathcal{K}}_k^T \tag{103}$$

$$\hat{\boldsymbol{x}}_k^\star = \hat{\boldsymbol{x}}_k^+ + \boldsymbol{\mathcal{K}}_k \left(\hat{\boldsymbol{x}}_{k+1}^\star - \hat{\boldsymbol{x}}_{k+1}^-\right). \tag{104}$$

Two things are worthy of discussion from these equations. First, note that the backward filter is no longer used. Second, the smoothed state is independent from the smoothed covariance. These two characteristics make the RTS smoother a more common approach from a practical standpoint [14].

B. Rauch-Tung-Striebel Stable Form Smoother

The stable form of the RTS smoother is obtained by expressing the prior covariance at time step $k+1$ in terms of the posterior covariance at time step k, that is

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\Phi}_k \boldsymbol{P}_k^+ \boldsymbol{\Phi}_k^T + \boldsymbol{G}_k \boldsymbol{Q}_k \boldsymbol{G}_k^T. \tag{105}$$

Therefore, Eq. (100) can be expressed as

$$\boldsymbol{P}_k^\star = \boldsymbol{P}_k^+ - \boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T \left(\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1} \left(\boldsymbol{P}_{k+1}^- - \boldsymbol{P}_{k+1}^\star\right) \left(\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1} \boldsymbol{\Phi}_k\boldsymbol{P}_k^+. \tag{106}$$

Factoring out $\boldsymbol{\Phi}_k^{-1}$ from both sides

$$\begin{aligned}\boldsymbol{P}_k^\star = {} & \boldsymbol{\Phi}_k^{-1}\left(\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T\right)\boldsymbol{\Phi}_k^{-T} \\ & - \boldsymbol{\Phi}_k^{-1}\left[\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T\left(\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T\right]\boldsymbol{\Phi}_k^{-T} \\ & + \boldsymbol{\Phi}_k^{-1}\left[\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T\left(\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\left(\boldsymbol{P}_{k+1}^\star\right)\left(\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T\right]\boldsymbol{\Phi}_k^{-T}.\end{aligned} \tag{107}$$

To unclutter notation, let $\boldsymbol{\mathcal{M}}_k^{-1} = \boldsymbol{\Phi}_k\boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T$, such that

$$\begin{aligned}\boldsymbol{P}_k^\star = {} & \boldsymbol{\Phi}_k^{-1}\boldsymbol{\mathcal{M}}_k^{-1}\boldsymbol{\Phi}_k^{-T} - \boldsymbol{\Phi}_k^{-1}\left[\boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{M}}_k^{-1} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\boldsymbol{\mathcal{M}}_k^{-1}\right]\boldsymbol{\Phi}_k^{-T} \\ & + \boldsymbol{\Phi}_k^{-1}\left[\boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{M}}_k^{-1} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\left(\boldsymbol{P}_{k+1}^\star\right)\left(\boldsymbol{\mathcal{M}}_k^{-1} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\boldsymbol{\mathcal{M}}_k^{-1}\right]\boldsymbol{\Phi}_k^{-T}.\end{aligned} \tag{108}$$

In case that the process noise covariance is not diagonal, its UDU factorization can be used, where

$$\boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T = \left(\boldsymbol{G}_k\boldsymbol{U}_{\boldsymbol{Q}}\right)\boldsymbol{D}_{\boldsymbol{Q}}\left(\boldsymbol{G}_k\boldsymbol{U}_{\boldsymbol{Q}}\right)^T = \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{D}_{\boldsymbol{Q}}\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T. \tag{109}$$

At this point, the matrix inversion lemma can be used to simplify the expression, such that

$$\boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{M}}_k^{-1} + \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{D}_{\boldsymbol{Q}}\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\right)^{-1} = \boldsymbol{I} - \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k, \tag{110}$$

where

$$\boldsymbol{\mathcal{N}}_k = \left(\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}} + \boldsymbol{D}_{\boldsymbol{Q}}^{-1}\right)^{-1}. \tag{111}$$

With this, Eq. (108) can be simplified to

$$\boldsymbol{P}_k^\star = \boldsymbol{\Phi}_k^{-1}\left[\left(\boldsymbol{I} - \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right)\boldsymbol{P}_{k+1}^\star\left(\boldsymbol{I} - \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right)^T + \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\right]\boldsymbol{\Phi}_k^{-T}. \tag{112}$$

By defining $\overline{\boldsymbol{G}}_k = \boldsymbol{\Phi}_k^{-1}\overline{\boldsymbol{G}}_{\boldsymbol{Q}}$ and $\overline{\boldsymbol{L}}_k = \left(\boldsymbol{P}_k^+\right)^{-1}\overline{\boldsymbol{G}}_k\boldsymbol{\mathcal{N}}_k$, the desired Joseph-like form is obtained, where

$$\boldsymbol{P}_k^\star = \left(\boldsymbol{I} - \overline{\boldsymbol{G}}_k\overline{\boldsymbol{L}}_k^T\right)\boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{k+1}^\star\boldsymbol{\Phi}_k^{-T}\left(\boldsymbol{I} - \overline{\boldsymbol{G}}_k\overline{\boldsymbol{L}}_k^T\right)^T + \overline{\boldsymbol{G}}_k\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_k^T. \tag{113}$$

To find the smoothed state, the prior covariance and state estimate at time step $k+1$ are once again expressed in terms of the posterior covariance and state estimate at time step k. Therefore

$$\hat{\boldsymbol{x}}_k^\star = \hat{\boldsymbol{x}}_k^+ + \boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T \left(\boldsymbol{\mathcal{M}}_k^{-1} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1} \left(\hat{\boldsymbol{x}}_{k+1}^\star - \boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_k^+\right) \tag{114}$$

$$= \hat{\boldsymbol{x}}_k^+ + \boldsymbol{P}_k^+\boldsymbol{\Phi}_k^T \left(\boldsymbol{\mathcal{M}}_k - \boldsymbol{\mathcal{M}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right) \left(\hat{\boldsymbol{x}}_{k+1}^\star - \boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_k^+\right) \tag{115}$$

$$= \boldsymbol{\Phi}_k^{-1}\boldsymbol{\mathcal{M}}_k^{-1} \left(\boldsymbol{\mathcal{M}}_k - \boldsymbol{\mathcal{M}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right)\hat{\boldsymbol{x}}_{k+1}^\star + \left(\boldsymbol{I} - \boldsymbol{\Phi}_k^{-1}\boldsymbol{\mathcal{M}}_k^{-1}\left(\boldsymbol{\mathcal{M}}_k - \boldsymbol{\mathcal{M}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right)\boldsymbol{\Phi}_k\right)\hat{\boldsymbol{x}}_k^+ \tag{116}$$

$$= \boldsymbol{\Phi}_k^{-1}\left[\left(\boldsymbol{I} - \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right)\hat{\boldsymbol{x}}_{k+1}^\star + \left(\boldsymbol{I} - \left(\boldsymbol{I} - \overline{\boldsymbol{G}}_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\right)\right)\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_k^+\right]. \tag{117}$$

Recalling the definitions of $\overline{\boldsymbol{G}}_k$ and $\overline{\boldsymbol{L}}_k$, Eq. (117) can be rewritten as

$$\hat{\boldsymbol{x}}_k^\star = \left(\boldsymbol{I} - \overline{\boldsymbol{G}}_k\overline{\boldsymbol{L}}_k^T\right)\boldsymbol{\Phi}_k^{-1}\hat{\boldsymbol{x}}_{k+1}^\star + \overline{\boldsymbol{G}}_k\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_k^T\left(\boldsymbol{P}_k^+\right)^{-1}\hat{\boldsymbol{x}}_k^+. \tag{118}$$

In both, Eq. (113) and Eq. (118), the inverse of the state covariance is used. Therefore, this new variant is used with the information formulation of the KF, as shown in Algorithm 4. It is important to note that Eq. (113) and Eq. (118) resemble the equations obtained by Watanabe and Tzafestas [17]. However, since the process noise covariance is not assumed to be diagonal in this derivation, the equations only share the same structure.

Algorithm 4 Rauch-Tung-Striebel Stable Form Smoother

Input: $\hat{\boldsymbol{\mathcal{X}}}_k^+$, $\boldsymbol{\mathcal{I}}_k^+$, $\hat{\boldsymbol{x}}_{k+1}^\star$, $\boldsymbol{P}_{k+1}^\star$, $\boldsymbol{\Phi}_k$, $\boldsymbol{G}_k$, $\boldsymbol{U}_{\boldsymbol{Q}}$, $\boldsymbol{D}_{\boldsymbol{Q}}$
Output: $\hat{\boldsymbol{x}}_k^\star$, $\boldsymbol{U}_k^\star$, $\boldsymbol{D}_k^\star$

