# Inverse Schmidt Estimators

*Kejian J. Wu and Stergios I. Roumeliotis*
{kejian|stergios}@cs.umn.edu

*Department of Computer Science & Engineering*
*University of Minnesota*

MARS LAB

Multiple Autonomous
Robotic Systems Labroratory

Technical Report
Number -2016-003
September 2016

Dept. of Computer Science & Engineering
University of Minnesota
4-192 EE/CS Building
200 Union St. S.E.
Minneapolis, MN 55455
Tel: (612) 625-2217
Fax: (612) 625-0572
URL: mars.cs.umn.edu

# Inverse Schmidt Estimators

Kejian J. Wu and Stergios I. Roumeliotis
{kejian|stergios}@cs.umn.edu

Department of Computer Science & Engineering
University of Minnesota

*Multiple Autonomous Robotic Systems Laboratory, TR-2016-003*
September 2016

## 1 Introduction

The Kalman filter (KF) and its information-domain equivalent, inverse filter (IF), are popular algorithms for localization and navigation applications [1]. And in order to improve the numerical stability, their square-root forms have been developed, i.e., the square-root Kalman filter (SR-KF) and the square-root inverse filter (SR-IF) [2]. When nuisance parameters exist, the Schmidt-Kalman filter (SKF) [3] can be used for reducing the dimensionality of the state estimate, while still considering the uncertainty of these parameters and ensuring the correctness of the covariance matrix, i.e., to guarantee estimation consistency. In this work, we introduce the SKF's information-domain equivalent in its square-root form, i.e., the square-root inverse Schmidt estimator (SR-ISE or ISE). To the best of our knowledge, this is the first time that the information form of the Schmidt filter has been developed.

Furthermore, we provide complexity analysis of this proposed exact ISE, and identify its limitations in terms of computational efficiency. In order to reduce the processing cost even more, we introduce further approximations based on the exact ISE and obtain other consistent alternatives. One important outcome is the resource-aware inverse Schmidt estimator (RISE), that enables trading estimation accuracy for computational efficiency, and hence is more flexible when employed to provide sufficiently accurate solutions with low processing and memory cost at the same time.

Potential applications of the proposed (exact or resource-aware) inverse Schmidt estimators includes, but is not limited to, the problem of simultaneous localization and mapping (SLAM). When considering large-scale SLAM problems, information-domain methods allow sparse structures in the Hessian matrix (and its square-root factor) [4] as compared to the dense covariance matrix, and hence are more efficient in terms of both computational and memory requirements. Meanwhile, as compared to the optimal KF or IF, their square-root equivalents improve the numerical stability. As a result, information-based square-root methods have been developed for the SLAM problem [5, 6]. These optimal solutions, however, cannot always be obtained in real time when the problem size grows, especially on resource-constrained mobile devices, and hence, approximations are required to reduce the computational complexity. In this regard, we propose to employ the idea of Schmidt, which provides a consistent approximation by choosing to update only a portion of the entire state vector. Therefore, one could potentially employ the inverse Schmidt estimators presented in this work, to obtain efficient approximate solutions for large-scale SLAM problems.

The paper is organized as follows: We start by providing background knowledge in Sec. 2 on estimation consistency, which is an important concept and ensured by the Schmidt approach, and hence a major objective we aim to achieve in our proposed methods. The problem of finding the exact equivalent of the Schmidt-Kalman filter (SKF) in the information domain is formulated in Sec. 3. Then, Sec. 4 presents in detail a complete derivation of the exact inverse Schmidt estimator (ISE) algorithm, as well as its complexity analysis. Further approximations for gaining extra computational efficiency, based on the exact ISE, are introduced in Sec. 5, featuring the resource-aware inverse Schmidt estimator (RISE) that balances between accuracy and efficiency. Finally, Sec. 6 concludes this paper.

# 2 Background: Estimation Consistency

## 2.1 Definition of Estimation Consistency

As defined in [7, 8], a state estimator is consistent if the estimation errors are zero-mean and have covariance matrix smaller than or equal to the one calculated by the estimator. For the purposes of this work, we focus on the covariance requirement, while the zero-mean error requirement is typically satisfied in general. In other words, an estimator is consistent, if the estimated covariance (or the inverse of the estimated Hessian matrix for estimators in the information form) is greater than or equal to, in the matrix positive-semidefinite sense, the true covariance computed by an optimal approach (e.g., KF or IF).

Consistency is an important concept in the theory of estimation, as it is a fundamental property of an estimator, with inconsistency typically indicating a bad performance or even failure of the estimator. Specifically, according to the definition, inconsistency means that the estimated covariance (or its information equivalent) is overly confident and does not represent correctly the uncertainty of the current estimate. Hence, the estimated covariance does not offer a reliable measure of the quality of the state estimate. More importantly, combining these overly optimistic estimates with new measurements later on will further degrade the accuracy of the results.

## 2.2 Inconsistent Approximations

In practice, however, in order to limit the computational cost and achieve efficient solutions, approximations that lead to inconsistent estimates are commonly used. For example, in the literature of simultaneous localization and mapping (SLAM), most state-of-the-art systems employ approximations where some previously-estimated states are assumed to be perfectly known, such as past keyframes involved in the system's frontend thread during relocalization. Apparently, assuming some imperfect states as perfectly-known ignores the uncertainty of these states, and hence the estimated covariance based on this assumption will become falsely optimistic as compared to the true uncertainty. Therefore, these approaches generate inconsistent estimates according to the definition, and the estimation accuracy will suffer, in exchange for faster processing speed.

In fact, the problem of inconsistency has been acknowledged in the SLAM community in the past, and remedies are often used to alleviate its negative impact on estimation accuracy, e.g., by artificially inflating the covariance of the noise corresponding to the relocalization visual observations [9, 10]. These heuristics, however, offer no guarantees on the estimation consistency or the systems performance. To address this issue, in this work, we focus on consistent approximations and present novel estimators that ensure estimation consistency.

# 3 Problem Formulation

## 3.1 The Kalman Filter (KF) and the Schmidt-Kalman Filter (SKF)

The standard Kalman filter (KF) estimates a state vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$ and the corresponding error covariance matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$. It is optimal in the minimum mean-squared-error (MMSE) sense, and has a quadratic complexity in the size of all estimated states, in both computational and memory requirements. To achieve more efficient solutions, approximations have been introduced to the KF. One of such existing methods is the Schmidt-Kalman filter (SKF), which reduces the computational cost down to linear, and more importantly, guarantees estimation consistency at the same time.

Specifically, if $\mathbf{x}$ is splitted into

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1^T & \mathbf{x}_2^T\end{bmatrix}^T, \quad \text{with } \mathbf{x}_1 \in \mathbb{R}^{n_1\times 1}, \ \mathbf{x}_2 \in \mathbb{R}^{n_2\times 1}, \text{ and } n_1+n_2=n \tag{1}$$

and correspondingly for the covariance matrix:

$$\mathbf{P} = \begin{bmatrix}\mathbf{P}_{11} & \mathbf{P}_{12}\\ \mathbf{P}_{21} & \mathbf{P}_{22}\end{bmatrix}, \quad \text{with } \mathbf{P}_{11} \in \mathbb{R}^{n_1\times n_1}, \ \mathbf{P}_{21}^T=\mathbf{P}_{12} \in \mathbb{R}^{n_1\times n_2}, \text{ and } \mathbf{P}_{22} \in \mathbb{R}^{n_2\times n_2} \tag{2}$$

where $\mathbf{x}_1$ consists of the states of interest (to be updated) and $\mathbf{x}_2$ consists of all other states (not to be updated), then the SKF performs the following update:

$$\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{I}_m \tag{3}$$

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T\mathbf{S}^{-1} \triangleq \begin{bmatrix}\mathbf{K}_1 \\ \mathbf{K}_2\end{bmatrix}, \quad \text{with} \;\; \mathbf{K}_1 \in \mathbb{R}^{n_1 \times m} \;\; \text{and} \;\; \mathbf{K}_2 \in \mathbb{R}^{n_2 \times m} \tag{4}$$

$$\hat{\mathbf{x}}^s = \begin{bmatrix}\hat{\mathbf{x}}_1 + \mathbf{K}_1\mathbf{r} \\ \hat{\mathbf{x}}_2\end{bmatrix} \tag{5}$$

$$\mathbf{P}^s = \begin{bmatrix}\mathbf{P}_{11} - \mathbf{K}_1\mathbf{S}\mathbf{K}_1^T & \mathbf{P}_{12} - \mathbf{K}_1\mathbf{S}\mathbf{K}_2^T \\ \mathbf{P}_{21} - \mathbf{K}_2\mathbf{S}\mathbf{K}_1^T & \mathbf{P}_{22}\end{bmatrix} \tag{6}$$

where $\mathbf{H} \in \mathbb{R}^{m \times n}$ denotes the pre-whitened measurement Jacobian (so that the measurement noise has covariance equal to the $m \times m$ identity matrix $\mathbf{I}_m$), $\mathbf{r} \in \mathbb{R}^{m \times 1}$ the pre-whitened measurement residual, $\mathbf{S}$ the residual covariance, $\mathbf{K}$ the Kalman gain, and $\hat{\mathbf{x}}$ the prior state estimate, respectively. Note that, after the SKF update, $\hat{\mathbf{x}}_1$, $\mathbf{P}_{11}$, and $\mathbf{P}_{12}$ become exactly the same as the result of the standard KF update, while $\hat{\mathbf{x}}_2$ and $\mathbf{P}_{22}$ remain unchanged. This way, the SKF updates only the states of interest and the corresponding covariance blocks, and hence, achieves computational savings as compared to the KF.

The SKF is an approximation to the KF (or equivalently the SR-IF), in the sense that it drops a certain amount of available information, i.e., the portion corresponding to $\mathbf{x}_2$. This can be shown analytically as follows: The posterior covariance after the KF update is given by:

$$\mathbf{P}^{\oplus} = \mathbf{P} - \mathbf{K}\mathbf{S}\mathbf{K}^T \tag{7}$$

Therefore, the SKF covariance update in (6) can be written as:

$$\mathbf{P}^s = \mathbf{P}^{\oplus} + \bar{\mathbf{K}}_2\mathbf{S}\bar{\mathbf{K}}_2^T, \quad \text{with} \;\; \bar{\mathbf{K}}_2 \triangleq \begin{bmatrix}\mathbf{0}_{n_1 \times m} \\ \mathbf{K}_2\end{bmatrix} \tag{8}$$

and hence, the corresponding information is:

$$\begin{aligned}\boldsymbol{\mathcal{H}}^s \triangleq \mathbf{P}^{s^{-1}} &= (\mathbf{P}^{\oplus} + \bar{\mathbf{K}}_2\mathbf{S}\bar{\mathbf{K}}_2^T)^{-1} = \mathbf{P}^{\oplus^{-1}} - \mathbf{P}^{\oplus^{-1}}\bar{\mathbf{K}}_2(\mathbf{S}^{-1} + \bar{\mathbf{K}}_2^T\mathbf{P}^{\oplus^{-1}}\bar{\mathbf{K}}_2)^{-1}\bar{\mathbf{K}}_2^T\mathbf{P}^{\oplus^{-1}} \\ &= \boldsymbol{\mathcal{H}}^{\oplus} - \boldsymbol{\mathcal{H}}^{\oplus}\bar{\mathbf{K}}_2(\mathbf{S}^{-1} + \bar{\mathbf{K}}_2^T\boldsymbol{\mathcal{H}}^{\oplus}\bar{\mathbf{K}}_2)^{-1}\bar{\mathbf{K}}_2^T\boldsymbol{\mathcal{H}}^{\oplus}, \quad \text{with} \;\; \boldsymbol{\mathcal{H}}^{\oplus} \triangleq \mathbf{P}^{\oplus^{-1}}\end{aligned} \tag{9}$$

If we define the Cholesky factorization of the following symmetric positive-definite (SPD) matrix:

$$\mathbf{S}^{-1} + \bar{\mathbf{K}}_2^T\boldsymbol{\mathcal{H}}^{\oplus}\bar{\mathbf{K}}_2 \triangleq \mathbf{L}_s\mathbf{L}_s^T \tag{10}$$

then (9) becomes:

$$\begin{aligned}\boldsymbol{\mathcal{H}}^s &= \boldsymbol{\mathcal{H}}^{\oplus} - \boldsymbol{\mathcal{H}}^{\oplus}\bar{\mathbf{K}}_2\mathbf{L}_s^{-T}\mathbf{L}_s^{-1}\bar{\mathbf{K}}_2^T\boldsymbol{\mathcal{H}}^{\oplus} \\ &= \boldsymbol{\mathcal{H}}^{\oplus} - \mathbf{J}^{s^T}\mathbf{J}^s, \quad \text{with} \;\; \mathbf{J}^s \triangleq \mathbf{L}_s^{-1}\bar{\mathbf{K}}_2^T\boldsymbol{\mathcal{H}}^{\oplus} \in \mathbb{R}^{m \times n}\end{aligned} \tag{11}$$

which shows that the information term $\mathbf{J}^{s^T}\mathbf{J}^s$ is discarded during the SKF update. Due to this fact, the SKF is a *consistent* approximation of the KF [see (11)], i.e.,

$$\mathbf{P}^s \geq \mathbf{P}^{\oplus} \tag{12}$$

in the matrix positive-semidefinite sense.

On the other hand, note that, among all possible approximate algorithms that do not update $\mathbf{x}_2$, the SKF is the "best" one in the sense that all the information on $\mathbf{x}_1$ has been absorbed. This is obvious as the updated state estimate and covariance of $\mathbf{x}_1$ are exactly the same as those of the optimal KF.

When applied to the SLAM problem, similarly to the optimal KF, the major drawback of the SKF is its high memory requirements: Quadratic in the size of *all* states due to the dense covariance matrix. Thus, the SKF cannot be employed in large-scale SLAM tasks.

