A Multipurpose Consider Covariance Analysis for Square-Root Information Filters

Joanna C. Hinks* and Mark L. Psiaki†
Cornell University, Ithaca, NY, 14853-7501

A new form of consider covariance analysis suitable for application to square-root information filters with a wide variety of model errors is presented and demonstrated. A special system formulation is employed, and the analysis draws on the algorithms of square-root information filtering to provide generality and compactness. This analysis enables one to investigate the estimation errors that arise when the filter's dynamics model, measurement model, assumed statistics, or some combination of these is incorrect. Such an investigation can improve filter design or characterize an existing filter's true accuracy. Areas of application include incorrect initial state covariance; incorrect, colored, or correlated noise statistics; unestimated states; and erroneous system matrices. Several simple, yet practical, examples are developed, and the consider analysis results for these examples are shown to agree closely with Monte Carlo simulations.

I. Introduction

THE field of estimation addresses the question of how to form the best possible estimates of a system's state given the available, imperfect models of that system. For most applications, it is equally important to determine the statistical uncertainty of the estimates. The Kalman filter (KF) family of estimation schemes, including its various extended, square-root, and unscented formulations, generates both optimal state estimates and their error covariances, or an equivalent. These methods lend themselves naturally to the study of estimation uncertainty by means of covariance analysis.

In covariance analysis, the covariance properties of the estimation error provide information about the filter and its accuracy. For example, one can examine the observability of particular subsets of states, or determine which measurements are most critical to filter performance. The covariance produced by a filter, however, is only a reliable metric of estimation error if the filter's model perfectly describes the true underlying system. When mismatch occurs between the filter model and the "truth" model, a more specialized alternative known as consider covariance analysis can sometimes provide valuable insight.

A consider covariance analysis allows one to "consider" the effects on estimation accuracy that result from certain types of mismatch between a filter's system model and the "truth" system. One can study the effects of model mismatch by computing the true covariance of the error between the filter's estimates and the "truth" states, which is distinct from the filter's reported covariance. The true covariances provide a metric for the accuracy of estimates produced by the mismodeled filter. In the context of this paper, the following definition for the phrase "consider covariance analysis" is adopted: A consider covariance analysis examines the errors of a filter that is based on an incorrect system model, specifically by *computing the "true" error covariance as dictated by an assumed "true" system model.* Note that the analysis is inherently hypothetical because it requires one to assume a specific "truth" model that may have errors of its own. The true covariances that the consider covariance analysis computes are relative to that assumed "truth" model. They are true in the sense that *if* the given filter were applied to a system with the proposed "true" system model, then the errors in the resulting estimates would have those covariances. While some consider analysis methods compute true covariance directly, others produce some equivalent from which covariance

*Postdoctoral Assistant, Department of Mechanical & Aerospace Engineering, University at Buffalo, State University of New York. Email: jchinks@buffalo.edu, Member, AIAA. (Formerly Graduate Student, Sibley School of Mechanical & Aerospace Engineering, Cornell University)

†Professor, Sibley School of Mechanical & Aerospace Engineering, 206 Upson Hall. Associate Fellow, AIAA.

may be calculated, such as the matrix difference between the true and filter-reported covariances. Some of this paper's references prefer the name "sensitivity analysis", because such methods enable one to explore the sensitivity of a filter to various kinds of model error. Still others employ the generic name "covariance analysis" regardless of whether the filter model is assumed to match the true system model. Both alternate names refer to analyses that fall within this paper's broad definition of consider covariance analysis.

Consider covariance analysis is a flexible tool that can be applied in different ways depending on the source of the filter model errors. Errors in filter models may occur unintentionally, especially in situations where the dynamics or disturbances are only poorly understood. They may also be intentional, as when a reduced-order filter or simplified equations are employed to mitigate computational burden. In the latter scenario, consider covariance analysis is an important step in the design process that indicates whether the suboptimal filter performance is likely to be acceptable. Different candidate filters may be evaluated and compared.