▷ *$\hat{\boldsymbol{\mathcal{X}}}_k^+, \hat{\boldsymbol{x}}_{k+1}^\star$ are $n \times 1$* ◁
▷ *$\boldsymbol{\mathcal{I}}_k^+, \boldsymbol{P}_{k+1}^\star, \boldsymbol{\Phi}_k$ are $n \times n$* ◁
▷ *$\boldsymbol{G}_k$ is $n \times p$ and $\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{D}_{\boldsymbol{Q}}\boldsymbol{U}_{\boldsymbol{Q}}^T = \boldsymbol{Q}_k \in \mathbb{R}^{p\times p}$* ◁
$\overline{\boldsymbol{G}}_{\boldsymbol{Q}} = \boldsymbol{G}_k\boldsymbol{U}_{\boldsymbol{Q}}$
$\overline{\boldsymbol{G}}_k = \boldsymbol{\Phi}_k^{-1}\overline{\boldsymbol{G}}_{\boldsymbol{Q}}$
$\boldsymbol{\mathcal{M}}_k = \boldsymbol{\Phi}_k^{-T}\boldsymbol{\mathcal{I}}_k^+\boldsymbol{\Phi}_k^{-1}$
$\Delta_{\boldsymbol{Q}} = \mathrm{diag}(1/\mathrm{diag}(\boldsymbol{D}_{\boldsymbol{Q}}))$
$\boldsymbol{\mathcal{N}}_k = (\overline{\boldsymbol{G}}_{\boldsymbol{Q}}^T\boldsymbol{\mathcal{M}}_k\overline{\boldsymbol{G}}_{\boldsymbol{Q}} + \Delta_{\boldsymbol{Q}})^{-1}$
$\overline{\boldsymbol{L}}_k = \boldsymbol{\mathcal{I}}_k^+\overline{\boldsymbol{G}}_k\boldsymbol{\mathcal{N}}_k$
▷ *Smoothed covariance* ◁
$\boldsymbol{P}_k^\star = (\boldsymbol{I}_{n\times n} - \overline{\boldsymbol{G}}_k\overline{\boldsymbol{L}}_k^T)\boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{k+1}^\star\boldsymbol{\Phi}_k^{-T}(\boldsymbol{I}_{n\times n} - \overline{\boldsymbol{G}}_k\overline{\boldsymbol{L}}_k^T)^T + \overline{\boldsymbol{G}}_k\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_k^T$
▷ *Smoothed state* ◁
$\hat{\boldsymbol{x}}_k^\star = (\boldsymbol{I}_{n\times n} - \overline{\boldsymbol{G}}_k\overline{\boldsymbol{L}}_k^T)\boldsymbol{\Phi}_k^{-1}\hat{\boldsymbol{x}}_{k+1}^\star + \overline{\boldsymbol{G}}_k\boldsymbol{\mathcal{N}}_k\overline{\boldsymbol{G}}_k^T\hat{\boldsymbol{\mathcal{X}}}_k^+$

C. Rauch-Tung-Striebel UDU Smoother

To derive the UDU formulation of the RTS smoother, Eq. (100) is first rewritten as

$$\boldsymbol{P}_k^\star = \boldsymbol{P}_k^+ - \boldsymbol{\mathcal{K}}_k\left(\boldsymbol{P}_{k+1}^- - \boldsymbol{P}_{k+1}^\star\right)\boldsymbol{\mathcal{K}}_k^T, \tag{119}$$

$$= \boldsymbol{\mathcal{K}}_k\left(\boldsymbol{P}_{k+1}^\star - \boldsymbol{P}_{k+1}^- + \boldsymbol{\mathcal{K}}_k^{-1}\boldsymbol{P}_k^+\boldsymbol{\mathcal{K}}_k^{-T}\right)\boldsymbol{\mathcal{K}}_k^T. \tag{120}$$

Note that,

$$\mathcal{K}_k^{-1} = \boldsymbol{P}_{k+1}^{-}\boldsymbol{\Phi}_k^{-T}\left(\boldsymbol{P}_k^{+}\right)^{-1}, \tag{121}$$

which makes Eq. (120) equal to

$$\boldsymbol{P}_k^{\star} = \mathcal{K}_k\left(\boldsymbol{P}_{k+1}^{\star} - \boldsymbol{P}_{k+1}^{-} + \boldsymbol{P}_{k+1}^{-}\boldsymbol{\Phi}_k^{-T}\left(\boldsymbol{P}_k^{+}\right)^{-1}\boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{k+1}^{-}\right)\mathcal{K}_k^{T}. \tag{122}$$

This is the starting form for the derivation of the RTS smoother in UDU form.

Just as with the FP, assume that the UDU formulation of the prior and posterior covariance are available

$$\boldsymbol{P}_k^{+} = \boldsymbol{U}_k^{+}\boldsymbol{D}_k^{+}\boldsymbol{U}_k^{+T}, \quad \boldsymbol{P}_{k+1}^{-} = \boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T}. \tag{123}$$

Substituting the UDU factorization for the both the prior and posterior covariance in Eq. (122) yields

$$\boldsymbol{P}_k^{\star} = \mathcal{K}_k\left[\boldsymbol{P}_{k+1}^{\star} - \boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T} + \boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T}\boldsymbol{\Phi}_k^{-T}\left(\boldsymbol{U}_k^{+}\boldsymbol{D}_k^{+}\boldsymbol{U}_k^{+T}\right)^{-1}\boldsymbol{\Phi}_k^{-1}\boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T}\right]\mathcal{K}_k^{T}. \tag{124}$$

In this case, the inverses of the UDU factors of the posterior covariance have to be obtained. This is of no concern as the inverse of the UDU factorization can be easily computed as [18]:

$$\left(\boldsymbol{U}_k^{+}\boldsymbol{D}_k^{+}\boldsymbol{U}_k^{+T}\right)^{-1} = \left(\boldsymbol{U}_k^{+}\right)^{-T}\left(\boldsymbol{D}_k^{+}\right)^{-1}\left(\boldsymbol{U}_k^{+}\right)^{-1} = \boldsymbol{V}_k^{+T}\boldsymbol{O}_k^{+}\boldsymbol{V}_k^{+}, \tag{125}$$

where calculating $\boldsymbol{O}_k^{+}$ only requires n scalar divisions, and $\boldsymbol{V}_k^{+}$ can be found by solving

$$\boldsymbol{V}_k^{+(i,j)} = -\left(\boldsymbol{U}_k^{+(i,j)} + \sum_{l=i+1}^{j-1}\boldsymbol{U}_k^{+(i,l)}\boldsymbol{V}_k^{+(l,j)}\right), \quad j = n, \ldots, 2, \quad i = j-1, \ldots, 1. \tag{126}$$

Therefore, by substituting the inverse

$$\boldsymbol{P}_k^{\star} = \mathcal{K}_k\left[\boldsymbol{P}_{k+1}^{\star} - \boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T} + \left(\boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T}\boldsymbol{\Phi}_k^{-T}\boldsymbol{V}_k^{+T}\right)\boldsymbol{O}_k^{+}\left(\boldsymbol{V}_k^{+}\boldsymbol{\Phi}_k^{-1}\boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T}\right)\right]\mathcal{K}_k^{T}. \tag{127}$$

To unclutter notation, let $\boldsymbol{\mathcal{S}}_k = \boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T}\boldsymbol{\Phi}_k^{-T}\boldsymbol{V}_k^{+T}$, such that

$$\boldsymbol{P}_k^{\star} = \mathcal{K}_k\left(\boldsymbol{P}_{k+1}^{\star} - \boldsymbol{U}_{k+1}^{-}\boldsymbol{D}_{k+1}^{-}\left(\boldsymbol{U}_{k+1}^{-}\right)^{T} + \boldsymbol{\mathcal{S}}_k\boldsymbol{O}_k^{+}\boldsymbol{\mathcal{S}}_k^{T}\right)\mathcal{K}_k^{T}. \tag{128}$$

Using the UDU factorization for $\boldsymbol{P}^\star_{k+1} = \boldsymbol{U}^\star_{k+1}\boldsymbol{D}^\star_{k+1}\boldsymbol{U}^{\star T}_{k+1}$ (coming from the previous iteration)

$$\boldsymbol{P}^\star_k = \mathcal{K}_k \left(\boldsymbol{U}^\star_{k+1}\boldsymbol{D}^\star_{k+1}\boldsymbol{U}^{\star T}_{k+1} - \boldsymbol{U}^-_{k+1}\boldsymbol{D}^-_{k+1}\left(\boldsymbol{U}^-_{k+1}\right)^T + \boldsymbol{\mathcal{S}}_k\boldsymbol{O}^+_k\boldsymbol{\mathcal{S}}^T_k\right)\mathcal{K}^T_k. \tag{129}$$

With this form, note that the UDU factorization of the inner parenthesis can be found by using the WHHR (Remark III.3), such that

$$\boldsymbol{P}^\star_k = \mathcal{K}_k\left(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T\right)\mathcal{K}^T_k. \tag{130}$$

And finally, either the MWGS or the WHHR algorithm (Remark III.4) can be used to find the final UDU factorization, where

$$\boldsymbol{U}^\star_k\boldsymbol{D}^\star_k\boldsymbol{U}^{\star T}_k = \mathcal{K}_k\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{U}}^T\mathcal{K}^T_k. \tag{131}$$

In this formulation, the smoothed state is found by using Eq. (101), with

$$\mathcal{K}_k = \boldsymbol{U}^+_k\boldsymbol{D}^+_k\boldsymbol{U}^{+T}_k\boldsymbol{\Phi}^T_k\boldsymbol{V}^{-T}_{k+1}\boldsymbol{O}^-_{k+1}\boldsymbol{V}^-_{k+1}, \tag{132}$$

$$\hat{\boldsymbol{x}}^\star_k = \hat{\boldsymbol{x}}^+_k + \boldsymbol{U}^+_k\boldsymbol{D}^+_k\boldsymbol{U}^{+T}_k\boldsymbol{\Phi}^T_k\boldsymbol{V}^{-T}_{k+1}\boldsymbol{O}^-_{k+1}\boldsymbol{V}^-_{k+1}\left(\hat{\boldsymbol{x}}^\star_{k+1} - \hat{\boldsymbol{x}}^-_{k+1}\right). \tag{133}$$

The pseudocode for implementing this smoother can be found in Algorithm 5.