### 3.2 The Inverse Filter (IF)

The inverse filter (IF) is the information-domain equivalent of the KF, where the Hessian matrix $\boldsymbol{\mathcal{H}} = \mathbf{P}^{-1}$ is maintained and estimated instead of the covariance $\mathbf{P}$, and hence the name. For updates, it simply adds new information contribution terms from the measurements to the prior information, as:

$$\boldsymbol{\mathcal{H}}^{\oplus} = \boldsymbol{\mathcal{H}} + \mathbf{H}^T\mathbf{H} \tag{13}$$

And the state is updated by solving the normal equation using the Cholesky factorization of $\boldsymbol{\mathcal{H}}^{\oplus}$.

It is well-known that the information domain solutions are more suitable for large-scale SLAM, as the Hessian matrix and its corresponding Cholesky factor are sparse [4].

### 3.3 The Square-Root Inverse Filter (SR-IF)

The square-root inverse filter (SR-IF) is the square-root form equivalent of the IF (hence equivalent to the KF as well), and it maintains the (upper-triangular) Cholesky factor $\mathbf{R} \in \mathbb{R}^{n\times n}$ of the Hessian matrix:

$$\mathbf{P}^{-1} = \boldsymbol{\mathcal{H}} = \mathbf{R}^T\mathbf{R} \tag{14}$$

At each update step, the SR-IF solves the following optimization problem:

$$\min_{\widetilde{\mathbf{x}}} \;\; ||\mathbf{R}\widetilde{\mathbf{x}} - \mathbf{r}_0||^2 + ||\mathbf{H}\widetilde{\mathbf{x}} - \mathbf{r}||^2 \tag{15}$$

where $\widetilde{\mathbf{x}} \triangleq \mathbf{x} - \hat{\mathbf{x}}$ denotes the error state. In the above cost function $\mathcal{C}$, the first term corresponds to the prior information, and the second term arises from the (linearized) measurement equations. Note that, the prior residual, $\mathbf{r}_0$, is nonzero if the linearization and solve are carried out iteratively, in order to reduce linearization errors.

To solve this least-squares problem, a QR factorization is performed on the stacked matrix of the prior information factor $\mathbf{R}$ and the measurement Jacobian $\mathbf{H}$:

$$\begin{bmatrix}\mathbf{R}\\ \mathbf{H}\end{bmatrix} = \mathbf{Q}^{\oplus}\begin{bmatrix}\mathbf{R}^{\oplus}\\ \mathbf{0}_{m\times n}\end{bmatrix}, \quad \text{with} \;\; {\mathbf{Q}^{\oplus}}^T\mathbf{Q}^{\oplus} = \mathbf{Q}^{\oplus}{\mathbf{Q}^{\oplus}}^T = \mathbf{I}_{n+m} \tag{16}$$

where $\mathbf{R}^{\oplus} \in \mathbb{R}^{n\times n}$ is the upper-triangular factor of the QR factorization. In terms of the cost function, this update process can be interpreted as:

$$\mathcal{C} = ||\mathbf{R}\widetilde{\mathbf{x}} - \mathbf{r}_0||^2 + ||\mathbf{H}\widetilde{\mathbf{x}} - \mathbf{r}||^2 = \left\|\begin{bmatrix}\mathbf{R}\\ \mathbf{H}\end{bmatrix}\widetilde{\mathbf{x}} - \begin{bmatrix}\mathbf{r}_0\\ \mathbf{r}\end{bmatrix}\right\|^2 \tag{17}$$

$$= \left\|{\mathbf{Q}^{\oplus}}^T\begin{bmatrix}\mathbf{R}\\ \mathbf{H}\end{bmatrix}\widetilde{\mathbf{x}} - {\mathbf{Q}^{\oplus}}^T\begin{bmatrix}\mathbf{r}_0\\ \mathbf{r}\end{bmatrix}\right\|^2 \tag{18}$$

$$= \left\|\begin{bmatrix}\mathbf{R}^{\oplus}\\ \mathbf{0}_{m\times n}\end{bmatrix}\widetilde{\mathbf{x}} - {\mathbf{Q}^{\oplus}}^T\begin{bmatrix}\mathbf{r}_0\\ \mathbf{r}\end{bmatrix}\right\|^2 \tag{19}$$

$$= \left\|\mathbf{R}^{\oplus}\widetilde{\mathbf{x}} - \mathbf{r}_1^{\oplus}\right\|^2 + \left\|\mathbf{r}_2^{\oplus}\right\|^2, \quad \text{with} \;\; \begin{bmatrix}\mathbf{r}_1^{\oplus}\\ \mathbf{r}_2^{\oplus}\end{bmatrix} \triangleq {\mathbf{Q}^{\oplus}}^T\begin{bmatrix}\mathbf{r}_0\\ \mathbf{r}\end{bmatrix}, \;\; \mathbf{r}_1^{\oplus} \in \mathbb{R}^{n\times 1} \;\; \text{and} \;\; \mathbf{r}_2^{\oplus} \in \mathbb{R}^{m\times 1} \tag{20}$$

where from (17) to (18) we have used the fact that multiplying the unitary matrix $\mathbf{Q}^{\oplus}$ (from the QR factorization) preserves the norm of a vector. By minimizing (20), the solution is given by:

$$\widetilde{\mathbf{x}} = {\mathbf{R}^{\oplus}}^{-1}\mathbf{r}_1^{\oplus} \tag{21}$$

and hence the updated state estimate and information factor are given by $\hat{\mathbf{x}}^{\oplus} = \hat{\mathbf{x}} + \widetilde{\mathbf{x}}$ and $\mathbf{R}^{\oplus}$, respectively. As mentioned before, this process can be repeated iteratively till convergence to reduce linearization errors.

As compared to the IF in the Hessian form, the SR-IF operates on the square-root factor domain, and the update employs QR factorization on the sqaure-root factor instead of Cholesky factorization on the Hessian. This provides better numerical stability and acuuracy, but at the expense of slower processing speed [11].

### 3.4 The Inverse Schmidt Estimator (ISE) Problem Formulation

Motivated by the potential processing savings of the SKF, as well as the low-memory requirements of the Hessian (or equivalently its Cholesky factorization) representation of the uncertainty, in this work, we seek to derive the Schmidt estimator, but in the information domain. Specifically, our objective is to find the exact inverse Schmidt estimator (ISE), i.e., to derive an algorithm that is exactly equivalent to the SKF, while operating in the information domain.

As mentioned in the previous subsections, an estimator in the information domain can take two forms: Either the Hessian form (e.g., the IF) or the square-root form (e.g., the SR-IF). For the task of finding the inverse equivalent of the SKF, however, the Hessian form does not seem to provide any computational savings, because all the blocks of the Hessian matrix will need to be updated. To be specific, if we compute the inverse of the SKF's updated covariance matrix in (6), the corresponding updated Schmidt Hessian matrix will require changes in all of its four blocks, as compared to the prior Hessian before the Schmidt update. This is in contrast to the SKF, as the (potentially very large) portion of $\mathbf{P}_{22}$ remains unchanged, from which the SKF achieves computational savings. As a result, compared to the optimal IF, the Schmidt approximation in the Hessian form will not lead to any processing savings, while its solution is only suboptimal, and hence is meaningless.

Instead, as shown later, the Schmidt approximation preserves the corresponding portion ($\mathbf{R}_{22}$) of the square-root information factor, and hence may bring potential computational benefits due to the saved work for this portion. Therefore, in this work, we focus on finding the information-domain Schmidt estimators in the square-root inverse form. And from this point on, by saying "inverse Schmidt estimator", we refer to only the square-root inverse form estimator that operates on the Cholesky factor of the Hessian.

So within the square-root inverse form, the problem of finding the exact square-root inverse Schmidt estimator (SR-ISE, or simply ISE) can be formulated as follows: Given the (invertible and upper-triangular) prior information factor $\mathbf{R}$ and the measurement Jacobian $\mathbf{H}$:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0}_{n_2 \times n_1} & \mathbf{R}_{22} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \tag{22}$$

where $\mathbf{R}_{11} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{R}_{12} \in \mathbb{R}^{n_1 \times n_2}$, $\mathbf{R}_{22} \in \mathbb{R}^{n_2 \times n_2}$, $\mathbf{H}_1 \in \mathbb{R}^{m \times n_1}$, and $\mathbf{H}_2 \in \mathbb{R}^{m \times n_2}$, find the algorithm that computes the updated Schmidt information factor $\mathbf{R}^s \in \mathbb{R}^{n \times n}$ (invertible and upper-triangular as well), such that it reflects the covariance matrix of the SKF [see (6)], i.e.,

$$\mathbf{P}^{s^{-1}} = \mathbf{R}^{s^T} \mathbf{R}^s, \quad \text{with } \mathbf{R}^s = \begin{bmatrix} \mathbf{R}^s_{11} & \mathbf{R}^s_{12} \\ \mathbf{0}_{n_2 \times n_1} & \mathbf{R}^s_{22} \end{bmatrix} \tag{23}$$

with the same block sizes as defined in $\mathbf{R}$. Similarly for the state update: The updated state estimate $\hat{\mathbf{x}}^s$ should be equal to that of the SKF [see (5)]. Note that there are several basic underlying requirements of the algorithm: At any step, the algorithm (i) should not compute explicitly the covariance matrix, since we seek an inverse (information-domain) approach, and (ii) should not compute explicitly the information matrix, since we require the square-root form.

## 4 The Exact Inverse Schmidt Estimator (ISE)

In this section, we present the detailed derivation of the exact ISE algorithm. First, we propose a generic framework that operates in the square-root information domain, inspired by the SKF and the SR-IF. Then, we derive several sufficient and necessary conditions, under which this framework becomes equivalent to the SKF. Last, based on one of these equivalent conditions, an efficient exact ISE algorithm is presented.

### 4.1 A Generic Framework

As a first step, our objective is to seek a square-root information-domain algorithm that shares two fundamental properties with the SKF: (i) The $\mathbf{x}_2$ part of the state vector is not updated during the Schmidt process, i.e., the state estimate, $\hat{\mathbf{x}}_2$, and the corresponding covariance block, $\mathbf{P}_{22}$, remain intact [see (5)

and (6)]; (ii) It is a consistent approximation of the optimal SR-IF, i.e., the resulting inferred covariance is conservative as compared to that of the SR-IF [see (12)].

Our proposed algorithm follows the approach of the SR-IF, where a unitary transformation, defined by the QR factor $\mathbf{Q}^{\oplus}$, is performed on the cost function [see (18)]. In our algorithm, instead of employing this specific unitary matrix $\mathbf{Q}^{\oplus}$, a generic unitary matrix $\breve{\mathbf{U}}$ is used. Specifically, in terms of the cost function to be minimized [see (17)]:

$$\mathcal{C} = ||\mathbf{R}\widetilde{\mathbf{x}} - \mathbf{r}_0||^2 + ||\mathbf{H}\widetilde{\mathbf{x}} - \mathbf{r}||^2 = \left\| \begin{bmatrix} \mathbf{R} \\ \mathbf{H} \end{bmatrix} \widetilde{\mathbf{x}} - \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r} \end{bmatrix} \right\|^2 \tag{24}$$

$$= \left\| \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{0}_{n_2 \times 1} \\ \mathbf{r} \end{bmatrix} \right\|^2 \tag{25}$$

$$= \left\| \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{r} \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} \right\|^2 \tag{26}$$

$$= \left\| \breve{\mathbf{U}}^T \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \breve{\mathbf{U}}^T \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{r} \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} \right\|^2 \tag{27}$$

where $\breve{\mathbf{U}} \in \mathbb{R}^{(n+m)\times(n+m)}$ is unitary, i.e., $\breve{\mathbf{U}}^T\breve{\mathbf{U}} = \breve{\mathbf{U}}\breve{\mathbf{U}}^T = \mathbf{I}_{n+m}$. Note that from (24) to (25), we have used the fact that $\mathbf{r}_0 = \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{0}_{n_2\times 1} \end{bmatrix}$, i.e., the prior residual has only a nonzero block corresponding to the $\mathbf{x}_1$ part, since $\mathbf{x}_2$ will not be updated in the process.