When model errors result from high uncertainty in the dynamics, the practitioner typically knows that model errors are likely, but he or she does not know exactly what they are. Consequently, there is no single "truth" model standard with respect to which filters may be analyzed. A common application of consider covariance analysis iteratively evaluates proposed filters, seeking one that is insensitive to changes in the uncertain system parameters. Such a filter is generally suboptimal for any particular realization of the "true" system, but maintains acceptable performance characteristics over the range of possible "true" systems.[1,2] Similarly, an outer Monte Carlo analysis may be wrapped around a consider covariance analysis to characterize the performance of a single proposed filter over some statistical distribution of hypothetical "truth" models. Alternatively, consider analysis can compare the relative magnitude of different potential error effects, so as to develop an error budget or investigate discrepancies in solutions obtained from multiple sources.[3,4] A final suggested application relates to multiple model filtering. This filtering approach assumes that the true system model is unknown and forms estimates as a weighted sum of the estimates corresponding to a set of candidate models. The philosophical similarities between consider covariance analysis and multiple model filtering hint at the possibility of using the former as a tool in the analysis or design of the latter. The exact form such an analysis would take is unclear.

Many previous consider covariance analyses are narrow in scope, and address only a few particular classes of filter model errors. One of the simplest forms of consider analysis focuses on filters with unmodeled constant random biases, particularly dynamics model parameter biases or measurement biases.[5–7] Another very simple analysis treats incorrect covariances of the process noise, measurement noise, or initial estimate.[7–9] Some less narrow efforts draw on both of these simple analyses to examine the effects of unmodeled coloring of the process noise or other unestimated dynamically-varying disturbance states.[10–12] This type of error is often encountered in the context of intentional model simplification and reduced-order filtering. Finally, a few works investigate the case where both the noise covariances and some or all of the state-space system matrices used by the filter are incorrect.[13–17] These sources do not explicitly allow for situations where some elements of the "truth" state vector are unestimated, although an experienced analyst could modify the algorithms to accommodate that case.

Algorithms that handle a subset of error classes can be useful, but a general approach that can simultaneously study all of these errors is preferable. Such general consider covariance analyses are available in Refs. 2, 18, 19. These algorithms, which are implemented in the covariance domain, augment the state vector of a traditional Kalman filter and perform several additional intermediate calculations. All of the previously-mentioned error classes can be studied, including incorrect process and measurement noise covariances, incorrect initial estimate covariance, unestimated random biases or dynamics, and erroneous system matrices. Notably, none of the previous consider covariance analyses is able to directly handle the situation of cross-correlated process and measurement noise. The single exception is Ref. 17, which suggests how such an analysis might be performed but does not derive the equations for that case. Many of the more general versions of consider analysis could, however, incorporate the analysis of non-independent process and measurement noise with some additional pre-processing steps.

None of the general consider analyses operate in the information domain or directly analyze a square-root information filter (SRIF). Square-root information filtering is theoretically equivalent to traditional Kalman filtering under the correct assumptions. The SRIF, however, is preferred in some applications that require good numerical properties or an infinite initial estimation error covariance. Of the aforementioned consider covariance methods, Refs. 11, 12 are the most capable analyses in square-root information form. Neither of these works can study systems where the unestimated states have dynamics with a non-invertible state

transition matrix, and they do not develop the algorithms for incorrect system matrices.

This paper presents a new discrete square-root information form of generalized consider covariance analysis. The following contributions are made: First, this paper's analysis is the most general consider covariance analysis for square-root information filters, and a single unifying framework encompasses filters with all the error classes discussed above. Second, the algorithms are demonstrated using simple numerical examples. These examples are selected to illustrate the application of the new analysis to various types of filter errors. Consequently, the inexperienced practitioner does not have to derive non-obvious implementation steps independently. As an added benefit, the examples highlight some situations in which consider covariance analysis may prove useful, such as when a mismodeled filter behaves in a counterintuitive way.

Note that this paper restricts its focus to consider *analysis*. This paper does not address consider *filtering*, which adjusts the filter estimates based on the consider covariance calculations.[20–22] It is beyond the scope of this work to define or develop the various reasonable classes of consider filters for this new framework.

Readers not familiar with the square-root information filter may struggle with this paper's consider covariance analysis, because of its heavy dependence on SRIF techniques. The generality and relative simplicity of this approach may outweigh such difficulties for many kinds of model error analyses. References 11 and 23 are recommended to the interested reader for gaining familiarity with SRIF methods.