Algorithm 5 Rauch-Tung-Striebel UDU Smoother

Input: $\hat{\boldsymbol{x}}^+_k$, $\boldsymbol{U}^+_k$, $\boldsymbol{D}^+_k$, $\hat{\boldsymbol{x}}^-_{k+1}$, $\boldsymbol{U}^-_{k+1}$, $\boldsymbol{D}^-_{k+1}$, $\hat{\boldsymbol{x}}^\star_{k+1}$, $\boldsymbol{U}^\star_{k+1}$, $\boldsymbol{D}^\star_{k+1}$, $\boldsymbol{\Phi}_k$
Output: $\hat{\boldsymbol{x}}^\star_k$, $\boldsymbol{U}^\star_k$, $\boldsymbol{D}^\star_k$

▷ $\hat{\boldsymbol{x}}^+_k, \hat{\boldsymbol{x}}^-_{k+1}, \hat{\boldsymbol{x}}^\star_{k+1}$ *are* $n \times 1$ ◁
▷ $\boldsymbol{U}^+_k, \boldsymbol{U}^-_{k+1}, \boldsymbol{U}^\star_{k+1}, \boldsymbol{D}^+_k, \boldsymbol{D}^-_{k+1}, \boldsymbol{D}^\star_{k+1}, \boldsymbol{\Phi}_k$ *are* $n \times n$ ◁
$\boldsymbol{V}^+_k = (\boldsymbol{U}^+_k)^{-1}$, $\boldsymbol{O}^+_k = (\boldsymbol{D}^+_k)^{-1}$
$\boldsymbol{V}^-_{k+1} = (\boldsymbol{U}^-_{k+1})^{-1}$, $\boldsymbol{O}^+_{k+1} = (\boldsymbol{D}^+_k)^{-1}$
$\boldsymbol{\mathcal{S}}_k = \boldsymbol{U}^-_{k+1}\boldsymbol{D}^-_{k+1}(\boldsymbol{U}^-_{k+1})^T\boldsymbol{\Phi}^{-T}_k\boldsymbol{V}^{+T}_k$
▷ *Smoothed covariance* ◁
$\boldsymbol{U} = \boldsymbol{U}^\star_{k+1}$, $\boldsymbol{D} = \boldsymbol{D}^\star_{k+1}$
$[\boldsymbol{U}, \boldsymbol{D}] \leftarrow \mathrm{WHHR}(\boldsymbol{U}, \boldsymbol{D}, \boldsymbol{U}^-_{k+1}, \boldsymbol{D}^-_{k+1}, \boldsymbol{\mathcal{S}}_k, \boldsymbol{O}^+_k)$ ▷ *Algorithm 1*
$\mathcal{K}_k = \boldsymbol{U}^+_k\boldsymbol{D}^+_k\boldsymbol{U}^{+T}_k\boldsymbol{\Phi}^T_k\boldsymbol{V}^{+T}_k\boldsymbol{O}^-_{k+1}\boldsymbol{V}^-_{k+1}$
$[\boldsymbol{U}, \boldsymbol{D}] \leftarrow \mathrm{WHHR}(\mathcal{K}_k\boldsymbol{U}, \boldsymbol{D}, [\,], [\,], [\,], [\,])$ ▷ *Algorithm 1*
$\boldsymbol{U}^\star_k = \boldsymbol{U}$, $\boldsymbol{D}^\star_k = \boldsymbol{D}$
▷ *Smoothed state* ◁
$\hat{\boldsymbol{x}}^\star_k = \hat{\boldsymbol{x}}^+_k + \mathcal{K}_k(\hat{\boldsymbol{x}}^\star_{k+1} - \hat{\boldsymbol{x}}^-_{k+1})$

VI. Numerical Example

This section provides a numerical example where all six smoother formulations presented are studied and compared. The purpose of this section is to evaluate the numerical equivalency of the different approaches. For ease of notation consider the following abbreviations:

- Fraser-Potter Smoother (Section IV.A) → FP
- Fraser-Potter Stable Form Smoother (Section IV.B) → FS
- Fraser-Potter UDU Smoother (Section IV.C) → FU
- Rauch-Tung-Striebel Smoother (Section V.A) → RT
- Rauch-Tung-Striebel Stable Form Smoother (Section V.B) → RS
- Rauch-Tung-Striebel UDU Smoother (Section V.C) → RU

To this end, consider the following system [18],

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{q}_k, \tag{134}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{\eta}_k, \tag{135}$$

with

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{I}_{2\times2} & \Delta t_k \boldsymbol{I}_{2\times2} \\ \boldsymbol{B}_k & \boldsymbol{I}_{2\times2} \end{bmatrix}, \quad \boldsymbol{B}_k = 0.1 \begin{bmatrix} \sin(t_k) - \sin(t_{k-1}) & -\cos(t_k) + \cos(t_{k-1}) \\ 0 & \sin(t_k) - \sin(t_{k-1}) \end{bmatrix}, \tag{136}$$

$$\boldsymbol{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{137}$$

$$\mathbb{E}(\boldsymbol{q}_k) = \boldsymbol{0}, \quad \mathbb{E}\left(\boldsymbol{q}_k \boldsymbol{q}_k^T\right) = \boldsymbol{Q}_k = \begin{bmatrix} 0.01\boldsymbol{I}_{2\times2} & 0.00 \\ 0.00 & 0.01\boldsymbol{I}_{2\times2} \end{bmatrix}, \tag{138}$$

$$\mathbb{E}(\boldsymbol{\eta}_k) = \boldsymbol{0}, \quad \mathbb{E}\left(\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T\right) = \boldsymbol{R}_k = \begin{bmatrix} 2.96 & 2.80 \\ 2.80 & 2.96 \end{bmatrix}, \tag{139}$$

and $\mathbb{E}\left(\boldsymbol{q}_k \boldsymbol{\eta}_k^T\right) = \boldsymbol{0}$.

To assess the performance of each smoother, 1000 Monte Carlo (MC) runs are performed by first solving the forward filtering problem for 100 time steps with $\Delta t_k = 1$, using both the KF and the UDU formulation of the KF. Each run starts from an initial condition defined as

$$\boldsymbol{x}_0 \sim \mathcal{N}\left(\boldsymbol{0}_{4\times1}, \boldsymbol{I}_{4\times4}\right). \tag{140}$$

Figure 1 shows the estimation error as a function of time obtained with the KF (the UDU-factorized KF formulation yields identical results). As it can be seen, the estimation error is unbiased and the filter's predicted covariance matches with the sample variance.

After performing the filtering problem, each of the smoothers discussed are used to find a smoothed state and covariance. The performance of the smoothers is evaluated using the time-averaged root mean square error (RMSE)

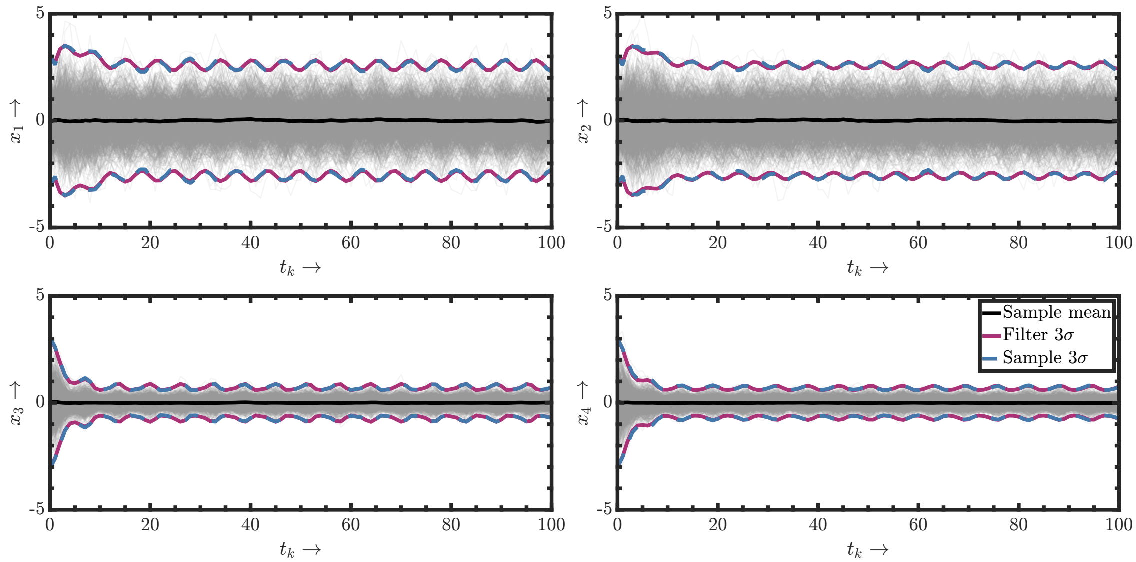


Fig. 1 Filtering error as a function of time steps. The gray lines show the MC runs.

between the smoothed state and the true state, and a comparison of the smoothed covariance matrices. In this work, the RMSE is defined as

$$\text{RMSE} = \sum_{k=1}^{n_T} \frac{1}{n_T} \sum_{j=1}^{n_{\text{MC}}} \frac{1}{n_{\text{MC}}} \sqrt{\frac{1}{n_x} \sum_{i=1}^{n_x} \left(\boldsymbol{x}_{k,j}^{(i)} - \hat{\boldsymbol{x}}_{k,j}^{\star(i)} \right)^2}, \tag{141}$$

where $n_T = 100$ is the total time steps, $n_{\text{MC}} = 1000$ is the total MC runs, and $n_x = 4$ is the state dimension. The obtained RMSEs for each smoother are then compared by calculating their relative difference, i.e. $|\text{RMSE}_i - \text{RMSE}_j|$ where i, j refers to the different smoothers.