As mentioned earlier, the first objective is to preserve the state $\mathbf{x}_2$. The following lemma states the condition when this requirement is satisfied:

**Lemma 1.** *Given an invertible and upper-triangular prior information factor* $\mathbf{R}$ *as in* (22)*, and the corresponding prior covariance matrix* $\mathbf{P}$ *as in* (2)*. An algorithm updates the factor into another invertible and upper-triangular matrix* $\mathbf{R}'$*, with the corresponding covariance* $\mathbf{P}'$*. Then the covariance corresponding to* $\mathbf{x}_2$ *remains unchanged, i.e.,* $\mathbf{P}'_{22} = \mathbf{P}_{22}$*, if and only if the corresponding information factor remains unchanged, i.e.,* $\mathbf{R}'_{22} = \mathbf{R}_{22}$ *(up to the sign of each row).*

*Proof.* From (14) and (22), using block matrix inversion, the (2,2) block of the covariance matrix can be written in terms of the information factor blocks as:

$$\mathbf{P} = (\mathbf{R}^T\mathbf{R})^{-1} = \begin{bmatrix} \mathbf{R}_{11}^T\mathbf{R}_{11} & \mathbf{R}_{11}^T\mathbf{R}_{12} \\ \mathbf{R}_{12}^T\mathbf{R}_{11} & \mathbf{R}_{12}^T\mathbf{R}_{12} + \mathbf{R}_{22}^T\mathbf{R}_{22} \end{bmatrix}^{-1} \Rightarrow \mathbf{P}_{22} = \mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T} \tag{28}$$

Similarly, we have $\mathbf{P}'_{22} = \mathbf{R}'^{-1}_{22}\mathbf{R}'^{-T}_{22}$. Hence, $\mathbf{P}'_{22} = \mathbf{P}_{22}$ is equivalent to $\mathbf{R}'^{-1}_{22}\mathbf{R}'^{-T}_{22} = \mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}$, or equivalently $\mathbf{R}'^{T}_{22}\mathbf{R}'_{22} = \mathbf{R}_{22}^{T}\mathbf{R}_{22}$. Since both $\mathbf{R}'_{22}$ and $\mathbf{R}_{22}$ are invertible and upper-triangular matrices, they are the Cholesky factors of the same SPD matrix. Hence, $\mathbf{R}'_{22} = \mathbf{R}_{22}$ (up to the sign of each row) from the uniqueness of the Cholesky factorization of a SPD matrix. □

Therefore, from Lemma 1, in order to preserve $\mathbf{x}_2$, the updated (upper-triangular) information factor should take the form:

$$\mathbf{R}' = \begin{bmatrix} \mathbf{R}'_{11} & \mathbf{R}'_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \tag{29}$$

with the $\mathbf{R}_{22}$ block stays unchanged. Based on this key result, we propose the matrix $\breve{\mathbf{U}}$ to take the following form:

$$\breve{\mathbf{U}} = \begin{bmatrix} \mathbf{U} & \\ & \mathbf{I}_{n_2} \end{bmatrix} \tag{30}$$

where $\mathbf{U} \in \mathbb{R}^{(n_1+m)\times(n_1+m)}$ is a unitary matrix. Note that here $\breve{\mathbf{U}}$ is unitary if and only if $\mathbf{U}$ is unitary. Substituting (30) into (27), the cost function $\mathcal{C}$ becomes:

$$\mathcal{C} = \left\| \begin{bmatrix} \mathbf{U}^T & \\ & \mathbf{I}_{n_2} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{H}_1 & \mathbf{H}_2 \\ \cdots\cdots & \cdots\cdots \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{U}^T & \\ & \mathbf{I}_{n_2} \end{bmatrix} \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{r} \\ \cdots\cdots \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 \tag{31}$$

Partition the columns of $\mathbf{U}$ as $\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix}$, with $\mathbf{U}_1 \in \mathbb{R}^{(n_1+m)\times n_1}$ and $\mathbf{U}_2 \in \mathbb{R}^{(n_1+m)\times m}$, and define:

$$\mathbf{R}'_{11} \triangleq \mathbf{U}_1^T \begin{bmatrix} \mathbf{R}_{11} \\ \mathbf{H}_1 \end{bmatrix} \in \mathbb{R}^{n_1\times n_1}, \quad \mathbf{R}'_{12} \triangleq \mathbf{U}_1^T \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{H}_2 \end{bmatrix} \in \mathbb{R}^{n_1\times n_2} \tag{32}$$

$$\mathbf{J}_1 \triangleq \mathbf{U}_2^T \begin{bmatrix} \mathbf{R}_{11} \\ \mathbf{H}_1 \end{bmatrix} \in \mathbb{R}^{m\times n_1}, \quad \mathbf{J}_2 \triangleq \mathbf{U}_2^T \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{H}_2 \end{bmatrix} \in \mathbb{R}^{m\times n_2}, \quad \mathbf{J} \triangleq \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 \end{bmatrix} \in \mathbb{R}^{m\times n} \tag{33}$$

$$\mathbf{r}' \triangleq \mathbf{U}_1^T \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{r} \end{bmatrix} \in \mathbb{R}^{n_1\times 1}, \quad \mathbf{r}_J \triangleq \mathbf{U}_2^T \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{r} \end{bmatrix} \in \mathbb{R}^{m\times 1} \tag{34}$$

$$\implies \begin{bmatrix} \mathbf{R}'_{11} & \mathbf{R}'_{12} \\ \mathbf{J}_1 & \mathbf{J}_2 \end{bmatrix} = \mathbf{U}^T \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{r}' \\ \mathbf{r}_J \end{bmatrix} = \mathbf{U}^T \begin{bmatrix} \mathbf{r}_0^1 \\ \mathbf{r} \end{bmatrix} \tag{35}$$

then the cost function $\mathcal{C}$ in (31) can be written as:

$$\mathcal{C} = \left\| \begin{bmatrix} \mathbf{R}'_{11} & \mathbf{R}'_{12} \\ \mathbf{J}_1 & \mathbf{J}_2 \\ \cdots\cdots & \cdots\cdots \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{r}' \\ \mathbf{r}_J \\ \cdots\cdots \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 \tag{36}$$

$$= \left\| \begin{bmatrix} \mathbf{R}'_{11} & \mathbf{R}'_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1 \\ \widetilde{\mathbf{x}}_2 \end{bmatrix} - \mathbf{r}_J \right\|^2 \tag{37}$$

$$= \left\| \mathbf{R}'\widetilde{\mathbf{x}} - \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 + \left\| \mathbf{J}\widetilde{\mathbf{x}} - \mathbf{r}_J \right\|^2 \tag{38}$$

with $\mathbf{R}'$ defined in (29). Note that, at this point, the cost function is still equal to the original cost $\mathcal{C}$ as in (17) (since any unitary transformation will not change the cost value), and minimizing it would generate the exact same update results as in the optimal SR-IF.

With the expression of the cost function $\mathcal{C}$ in (38), we are ready to introduce the necessary approximation step in our proposed algorithm, in order to imitate the SKF: The second term in the cost function is discarded. As a result, the new cost function $\mathcal{C}'$ consists of only the first term in (38), i.e.,

$$\mathcal{C} = \left\| \mathbf{R}'\widetilde{\mathbf{x}} - \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 + \left\| \mathbf{J}\widetilde{\mathbf{x}} - \mathbf{r}_J \right\|^2 \quad \implies \quad \mathcal{C}' = \left\| \mathbf{R}'\widetilde{\mathbf{x}} - \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 \tag{39}$$

which leads to the new optimization problem:

$$\min_{\widetilde{\mathbf{x}}} \quad \left\| \mathbf{R}'\widetilde{\mathbf{x}} - \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \right\|^2 \tag{40}$$

and from which the solution (state correction) is given by:

$$\widetilde{\mathbf{x}}^\star = \mathbf{R}'^{-1} \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}'_{11} & \mathbf{R}'_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}'^{-1}_{11}\mathbf{r}' \\ \mathbf{0}_{n_2\times 1} \end{bmatrix} \tag{41}$$

and hence the state update can be written as:

$$\hat{\mathbf{x}}' = \hat{\mathbf{x}} + \widetilde{\mathbf{x}}^\star = \begin{bmatrix} \hat{\mathbf{x}}_1 + \mathbf{R}'^{-1}_{11}\mathbf{r}' \\ \hat{\mathbf{x}}_2 \end{bmatrix} \tag{42}$$

Clearly, only the $\mathbf{x}_1$ portion of the state vector has been changed, while $\mathbf{x}_2$ remains the same. Meanwhile, the updated information factor is given by $\mathbf{R}'$ [from (40)], and from its expression in (29), the block $\mathbf{R}_{22}$ stays unchanged, and hence, so as the corresponding covariance block $\mathbf{P}_{22}$ due to Lemma 1. Note that, in the above analysis, we have assumed that the updated factor block $\mathbf{R}'_{11}$ is invertible, which is a fundamental requirement for the proposed procedure to be a valid update.

Lastly, as desired, this approach is a consistent approximation to the optimal SR-IF, due to the step of discarding the second cost term in (38). Specifically, dropping a cost term corresponds to ignoring the information contained in it, and hence leads to less amount of information to be absorbed in the update, or equivalently larger uncertainty (covariance) in the updated estimates. This result matches the fact that the SKF drops information as well, as shown in (11), and hence justifies the necessity of this approximation step. The consistency of the proposed algorithm is stated formally in the following lemma:

**Lemma 2.** *Given an invertible prior information factor* $\mathbf{R}$ *and a measurement Jacobian* $\mathbf{H}$*. Let* $\mathbf{P}^{\oplus}$ *denote the posterior covariance matrix given by an optimal approach (e.g., KF or SR-IF), and let* $\mathbf{P}'$ *denote the covariance matrix corresponding to the updated information factor* $\mathbf{R}'$ *in* (40)*. Then* $\mathbf{P}' \geq \mathbf{P}^{\oplus}$*, in the matrix positive-semidefinite sense.*

*Proof.* The posterior covariance matrix represents the uncertainty of the posterior state estimates considering all available information, i.e.,

$$\mathbf{P}^{\oplus} = (\mathbf{R}^T\mathbf{R} + \mathbf{H}^T\mathbf{H})^{-1} \tag{43}$$

and by the same arguments from (24) to (38), i.e., by introducing the unitary transformation defined in (30), the posterior covariance matrix is equal to:

$$\mathbf{P}^{\oplus} = (\mathbf{R}^T\mathbf{R} + \mathbf{H}^T\mathbf{H})^{-1} = (\mathbf{R}'^T\mathbf{R}' + \mathbf{J}^T\mathbf{J})^{-1} \tag{44}$$

Meanwhile, by definition, $\mathbf{P}' = (\mathbf{R}'^T\mathbf{R}')^{-1}$ from (40). Hence, by the matrix inversion lemma, we obtain:

$$\mathbf{P}' - \mathbf{P}^{\oplus} = (\mathbf{R}'^T\mathbf{R}')^{-1} - (\mathbf{R}'^T\mathbf{R}' + \mathbf{J}^T\mathbf{J})^{-1} = (\mathbf{R}'^T\mathbf{R}')^{-1}\mathbf{J}^T\left[\mathbf{I}_m + \mathbf{J}(\mathbf{R}'^T\mathbf{R}')^{-1}\mathbf{J}^T\right]^{-1}\mathbf{J}(\mathbf{R}'^T\mathbf{R}')^{-1} \geq \mathbf{0} \tag{45}$$

□

**Algorithm 1** A Generic Framework

1: **Input:** Current state estimate $\hat{\mathbf{x}}$, prior information factor $\mathbf{R}$ and residual $\mathbf{r}_0$, pre-whitened measurement Jacobian $\mathbf{H}$ and residual $\mathbf{r}$
2: **procedure** UPDATE
3: Obtain a unitary matrix: $\mathbf{U} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}$, such that $\mathbf{R}'_{11}$ is invertible and upper-triangular
4: Perform the unitary transformation: $\mathbf{R}'_{11} \leftarrow \mathbf{U}_1^T\begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix}$, $\mathbf{R}'_{12} \leftarrow \mathbf{U}_1^T\begin{bmatrix}\mathbf{R}_{12}\\ \mathbf{H}_2\end{bmatrix}$, $\mathbf{r}' \leftarrow \mathbf{U}_1^T\begin{bmatrix}\mathbf{r}_0^1\\ \mathbf{r}\end{bmatrix}$
5: Information factor update: $\mathbf{R}' \leftarrow \begin{bmatrix}\mathbf{R}'_{11} & \mathbf{R}'_{12}\\ \mathbf{0} & \mathbf{R}_{22}\end{bmatrix}$
6: State update: $\hat{\mathbf{x}}' \leftarrow \begin{bmatrix}\hat{\mathbf{x}}_1 + \mathbf{R}'^{-1}_{11}\mathbf{r}'\\ \hat{\mathbf{x}}_2\end{bmatrix}$
7: **end procedure**
8: **Output:** Updated state estimate $\hat{\mathbf{x}}'$ and information factor $\mathbf{R}'$

To summarize, our proposed approach first combines the prior knowledge with the measurement information [see (24)], then decomposes part of the combined information into two pieces by projecting it onto the two block columns of a unitary matrix [see (31) - (38)], and last drops one of these two resulting information pieces to obtain the final updated factor [see (39)]. This procedure is shown in Alg. 1. Note that, as mentioned before and written out in Alg. 1, we need to guarantee that the updated information factor is meaningful and structured, i.e., the final factor $\mathbf{R}'$ is invertible and upper-triangular. Based on the assumption that the input factor $\mathbf{R}$, and hence the $\mathbf{R}_{22}$ block, is invertible and upper-triangular, this requirement is satisfied if and only if the resulting $\mathbf{R}'_{11}$ block is invertible and upper-triangular. As a result, this requirement poses basic constraints on the matrix $\mathbf{U}_1$, in addition to having orthonormal columns.

To conclude, Alg. 1 satisfies the following desired properties (regardless of the choice of the unitary matrix $\mathbf{U}$): (i) It preserves the estimate and the information factor block of $\mathbf{x}_2$ (and hence the corresponding covariance block), and (ii) it is a consistent approximation of the optimal SR-IF. These are the two fundamental properties shared with the SKF. Note that, Alg. 1 presents a generic framework in the sense that it represents a family of algorithms by choosing a different matrix $\mathbf{U}$. One could potentially employ this framework to obtain an algorithm that generates an approximate solution and achieves computational savings, as compared to the SR-IF, due to the fact that $\mathbf{x}_2$ and $\mathbf{R}_{22}$ are not updated. In what follows, we show the procedure to obtain the specific $\mathbf{U}$ (either explicitly or implicitly) that leads to the exact square-root inverse Schmidt estimator, which was our initial goal. Later on, we also present alternative algorithms, special cases of this generic framework that have lower processing requirements and cause fewer fill-ins, when the factor is sparse.

## 4.2 Exact ISE: Sufficient and Necessary Conditions

Under the framework of Alg. 1, our objective is to find the specific matrix $\mathbf{U}$ such that this algorithm becomes the exact ISE, i.e., the updated state estimates and covariance matrix are equal to those of the SKF. Note that, since Alg. 1 is a square-root inverse approach, it does not compute explicitly the covariance matrix, but instead, the information factor. Hence, by saying the updated covariance of Alg. 1, we refer to the underlying covariance matrix corresponding to the updated information factor, as can be computed using (14). Moreover, from this point on, we assume that the input prior factor $\mathbf{R}$ is invertible and upper-triangular, and will not repeat this assumption in our subsequent statements.