The remainder of this paper is organized as follows: Section II briefly reviews a simple example of consider covariance analysis to illustrate the basic concept. Section III presents a special consider form of the state-space system upon which this paper's algorithms operate. In Section IV, the consider analysis algorithms are developed. Section V demonstrates how some useful example problems can be translated into the consider form of Section III, and the consider algorithms are validated for these examples by Monte Carlo analysis. Conclusions are drawn in Section VI.

II. Introductory Example

This section's simple example illustrates the principles of consider covariance analysis in a more traditional form. The goal of this example is to help the reader to grasp the "big picture" of consider analysis before moving on to more complicated systems and algorithms.

The estimation scenario of interest is a basic static system with a batch measurement equation given by

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{\nu} \tag{1}$$

where $\boldsymbol{x}$ is a static vector that is to be estimated, $\boldsymbol{y}$ is a vector of measurements, H is the matrix relating the measurements to the unknown $\boldsymbol{x}$ vector, and $\boldsymbol{\nu}$ is the measurement noise. The measurement noise has been *incorrectly* assumed to have covariance $P_{\nu\nu a}$, whereas the true measurement noise covariance is given by $\mathrm{E}\left[\boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}\right] = P_{\nu\nu}$.

The standard weighted least-squares solution to the static batch problem of Eq. (1) is well-known; it takes the form

$$\hat{\boldsymbol{x}} = \left(H^{\mathrm{T}}P_{\nu\nu a}^{-1}H\right)^{-1}H^{\mathrm{T}}P_{\nu\nu a}^{-1}\boldsymbol{y} \tag{2}$$

where the measurements have been weighted by the inverse of the assumed noise covariance $P_{\nu\nu a}$. The estimation error in this case is

$$\begin{aligned}(\hat{\boldsymbol{x}} - \boldsymbol{x}) &= \left(H^{\mathrm{T}}P_{\nu\nu a}^{-1}H\right)^{-1}H^{\mathrm{T}}P_{\nu\nu a}^{-1}\boldsymbol{y} - \boldsymbol{x} \\ &= \left(H^{\mathrm{T}}P_{\nu\nu a}^{-1}H\right)^{-1}H^{\mathrm{T}}P_{\nu\nu a}^{-1}\left(H\boldsymbol{x} + \boldsymbol{\nu}\right) - \boldsymbol{x} \\ &= \left(H^{\mathrm{T}}P_{\nu\nu a}^{-1}H\right)^{-1}H^{\mathrm{T}}P_{\nu\nu a}^{-1}\boldsymbol{\nu}\end{aligned} \tag{3}$$

The expression for estimation error in Eq. (3) can be used to derive the estimation error covariance, which should be small if $\hat{\boldsymbol{x}}$ is a good estimate of $\boldsymbol{x}$:

$$\mathrm{E}\left[(\hat{\boldsymbol{x}} - \boldsymbol{x})(\hat{\boldsymbol{x}} - \boldsymbol{x})^{\mathrm{T}}\right] = \left(H^{\mathrm{T}}P_{\nu\nu a}^{-1}H\right)^{-1}H^{\mathrm{T}}P_{\nu\nu a}^{-1}\mathrm{E}\left[\boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}\right]P_{\nu\nu a}^{-1}H\left(H^{\mathrm{T}}P_{\nu\nu a}^{-1}H\right)^{-1} \tag{4}$$

In Eq. (4), it has been recognized that covariance matrices are symmetric so that there is no need to write $P_{\nu\nu a}^{-\mathrm{T}}$ in place of $P_{\nu\nu a}^{-1}$. Note, $(\,)^{-\mathrm{T}}$ refers to the transpose of the inverse of the matrix in question.