To compare the smoothed covariance matrices, the covariance sequence of each smoother is first averaged over time and MC runs to obtain an averaged covariance matrix,

$$\bar{\boldsymbol{P}}^{\star} = \sum_{k=1}^{n_T} \frac{1}{n_T} \sum_{j=1}^{n_{\text{MC}}} \frac{1}{n_{\text{MC}}} \boldsymbol{P}_{k,j}^{\star}. \tag{142}$$

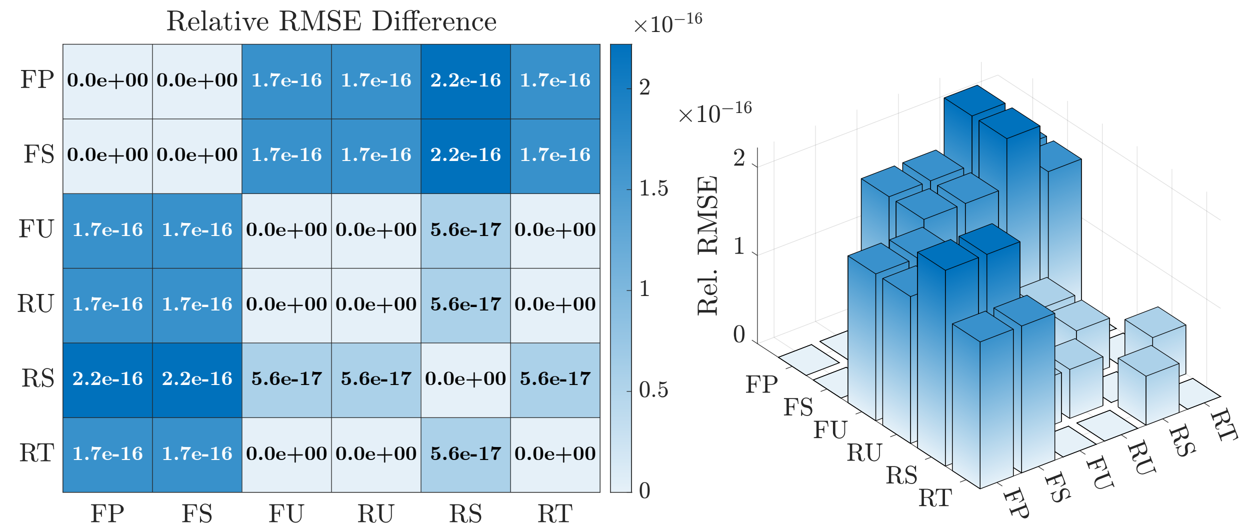
The averaged covariances are then compared using the Frobenius norm of their differences, such that

$$||\bar{\boldsymbol{P}}_i^{\star} - \bar{\boldsymbol{P}}_j^{\star}||_F = \sqrt{\text{Tr}\left[(\bar{\boldsymbol{P}}_i^{\star} - \bar{\boldsymbol{P}}_j^{\star})(\bar{\boldsymbol{P}}_i^{\star} - \bar{\boldsymbol{P}}_j^{\star})^T \right]}, \tag{143}$$

where Tr $[\cdot]$ represents the trace operator and i, j refers to the different smoothers.

Figure 2 shows the relative RMSE difference for all smoothers considered, as a heat map and a bar plot. From this figure, it can be seen that all variants perform nearly equivalent to each other,[¶] validating the new UDU-factorized smoothers and the stable version of the RTS smoother. Figure 3 shows the relative Frobenius norm difference for all

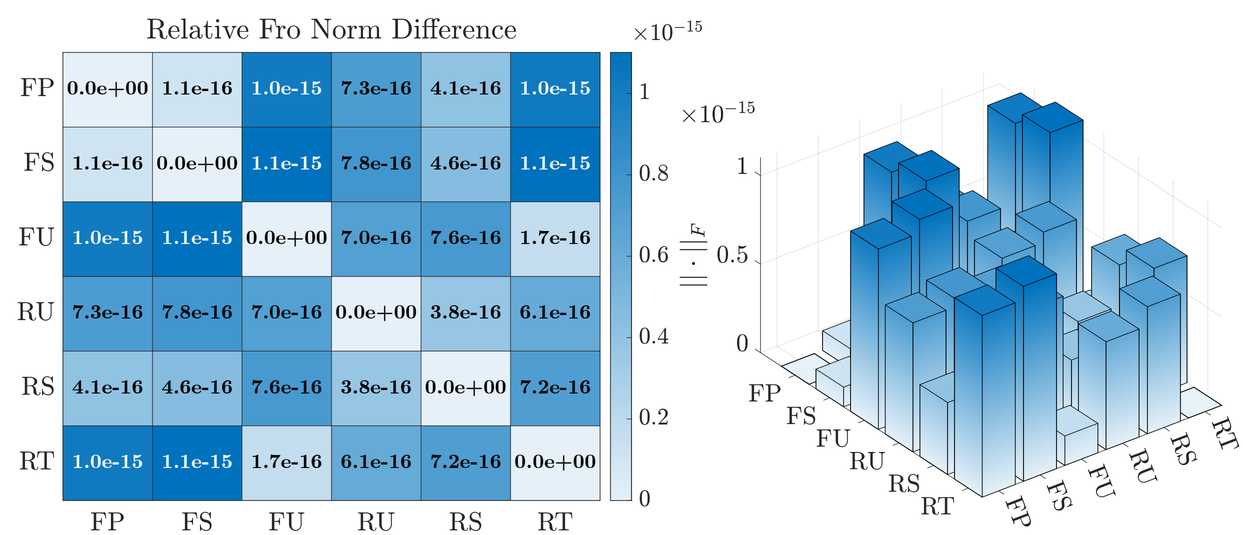
[¶]The differences fall within machine precision for double-precision numbers, as expected for the linear system used in this example.



(a) Heat map **(b) Bar plot**

Fig. 2 Relative RMSE difference

smoothers considered, also as a heat map and a bar plot. From this figure, a similar conclusion can be drawn. As expected, the relative differences are one order of magnitude higher. However, the differences are well below common numerical tolerances, validating the implemented algorithms. Figure 4 shows the smoothed error as a function of time obtained with the RU (all other smoothers, as seen from the previous figures, yield visually identical results). Smoothing tightens the error bounds as expected.



(a) Heat map **(b) Bar plot**

Fig. 3 Relative Frobenius norm difference

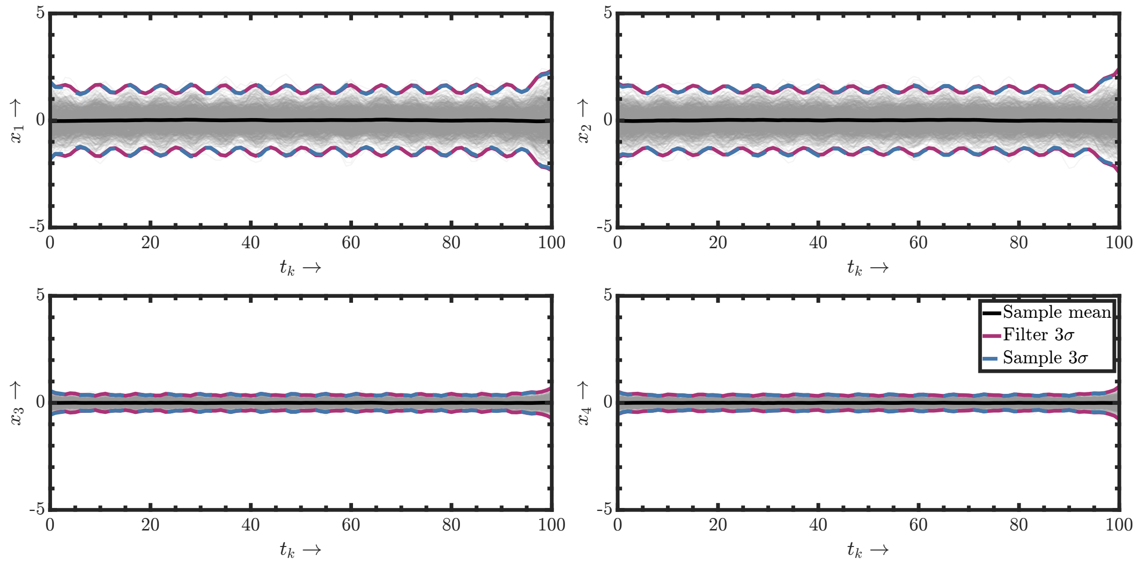


Fig. 4 Smoothed error as a function of time steps. The gray lines show the MC runs.

VII. Final Remarks

A. A Closer Look at the Functionality of the WHHR

The main goal of the WHHR is to compute the UDU factors of a UDU-factorized covariance matrix that is simultaneously updated and downdated, as described in Eq. (27). A secondary application of the WHHR is to compute the UDU factors of a UDU-factorized covariance matrix that has been pre-multiplied by an arbitrary matrix and post-multiplied by its transpose, as presented in Eq. (13). These two applications serve as the tools for deriving the RTS smoother in a UDU framework. However, there is another application of the WHHR that is worth discussing.

The covariance operations in every variant of the full-covariance KF discussed in this work (information formulation, forward filter, and backward filter) can be expressed in the form

$$\tilde{\boldsymbol{P}} = \boldsymbol{\mathcal{A}} \boldsymbol{P} \boldsymbol{\mathcal{A}}^T + \boldsymbol{\mathcal{B}} \boldsymbol{C} \boldsymbol{\mathcal{B}}^T . \tag{144}$$

For example, the covariance propagation in the KF can be represented by setting $\boldsymbol{\mathcal{A}} = \boldsymbol{\Phi}_k$, $\boldsymbol{P} = \boldsymbol{P}_k^+$, $\boldsymbol{\mathcal{B}} = \boldsymbol{G}_k$, and $\boldsymbol{C} = \boldsymbol{Q}_k$. The covariance update in the KF, in Joseph form, can be expressed by setting $\boldsymbol{\mathcal{A}} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)$, $\boldsymbol{P} = \boldsymbol{P}_k^-$, $\boldsymbol{\mathcal{B}} = \boldsymbol{K}_k$, and $\boldsymbol{C} = \boldsymbol{R}_k$. In fact, any covariance operation that resembles a Joseph form, follows the structure in Eq. (144). This implies that stable versions of the FP and the RTS smoothers can also be written in the form of Eq. (144).