So far, we have considered only the equivalence with respect to the $\mathbf{x}_2$ portion of the entire state (i.e., the *unchanged* part). What remains to be accomplished is to establish the same equivalence with respect to $\mathbf{x}_1$ (i.e., the *changed*, or updated part). For this reason, we analyze the condition on the equivalence of the covariance blocks that are changed in the SKF update, i.e., the $(1,1)$, $(1,2)$, and $(2,1)$ blocks. The result is stated in the following lemma:

**Lemma 3.** *Let $\mathbf{P}' \triangleq (\mathbf{R}'^T\mathbf{R}')^{-1}$ denote the covariance matrix corresponding to the updated information factor $\mathbf{R}'$ of Alg. 1, and let $\mathbf{P}^s$ denote the updated covariance matrix of the SKF. Then*

$$\mathbf{P}'_{11} = \mathbf{P}^s_{11}, \quad \mathbf{P}'_{12} = \mathbf{P}^s_{12}, \quad \textit{and} \quad \mathbf{P}'_{21} = \mathbf{P}^s_{21} \tag{46}$$

*if and only if $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0}_{m\times n_1}$, where $\mathbf{P}'_{(:,1)} \triangleq \begin{bmatrix}\mathbf{P}'_{11}\\ \mathbf{P}'_{21}\end{bmatrix}$ denotes the first block column of the matrix $\mathbf{P}'$.*

*Proof.* In the proof of Lemma 2, we have shown that [see (45)]:

$$\mathbf{P}' - \mathbf{P}^{\oplus} = (\mathbf{R}'^T\mathbf{R}')^{-1}\mathbf{J}^T\left[\mathbf{I}_m + \mathbf{J}(\mathbf{R}'^T\mathbf{R}')^{-1}\mathbf{J}^T\right]^{-1}\mathbf{J}(\mathbf{R}'^T\mathbf{R}')^{-1} \tag{47}$$

where $\mathbf{P}^{\oplus}$ denotes the posterior covariance after the KF update and $\mathbf{J}$ is defined in (33). Substituting $\mathbf{P}' = (\mathbf{R}'^T\mathbf{R}')^{-1}$ gives:

$$\mathbf{P}' - \mathbf{P}^{\oplus} = \mathbf{P}'\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}' \tag{48}$$

If we partition the block columns of $\mathbf{P}'$ corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ as $\mathbf{P}' = \begin{bmatrix}\mathbf{P}'_{(:,1)} & \mathbf{P}'_{(:,2)}\end{bmatrix}$, then we obtain:

$$\begin{aligned}\mathbf{P}' - \mathbf{P}^{\oplus} &= \begin{bmatrix}\mathbf{P}'^T_{(:,1)}\\ \mathbf{P}'^T_{(:,2)}\end{bmatrix}\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\begin{bmatrix}\mathbf{P}'_{(:,1)} & \mathbf{P}'_{(:,2)}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{P}'^T_{(:,1)}\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}'_{(:,1)} & \mathbf{P}'^T_{(:,1)}\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}'_{(:,2)}\\ \mathbf{P}'^T_{(:,2)}\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}'_{(:,1)} & \mathbf{P}'^T_{(:,2)}\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}'_{(:,2)}\end{bmatrix}\end{aligned} \tag{49}$$

On the other hand, the SKF and the KF have the same updated covariance blocks $(1,1)$, $(1,2)$, and $(2,1)$ [see (6)]:

$$\mathbf{P}^s_{11} = \mathbf{P}^{\oplus}_{11}, \quad \mathbf{P}^s_{12} = \mathbf{P}^{\oplus}_{12}, \quad \text{and} \quad \mathbf{P}^s_{21} = \mathbf{P}^{\oplus}_{21} \tag{50}$$

Therefore, (46) is equivalent to:

$$\mathbf{P}'_{11} = \mathbf{P}^{\oplus}_{11}, \quad \mathbf{P}'_{12} = \mathbf{P}^{\oplus}_{12}, \quad \text{and} \quad \mathbf{P}'_{21} = \mathbf{P}^{\oplus}_{21} \tag{51}$$

or in the block matrix form as:

$$\mathbf{P}' - \mathbf{P}^{\oplus} = \begin{bmatrix} \mathbf{0}_{n_1 \times n_1} & \mathbf{0}_{n_1 \times n_2} \\ \mathbf{0}_{n_2 \times n_1} & * \end{bmatrix} \tag{52}$$

which holds true if and only if $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0}$ in (49), since $(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}$ is a SPD matrix. □

As Lemma 3 demands, in order to absorb all the information of $\mathbf{x}_1$ (as is the case of the SKF), the Jacobian matrix $\mathbf{J}$ of the discarded information must be orthogonal to the directions of the updated covariance blocks corresponding to $\mathbf{x}_1$.

At this point, with Lemma 1 and Lemma 3 in place, we are ready to state our first main result, that presents a *sufficient and necessary* condition for finding the exact ISE as a mathematical equivalent to the SKF, under the framework of Alg. 1:

**Theorem 1.** *Given the same input, Alg. 1 and the SKF output the same updated state estimates and covariance matrices, respectively, i.e.,* $\hat{\mathbf{x}}' = \hat{\mathbf{x}}^s$ *and* $\mathbf{P}' = \mathbf{P}^s$*, if and only if* $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0}$.

*Proof.* (i) Necessity: The necessity of the condition follows directly from Lemma 3: If $\mathbf{P}' = \mathbf{P}^s$, then the three sub-blocks must equal, as in (46). Hence, the condition $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0}$ holds from its necessity in Lemma 3.

(ii) Sufficiency: (a) As for the covariance, from the condition's sufficiency in Lemma 3, we know that (46) holds, i.e., the $(1,1)$, $(1,2)$, and $(2,1)$ blocks of the covariance are equal. The last remaining block, $(2,2)$, are equal as a property of Alg. 1 that we have shown earlier: Since $\mathbf{R}'_{22} = \mathbf{R}_{22}$ (see the information factor update in Alg. 1), $\mathbf{P}'_{22} = \mathbf{P}_{22} = \mathbf{P}^s_{22}$ holds true from Lemma 1 and the property of the SKF [see (6)]. Hence, $\mathbf{P}' = \mathbf{P}^s$ holds as all the (four) sub-blocks of the covariance are equal. (b) As for the state estimates, $\hat{\mathbf{x}}'_2 = \hat{\mathbf{x}}_2 = \hat{\mathbf{x}}^s_2$ holds, again because of the property of Alg. 1 (see the state update in Alg. 1) and the SKF [see (5)]. To show that $\hat{\mathbf{x}}'_1 = \hat{\mathbf{x}}^s_1$, we start from the fact that $\hat{\mathbf{x}}^s_1 = \hat{\mathbf{x}}^{\oplus}_1$, where $\hat{\mathbf{x}}^{\oplus}$ denotes the updated state estimates of the KF [see (5)], or equivalently of the SR-IF. As we have shown earlier, the SR-IF updates the state by solving the optimization problem (15), where the cost function $\mathcal{C}$ can be written equivalently as in (38) [see the arguments from (24) to (38)]. By minimizing this least-squares cost function, the state correction computed by the SR-IF can be obtained through the normal equation as:

$$\begin{aligned} \widetilde{\mathbf{x}}^{\oplus} &= (\mathbf{R}'^T\mathbf{R}' + \mathbf{J}^T\mathbf{J})^{-1}(\mathbf{R}'^T \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} + \mathbf{J}^T\mathbf{r}_J) \\ &= (\mathbf{P}'^{-1} + \mathbf{J}^T\mathbf{J})^{-1}(\mathbf{R}'^T \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} + \mathbf{J}^T\mathbf{r}_J) \\ &= \left[\mathbf{P}' - \mathbf{P}'\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}'\right] (\mathbf{R}'^T \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} + \mathbf{J}^T\mathbf{r}_J) \end{aligned} \tag{53}$$

From the condition $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0} \iff \mathbf{P}'_{(1,:)}\mathbf{J}^T = \mathbf{0}$, since $\mathbf{P}'$ is symmetric $\iff \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \mathbf{P}'\mathbf{J}^T = \mathbf{0}$, and the state partitioning $\widetilde{\mathbf{x}}^{\oplus} = \begin{bmatrix} \widetilde{\mathbf{x}}^{\oplus}_1 \\ \widetilde{\mathbf{x}}^{\oplus}_2 \end{bmatrix} \Longrightarrow \widetilde{\mathbf{x}}^{\oplus}_1 = \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \widetilde{\mathbf{x}}^{\oplus}$, we obtain:

$$\begin{aligned} \widetilde{\mathbf{x}}^{\oplus}_1 &= \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \widetilde{\mathbf{x}}^{\oplus} = \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \left[\mathbf{P}' - \mathbf{P}'\mathbf{J}^T(\mathbf{I}_m + \mathbf{J}\mathbf{P}'\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{P}'\right] (\mathbf{R}'^T \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} + \mathbf{J}^T\mathbf{r}_J) \\ &= \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \mathbf{P}'\mathbf{R}'^T \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} (\mathbf{R}'^T\mathbf{R}')^{-1}\mathbf{R}'^T \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \mathbf{R}'^{-1} \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_1} & \mathbf{0}_{n_1 \times n_2} \end{bmatrix} \begin{bmatrix} \mathbf{R}'_{11} & \mathbf{R}'_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}' \\ \mathbf{0}_{n_2 \times 1} \end{bmatrix} = \mathbf{R}'^{-1}_{11}\mathbf{r}' \end{aligned} \tag{54}$$

which leads to the updated state estimate $\hat{\mathbf{x}}^{\oplus}_1$ of the SR-IF as:

$$\hat{\mathbf{x}}^{\oplus}_1 = \hat{\mathbf{x}}_1 + \widetilde{\mathbf{x}}^{\oplus}_1 = \hat{\mathbf{x}}_1 + \mathbf{R}'^{-1}_{11}\mathbf{r}' \tag{55}$$

Comparing this to the state update in Alg. 1, we can see that $\hat{\mathbf{x}}'_1 = \hat{\mathbf{x}}^{\oplus}_1 = \hat{\mathbf{x}}^s_1$. Combining it with the previous result $\hat{\mathbf{x}}'_2 = \hat{\mathbf{x}}^s_2$, we conclude that $\hat{\mathbf{x}}' = \hat{\mathbf{x}}^s$, as both (two) sub-blocks of the state estimates are equal. □

Recall that, the one and only degree of freedom in Alg. 1 is the unitary matrix $\mathbf{U}$, and by choosing different $\mathbf{U}$ matrices, it results in different matrices $\mathbf{J}$ and $\mathbf{R}'$ (hence $\mathbf{P}'$) [see (35)]. Therefore, as Theorem 1 demands, in order to find the exact ISE as a specific realization of Alg. 1, the task now becomes to seek a unitary matrix $\mathbf{U}$ such that the condition $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0}$ is satisfied. This condition poses an *implicit* constraint on the matrix $\mathbf{U}$, and in order to come up with an actual algorithm, we have further derived several other equivalent but *explicit* constraints, based on this one. The result is stated in the following lemma:

**Lemma 4.** *Assume that* $\mathbf{U} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}$ *is a unitary matrix. Define the following matrices:*

$$\mathbf{G}_1 \triangleq \begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix} \in \mathbb{R}^{(n_1+m)\times n_1}, \quad \mathbf{G}_2 \triangleq \begin{bmatrix}\mathbf{R}_{12}\\ \mathbf{H}_2\end{bmatrix} \in \mathbb{R}^{(n_1+m)\times n_2}, \quad \mathbf{A} \triangleq \mathbf{R}_{22}^{-T}\mathbf{G}_2^T \in \mathbb{R}^{n_2\times(n_1+m)} \tag{56}$$

*Then the following statements are equivalent:*

*(C1)* $\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0}$.

*(C2)* $\left[\mathbf{G}_1(\mathbf{U}_1^T\mathbf{G}_1)^{-1}\mathbf{U}_1^T - \mathbf{I}_{n_1+m}\right](\mathbf{I}_{n_1+m} + \mathbf{A}^T\mathbf{A})\,\mathbf{U}_1 = \mathbf{0}$.

*(C3) The columns of* $\mathbf{U}_1$ *form an orthonormal basis for the column space of* $(\mathbf{I}_{n_1+m} + \mathbf{A}^T\mathbf{A})^{-1}\mathbf{G}_1$.

*(C4) The columns of* $\mathbf{U}_1$ *form an orthonormal basis for the right null space of* $\mathbf{Q}_2^T(\mathbf{I}_{n_1+m} + \mathbf{A}^T\mathbf{A})$*, where the columns of* $\mathbf{Q}_2$ *form a basis for the left null space of* $\mathbf{G}_1$.