At this point, the distinction between covariance analysis and consider covariance analysis can be demonstrated. A traditional covariance analysis would assume that the batch estimator's model of measurement

noise covariance, $P_{\nu\nu a}$, was correct. It would then use this assumed value in Eq. (4) to compute the estimator's assumed estimation error covariance, P_{xxa}:

$$P_{xxa} = \left(H^{\mathrm{T}} P_{\nu\nu a}^{-1} H\right)^{-1} H^{\mathrm{T}} P_{\nu\nu a}^{-1} P_{\nu\nu a} P_{\nu\nu a}^{-1} H \left(H^{\mathrm{T}} P_{\nu\nu a}^{-1} H\right)^{-1} = \left(H^{\mathrm{T}} P_{\nu\nu a}^{-1} H\right)^{-1} \tag{5}$$

The covariance P_{xxa} is useful in its own right. It gives an indication of how accurate the estimate $\hat{\boldsymbol{x}}$ is expected to be, given correct assumptions about the measurements. It is not, however, the true covariance of the estimation errors. The true covariance P_{xx} is computed by a simple form of consider analysis, substituting the true measurement noise covariance $P_{\nu\nu}$ into Eq. (4) to form

$$P_{xx} = \left(H^{\mathrm{T}} P_{\nu\nu a}^{-1} H\right)^{-1} H^{\mathrm{T}} P_{\nu\nu a}^{-1} P_{\nu\nu} P_{\nu\nu a}^{-1} H \left(H^{\mathrm{T}} P_{\nu\nu a}^{-1} H\right)^{-1} \tag{6}$$

This expression does not simplify in the same way as Eq. (5). It provides a metric for the actual level of error expected from the batch least squares problem in the presence of incorrectly modeled noise.

Note, it is possible for P_{xx} to be greater than P_{xxa}, less than P_{xxa}, or to have some components that are greater while others are less. The definitions of "greater" and "less" relate to the positive definiteness of $P_{xx} - P_{xxa}$. For example, if $P_{\nu\nu a} = \rho P_{\nu\nu}$ for some positive scalar ρ, then $P_{xx} - P_{xxa}$ is positive definite if $\rho < 1$ and negative definite if $\rho > 1$.

III. Consider System Formulation

One of the primary challenges in deriving highly general algorithms for consider covariance analysis is that different kinds of filter model errors enter the system equations in different ways. This section presents a new standardized system formulation for consider analysis by which all the discussed varieties of filter modeling errors can be represented. The consider algorithms derived in Section IV manipulate this standard set of equations. Therefore, any system that can be cast into this framework can be analyzed using the general algorithms of Section IV, without further customization.

It may be difficult to see how an arbitrary problem can be rewritten in the defined consider form. In order to bridge this gap between theory and practice, Section V casts several simple yet widely applicable examples into this section's standard form.

The consider form defined here must represent both the "true" behavior of the system and the filter's estimate of that behavior. To that end, this new system form must have three mathematical components. First, it must have state-space models of the system's dynamic behavior and of the measurements that are to be used for estimation. Second, it must have a complete statistical description of the process and measurement noise, including any correlation between noise components or between noise at different times. Finally, it must have initializations of both the state estimate and the uncertainty in that estimate.

Before any consideration of consider covariance analysis, the square-root information filter assumes that the system dynamics and measurement models are given by:

$$\boldsymbol{x}_{k+1} = \Phi_{fk} \boldsymbol{x}_k + \Gamma_{fk} \boldsymbol{w}_k \tag{7a}$$

$$\boldsymbol{y}_k = H_{fk} \boldsymbol{x}_k + \boldsymbol{\nu}_k \tag{7b}$$

In these equations, $\boldsymbol{x}_k$ is the state vector at discrete sample time k, Φ_{fk} is the state transition matrix from sample k to sample $k+1$, $\boldsymbol{w}_k$ is process noise, and Γ_{fk} is the process noise influence matrix. The vector $\boldsymbol{y}_k$ contains the measurements that apply at sample k, H_{fk} is the measurement sensitivity matrix, and the measurement noise vector is given by $\boldsymbol{\nu}_k$. The measurements are assumed to have been normalized such that $\boldsymbol{\nu}_k$ is a zero-mean, identity-covariance, white, Gaussian random vector:

$$\mathrm{E}[\boldsymbol{\nu}_k] = 0, \quad \mathrm{E}\left[\boldsymbol{\nu}_k \boldsymbol{\nu}_k^{\mathrm{T}}\right] = R_{fk} = \mathrm{I}, \quad \mathrm{E}\left[\boldsymbol{\nu}_k \boldsymbol{\nu}_j^{\mathrm{T}}\right] = 0 \quad \forall\, k \neq j \tag{8}$$

Such normalization is always possible via suitable transformation of raw measurements,[23] and it is a prerequisite for the standard form of the square-root information filtering algorithms. The Gaussian process noise $\boldsymbol{w}_k$ is also zero-mean and white, with nominal covariance given by

$$\mathrm{E}\left[\boldsymbol{w}_k \boldsymbol{w}_k^{\mathrm{T}}\right] = Q_{fk} \tag{9}$$

Process noise and measurement noise are further assumed to be uncorrelated with each other.