As presented, the WHHR cannot be directly used to compute the UDU factors of $\tilde{\boldsymbol{P}}$ in Eq. (144), since $\boldsymbol{P}$ and $\boldsymbol{C}$ are not diagonal matrices. Therefore, Eq. (144) does not exactly match the structure of Eq. (27). Nevertheless, if the UDU factorizations of $\boldsymbol{P}$ and $\boldsymbol{C}$ are used, Eq. (144) can be reformulated to resemble Eq. (27). This is valid because the UDU factors of $\boldsymbol{P}$ are the quantities of interest in this work, and $\boldsymbol{C}$ typically represents a process or measurement noise covariance matrix. Then, by using the UDU factorization of $\boldsymbol{P}$ and $\boldsymbol{C}$, Eq. (144) can be expressed as

$$\tilde{\boldsymbol{P}} = \boldsymbol{\mathcal{A}}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{\mathcal{A}}^T + \boldsymbol{\mathcal{B}}\boldsymbol{U}_C\boldsymbol{D}_C\boldsymbol{U}_C^T\boldsymbol{\mathcal{B}}^T. \tag{145}$$

Now, factoring $\boldsymbol{\mathcal{A}}$ from the left and $\boldsymbol{\mathcal{A}}^T$ from the right results in

$$\tilde{\boldsymbol{P}} = \boldsymbol{\mathcal{A}}\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T + \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}}\boldsymbol{U}_C\boldsymbol{D}_C\boldsymbol{U}_C^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{A}}^{-T}\right)\boldsymbol{\mathcal{A}}^T. \tag{146}$$

And finally, by letting $\boldsymbol{B} = \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}}\boldsymbol{U}_C$ and $\boldsymbol{F} = \boldsymbol{D}_C$, then

$$\tilde{\boldsymbol{P}} = \boldsymbol{\mathcal{A}}\left(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T + \boldsymbol{B}\boldsymbol{F}\boldsymbol{B}^T\right)\boldsymbol{\mathcal{A}}^T, \tag{147}$$

where the term in parenthesis is equal to Eq. (27) with $\boldsymbol{A} = \boldsymbol{E} = \boldsymbol{0}$. Therefore, the UDU factors of the term in parentheses can be computed using the WHHR, as described in Remark III.3. Once this is done, the UDU factors of $\tilde{\boldsymbol{P}}$ can then be obtained using the WHHR, as outlined in Remark III.4.

This result shows that, if desired, the WHHR could serve as the only tool required to perform UDU-factorized filtering and smoothing, replacing both rank-1 updates and the MWGS algorithm. However, care must be taken to ensure that $\boldsymbol{\mathcal{A}}$ is invertible.

B. The FP UDU Smoother vs. the RTS UDU Smoother

As mentioned previously, the RTS smoother is more commonly used than the FP smoother, as it does not require a backward filter, and the smoothed state is independent of the smoothed covariance. The same holds true for the UDU versions presented in this work. The RTS UDU smoother primarily requires two function calls to the WHHR algorithm. In contrast, the FP UDU smoother requires $2n + m$ Fletcher–Powell rank-1 updates and one function call to the WHHR algorithm (which includes the execution of the backward information UDU KF as well). While no formal complexity analysis is presented here, observations from the numerical example suggest that the RTS UDU smoother is approximately 6% faster than the FP UDU smoother.[‖] Therefore, given the mathematical equivalence of the two smoothers and similar runtime, the RTS smoother may be the more practical choice for implementation in UDU form, as it does not require a backward filter and the smoothed state is independent of the smoothed covariance.

VIII. Conclusions

In this work, the UDU-factorized versions of the Fraser-Potter (FP) smoother and the Rauch-Tung-Striebel (RTS) smoother are proposed. For the UDU version of the FP smoother, previously developed tools such as rank-1 updates are

[‖] The smoothers were run on a MacBook Pro with an Apple M1 Pro chip and 16 GB of RAM, implemented in MATLAB R2025b.

used. For the development of the RTS smoother in UDU form, the weighted hyperbolic Householder reflector (WHHR) is derived. The use of the WHHR ensures that the covariance matrix maintains its UDU form throughout the different steps in the algorithm. In addition, a stable form of the RTS smoother is presented, where matrix subtraction is avoided by reformulating the covariance equations to resemble a Joseph-like form.

All derived algorithms, along with the traditional formulations, are compared on a numerical example, demonstrating their equivalence and validating the newly proposed methods. Additionally, the utility of the WHHR is discussed, proposing it as a replacement for other tools commonly used in the UDU-factorized versions of the Kalman filter (KF). Finally, given that both the FP smoother and the RTS smoother are equivalent, the UDU form of the RTS is recommended as a more practical choice for implementation.

Acknowledgment

This material is based on research sponsored by NASA under grant number: 80NSSC20M0211.

Appendix

In this appendix, the equivalence between the FP and RTS smoothers is demonstrated. The derivation is divided into three parts. First, useful relations are established. Second, the equivalence between the smoothed covariances is demonstrated. Finally, the equivalence between the smoothed states is presented. The process for proving the equivalence of the two smoothers is inspired by and follows some of the steps presented in [14].

Preliminaries

Starting from the state propagation equation,

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{q}_k, \tag{148}$$

the previous state, $\boldsymbol{x}_k$, can be expressed as

$$\boldsymbol{x}_k = \boldsymbol{\Phi}_k^{-1} \boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k^{-1} \boldsymbol{G}_k \boldsymbol{q}_k. \tag{149}$$

Using this relation, the backward propagation of the state estimate is given by

$$\hat{\boldsymbol{x}}_{b,k}^{-} = \boldsymbol{\Phi}_k^{-1} \hat{\boldsymbol{x}}_{b,k+1}^{+}. \tag{150}$$

The corresponding estimation error is defined as

$$\begin{aligned}
\boldsymbol{e}_{b,k}^{-} &= \hat{\boldsymbol{x}}_{b,k}^{-} - \boldsymbol{x}_k && (151)\\
&= \boldsymbol{\Phi}_k^{-1}\hat{\boldsymbol{x}}_{b,k+1}^{+} - \left(\boldsymbol{\Phi}_k^{-1}\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k^{-1}\boldsymbol{G}_k\boldsymbol{q}_k\right) && (152)\\
&= \boldsymbol{\Phi}_k^{-1}\left(\hat{\boldsymbol{x}}_{b,k+1}^{+} - \boldsymbol{x}_{k+1}\right) + \boldsymbol{\Phi}_k^{-1}\boldsymbol{G}_k\boldsymbol{q}_k && (153)\\
&= \boldsymbol{\Phi}_k^{-1}\boldsymbol{e}_{b,k+1}^{+} + \boldsymbol{\Phi}_k^{-1}\boldsymbol{G}_k\boldsymbol{q}_k, && (154)
\end{aligned}$$

where $\boldsymbol{e}_{b,k+1}^{+} = \hat{\boldsymbol{x}}_{b,k+1}^{+} - \boldsymbol{x}_{k+1}$ denotes the backward estimation error at time $k+1$. Assuming that $\boldsymbol{e}_{b,k+1}^{+}$ and $\boldsymbol{q}_k$ are uncorrelated, the covariance of the backward estimation error becomes

$$\boldsymbol{P}_{b,k}^{-} = \boldsymbol{\Phi}_k^{-1}\left(\boldsymbol{P}_{b,k+1}^{+} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)\boldsymbol{\Phi}_k^{-T}. \tag{155}$$

Equivalently, using the matrix inversion lemma (Eq. (73)) with $\boldsymbol{A} = \boldsymbol{I}$, $\boldsymbol{B} = \mathcal{I}_{b,k+1}^{+}\boldsymbol{G}_k$, $\boldsymbol{C} = \boldsymbol{Q}_k$, and $\boldsymbol{D} = \boldsymbol{G}_k^T$, the inverse of Eq. (155) can be expressed as

$$\mathcal{I}_{b,k}^{-} = \boldsymbol{\Phi}_k^T\left(\boldsymbol{I} - \boldsymbol{K}_{b,k}\boldsymbol{G}_k^T\right)\mathcal{I}_{b,k+1}^{+}\boldsymbol{\Phi}_k, \tag{156}$$

where

$$\boldsymbol{K}_{b,k} = \mathcal{I}_{b,k+1}^{+}\boldsymbol{G}_k\left(\boldsymbol{G}_k^T\mathcal{I}_{b,k+1}^{+}\boldsymbol{G}_k + \boldsymbol{Q}_k^{-1}\right)^{-1}. \tag{157}$$

Additionally, recall from the Kalman update

$$\boldsymbol{P}_{f,k}^{+} = \left(\left(\boldsymbol{P}_{f,k}^{-}\right)^{-1} + \boldsymbol{H}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{H}_k\right)^{-1}, \tag{158}$$

so that

$$\mathcal{I}_{f,k}^{+} = \mathcal{I}_{f,k}^{-} + \boldsymbol{H}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{H}_k. \tag{159}$$

Similarly, for the backward filter

$$\mathcal{I}_{b,k}^{+} = \mathcal{I}_{b,k}^{-} + \boldsymbol{H}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{H}_k, \tag{160}$$

which yields

$$\mathcal{I}_{f,k}^{+} - \mathcal{I}_{f,k}^{-} = \mathcal{I}_{b,k}^{+} - \mathcal{I}_{b,k}^{-}. \tag{161}$$