*Proof. (C1)* $\Longleftrightarrow$ *(C2)*: From the definition of $\mathbf{P}'$, and the expression of $\mathbf{R}'$ in (29), we obtain:

$$\mathbf{P}' = (\mathbf{R}'^T\mathbf{R}')^{-1} = \begin{bmatrix}\mathbf{R}_{11}'^{-1}\mathbf{R}_{11}'^{-T} + \mathbf{R}_{11}'^{-1}\mathbf{R}_{12}'\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\mathbf{R}_{12}'^T\mathbf{R}_{11}'^{-T} & -\mathbf{R}_{11}'^{-1}\mathbf{R}_{12}'\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\\ -\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\mathbf{R}_{12}'^T\mathbf{R}_{11}'^{-T} & \mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\end{bmatrix} \tag{57}$$

which gives the expression of the first block column of $\mathbf{P}'$, i.e., $\mathbf{P}'_{(:,1)}$. Hence,

$$\begin{aligned}
\mathbf{J}\mathbf{P}'_{(:,1)} = \mathbf{0} \quad &\Longleftrightarrow \quad \mathbf{J}_1(\mathbf{R}_{11}'^{-1}\mathbf{R}_{11}'^{-T} + \mathbf{R}_{11}'^{-1}\mathbf{R}_{12}'\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\mathbf{R}_{12}'^T\mathbf{R}_{11}'^{-T}) - \mathbf{J}_2\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\mathbf{R}_{12}'^T\mathbf{R}_{11}'^{-T} = \mathbf{0}\\
&\Longleftrightarrow \quad \mathbf{J}_1(\mathbf{R}_{11}'^{-1} + \mathbf{R}_{11}'^{-1}\mathbf{R}_{12}'\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\mathbf{R}_{12}'^T) - \mathbf{J}_2\mathbf{R}_{22}^{-1}\mathbf{R}_{22}^{-T}\mathbf{R}_{12}'^T = \mathbf{0} \qquad \text{(since } \mathbf{R}_{11}' \text{ is invertible)}\\
&\Longleftrightarrow \quad \mathbf{U}_2^T\underbrace{\left[\mathbf{G}_1(\mathbf{U}_1^T\mathbf{G}_1)^{-1}\mathbf{U}_1^T(\mathbf{I}+\mathbf{A}^T\mathbf{A})\mathbf{U}_1 - \mathbf{A}^T\mathbf{A}\mathbf{U}_1\right]}_{\triangleq\,\mathbf{\Gamma}_1} = \mathbf{0} \qquad \text{(from (32), (33), and (56))}\\
&\Longleftrightarrow \quad \mathbf{U}_2^T\mathbf{\Gamma}_1 = \mathbf{0}\\
&\Longleftrightarrow \quad \mathbf{\Gamma}_1 = \mathbf{U}_1 \qquad \text{(since } \mathbf{U} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix} \text{ is unitary and } \mathbf{U}_1^T\mathbf{\Gamma}_1 = \mathbf{I})\\
&\Longleftrightarrow \quad \mathbf{G}_1(\mathbf{U}_1^T\mathbf{G}_1)^{-1}\mathbf{U}_1^T(\mathbf{I}+\mathbf{A}^T\mathbf{A})\mathbf{U}_1 - \mathbf{A}^T\mathbf{A}\mathbf{U}_1 = \mathbf{U}_1 \qquad \text{(from the definition of } \mathbf{\Gamma}_1)\\
&\Longleftrightarrow \quad \left[\mathbf{G}_1(\mathbf{U}_1^T\mathbf{G}_1)^{-1}\mathbf{U}_1^T - \mathbf{I}\right](\mathbf{I}+\mathbf{A}^T\mathbf{A})\,\mathbf{U}_1 = \mathbf{0}.
\end{aligned}$$

*(C2)* $\Longleftrightarrow$ *(C3)*: Define $\mathbf{\Gamma}_2 \triangleq \mathbf{G}_1(\mathbf{U}_1^T\mathbf{G}_1)^{-1}\mathbf{U}_1^T - \mathbf{I} \in \mathbb{R}^{(n_1+m)\times(n_1+m)}$. We can show that:

$$\begin{aligned}
&\begin{cases}\mathbf{\Gamma}_2\mathbf{G}_1 = \mathbf{0} \implies \dim\{null(\mathbf{\Gamma}_2)\} \geq n_1 & \text{(since } rank(\mathbf{G}_1) = n_1 \text{ because } \mathbf{R}_{11} \text{ is invertible)}\\ \mathbf{\Gamma}_2\mathbf{U}_2 = -\mathbf{U}_2 \implies rank(\mathbf{\Gamma}_2) \geq m & \text{(since } \mathbf{U} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix} \text{ is unitary and } rank(\mathbf{U}_2) = m)\\ rank(\mathbf{\Gamma}_2) + \dim\{null(\mathbf{\Gamma}_2)\} = n_1 + m & \text{(rank-nullity theorem)}\end{cases}\\
&\qquad\qquad \implies \quad \dim\{null(\mathbf{\Gamma}_2)\} = n_1 \qquad (58)\\
&\qquad\qquad \implies \quad col(\mathbf{G}_1) = null(\mathbf{\Gamma}_2) \qquad (59)
\end{aligned}$$

where $col(\cdot)$ denotes the column space of a matrix, while $null(\cdot)$ the right null space. Hence,

$$
\begin{aligned}
& \left[\mathbf{G}_1(\mathbf{U}_1^T\mathbf{G}_1)^{-1}\mathbf{U}_1^T - \mathbf{I}\right](\mathbf{I}+\mathbf{A}^T\mathbf{A})\,\mathbf{U}_1 = \mathbf{0} \\
\iff\ & \mathbf{\Gamma}_2\,(\mathbf{I}+\mathbf{A}^T\mathbf{A})\,\mathbf{U}_1 = \mathbf{0} && \text{(from the definition of } \mathbf{\Gamma}_2) \\
\iff\ & col((\mathbf{I}+\mathbf{A}^T\mathbf{A})\,\mathbf{U}_1) = null(\mathbf{\Gamma}_2) && \text{(from (58) and } rank((\mathbf{I}+\mathbf{A}^T\mathbf{A})\,\mathbf{U}_1) = n_1) \\
\iff\ & col((\mathbf{I}+\mathbf{A}^T\mathbf{A})\,\mathbf{U}_1) = col(\mathbf{G}_1) && \text{(from (59))} \\
\iff\ & col(\mathbf{U}_1) = col((\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{G}_1) && \text{(since } \mathbf{I}+\mathbf{A}^T\mathbf{A} \text{ is invertible)} \\
\iff\ & \text{The columns of } \mathbf{U}_1 \text{ form an orthonormal basis for } col((\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{G}_1). \\
& && \text{(since } \mathbf{U}_1 \text{ has orthonormal columns)}
\end{aligned}
$$

*(C3)* $\iff$ *(C4)*: Given the matrix $\mathbf{Q}_2$, whose columns form a basis for the left null space of $\mathbf{G}_1$, i.e., $\mathbf{Q}_2^T\mathbf{G}_1 = \mathbf{0}$. Since $\mathbf{G}_1 \in \mathbb{R}^{(n_1+m)\times n_1}$ and is of full column rank, the dimension of its left null space must be $m$, and hence, we have $\mathbf{Q}_2 \in \mathbb{R}^{(n_1+m)\times m}$ and is of full column rank as a basic requirement of being a basis. Moreover, since $\mathbf{I}+\mathbf{A}^T\mathbf{A}$ is invertible, we obtain:

$$
\begin{aligned}
&\begin{cases}
(\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{G}_1 \in \mathbb{R}^{(n_1+m)\times n_1}, \text{ with full column rank } n_1 \\
\mathbf{Q}_2^T(\mathbf{I}+\mathbf{A}^T\mathbf{A}) \in \mathbb{R}^{m\times(n_1+m)}, \text{ with full row rank } m \implies \dim\{null(\mathbf{Q}_2^T(\mathbf{I}+\mathbf{A}^T\mathbf{A}))\} = n_1 \\
\mathbf{Q}_2^T\mathbf{G}_1 = [\mathbf{Q}_2^T(\mathbf{I}+\mathbf{A}^T\mathbf{A})]\left[(\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{G}_1\right] = \mathbf{0}
\end{cases} \\
&\qquad\qquad \implies \quad null(\mathbf{Q}_2^T(\mathbf{I}+\mathbf{A}^T\mathbf{A})) = col((\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{G}_1) \qquad (60)
\end{aligned}
$$

which shows that the right null space of $\mathbf{Q}_2^T(\mathbf{I}+\mathbf{A}^T\mathbf{A})$ is the same subspace as the column space of $(\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{G}_1$. Therefore, if the columns of $\mathbf{U}_1$ form an orthonormal basis for either one subspace, so they do for the other. □

Among those conditions listed in Lemma 4, the first one is our starting point (from Theorem 1) as an implicit constraint on the matrix $\mathbf{U}$, the second presents an explicit constraint involving just the $\mathbf{U}_1$ portion, which is the only necessary part needed in Alg. 1, while the last two conditions are direct statements on how to obtain such a matrix $\mathbf{U}_1$. Hence, if one computes a matrix $\mathbf{U}_1$, based on condition *(C3)* or *(C4)*, and substitute it into Alg. 1, then this would give a square-root inverse algorithm that is equivalent to the SKF. On the other hand, in addition to being equivalent to the SKF, our desired ISE in the square-root form has further requirements to be satisfied as mentioned before, i.e., the resulting information factor $\mathbf{R}'$ should be invertible and upper-triangular. As for the invertibility, it turns out that this is automatically guaranteed by the condition *(C3)* or *(C4)*, as stated in the following lemma:

**Lemma 5.** *In Alg. 1, if* $\mathbf{U}_1$ *satisfies the condition (C3) or (C4) in* Lemma 4*, then the updated factor* $\mathbf{R}'$ *is invertible.*

*Proof.* Since the condition *(C3)* and *(C4)* are equivalent [see (60)], it suffices to show the proof for just one of them, and here we choose *(C3)* for simplicity: Since both $\mathbf{G}_1$ and $\mathbf{U}_1$ have full column rank, from the condition *(C3)*, there must exist an invertible matrix $\mathbf{E}$, such that:

$$
\begin{aligned}
& (\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{G}_1 = \mathbf{U}_1\mathbf{E} \\
\implies\ & \mathbf{G}_1 = (\mathbf{I}+\mathbf{A}^T\mathbf{A})\mathbf{U}_1\mathbf{E} \\
\implies\ & \mathbf{R}'_{11} = \mathbf{U}_1^T\mathbf{G}_1 = \mathbf{U}_1^T(\mathbf{I}+\mathbf{A}^T\mathbf{A})\mathbf{U}_1\mathbf{E} = (\mathbf{I}+\mathbf{U}_1^T\mathbf{A}^T\mathbf{A}\mathbf{U}_1)\mathbf{E} \qquad (61)
\end{aligned}
$$

based on the definition of $\mathbf{R}'_{11}$ in (32) and that $\mathbf{U}_1$ has orthonormal columns as state in condition *(C3)*. Therefore, from (61), we can see that $\mathbf{R}'_{11}$ must be invertible as a product of two invertible matrices. Moreover, given the assumption that $\mathbf{R}$ is invertible, and hence $\mathbf{R}_{22}$ as well, we conclude that $\mathbf{R}'$ is invertible as a block upper-triangular matrix with invertible diagonal blocks [see its expression in (29)]. □

As for the upper-triangular structure of $\mathbf{R}'$, it poses an additional constraint on the matrix $\mathbf{U}_1$. Note that, the conditions in Lemma 4 do not give a unique choice of $\mathbf{U}_1$. In fact, there are infinitely many matrices that will satisfy these conditions: It can be verified that, if a matrix $\mathbf{U}_1^\star$ satisfies the conditions in Lemma 4,

then so does the matrix $\mathbf{U}_1^\star\mathbf{Q}$, where $\mathbf{Q} \in \mathbb{R}^{n_1\times n_1}$ is any arbitrary unitary matrix. This is reasonable because, for the underlying Schmidt Hessian matrix, its corresponding square-root factor [see (14)] is nonunique, if no other requirement of the factor is specified. With the additional restriction that $\mathbf{R}'$ be upper-triangular, or equivalently $\mathbf{R}'_{11}$, the choice of $\mathbf{U}_1$ is then unique (up to the sign of each column), due to the uniqueness of the Cholesky factorization of a SPD matrix (i.e., the Schmidt Hessian). Hence, as our goal is to obtain an exact ISE in the square-root form that maintains an upper-triangular information factor, we need to add this additional constraint on $\mathbf{U}_1$.

To conclude, we summarize the results of our previous analyses with the following statement:

**Theorem 2.** *Alg. 1 is an exact ISE in the square-root form, with the updated information factor being invertible and upper-triangular, if and only if the matrix* $\mathbf{U}_1$ *satisfies the condition (C3) or (C4) in* Lemma 4*, and* $\mathbf{R}'_{11} = \mathbf{U}_1^T\mathbf{G}_1$ *is upper-triangular.*

Theorem 2 is a direct combination of the results of Theorem 1, Lemma 4, and Lemma 5, hence no further proof is needed. The sufficient and necessary condition stated here consists of two parts: The first part guarantees the equivalence to the SKF and the invertibility of the updated factor, while the second regularizes the upper-triangular structure of the factor as required. Theorem 2 provides useful guidelines for realizing the exact ISE algorithm. In what follows, we utilize this result and present our proposed exact ISE algorithm that is both computationally efficient and numerically stable.

### 4.3 Exact ISE: The Algorithm

The condition stated in Theorem 2 can lead directly to the exact ISE algorithm. In fact, starting from either one of the constraints *(C3)* and *(C4)* on the matrix $\mathbf{U}_1$, we have found several versions of the algorithm. All these realizations are mathematical equivalents of the same exact ISE, but meanwhile, they differ in terms of the computational efficiency and numerical stability.