According to standard SRIF nomenclature, the filter's process noise information equation, which encapsulates the information about the process noise statistics in a convenient form, is given by

$$\mathcal{R}_{fwwk}\boldsymbol{w}_k = -\boldsymbol{\nu}_{wk} \tag{10}$$

where $\boldsymbol{\nu}_{wk}$ is a zero-mean, identity-covariance, Gaussian random vector, and $\mathcal{R}_{fwwk}$ is related to the nominal process noise covariance according to

$$\mathcal{R}_{fwwk}^{-1}\mathcal{R}_{fwwk}^{-\mathrm{T}} = Q_{fk} \tag{11}$$

The notation "$\mathcal{R}$" is used throughout this papter for all square-root information matrices. This differs from the standard notation of Ref. 11, which uses "R" for square-root information matrices. The symbol "R" is reserved for the measurement noise covariance in the present paper, consistent with Eq. (8). The square-root information filter's estimate and covariance are initialized at $k = 0$ by assuming the standard *a priori* state information equation:

$$\bar{\mathcal{R}}_{fxx0}\boldsymbol{x}_0 = \bar{\boldsymbol{z}}_0 - \bar{\boldsymbol{\nu}}_{x0} \tag{12}$$

In this equation, $\bar{\boldsymbol{\nu}}_{x0}$ is a zero-mean, identity-covariance, Gaussian random vector, and the information state $\bar{\boldsymbol{z}}_0$ parameterizes the initial state estimate according to the equation

$$\bar{\boldsymbol{x}}_0 = \bar{\mathcal{R}}_{fxx0}^{-1}\bar{\boldsymbol{z}}_0 \tag{13}$$

The matrix $\bar{\mathcal{R}}_{fxx0}$ is the square-root information matrix for the initial state estimate; it is related to the initial estimation error covariance by

$$\bar{\mathcal{R}}_{fxx0}^{-1}\bar{\mathcal{R}}_{fxx0}^{-\mathrm{T}} = \overline{P}_{fxx0} \tag{14}$$

The overbar notation employed in Eqs. (12)-(14) and throughout this paper designates *a priori* estimates or the matrices and vectors associated with those estimates. Likewise, the notation "$\hat{}$" will be used to denote *a posteriori* estimates and quantities.

The matrices Φ_{fk}, Γ_{fk}, and H_{fk} are specifically the versions of those system matrices assumed by the filter's model. They may differ from the "truth" system matrices Φ_k, Γ_k, and H_k. In many cases, however, the two sets of matrices will be identical. In the remainder of this paper, the symbols Φ_k, Γ_k, and H_k will be used for both the filter and "truth" system whenever the corresponding matrices are the same. Likewise, the covariance and square-root information matrices Q_{fk}, R_{fk}, $\overline{P}_{fxx0}$, $\mathcal{R}_{fwwk}$, $\mathcal{R}_{f\nu\nu k}$, and $\bar{\mathcal{R}}_{fxx0}$ are specifically the versions of those matrices used by the filter. The filter's measurement noise covariance and square-root information matrices R_{fk} and $\mathcal{R}_{f\nu\nu k}$ are always assumed to be identity matrices. The "truth" covariance and square-root information matrices are designated by Q_k, R_k, $\overline{P}_{xx0}$, $\mathcal{R}_{wwk}$, $\mathcal{R}_{\nu\nu k}$, and $\bar{\mathcal{R}}_{xx0}$. Except where they differ or could potentially differ, the symbols for the "truth" matrices will be used.