Finally, consider the backward propagation of the state in information form, such that

$$\hat{\mathcal{X}}_{b,k}^{-} = \mathcal{I}_{b,k}^{-}\boldsymbol{\Phi}_k^{-1}\boldsymbol{P}_{b,k+1}^{+}\hat{\mathcal{X}}_{b,k+1}^{+}, \tag{162}$$

with $\hat{\mathcal{X}}^-_{b,k} = \mathcal{I}^-_{b,k}\hat{\boldsymbol{x}}^-_{b,k}$ and $\hat{\mathcal{X}}^+_{b,k+1} = \mathcal{I}^+_{b,k}\hat{\boldsymbol{x}}^+_{b,k+1}$. Therefore

$$\hat{\mathcal{X}}^-_{b,k} = \boldsymbol{\Phi}_k^T \left(\boldsymbol{I} - \boldsymbol{K}_{b,k}\boldsymbol{G}_k^T\right) \hat{\mathcal{X}}^+_{b,k+1}, \tag{163}$$

and

$$\hat{\mathcal{X}}^+_{b,k+1} = \mathcal{I}^+_{b,k+1}\boldsymbol{\Phi}_k \boldsymbol{P}^-_{b,k}\hat{\mathcal{X}}^-_{b,k}, \tag{164}$$

which yields

$$\hat{\mathcal{X}}^+_{b,k+1} = \left(\boldsymbol{I} + \mathcal{I}^+_{b,k+1}\boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)\boldsymbol{\Phi}_k^{-T}\hat{\mathcal{X}}^-_{b,k}. \tag{165}$$

Covariance Equivalence

The covariance equivalence proof begins with the FP smoother covariance, given by

$$\boldsymbol{P}^\star_k = \left(\left(\boldsymbol{P}^+_{f,k}\right)^{-1} + \left(\boldsymbol{P}^-_{b,k}\right)^{-1}\right)^{-1} = \left(\mathcal{I}^+_{f,k} + \mathcal{I}^-_{b,k}\right)^{-1}. \tag{166}$$

Applying the matrix inversion lemma with $\boldsymbol{A} = \left(\boldsymbol{P}^+_{f,k}\right)^{-1}$, $\boldsymbol{B} = \boldsymbol{I}$, $\boldsymbol{C} = \left(\boldsymbol{P}^-_{b,k}\right)^{-1}$, and $\boldsymbol{D} = \boldsymbol{I}$, yields

$$\boldsymbol{P}^\star_k = \boldsymbol{P}^+_{f,k} - \boldsymbol{P}^+_{f,k}\left(\boldsymbol{P}^+_{f,k} + \boldsymbol{P}^-_{b,k}\right)^{-1}\boldsymbol{P}^+_{f,k}. \tag{167}$$

Note that factoring out $\boldsymbol{P}^-_{b,k}$ from the second term, recovers Eq. (65). Working with the inner term in Eq. (167) and substituting $\boldsymbol{P}^-_{b,k}$ from Eq. (155), results in

$$\begin{aligned}
\left(\boldsymbol{P}^+_{f,k} + \boldsymbol{P}^-_{b,k}\right)^{-1} &= \left[\boldsymbol{P}^+_{f,k} + \boldsymbol{\Phi}_k^{-1}\left(\boldsymbol{P}^+_{b,k+1} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)\boldsymbol{\Phi}_k^{-T}\right]^{-1}, \\
&= \boldsymbol{\Phi}_k^T\left(\boldsymbol{\Phi}_k\boldsymbol{P}^+_{f,k}\boldsymbol{\Phi}_k^T + \boldsymbol{P}^+_{b,k+1} + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T\right)^{-1}\boldsymbol{\Phi}_k, \\
&= \boldsymbol{\Phi}_k^T\left(\boldsymbol{P}^-_{f,k+1} + \boldsymbol{P}^+_{b,k+1}\right)^{-1}\boldsymbol{\Phi}_k.
\end{aligned} \tag{168}$$

Now, from Eq. (166), the smoothed covariance can be rewritten as

$$\boldsymbol{P}^\star_{k+1} = \left(\mathcal{I}^+_{f,k+1} + \mathcal{I}^-_{b,k+1}\right)^{-1} \quad\Longrightarrow\quad \left(\boldsymbol{P}^\star_{k+1}\right)^{-1} = \mathcal{I}^+_{f,k+1} + \mathcal{I}^-_{b,k+1}, \tag{169}$$

which yields

$$\boldsymbol{P}^+_{b,k+1} = \left(\left(\boldsymbol{P}^\star_{k+1}\right)^{-1} - \mathcal{I}^-_{f,k+1}\right)^{-1}. \tag{170}$$

By substituting $\boldsymbol{P}^{+}_{b,k+1}$ into Eq. (168), the following relation is obtained

$$\left(\boldsymbol{P}^{+}_{f,k} + \boldsymbol{P}^{-}_{b,k}\right)^{-1} = \boldsymbol{\Phi}^{T}_{k}\left[\boldsymbol{P}^{-}_{f,k+1} + \left(\left(\boldsymbol{P}^{\star}_{k+1}\right)^{-1} - \boldsymbol{\mathcal{I}}^{-}_{f,k+1}\right)^{-1}\right]^{-1}\boldsymbol{\Phi}_{k}. \tag{171}$$

Pre- and post-factoring out $\boldsymbol{P}^{-}_{f,k+1}$ from the inner term, yields

$$\left(\boldsymbol{P}^{+}_{f,k} + \boldsymbol{P}^{-}_{b,k}\right)^{-1} = \boldsymbol{\Phi}^{T}_{k}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\left[\boldsymbol{\mathcal{I}}^{-}_{f,k+1} + \boldsymbol{\mathcal{I}}^{-}_{f,k+1}\left(\left(\boldsymbol{P}^{\star}_{k+1}\right)^{-1} - \boldsymbol{\mathcal{I}}^{-}_{f,k+1}\right)^{-1}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\right]^{-1}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\boldsymbol{\Phi}_{k}. \tag{172}$$

By invoking the matrix inversion lemma one more time, with $\boldsymbol{A} = \boldsymbol{P}^{-}_{f,k+1}$, $\boldsymbol{B} = \boldsymbol{I}$, $\boldsymbol{C} = -\boldsymbol{P}^{\star}_{k+1}$, and $\boldsymbol{D} = \boldsymbol{I}$, the following expression is obtained

$$\left[\boldsymbol{\mathcal{I}}^{-}_{f,k+1} + \boldsymbol{\mathcal{I}}^{-}_{f,k+1}\left(\left(\boldsymbol{P}^{\star}_{k+1}\right)^{-1} - \boldsymbol{\mathcal{I}}^{-}_{f,k+1}\right)^{-1}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\right] = \left(\boldsymbol{P}^{-}_{f,k+1} - \boldsymbol{P}^{\star}_{k+1}\right)^{-1}, \tag{173}$$

so that

$$\left(\boldsymbol{P}^{+}_{f,k} + \boldsymbol{P}^{-}_{b,k}\right)^{-1} = \boldsymbol{\Phi}^{T}_{k}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\left(\boldsymbol{P}^{-}_{f,k+1} - \boldsymbol{P}^{\star}_{k+1}\right)\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\boldsymbol{\Phi}_{k}. \tag{174}$$

Finally, substituting this into Eq. (167)

$$\boldsymbol{P}^{\star}_{k} = \boldsymbol{P}^{+}_{f,k} - \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\left(\boldsymbol{P}^{-}_{f,k+1} - \boldsymbol{P}^{\star}_{k+1}\right)\boldsymbol{\mathcal{I}}^{-}_{f,k+1}\boldsymbol{\Phi}_{k}\boldsymbol{P}^{+}_{f,k}, \tag{175}$$

and defining $\boldsymbol{\mathcal{K}}_{k}$ as

$$\boldsymbol{\mathcal{K}}_{k} = \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\boldsymbol{\mathcal{I}}^{-}_{f,k+1}, \tag{176}$$

the following expression is obtained

$$\boldsymbol{P}^{\star}_{k} = \boldsymbol{P}^{+}_{f,k} - \boldsymbol{\mathcal{K}}_{k}\left(\boldsymbol{P}^{-}_{f,k+1} - \boldsymbol{P}^{\star}_{k+1}\right)\boldsymbol{\mathcal{K}}^{T}_{k}, \tag{177}$$

which is the RTS smoothed covariance. Thus, the equivalence between the FP and RTS covariances is established.

State Equivalence

The state equivalence starts with the FP smoothed state, given by

$$\boldsymbol{x}^{\star}_{k} = \boldsymbol{P}^{\star}_{k}\left[\left(\boldsymbol{P}^{+}_{f,k}\right)^{-1}\hat{\boldsymbol{x}}^{+}_{f,k} + \left(\boldsymbol{P}^{-}_{b,k}\right)^{-1}\hat{\boldsymbol{x}}^{-}_{b,k}\right], \tag{178}$$

$$= \boldsymbol{P}^{\star}_{k}\boldsymbol{\mathcal{I}}^{+}_{f,k}\hat{\boldsymbol{x}}^{+}_{f,k} + \boldsymbol{P}^{\star}_{k}\boldsymbol{\mathcal{I}}^{-}_{b,k}\hat{\boldsymbol{x}}^{-}_{b,k}. \tag{179}$$

Using the information formulation ($\hat{\boldsymbol{\mathcal{X}}}^-_{b,k} = \boldsymbol{\mathcal{I}}^-_{b,k}\hat{\boldsymbol{x}}^-_{b,k}$), yields

$$\boldsymbol{x}^\star_k = \boldsymbol{P}^\star_k \boldsymbol{\mathcal{I}}^+_{f,k}\hat{\boldsymbol{x}}^+_{f,k} + \boldsymbol{P}^\star_k \hat{\boldsymbol{\mathcal{X}}}^-_{b,k}. \tag{180}$$

Note that, from Eq. (166), $\boldsymbol{P}^\star_k \boldsymbol{\mathcal{I}}^+_{f,k} = (\boldsymbol{I} + \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k})^{-1}$, and using the matrix inversion lemma with $\boldsymbol{A} = \boldsymbol{I}$, $\boldsymbol{B} = \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}$, $\boldsymbol{C} = \boldsymbol{I}$, and $\boldsymbol{D} = \boldsymbol{I}$, the state equation can be reformulated as

$$\boldsymbol{x}^\star_k = \left[\boldsymbol{I} - \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\left(\boldsymbol{I} + \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k} + \boldsymbol{P}^\star_k \hat{\boldsymbol{\mathcal{X}}}^-_{b,k}, \tag{181}$$

which recovers Eq. (66). Now, substituting $\boldsymbol{P}^\star_k$ with the RTS formulation (since it was proved that the RTS and FP smoothed covariances are equivalent)

$$\boldsymbol{x}^\star_k = \left[\boldsymbol{I} - \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\left(\boldsymbol{I} + \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k} + \left[\boldsymbol{P}^+_{f,k} - \boldsymbol{\mathcal{K}}_k\left(\boldsymbol{P}^-_{f,k+1} - \boldsymbol{P}^\star_{k+1}\right)\boldsymbol{\mathcal{K}}^T_k\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k}. \tag{182}$$