As an example, since the condition in Theorem 2 consists of two parts, one approach would accordingly decompose the matrix $\mathbf{U}_1$ into the following form:

$$\mathbf{U}_1 = \mathbf{U}'_1\mathbf{U}''_1, \quad \text{with} \;\; \mathbf{U}'_1 \in \mathbb{R}^{(n_1+m)\times n_1} \;\; \text{and} \;\; \mathbf{U}''_1 \in \mathbb{R}^{n_1\times n_1} \tag{62}$$

where $\mathbf{U}'_1$ satisfies the constraint *(C3)* or *(C4)*, while the unitary matrix $\mathbf{U}''_1$ is chosen such that the resulting factor is upper-triangular. It is straightforward to verify that their product $\mathbf{U}_1$ obeys the condition in Theorem 2. Therefore, this is a valid approach, and the corresponding exact ISE algorithm can be carried out as (if the constraint *(C3)* is selected, for instance):

1. Obtain $\mathbf{G}_1 \leftarrow \begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix}$ and $\mathbf{G}_2 \leftarrow \begin{bmatrix}\mathbf{R}_{12}\\ \mathbf{H}_2\end{bmatrix}$
2. Compute $\mathbf{A} \leftarrow \mathbf{R}_{22}^{-T}\mathbf{G}_2^T$
3. Compute $\mathbf{\Gamma} \leftarrow (\mathbf{I}_{n_1+m} + \mathbf{A}^T\mathbf{A})^{-1}\mathbf{G}_1$
4. Compute the thin QR factorization of $\mathbf{\Gamma} = \mathbf{U}'_1\mathbf{R}_\Gamma$, then compute $\mathbf{R}'_{11} \leftarrow \mathbf{U}_1'^T\mathbf{G}_1$, $\mathbf{R}'_{12} \leftarrow \mathbf{U}_1'^T\mathbf{G}_2$, and $\mathbf{r}' \leftarrow \mathbf{U}_1'^T\begin{bmatrix}\mathbf{r}_0^1\\ \mathbf{r}\end{bmatrix}$
5. Compute the QR factorization of $\mathbf{R}'_{11} = \mathbf{U}''_1\mathbf{R}^s_{11}$, then compute $\mathbf{R}^s_{12} \leftarrow \mathbf{U}_1''^T\mathbf{R}'_{12}$ and $\mathbf{r}^s \leftarrow \mathbf{U}_1''^T\mathbf{r}'$
6. Information factor update: $\mathbf{R}^s \leftarrow \begin{bmatrix}\mathbf{R}^s_{11} & \mathbf{R}^s_{12}\\ \mathbf{0}_{n_2\times n_1} & \mathbf{R}_{22}\end{bmatrix}$
7. State update: $\hat{\mathbf{x}}^s \leftarrow \begin{bmatrix}\hat{\mathbf{x}}_1 + \mathbf{R}_{11}^{s^{-1}}\mathbf{r}^s\\ \hat{\mathbf{x}}_2\end{bmatrix}$

where the superscript $s$ is used to denote the fact that this algorithm is the Schmidt estimator, i.e., it is equivalent to the SKF (see the definitions in Sec. 3.4). As evident, this procedure is a special realization of Alg. 1, where a specific $\mathbf{U}_1$ matrix is obtained implicitly based on Theorem 2, as the product of two

matrices: (i) $\mathbf{U}_1'$, whose columns, due to the thin QR factorization in Step 4, become an orthonormal basis for the column space of $\mathbf{\Gamma}$, following the constraint *(C3)*, and (ii) $\mathbf{U}_1''$, which transforms the previous Schmidt factor block $\mathbf{R}_{11}'$ into another equivalent but upper-triangular factor block $\mathbf{R}_{11}^s$, due to the QR factorization in Step 5.

Although this is a correct exact ISE algorithm, it has several drawbacks for numerical implementations: Firstly, the matrices $\mathbf{U}_1'$ and $\mathbf{U}_1''$ are formed explicitly in the two QR processes, hence reducing the computational efficiency. Second, the matrix inversion in Step 3 can cause loss in numerical accuracy. Lastly, the upper-triangular structure of the input factor block $\mathbf{R}_{11}$ is destroyed during the process, i.e., $\mathbf{R}_{11}'$ is in general dense, and hence results in the need of the triangularization step (see Step 5), which brings extra computational cost. While the first issue can be solved by using in-place QR operations so as to gain speed, the last two problems are caused by the approach itself, and cannot be eliminated easily without switching to a different scheme. In what follows, we present the "best" algorithm in terms of the computational efficiency and numerical stability, among many potential algorithms that we have found based on Theorem 2. This proposed method consists of several QR factorizations as its major operations, ensuring superior numerical behaviors. Moreover, the upper-triangular structure of the factor is respected throughout the entire process, achieving high efficiency by taking full advantage of this property.

We start from the constraint *(C4)*. Instead of a direct usage of this constraint to find $\mathbf{U}_1$, however, we first apply a unitary transformation on $\mathbf{G}_1$ and $\mathbf{G}_2$, and then follow the conditions in *(C4)*, but now with respect to the new resulting matrices. Specifically, we first perform the QR factorization of $\mathbf{G}_1$ as:

$$\mathbf{G}_1 = \begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix} = \mathbf{Q}_{G_1}\begin{bmatrix}\bar{\mathbf{R}}_{11}\\ \mathbf{0}_{m\times n_1}\end{bmatrix} \tag{63}$$

$$\implies \left[\mathbf{G}_1 \vdots \mathbf{G}_2\right] \xrightarrow{\text{QR}} \mathbf{Q}_{G_1}^T\left[\mathbf{G}_1 \vdots \mathbf{G}_2\right] = \begin{bmatrix}\bar{\mathbf{R}}_{11} & \vdots \bar{\mathbf{R}}_{12}\\ \mathbf{0}_{m\times n_1} & \vdots \bar{\mathbf{H}}_2\end{bmatrix} = \left[\bar{\mathbf{G}}_1 \vdots \bar{\mathbf{G}}_2\right] \tag{64}$$

$$\text{with} \quad \bar{\mathbf{G}}_1 \triangleq \begin{bmatrix}\bar{\mathbf{R}}_{11}\\ \mathbf{0}_{m\times n_1}\end{bmatrix} = \mathbf{Q}_{G_1}^T\mathbf{G}_1 \ \text{ and } \ \bar{\mathbf{G}}_2 \triangleq \begin{bmatrix}\bar{\mathbf{R}}_{12}\\ \bar{\mathbf{H}}_2\end{bmatrix} \triangleq \mathbf{Q}_{G_1}^T\mathbf{G}_2 \tag{65}$$

where the QR factor $\bar{\mathbf{R}}_{11} \in \mathbb{R}^{n_1\times n_1}$ is guaranteed to be a full-rank upper-triangular matrix, since $\mathbf{G}_1$ is of full column rank. Define the following matrix $\bar{\mathbf{A}}$, similarly to the matrix $\mathbf{A}$ in (56), as:

$$\bar{\mathbf{A}} \triangleq \mathbf{R}_{22}^{-T}\bar{\mathbf{G}}_2^T = \mathbf{A}\mathbf{Q}_{G_1} \tag{66}$$

Now we apply the statement in the constraint *(C4)* to these new inputs, i.e., we find a matrix $\bar{\mathbf{U}}_1$, whose columns form an orthonormal basis for the right null space of $\bar{\mathbf{Q}}_2^T(\mathbf{I} + \bar{\mathbf{A}}^T\bar{\mathbf{A}})$, where the columns of $\bar{\mathbf{Q}}_2$ form a basis for the left null space of $\bar{\mathbf{G}}_1$. The advantage brought by this pre-transformation procedure is twofold: (i) Given the expression of $\bar{\mathbf{G}}_1$, its left null space can be found in closed-form as $\bar{\mathbf{Q}}_2 = \begin{bmatrix}\mathbf{0}_{m\times n_1} & \mathbf{I}_m\end{bmatrix}^T$, and (ii) given the upper-triangular structure of $\bar{\mathbf{G}}_1$, it turns out that there is an efficient way to compute the updated factor block $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_1$, where the upper-triangular structure is always maintained. Specifically, we have:

$$\bar{\mathbf{Q}}_2 = \begin{bmatrix}\mathbf{0}_{m\times n_1} & \mathbf{I}_m\end{bmatrix}^T \implies \bar{\mathbf{Q}}_2^T(\mathbf{I} + \bar{\mathbf{A}}^T\bar{\mathbf{A}}) = \begin{bmatrix}\mathbf{0}_{m\times n_1} & \mathbf{I}_m\end{bmatrix}(\mathbf{I}_{n_1+m} + \bar{\mathbf{A}}^T\bar{\mathbf{A}}) = \bar{\mathbf{B}}^T \tag{67}$$

where we have defined:

$$\bar{\mathbf{A}} \triangleq \begin{bmatrix}\bar{\mathbf{A}}_1 & \bar{\mathbf{A}}_2\end{bmatrix}, \quad \text{with } \ \bar{\mathbf{A}}_1 \in \mathbb{R}^{n_2\times n_1} \ \text{ and } \ \bar{\mathbf{A}}_2 \in \mathbb{R}^{n_2\times m} \tag{68}$$

$$\bar{\mathbf{B}} \triangleq \begin{bmatrix}\bar{\mathbf{B}}_1\\ \bar{\mathbf{B}}_2\end{bmatrix}, \quad \text{with } \ \bar{\mathbf{B}}_1 \triangleq \bar{\mathbf{A}}_1^T\bar{\mathbf{A}}_2 \in \mathbb{R}^{n_1\times m} \ \text{ and } \ \bar{\mathbf{B}}_2 \triangleq \mathbf{I}_m + \bar{\mathbf{A}}_2^T\bar{\mathbf{A}}_2 \in \mathbb{R}^{m\times m} \tag{69}$$

As mentioned above, the columns of $\bar{\mathbf{U}}_1$ should form an orthonormal basis for the right null space of $\bar{\mathbf{Q}}_2^T(\mathbf{I} + \bar{\mathbf{A}}^T\bar{\mathbf{A}}) = \bar{\mathbf{B}}^T$, or equivalently, the left null space of $\bar{\mathbf{B}}$. Note that $\bar{\mathbf{B}} \in \mathbb{R}^{(n_1+m)\times m}$ is of full column rank, since $\bar{\mathbf{B}}_2$ is SPD [see (69)]. Therefore, after $\bar{\mathbf{B}}$ is computed, our next task is to obtain the updated factor blocks $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_1$ and $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_2$, where $\bar{\mathbf{U}}_1$ forms the left null space of $\bar{\mathbf{B}}$. A naive approach would first compute $\bar{\mathbf{U}}_1$ explicitly, and then obtain the factor blocks through matrix multiplications. This is inefficient, and it can be improved by performing a QR factorization on $\bar{\mathbf{B}}$, while computing $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_1$ and $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_2$ through

in-place operations. Moreover, special care must be taken for this QR process, in order to take full advantage of the specific structure of $\bar{\mathbf{G}}_1$ for efficiency. Specifically, a standard QR (e.g., through Householder transformations [11]) would destroy the upper-triangular structure of $\bar{\mathbf{G}}_1$, and results in a dense factor block $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_1$, and hence, a follow-up step will be required to triangularize it. It would be ideal if we can find a way to compute the factor blocks such that the structure is fully exploited, and furthermore, result in an upper-triangular matrix for $\bar{\mathbf{U}}_1^T\bar{\mathbf{G}}_1$ automatically at no extra cost. Indeed, we find that this is possible, through the following two-step approach: First, a QR factorization is performed only on the lower portion of $\bar{\mathbf{B}}$, i.e., the $\bar{\mathbf{B}}_2$ matrix, to triangularize it:

$$\left[\bar{\mathbf{B}} \vdots \bar{\mathbf{G}}_1 \vdots \bar{\mathbf{G}}_2\right] = \begin{bmatrix} \bar{\mathbf{B}}_1 \vdots & \bar{\mathbf{R}}_{11} & \vdots \bar{\mathbf{R}}_{12} \\ \bar{\mathbf{B}}_2 \vdots & \mathbf{0}_{m\times n_1} & \vdots \bar{\mathbf{H}}_2 \end{bmatrix}, \quad \bar{\mathbf{B}}_2 = \bar{\mathbf{Q}}_{B_2}\bar{\mathbf{R}}_{B_2} \tag{70}$$

$$\implies \left[\bar{\mathbf{B}} \vdots \bar{\mathbf{G}}_1 \vdots \bar{\mathbf{G}}_2\right] \xrightarrow{\text{QR}} \begin{bmatrix} \mathbf{I}_{n_1} & \\ & \bar{\mathbf{Q}}_{B_2}^T \end{bmatrix} \left[\bar{\mathbf{B}} \vdots \bar{\mathbf{G}}_1 \vdots \bar{\mathbf{G}}_2\right] = \begin{bmatrix} \bar{\mathbf{B}}_1 \vdots & \bar{\mathbf{R}}_{11} & \vdots \bar{\mathbf{R}}_{12} \\ \bar{\mathbf{R}}_{B_2} \vdots & \mathbf{0}_{m\times n_1} & \vdots \bar{\mathbf{H}}_2' \end{bmatrix} \tag{71}$$

$$\text{with} \quad \bar{\mathbf{H}}_2' \triangleq \bar{\mathbf{Q}}_{B_2}^T\bar{\mathbf{H}}_2 \tag{72}$$

where the QR factor $\bar{\mathbf{R}}_{B_2} \in \mathbb{R}^{m\times m}$ is guaranteed to be a full-rank upper-triangular matrix, since $\bar{\mathbf{B}}_2$ is SPD. Note that, in this step, we have utilized the fact that the lower portion of $\bar{\mathbf{G}}_1$ is zero, and hence it remains to be zero during the in-place QR process. This way, with $\bar{\mathbf{G}}_1$ unchanged, this first step can be carried out with minimal operations. Next, in the second step, we aim to zero out the $\bar{\mathbf{B}}_1$ block, through another in-place QR factorization, to obtain the desired left null space and its multiplication with the stacked factor blocks, but in an implicit way for efficiency. Moreover, as mentioned before, the resulting factor block $(1,1)$ should be guaranteed to be upper-triangular. To achieve this, we employ the QR with Givens rotations, which has the flexibility in terms of the rows involved in each update and also the order in which the zeros are introduced [11], and hence keeping the desired upper-triangular structure. To be specific, based on the fact that $\bar{\mathbf{G}}_1$ and $\bar{\mathbf{R}}_{B_2}$ are both upper-triangular, we propose a Given process with the following rule:

(R1) *Use the diagonal elements of $\bar{\mathbf{R}}_{B_2}$ to zero out all the elements of $\bar{\mathbf{B}}_1$, in the following order: From left to right in terms of columns, and within each column, from bottom to top in terms of rows. Specifically, for each element of $\bar{\mathbf{B}}_1$ with row index $i$ and column index $j$, $i \in \{1,2,\ldots,n_1\}$ and $j \in \{1,2,\ldots,m\}$, perform a Givens rotation to zero out this element, involving the following two rows: The $i$-th row of $\bar{\mathbf{B}}_1$ and the $j$-th row of $\bar{\mathbf{R}}_{B_2}$.*

By tracing the evolution of the non-zero pattern during the Givens process following this rule, it can be verified that, the upper-triangular structure of the $\bar{\mathbf{R}}_{11}$ block is kept all the way to the end, i.e., when $\bar{\mathbf{B}}_1$ becomes completely zero. Mathematically, this Givens QR process can be presented as:

$$\begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{R}}_{B_2} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{U}}_B' & \bar{\mathbf{Q}}_B' \end{bmatrix} \begin{bmatrix} \mathbf{0}_{n_1\times m} \\ \bar{\mathbf{R}}_{B_2}' \end{bmatrix} \tag{73}$$

$$\implies \begin{bmatrix} \bar{\mathbf{B}}_1 \vdots & \bar{\mathbf{R}}_{11} & \vdots \bar{\mathbf{R}}_{12} \\ \bar{\mathbf{R}}_{B_2} \vdots & \mathbf{0}_{m\times n_1} & \vdots \bar{\mathbf{H}}_2' \end{bmatrix} \xrightarrow{\text{QR}} \begin{bmatrix} \bar{\mathbf{U}}_B' & \bar{\mathbf{Q}}_B' \end{bmatrix}^T \begin{bmatrix} \bar{\mathbf{B}}_1 \vdots & \bar{\mathbf{R}}_{11} & \vdots \bar{\mathbf{R}}_{12} \\ \bar{\mathbf{R}}_{B_2} \vdots & \mathbf{0}_{m\times n_1} & \vdots \bar{\mathbf{H}}_2' \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n_1\times m} \vdots & \mathbf{R}_{11}^s \vdots & \mathbf{R}_{12}^s \\ \bar{\mathbf{R}}_{B_2}' \vdots & \mathbf{J}_1^s \vdots & \mathbf{J}_2^s \end{bmatrix} \tag{74}$$

$$\begin{aligned} \text{with} \quad & \mathbf{R}_{11}^s \triangleq \bar{\mathbf{U}}_B'^T \begin{bmatrix} \bar{\mathbf{R}}_{11} \\ \mathbf{0}_{m\times n_1} \end{bmatrix} \in \mathbb{R}^{n_1\times n_1}, \quad \mathbf{R}_{12}^s \triangleq \bar{\mathbf{U}}_B'^T \begin{bmatrix} \bar{\mathbf{R}}_{12} \\ \bar{\mathbf{H}}_2' \end{bmatrix} \in \mathbb{R}^{n_1\times n_2} \\ & \mathbf{J}_1^s \triangleq \bar{\mathbf{Q}}_B'^T \begin{bmatrix} \bar{\mathbf{R}}_{11} \\ \mathbf{0}_{m\times n_1} \end{bmatrix} \in \mathbb{R}^{m\times n_1}, \quad \mathbf{J}_2^s \triangleq \bar{\mathbf{Q}}_B'^T \begin{bmatrix} \bar{\mathbf{R}}_{12} \\ \bar{\mathbf{H}}_2' \end{bmatrix} \in \mathbb{R}^{m\times n_2} \end{aligned} \tag{75}$$

where $\bar{\mathbf{U}}_B' \in \mathbb{R}^{(n_1+m)\times n_1}$ spans the left null space of the matrix $\begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{R}}_{B_2} \end{bmatrix}$, while $\bar{\mathbf{Q}}_B' \in \mathbb{R}^{(n_1+m)\times m}$ spans the column space of it, and they together form a unitary matrix, whose transpose represents the product of the sequence of the Givens rotations, according to the order in (R1). As mentioned earlier, due to this specific Given process, the resulting factor block $\mathbf{R}_{11}^s$ is guaranteed to be upper-triangular. Note that, this is possible because of the upper-triangular structure of both $\bar{\mathbf{G}}_1$ and $\bar{\mathbf{R}}_{B_2}$, thanks to the previous two QR

processes, in (64) and (71). Finally, after all these three QR steps, the updated information factor blocks are given by $\mathbf{R}_{11}^s$ and $\mathbf{R}_{12}^s$ in (75), while $\mathbf{J}_1^s$ and $\mathbf{J}_2^s$ are the Jacobian blocks corresponding to the dropped information [see (32) – (39) for correspondence].

Now we prove the correctness of this proposed approach, i.e., it is indeed an exact ISE algorithm. Based on the generic framework of Alg. 1, we first find the specific underlying $\mathbf{U}_1$ matrix, that describes the entire unitary transformation defined in the above procedure, and then show that it satisfies the condition in Theorem 2. Specifically, as described above, our approach consists of three consecutive QR steps, where each of them defines a unitary transformation, converting the input information Jacobians, step by step, into the final updated factor blocks [see (64), (71), and (74)]:

$$\begin{bmatrix}\mathbf{G}_1 \vdots \mathbf{G}_2\end{bmatrix} \xrightarrow{\mathbf{Q}_{G_1}^T} \begin{bmatrix}\bar{\mathbf{G}}_1 \vdots \bar{\mathbf{G}}_2\end{bmatrix} \xrightarrow{\begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}^T\end{bmatrix}} \begin{bmatrix}\bar{\mathbf{R}}_{11} \vdots \bar{\mathbf{R}}_{12} \\ \mathbf{0} \vdots \bar{\mathbf{H}}_2'\end{bmatrix} \xrightarrow{\begin{bmatrix}\bar{\mathbf{U}}_B' & \bar{\mathbf{Q}}_B'\end{bmatrix}^T} \begin{bmatrix}\mathbf{R}_{11}^s \vdots \mathbf{R}_{12}^s \\ \mathbf{J}_1^s \vdots \mathbf{J}_2^s\end{bmatrix} \tag{76}$$

$$\implies \begin{bmatrix}\mathbf{R}_{11}^s \vdots \mathbf{R}_{12}^s\end{bmatrix} = \bar{\mathbf{U}}_B'^T \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}^T\end{bmatrix} \mathbf{Q}_{G_1}^T \begin{bmatrix}\mathbf{G}_1 \vdots \mathbf{G}_2\end{bmatrix} = \left(\mathbf{Q}_{G_1} \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B'\right)^T \begin{bmatrix}\mathbf{G}_1 \vdots \mathbf{G}_2\end{bmatrix} \tag{77}$$

Hence, if we define:

$$\mathbf{U}_1 \triangleq \mathbf{Q}_{G_1} \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B', \quad \text{with } \mathbf{U}_1 \in \mathbb{R}^{(n_1+m)\times n_1} \text{ and } \mathbf{U}_1^T\mathbf{U}_1 = \mathbf{I}_{n_1} \tag{78}$$

then $\mathbf{R}_{11}^s = \mathbf{U}_1^T\mathbf{G}_1$ and $\mathbf{R}_{12}^s = \mathbf{U}_1^T\mathbf{G}_2$. Comparing this result to Step 4 of Alg. 1, it is evident that this specific $\mathbf{U}_1$ in (78) is exactly the underlying matrix representing our entire unitary transformation. As expected, this orthonormal matrix $\mathbf{U}_1$ is the combination of the three orthonormal Q factors, corresponding to the three QR steps in our proposed approach. Next, we show that this specific $\mathbf{U}_1$ satisfies the condition stated in Theorem 2. First of all, as required by the second part of the condition, the resulting factor block $\mathbf{R}_{11}^s = \mathbf{U}_1^T\mathbf{G}_1$ is guaranteed to be upper-triangular, by the design of the last Given QR with the specific rule (R1). As for the first part of the condition, we show that *(C4)* holds true for this specific $\mathbf{U}_1$, i.e., the columns of $\mathbf{U}_1$ form an orthonormal basis for the right null space of $\mathbf{Q}_2^T(\mathbf{I} + \mathbf{A}^T\mathbf{A})$, where $\mathbf{Q}_2$ spans the left null space of $\mathbf{G}_1$. Given that $\mathbf{Q}_{G_1}$ is the Q factor of the QR factorization of $\mathbf{G}_1$, from (63), we choose $\mathbf{Q}_2$ to be the second block column of $\mathbf{Q}_{G_1}$, hence it spans the left null space of $\mathbf{G}_1$ as required. Moreover, since $\mathbf{Q}_{G_1}$ is unitary, i.e., $\mathbf{Q}_{G_1}^T\mathbf{Q}_{G_1} = \mathbf{I}$, taking the second block row of this equation gives $\mathbf{Q}_2^T\mathbf{Q}_{G_1} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}$. Now we can show that:

$$\begin{aligned}
&\mathbf{Q}_2^T(\mathbf{I} + \mathbf{A}^T\mathbf{A})\mathbf{U}_1 \\
&= \mathbf{Q}_2^T(\mathbf{I} + \mathbf{A}^T\mathbf{A})\mathbf{Q}_{G_1} \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B' && \text{(from the definition of } \mathbf{U}_1 \text{ in (78))} \\
&= \mathbf{Q}_2^T\mathbf{Q}_{G_1}(\mathbf{I} + \mathbf{Q}_{G_1}^T\mathbf{A}^T\mathbf{A}\mathbf{Q}_{G_1}) \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B' && \text{(since } \mathbf{Q}_{G_1}\mathbf{Q}_{G_1}^T = \mathbf{I} \text{ because } \mathbf{Q}_{G_1} \text{ is unitary)} \\
&= \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}(\mathbf{I} + \mathbf{Q}_{G_1}^T\mathbf{A}^T\mathbf{A}\mathbf{Q}_{G_1}) \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B' && \text{(since } \mathbf{Q}_2^T\mathbf{Q}_{G_1} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}) \\
&= \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}(\mathbf{I} + \bar{\mathbf{A}}^T\bar{\mathbf{A}}) \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B' && \text{(from the definition of } \bar{\mathbf{A}} \text{ in (66))} \\
&= \bar{\mathbf{B}}^T \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}\end{bmatrix} \bar{\mathbf{U}}_B' = \left(\bar{\mathbf{U}}_B'^T \begin{bmatrix}\mathbf{I} & \\ & \bar{\mathbf{Q}}_{B_2}^T\end{bmatrix} \begin{bmatrix}\bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_2\end{bmatrix}\right)^T && \text{(from the definition of } \bar{\mathbf{B}} \text{ in (67) and (69))} \\
&= \left(\bar{\mathbf{U}}_B'^T \begin{bmatrix}\bar{\mathbf{B}}_1 \\ \bar{\mathbf{R}}_{B_2}\end{bmatrix}\right)^T && \text{(from the QR factorization in (70))} \\
&= \mathbf{0} && \text{(from the QR factorization in (73))}
\end{aligned}$$

Hence, $\mathbf{U}_1$ belongs to the right null space of $\mathbf{Q}_2^T(\mathbf{I} + \mathbf{A}^T\mathbf{A})$. Moreover, since this null space is of dimension $n_1$ as shown in (60), and $\mathbf{U}_1$ has $n_1$ orthonormal columns, these columns must form an orthonormal basis for this null space. Therefore, by Theorem 2 with *(C4)*, we conclude that this specific $\mathbf{U}_1$ in (78), which is a summarized representation of our proposed approach, leads to a valid exact ISE algorithm.

**Algorithm 2** The Exact Inverse Schmidt Estimator (ISE)

1: **Input:** Current state estimate $\hat{\mathbf{x}}$, prior information factor $\mathbf{R}$ and residual $\mathbf{r}_0$, pre-whitened measurement Jacobian $\mathbf{H}$ and residual $\mathbf{r}$

2: **procedure** UPDATE

3: Perform the in-place QR of $\begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix}$ as: $\begin{bmatrix}\mathbf{R}_{11} & \vdots & \mathbf{R}_{12} & \vdots & \mathbf{r}_0^1\\ \mathbf{H}_1 & \vdots & \mathbf{H}_2 & \vdots & \mathbf{r}\end{bmatrix} \xrightarrow{\text{QR}} \begin{bmatrix}\bar{\mathbf{R}}_{11} & \vdots & \bar{\mathbf{R}}_{12} & \vdots & \bar{\mathbf{r}}_1\\ \mathbf{0} & \vdots & \bar{\mathbf{H}}_2 & \vdots & \bar{\mathbf{r}}_2\end{bmatrix}$

4: Compute $\begin{bmatrix}\bar{\mathbf{A}}_1 & \vdots & \bar{\mathbf{A}}_2\end{bmatrix} \leftarrow \mathbf{R}_{22}^{-T}\begin{bmatrix}\bar{\mathbf{R}}_{12}^T & \vdots & \bar{\mathbf{H}}_2^T\end{bmatrix}$, then compute $\bar{\mathbf{B}}_1 \leftarrow \bar{\mathbf{A}}_1^T\bar{\mathbf{A}}_2$ and $\bar{\mathbf{B}}_2 \leftarrow \mathbf{I} + \bar{\mathbf{A}}_2^T\bar{\mathbf{A}}_2$

5: Perform the in-place QR of $\bar{\mathbf{B}}_2$ as: $\begin{bmatrix}\bar{\mathbf{B}}_2 & \vdots & \bar{\mathbf{H}}_2 & \vdots & \bar{\mathbf{r}}_2\end{bmatrix} \xrightarrow{\text{QR}} \begin{bmatrix}\bar{\mathbf{R}}_{B_2} & \vdots & \bar{\mathbf{H}}_2' & \vdots & \bar{\mathbf{r}}_2'\end{bmatrix}$