The consider system form allows the user to describe a large variety of modeling errors within a common framework. It does this by keeping track of the true statistics of the system uncertainty. Uncertainty can arise from the correctly- or incorrectly-modeled process noise, measurement noise, or initial estimation error covariance. Additional sources of uncertainty include errors in the matrices that specify the dynamics or measurement models used by the filter and unestimated biases or time-varying disturbances. All of these sources of uncertainty, with the exception of any deterministic unestimated biases, are contained in the components of a single "consider state vector" $\boldsymbol{x}_{\boldsymbol{c}k}$. The consider analysis computes the contribution of $\boldsymbol{x}_{\boldsymbol{c}k}$ to the true estimation error covariance by tracking its covariance and the sensitivity matrices that specify how it enters the filter equations. All stochastic contributions are contained within this single vector. Therefore, the analysis does not have to additionally compute any cross-correlations, and computationally efficient techniques can be applied to the statistical calculations.

The consider system begins by defining perturbed versions of the filter's dynamics and measurement equations:

$$\boldsymbol{x}_{k+1} = \Phi_{fk}\boldsymbol{x}_k + \Gamma_{fk}\boldsymbol{w}_k + \Gamma_{xck}\boldsymbol{x}_{\boldsymbol{c}k} + \boldsymbol{b}_{xk} \tag{15a}$$

$$\boldsymbol{y}_k = H_{fk}\boldsymbol{x}_k + H_{ck}\boldsymbol{x}_{\boldsymbol{c}k} + \boldsymbol{b}_{yk} \tag{15b}$$

The matrix Γ_{xck} describes the influence of the consider state vector on the filter's dynamics model, and the matrix H_{ck} describes the influence of the consider state vector on the measurements. Another difference is the presence of the bias vectors $\boldsymbol{b}_{xk}$ and $\boldsymbol{b}_{yk}$. These are deterministic non-zero values that disturb the system from what the filter assumes. They may be time-varying, but are not driven by stochastic noise. In many cases, these biases are zero and can be neglected in the equations.

Equations (15a) and (15b) represent the consider generalizations of Eqs. (7a) and (7b). The generalized dynamics in Eq. (15a) simply adds the two new terms $\Gamma_{xck}\boldsymbol{x}_{\boldsymbol{c}k}$ and $\boldsymbol{b}_{xk}$ to the right-hand side. The generalized measurement equation (15b) replaces the zero-mean, identity-covariance measurement noise $\boldsymbol{\nu}_k$ with the sum of the two terms $H_{ck}\boldsymbol{x}_{\boldsymbol{c}k}$ and $\boldsymbol{b}_{yk}$. All of the random part of the "truth" measurement noise is modeled within $H_{ck}\boldsymbol{x}_{\boldsymbol{c}k}$.

The vector of consider states, $\boldsymbol{x}_{\boldsymbol{c}k}$, is defined differently for each individual system depending on what types of modeling errors are present. It has its own dynamics model given by

$$\boldsymbol{x}_{\boldsymbol{c}k+1} = \Phi_{ck}\boldsymbol{x}_{\boldsymbol{c}k} + \Gamma_{cck}\boldsymbol{w}_{\boldsymbol{c}k} \tag{16}$$

In Eq. (16), Φ_{ck} is the state transition matrix for the consider state vector dynamics. It need not be invertible or even square, as the number of elements in $\boldsymbol{x}_{\boldsymbol{c}k}$ may change. The matrix Γ_{cck} describes the influence of the driving consider process noise vector $\boldsymbol{w}_{\boldsymbol{c}k}$ on the consider state vector $\boldsymbol{x}_{\boldsymbol{c}k+1}$. Without loss of generality, both $\boldsymbol{w}_{\boldsymbol{c}k}$ and the initial consider state $\boldsymbol{x}_{\boldsymbol{c}0}$ are constrained to be zero-mean, identity-covariance, Gaussian random vectors, $\boldsymbol{w}_{\boldsymbol{c}k}$ is a white noise sequence, and $\boldsymbol{x}_{\boldsymbol{c}0}$ is uncorrelated with that sequence.

By means of the consider vector $\boldsymbol{x}_{\boldsymbol{c}k}$, it is possible to represent process noise and measurement noise with statistics that do not conform to the filter's assumptions. The noise intensities may be different from the nominal values, the process and measurement noise may be correlated, or the noise may be colored, i.e., correlated with noise at other times. Noise terms with non-zero mean values can be modeled by proper use of the bias vectors $\boldsymbol{b}_{xk}$ and $\boldsymbol{b}_{yk}$.