Plugging in for $\boldsymbol{\mathcal{K}}_k$ yields

$$\begin{aligned}\boldsymbol{x}^\star_k = &\left[\boldsymbol{I} - \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\left(\boldsymbol{I} + \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k}\\ &+ \left[\boldsymbol{P}^+_{f,k} - \boldsymbol{P}^+_{f,k}\boldsymbol{\Phi}^T_k\boldsymbol{\mathcal{I}}^-_{f,k+1}\left(\boldsymbol{P}^-_{f,k+1} - \boldsymbol{P}^\star_{k+1}\right)\boldsymbol{P}^+_{f,k}\boldsymbol{\Phi}_k\boldsymbol{\mathcal{I}}^-_{f,k+1}\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k},\end{aligned} \tag{183}$$

and factoring out $\boldsymbol{P}^+_{f,k}$

$$\begin{aligned}\boldsymbol{x}^\star_k = &\boldsymbol{P}^+_{f,k}\left[\boldsymbol{\mathcal{I}}^+_{f,k} - \boldsymbol{\mathcal{I}}^-_{b,k}\left(\boldsymbol{I} + \boldsymbol{P}^+_{f,k}\boldsymbol{\mathcal{I}}^-_{b,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k}\\ &+ \boldsymbol{P}^+_{f,k}\left[\boldsymbol{I} - \boldsymbol{\Phi}^T_k\boldsymbol{\mathcal{I}}^-_{f,k+1}\left(\boldsymbol{P}^-_{f,k+1} - \boldsymbol{P}^\star_{k+1}\right)\boldsymbol{\mathcal{I}}^-_{f,k+1}\boldsymbol{\Phi}_k\boldsymbol{P}^+_{f,k}\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k},\end{aligned} \tag{184}$$

$$\begin{aligned}\boldsymbol{x}^\star_k = &\boldsymbol{P}^+_{f,k}\left[\boldsymbol{\mathcal{I}}^+_{f,k} - \left(\boldsymbol{P}^-_{b,k} + \boldsymbol{P}^+_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k}\\ &+ \boldsymbol{P}^+_{f,k}\left[\boldsymbol{I} - \boldsymbol{\Phi}^T_k\boldsymbol{\mathcal{I}}^-_{f,k+1}\left(\boldsymbol{P}^-_{f,k+1} - \boldsymbol{P}^\star_{k+1}\right)\boldsymbol{\mathcal{I}}^-_{f,k+1}\boldsymbol{\Phi}_k\boldsymbol{P}^+_{f,k}\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k}.\end{aligned} \tag{185}$$

Using Eq. (174) and Eq. (168), the smoothed state becomes

$$\boldsymbol{x}^\star_k = \boldsymbol{P}^+_{f,k}\left[\boldsymbol{\mathcal{I}}^+_{f,k} - \left(\boldsymbol{P}^-_{b,k} + \boldsymbol{P}^+_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k} + \boldsymbol{P}^+_{f,k}\left[\boldsymbol{I} - \left(\boldsymbol{P}^-_{b,k} + \boldsymbol{P}^+_{f,k}\right)^{-1}\boldsymbol{P}^+_{f,k}\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k}, \tag{186}$$

$$= \boldsymbol{P}^+_{f,k}\left[\boldsymbol{\mathcal{I}}^+_{f,k} - \left(\boldsymbol{P}^-_{b,k} + \boldsymbol{P}^+_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k} + \boldsymbol{P}^+_{f,k}\left[\boldsymbol{I} - \boldsymbol{\Phi}^T_k\left(\boldsymbol{P}^-_{f,k+1} + \boldsymbol{P}^+_{b,k+1}\right)^{-1}\boldsymbol{\Phi}_k\boldsymbol{P}^+_{f,k}\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k}, \tag{187}$$

$$\begin{aligned}= &\boldsymbol{P}^+_{f,k}\left[\boldsymbol{\mathcal{I}}^+_{f,k} - \left(\boldsymbol{P}^-_{b,k} + \boldsymbol{P}^+_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^+_{f,k}\\ &+ \boldsymbol{P}^+_{f,k}\boldsymbol{\Phi}^T_k\left(\boldsymbol{P}^-_{f,k+1} + \boldsymbol{P}^+_{b,k+1}\right)^{-1}\left[\left(\boldsymbol{P}^-_{f,k+1} + \boldsymbol{P}^+_{b,k+1}\right)\boldsymbol{\Phi}^{-T}_k - \boldsymbol{\Phi}_k\boldsymbol{P}^+_{f,k}\right]\hat{\boldsymbol{\mathcal{X}}}^-_{b,k}.\end{aligned} \tag{188}$$

Substituting for $\boldsymbol{P}^{-}_{f,k+1}$ and factoring out $\boldsymbol{P}^{+}_{b,k+1}$ from the second term, yields

$$\begin{aligned}
\boldsymbol{x}^{\star}_{k} = \boldsymbol{P}^{+}_{f,k}&\left[\mathcal{I}^{+}_{f,k} - \left(\boldsymbol{P}^{-}_{b,k} + \boldsymbol{P}^{+}_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^{+}_{f,k} \\
&+ \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\left(\boldsymbol{P}^{-}_{f,k+1} + \boldsymbol{P}^{+}_{b,k+1}\right)^{-1}\boldsymbol{P}^{+}_{b,k+1}\left(\mathcal{I}^{+}_{b,k+1}\boldsymbol{G}_{k}\boldsymbol{Q}_{k}\boldsymbol{G}^{T}_{k} + \boldsymbol{I}\right)\boldsymbol{\Phi}^{-T}_{k}\hat{\boldsymbol{\mathcal{X}}}^{-}_{b,k},
\end{aligned} \tag{189}$$

and using Eq. (165), simplifies to

$$\boldsymbol{x}^{\star}_{k} = \boldsymbol{P}^{+}_{f,k}\left[\mathcal{I}^{+}_{f,k} - \left(\boldsymbol{P}^{-}_{b,k} + \boldsymbol{P}^{+}_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^{+}_{f,k} + \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\left(\boldsymbol{P}^{-}_{f,k+1} + \boldsymbol{P}^{+}_{b,k+1}\right)^{-1}\boldsymbol{P}^{+}_{b,k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}. \tag{190}$$

Now, to finish the derivation, the RTS smoothed state will be constructed from the FP smoothed state, and will be shown to be equal to Eq. (190). First, using Eq. (180)

$$\begin{aligned}
\boldsymbol{x}^{\star}_{k+1} &= \boldsymbol{P}^{\star}_{k+1}\mathcal{I}^{+}_{f,k+1}\hat{\boldsymbol{x}}^{+}_{f,k+1} + \boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{-}_{b,k+1}, && (191)\\
&= \boldsymbol{P}^{\star}_{k+1}\mathcal{I}^{+}_{f,k+1}\hat{\boldsymbol{x}}^{+}_{f,k+1} + \boldsymbol{P}^{\star}_{k+1}\left(\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1} - \boldsymbol{H}^{T}_{k+1}\boldsymbol{R}^{-1}_{k+1}\boldsymbol{y}_{k+1}\right), && (192)\\
&= \boldsymbol{P}^{\star}_{k+1}\mathcal{I}^{+}_{f,k+1}\left[\hat{\boldsymbol{x}}^{-}_{f,k+1} + \boldsymbol{P}^{+}_{f,k+1}\boldsymbol{H}^{T}_{k+1}\boldsymbol{R}^{-1}_{k+1}\left(\boldsymbol{y}_{k+1} - \boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}^{-}_{f,k+1}\right)\right] + \boldsymbol{P}^{\star}_{k+1}\left(\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1} - \boldsymbol{H}^{T}_{k+1}\boldsymbol{R}^{-1}_{k+1}\boldsymbol{y}_{k+1}\right), && (193)\\
&= \boldsymbol{P}^{\star}_{k+1}\left(\mathcal{I}^{+}_{f,k+1} - \boldsymbol{H}^{T}_{k+1}\boldsymbol{R}^{-1}_{k+1}\boldsymbol{H}_{k+1}\right)\hat{\boldsymbol{x}}^{-}_{f,k+1} + \boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}, && (194)\\
&= \boldsymbol{P}^{\star}_{k+1}\mathcal{I}^{-}_{f,k+1}\hat{\boldsymbol{x}}^{-}_{f,k+1} + \boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}, && (195)
\end{aligned}$$

where the second, third, and fifth lines come from the update in the KF and the information formulation of the KF. Subtracting $\hat{\boldsymbol{x}}^{-}_{f,k+1}$ and pre-mutliplying by $\mathcal{K}_{k}$, yields

$$\mathcal{K}_{k}\left(\boldsymbol{x}^{\star}_{k+1} - \hat{\boldsymbol{x}}^{-}_{f,k+1}\right) = \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\mathcal{I}^{-}_{f,k+1}\left[\left(\boldsymbol{P}^{\star}_{k+1} - \boldsymbol{P}^{-}_{f,k+1}\right)\mathcal{I}^{-}_{f,k+1}\hat{\boldsymbol{x}}^{-}_{f,k+1} + \boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}\right]. \tag{196}$$