6: Perform the in-place QR of $\begin{bmatrix}\bar{\mathbf{B}}_1\\ \bar{\mathbf{R}}_{B_2}\end{bmatrix}$ as: $\begin{bmatrix}\bar{\mathbf{B}}_1 & \vdots & \bar{\mathbf{R}}_{11} & \vdots & \bar{\mathbf{R}}_{12} & \vdots & \bar{\mathbf{r}}_1\\ \bar{\mathbf{R}}_{B_2} & \vdots & \mathbf{0} & \vdots & \bar{\mathbf{H}}_2' & \vdots & \bar{\mathbf{r}}_2'\end{bmatrix} \xrightarrow{\text{QR}} \begin{bmatrix}\mathbf{0} & \vdots & \mathbf{R}_{11}^s & \vdots & \mathbf{R}_{12}^s & \vdots & \mathbf{r}^s\\ \bar{\mathbf{R}}_{B_2}' & \vdots & \mathbf{J}_1^s & \vdots & \mathbf{J}_2^s & \vdots & \mathbf{r}_J^s\end{bmatrix}$, by Givens rotations following (R1)

7: Information factor update: $\mathbf{R}^s \leftarrow \begin{bmatrix}\mathbf{R}_{11}^s & \mathbf{R}_{12}^s\\ \mathbf{0} & \mathbf{R}_{22}\end{bmatrix}$

8: State update: $\hat{\mathbf{x}}^s \leftarrow \begin{bmatrix}\hat{\mathbf{x}}_1 + {\mathbf{R}_{11}^s}^{-1}\mathbf{r}^s\\ \hat{\mathbf{x}}_2\end{bmatrix}$

9: **end procedure**

10: **Output:** Updated Schmidt state estimate $\hat{\mathbf{x}}^s$ and information factor $\mathbf{R}^s$

The proposed exact ISE algorithm is summarized in Alg. 2.[1] By Theorem 2, this estimator is mathematically equivalent to the SKF but in the square-root inverse form (see the definitions in Sec. 3.4), and the updated information factor is guaranteed to be invertible and upper-triangular. Note that, it is a special realization of the generic framework in Alg. 1, i.e., Steps 3 – 6 in Alg. 2 are particular realizations of Steps 3 – 4 in Alg. 1, with the specific $\mathbf{U}_1$ as defined in (78), but obtained implicitly through three consecutive QR factorizations.

In order to achieve efficient implementations, all the QR factorizations should be performed in place, i.e., the transpose of the Q factor is applied in its factored form to the stacked matrices, without explicitly computing the Q factor. Furthermore, as designed, the upper-triangular structure of the information factor's $(1,1)$ block is maintained throughout the entire algorithm, and hence should be taken into account during each QR process. Lastly, the two matrix-inversion operations in Steps 4 and 8 involve only upper (or lower) triangular matrices, and hence can be computed efficiently by backward (or forward) substitutions.

As for the computational complexity, it can be shown that the total complexity of Alg. 2 sums up to $\mathcal{O}\left((n_1+m)[n_2^2+m(n_2+n_1+m)]\right)$. Typically, in practice, we would apply the Schmidt estimator when a major portion of the state vector is to remain the same, i.e., when $n_2 \gg n_1+m$. Under this assumption, the total cost is simplified to $\mathcal{O}\left((n_1+m)n_2^2\right)$. This quadratic cost in $n_2$ is due to the solve operation with the upper-triangular matrix $\mathbf{R}_{22} \in \mathbb{R}^{n_2\times n_2}$ in Step 4, which in this case becomes the dominant step in terms of the computational cost. Furthermore, if in addition, the input factor block $\mathbf{R}_{22}$ is sparse, and the solve can be done in linear time in $n_2$, then the total cost is reduced to $\mathcal{O}\left(m(n_1+m)n_2\right)$, which is now linear with respect to $n_2$. One such practical example is the problem of SLAM, where the square-root information factor is indeed sparse [4], and hence potentially may allow efficient approximate solutions by using the exact ISE.

# 5 The Approximate Inverse Schmidt Estimators

So far we have successfully derived the exact inverse Schmidt estimator (ISE), which is the information-domain equivalent of the SKF. And as mentioned before, the exact ISE is a special realization of the generic framework presented in Alg. 1. Furthermore, among all those algorithms that this generic framework covers, the exact ISE is the optimal one, in the sense that it minimizes the mean squared error of the posterior

[1] As mentioned before, in order to reduce linearization errors, this update procedure (including the linearization of the measurement equations) can be repeated iteratively till convergence.

estimate of $\mathbf{x}_1$, as is also the case for its covariance-domain equivalent SKF. In other words, the exact ISE absorbs all the information of $\mathbf{x}_1$, and updates its estimate to be the same as that of the optimal KF or SR-IF. Due to this optimality, the computational complexity of the exact ISE algorithm, as mentioned earlier, is still high (up to quadratic in the size of $\mathbf{x}_2$). Hence, we would like to further relax this optimality constraint, and seek to obtain alternative algorithms that are further approximations with respect to the exact ISE, while gaining in efficiency. Specifically, while $\mathbf{x}_2$ remains unaltered as before, we propose to update $\mathbf{x}_1$ only *approximately*, instead of *optimally* as in the case of the exact ISE, so as to reduce the computational cost.

To achieve this, we first identify the computationally-dominant step of the exact ISE algorithm, i.e., the back-solve operation with $\mathbf{R}_{22}$ in Step 4 of Alg. 2, since the size of $\mathbf{R}_{22}$ can be potentially very large. Then, instead of using the exact factor block $\mathbf{R}_{22}$, we propose to employ various approximate versions of it in the solving operation so that this step (as well as other steps) becomes less expensive. In what follows, we present two such algorithms, both of which are derived as further approximations of the exact ISE using this idea. Meanwhile, they are also special realizations of the generic framework in Alg. 1, and because of this, they inherit the properties of Alg. 1, i.e., $\mathbf{x}_2$ is preserved and the estimates are consistent.

## 5.1 The Resource-aware Inverse Schmidt Estimator (RISE)

The first approximate algorithm is obtained by simply setting $\mathbf{R}_{22} = \infty$ in Step 4 of the exact ISE algorithm (see Alg. 2). This immediately eliminates Step 4-6 from the procedure. Hence, comparing this to the three-step QR procedure in Alg. 2, we can see that this algorithm only takes the first QR step from the exact ISE, while removing the next two. The proposed algorithm is presented in Alg. 3. In fact, this algorithm is a special realization of the generic framework in Alg. 1, as it can be obtained by choosing $\mathbf{U} = \mathbf{Q}_{G_1}$, where $\mathbf{Q}_{G_1}$ is the Q factor of the QR factorization of $\mathbf{G}_1$, as defined in (63).

**Algorithm 3** The Resource-aware Inverse Schmidt Estimator (RISE)

1: **Input:** Current state estimate $\hat{\mathbf{x}}$, prior information factor $\mathbf{R}$ and residual $\mathbf{r}_0$, pre-whitened measurement Jacobian $\mathbf{H}$ and residual $\mathbf{r}$
2: **procedure** UPDATE
3: Perform the in-place QR of $\begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix}$ as: $\left[\begin{array}{c:c:c}\mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{r}_0^1\\ \hdashline \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{r}\end{array}\right] \xrightarrow{\text{QR}} \left[\begin{array}{c:c:c}\bar{\mathbf{R}}_{11} & \bar{\mathbf{R}}_{12} & \bar{\mathbf{r}}_1\\ \hdashline \mathbf{0} & \bar{\mathbf{H}}_2 & \bar{\mathbf{r}}_2\end{array}\right]$
4: Information factor update: $\mathbf{R}^r \leftarrow \begin{bmatrix}\bar{\mathbf{R}}_{11} & \bar{\mathbf{R}}_{12}\\ \mathbf{0} & \mathbf{R}_{22}\end{bmatrix}$
5: State update: $\hat{\mathbf{x}}^r \leftarrow \begin{bmatrix}\hat{\mathbf{x}}_1 + \bar{\mathbf{R}}_{11}^{-1}\bar{\mathbf{r}}_1\\ \hat{\mathbf{x}}_2\end{bmatrix}$
6: **end procedure**
7: **Output:** Updated state estimate $\hat{\mathbf{x}}^r$ and information factor $\mathbf{R}^r$

The gain in speed of this algorithm is obvious: Since it is a "trimmed" version of Alg. 2, the computational cost is strictly lower than that of the exact ISE. In fact, the block $\mathbf{R}_{22}$ is never involved in the procedure, and the total cost is at most linear in $n_2$. If in addition, $\mathbf{R}_{12}$ and $\mathbf{H}_2$ are sparse, then the complexity of this entire algorithm becomes constant (depending on $n_1$) with respect to $n_2$. Hence, the advantage of Alg. 3 is its fast speed. The disadvantage, however, is that this estimator does not provide any performance guarantee in terms of the accuracy of the updated state $\mathbf{x}_1$. Specifically, unlike the exact ISE algorithm in Alg. 2, the state update in Alg. 3 only depends on the stacked Jacobian matrix $\begin{bmatrix}\mathbf{R}_{11}\\ \mathbf{H}_1\end{bmatrix}$ corresponding to $\mathbf{x}_1$, as well as the residual vector $\begin{bmatrix}\mathbf{r}_0^1\\ \mathbf{r}\end{bmatrix}$, while $\mathbf{R}_{12}$, $\mathbf{R}_{22}$, and $\mathbf{H}_2$ have no impact on the state update. Hence, this state update step is equivalent to that of the *inconsistent* approach, where $\mathbf{R}_{12}$, $\mathbf{R}_{22}$, and $\mathbf{H}_2$ are ignored, i.e., $\mathbf{x}_2$ is assumed to be *known* perfectly with zero uncertainty. The key difference, however, is that by updating and keeping track of the cross term $\mathbf{R}_{12}$, Alg. 3 maintains the correct information factor and thus achieves *consistent* estimates.

More importantly, this proposed algorithm is resource-aware, i.e., it allows trading estimation accuracy for computational efficiency according to the availability of processing resources, by adjusting the size of

the window of the states selected to be updated (i.e., the size of $\mathbf{x}_1$ with respect to the entire $\mathbf{x}$). Smaller windows provide less accurate estimates at higher rates, while larger windows allow for higher accuracy at the cost of longer processing times. In the extreme case when all states are chosen for update (i.e., $\mathbf{x}_1 = \mathbf{x}$), this algorithm becomes exactly equivalent to the optimal SR-IF without any information loss, since in this case the $\mathbf{x}_2$ part vanishes and no cost term is dropped. For this reason, we name it the resource-aware inverse Schmidt estimator, or RISE.

### 5.2 Other Approximations

The second approximate algorithm is obtained by setting $\mathbf{R}_{22}$ to be its reduced form in Step 4 of the exact ISE algorithm (see Alg. 2). Specifically, if the $\mathbf{R}_{22}$ matrix is (block) diagonally dominant, one way would be to use a banded version of it, taking the original upper-triangular matrix but only up to a certain bandwidth. This bandwidth can be an adjustable parameter, which controls the balancing between accuracy and speed: In general, a larger bandwidth would lead to smaller discrepancy between the results of this approximate algorithm and those of the exact ISE, and hence more accurate estimates, but at a slower speed. Other approximate forms of the $\mathbf{R}_{22}$ matrix are also possible, depending on the specific characteristics of this matrix. The procedure of this proposed algorithm is omitted here, as it is very similar to Alg. 2, with the only change being that the $\mathbf{R}_{22}$ matrix in Step 4 is replaced by its approximate as described above.

## 6 Conclusions

In this work, we presented several novel inverse Schmidt estimators. First, we derived the exact inverse Schmidt estimator (ISE), which is mathematically equivalent to the Schmidt-Kalman filter (SKF), but in the square-root information form. A detailed derivation of our proposed exact ISE algorithm was presented. Specifically, we started by proposing a generic framework, which shares some fundamental properties with the SKF, such as preserving some portion of the state and ensuring estimation consistency. Next, we proved several sufficient and necessary conditions, under which the generic framework becomes the desired exact ISE. Then, based on these conditions, a version of the exact ISE algorithm was obtained. Our proposed algorithm consists of three QR factorizations for numerical stability, while the upper-triangular structure of the information factor is maintained throughout the entire process and fully exploited for efficiency. In addition, complexity analysis of the proposed algorithm is presented.

Moreover, to further improve the computational efficiency, we proposed several approximate algorithms to the exact ISE. One noticeable result is the resource-aware inverse Schmidt estimator (RISE), which provides a mechanism to trade estimation accuracy for computational efficiency, by adjusting the size of the window of the states to be updated. We hypothesize that these proposed (exact or approximate) inverse Schmidt estimators can be employed to provide efficient, accurate, and consistent estimates to real-world problems, such as simultaneous localization and mapping (SLAM) in large areas.

## References


[1] P. S. Maybeck, *Stochastic Models, Estimation and Control*, ser. Mathematics in Science and Engineering. New York, NY: Academic Press, 1979, vol. 1.

[2] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, ser. Mathematics in Science and Engineering. New York, NY: Academic Press, 1977, vol. 128.

[3] S. F. Schmidt, "Applications of state space methods to navigation problems," *Advances in Control Systems*, vol. 3, pp. 293–340, 1966.

[4] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proc. of the International Workshop on Vision Algorithms*, Springer, Verlag, 1999, pp. 298–372.

[5] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[6] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, February 2012.

[7] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY: Academic Press, 1970.

[8] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. of the American Control Conference*, vol. 4, Albuquerque, NM, Jun. 4–6 1997, pp. 2369–2373.

[9] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Trans. on Robotics*, vol. 25, no. 2, pp. 264–280, Apr. 2009.

[10] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Proc. of Robotics: Science and Systems*, Rome, Italy, Jul. 13-17 2015.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. JHU Press, 2012.