Using Eq. (174), Eq. (196) can be rewritten as

$$\begin{aligned}
\mathcal{K}_{k}\left(\boldsymbol{x}^{\star}_{k+1} - \hat{\boldsymbol{x}}^{-}_{f,k+1}\right) &= -\boldsymbol{P}^{+}_{f,k}\left(\boldsymbol{P}^{+}_{f,k} + \boldsymbol{P}^{-}_{b,k}\right)^{-1}\boldsymbol{\Phi}^{-1}_{k}\hat{\boldsymbol{x}}^{-}_{f,k+1} + \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\mathcal{I}^{-}_{f,k+1}\boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}, && (197)\\
&= -\boldsymbol{P}^{+}_{f,k}\left(\boldsymbol{P}^{+}_{f,k} + \boldsymbol{P}^{-}_{b,k}\right)^{-1}\hat{\boldsymbol{x}}^{+}_{f,k} + \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\mathcal{I}^{-}_{f,k+1}\boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}. && (198)
\end{aligned}$$

Working with the second term

$$\boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\mathcal{I}^{-}_{f,k+1}\boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1} = \boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\left(\boldsymbol{P}^{-}_{f,k+1} + \boldsymbol{P}^{+}_{b,k+1}\right)^{-1}\boldsymbol{P}^{+}_{b,k+1}\mathcal{I}^{+}_{b,k+1}\left(\boldsymbol{P}^{-}_{f,k+1} + \boldsymbol{P}^{+}_{b,k+1}\right)\mathcal{I}^{-}_{f,k+1}\boldsymbol{P}^{\star}_{k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}. \tag{199}$$

Note that, using Eq. (166) and Eq. (161)

$$\mathcal{I}^{+}_{b,k+1}\left(\boldsymbol{P}^{-}_{f,k+1}+\boldsymbol{P}^{+}_{b,k+1}\right)\mathcal{I}^{-}_{f,k+1}\boldsymbol{P}^{\star}_{k+1} = \mathcal{I}^{+}_{b,k+1}\left(\boldsymbol{P}^{-}_{f,k+1}+\boldsymbol{P}^{+}_{b,k+1}\right)\mathcal{I}^{-}_{f,k+1}\left(\mathcal{I}^{+}_{f,k+1}+\mathcal{I}^{-}_{b,k+1}\right)^{-1}, \tag{200}$$

$$= \left(\mathcal{I}^{-}_{b,k+1}+\mathcal{I}^{+}_{f,k+1}\right)\left(\mathcal{I}^{+}_{f,k+1}+\mathcal{I}^{-}_{b,k+1}\right)^{-1}, \tag{201}$$

$$= \boldsymbol{I}. \tag{202}$$

Therefore, Eq. (198) becomes

$$\mathcal{K}_k\left(\boldsymbol{x}^{\star}_{k+1}-\hat{\boldsymbol{x}}^{-}_{f,k+1}\right) = -\boldsymbol{P}^{+}_{f,k}\left(\boldsymbol{P}^{+}_{f,k}+\boldsymbol{P}^{-}_{b,k}\right)^{-1}\hat{\boldsymbol{x}}^{+}_{f,k}+\boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\left(\boldsymbol{P}^{-}_{f,k+1}+\boldsymbol{P}^{+}_{b,k+1}\right)^{-1}\boldsymbol{P}^{+}_{b,k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}. \tag{203}$$

Finally, adding $\hat{\boldsymbol{x}}^{+}_{f,k}$ on both sides,

$$\hat{\boldsymbol{x}}^{+}_{f,k}+\mathcal{K}_k\left(\boldsymbol{x}^{\star}_{k+1}-\hat{\boldsymbol{x}}^{-}_{f,k+1}\right) = \hat{\boldsymbol{x}}^{+}_{f,k}-\boldsymbol{P}^{+}_{f,k}\left(\boldsymbol{P}^{+}_{f,k}+\boldsymbol{P}^{-}_{b,k}\right)^{-1}\hat{\boldsymbol{x}}^{+}_{f,k}+\boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\left(\boldsymbol{P}^{-}_{f,k+1}+\boldsymbol{P}^{+}_{b,k+1}\right)^{-1}\boldsymbol{P}^{+}_{b,k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}, \tag{204}$$

where the left-hand side is the RTS smoothed state equation, such that

$$\boldsymbol{x}^{\star}_{k} = \boldsymbol{P}^{+}_{f,k}\left[\mathcal{I}^{+}_{f,k}-\left(\boldsymbol{P}^{-}_{b,k}+\boldsymbol{P}^{+}_{f,k}\right)^{-1}\right]\hat{\boldsymbol{x}}^{+}_{f,k}+\boldsymbol{P}^{+}_{f,k}\boldsymbol{\Phi}^{T}_{k}\left(\boldsymbol{P}^{-}_{f,k+1}+\boldsymbol{P}^{+}_{b,k+1}\right)^{-1}\boldsymbol{P}^{+}_{b,k+1}\hat{\boldsymbol{\mathcal{X}}}^{+}_{b,k+1}, \tag{205}$$

which is exactly Eq. (190). Thus the RTS smoother is obtained from the FP smoother and has been proven to be mathematically equivalent to it.

References

[1] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc, 2001.

[2] Jaynes, E., "On the rationale of maximum-entropy methods," *Proceedings of the IEEE*, Vol. 70, No. 9, 1982, pp. 939–952. https://doi.org/10.1109/PROC.1982.12425.

[3] Cover, T. M., and Thomas, J. A., *Elements of Information Theory*, Wiley-Interscience, 2006.

[4] Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, 1977.

[5] Kaminski, P., Bryson, A., and Schmidt, S., "Discrete square root filtering: A survey of current techniques," *IEEE Transactions on Automatic Control*, Vol. 16, No. 6, 1971, pp. 727–736. https://doi.org/10.1109/TAC.1971.1099816.

[6] Carpenter, J. R., and D'Souza, C., *Navigation Filter Best Practices*, NASA/TP–2018–219822/Revision, 2025.

[7] Agee, W. S., and Turner, R. H., "Triangular decomposition of a positive definite matrix plus a symmetric dyad with application to kalman filtering." Tech. rep., White Sands Missile Range, 1972.

[8] Carlson, N., "Fast triangular factorization of the square root filter," *AIAA Journal*, Vol. 11, No. 9, 1973, pp. 1259–1265. https://doi.org/10.2514/3.6907.

[9] Fletcher, R., and Powell, M. J. D., "On the modification of LDL factorizations," *Mathematics of Computation*, Vol. 28, No. 128, 1974, pp. 1067–1087. https://doi.org/10.2307/2005366.

[10] Thornton, C. L., and Bierman, G. J., "UDUT Covariance Factorization for Kalman Filtering," *Advances in Theory and Application*, Control and Dynamic Systems, Vol. 16, edited by C. Leondes, Academic Press, 1980, pp. 177–248.

[11] Fraser, D. C., "A New Technique for the Optimal Smoothing of Data," Ph.D. thesis, Massachusetts Institute of Technology, 1967.

[12] Fraser, D. C., and Potter, J. E., "The optimum smoother as a combination of two opimum linear filters," *IEEE Transactions on Automatic Control*, Vol. AC-14, No. 4, 1969, pp. 387–390. https://doi.org/10.1109/TAC.1969.1099196.

[13] Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, Vol. 3, No. 8, 1965, pp. 1445–1450. https://doi.org/10.2514/3.3166.

[14] Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Chapman & Hall, 2011.

[15] Bierman, G., "A new computationally efficient fixed-interval, discrete-time smoother," *Automatica*, Vol. 19, No. 5, 1983, pp. 503–511. https://doi.org/10.1016/0005-1098(83)90005-5.

[16] McReynolds, S., "Covariance factorization algorithms for fixed-interval smoothing of linear discrete dynamic systems," *IEEE Transactions on Automatic Control*, Vol. 35, No. 10, 1990, pp. 1181–1183. https://doi.org/10.1109/9.58568.

[17] Watanabe, K., and Tzafestas, S., "New computationally efficient formula for backward-pass fixed-interval smoother and its UD factorisation algorithm," *IEE Proceedings D (Control Theory and Applications)*, Vol. 136, No. 2, 1989, pp. 73–78. https://doi.org/10.1049/ip-d.1989.0011.

[18] D'Souza, C., and Zanetti, R., "Information Formulation of the UDU Kalman Filter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 55, No. 1, 2019. https://doi.org/10.1109/TAES.2018.2850379.

[19] Householder, A. S., "Unitary triangularization of a nonsymmetric matrix," *Journal of the Association for Computing Machinery*, Vol. 5, No. 4, 1958, pp. 339–342. https://doi.org/10.1145/320941.320947.

[20] Rader, C., and Steinhardt, A., "Hyperbolic Householder transformations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 34, No. 6, 1986. https://doi.org/10.1109/TASSP.1986.1164998.

[21] Zanetti, R., Holt, G., Gay, R., D'Souza, C., Sud, J., Mamich, H., and Gillis, R., "Design and Flight Performance of the Orion Prelaunch Navigation System," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 9, 2017, pp. 2289–2300. https://doi.org/10.2514/1.G002666.

[22] Zanetti, R., Holt, G., Gay, R., D'Souza, C., Sud, J., Mamich, H., Begley, M., King, E., and Clark, F. D., "Absolute Navigation Performance of the Orion Exploration Flight Test 1," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 5, 2018, pp. 1106–1116. https://doi.org/10.2514/1.G002371.

[23] McCabe, J. S., "A Discussion on UDU Kalman Filters and Measurement Underweighting," *Proceedings of the International Conference On Multisensor Fusion And Integration For Intelligent Systems (MFI)*, 2025, pp. 1–6. https://doi.org/10.1109/MFI67357.2025.11259160.

[24] McCabe, J. S., "Multitarget tracking and terrain-aided navigation using square-root consider filters," Ph.D. thesis, Missouri University of Science and Technology, 2018.

[25] Grewal, M. S., and Andrews, A. P., *Kalman filtering: Theory and Practice with MATLAB*, John Wiley & Sons, 2014.