In place of Eq. (10), the consider analysis models the process noise statistics using the following square-root information equation:

$$\mathcal{R}_{fwwk}\boldsymbol{w}_k = -S_{wck}\boldsymbol{x}_{\boldsymbol{c}k} - \boldsymbol{b}_{wk} \tag{17}$$

In this equation, S_{wck} is a sensitivity matrix, and $\boldsymbol{b}_{wk}$ is a bias. Note how Eq. (17) replaces the simple noise term $\boldsymbol{\nu}_{wk}$ of Eq. (10) by the sum of the terms $S_{wck}\boldsymbol{x}_{\boldsymbol{c}k}$ and $\boldsymbol{b}_{wk}$. This replacement enables the consider process noise model to be non-white, biased, or correlated with the measurement noise. It also allows the "truth" process noise covariance to differ from the filter's assumed value.

In the same way, the initial state information equation of the consider analysis differs from the filter's Eq. (12):

$$\bar{\mathcal{R}}_{fxx0}\boldsymbol{x}_0 = \bar{\boldsymbol{z}}_0 - \bar{S}_{xc0}\boldsymbol{x}_{\boldsymbol{c}0} - \bar{\boldsymbol{b}}_{c0} \tag{18}$$

That is, $\boldsymbol{\nu}_{x0}$ of Eq. (12) is replaced by $\bar{S}_{xc0}\boldsymbol{x}_{\boldsymbol{c}0} + \bar{\boldsymbol{b}}_{c0}$. The bias $\bar{\boldsymbol{b}}_{c0}$ and sensitivity matrix $\bar{S}_{xc0}$ model the manner in which the "truth" initial mean and covariance of the state differ from the filter's assumed initial mean and covariance.

In summary, the consider model is a generalized form of the filter model. It consists of Eqs. (15)-(18). The consider state dynamics model, Eq. (16), is a completely new equation. All of the other consider model equations are generalizations of the original filter model equations, Eqs. (7), (10), and (12). An important feature of Eq. (16) is the complete flexibility with which Φ_{ck} can be defined. In particular, Φ_{ck} can be non-square and can have a non-trivial nullspace, e.g., columns of zeros. This feature allows components of $\boldsymbol{x}_{\boldsymbol{c}k}$ to be white noise, as would be required to reproduce the original filter model.

The consider covariance analysis is driven by a simple philosophy: In each filter equation, replace the filter's simplistic random terms with the "truth" random terms from the consider analysis. This has been done explicitly in consider model Eqs. (15b), (17), and (18), and implicitly in Eq. (15a). The same philosophy carries over to the square-root information equations that are recursively updated by the consider analysis. At each sample time, the filter forms the state information equation

$$\mathcal{R}_{xxk}\boldsymbol{x}_k = \boldsymbol{z}_k - \boldsymbol{\nu}_{xk} \tag{19}$$

In the consider analysis, the simplistic random term $\boldsymbol{\nu}_{xk}$ is replaced by a more complicated expression related to the history of the consider vector $\boldsymbol{x}_{\boldsymbol{c}k}$. This expression's "truth" mean and covariance replace the zero mean and identity covariance of $\boldsymbol{\nu}_{xk}$ in the consider analysis covariance calculations. These calculations are given in detail in Section IV.

IV. Consider Analysis Algorithms

The consider covariance analysis algorithms presented here largely follow standard SRIF procedures. They differ from standard filtering equations in that they keep track of some additional sensitivity matrices

and perform a few extra factorizations. The analysis performs the original SRIF calculations and appends new calculations that capture all the model error effects. In most cases, only minimal modification is necessary from the original filter implementation. All the derivations in this section assume that the system has already been placed in the consider form defined in Section III.

A. Initialization and First Steps

The first part of the consider analysis requires several special initialization steps. After initialization, the analysis can proceed recursively with a generic form for each dynamic propagation and measurement update. The analysis is presumed to start from the state information equation, Eq. (18), repeated here:

$$\bar{\mathcal{R}}_{fxx0}\boldsymbol{x}_0 = \bar{\boldsymbol{z}}_0 - \bar{S}_{xc0}\boldsymbol{x}_{\boldsymbol{c}0} - \bar{\boldsymbol{b}}_{c0} \tag{20}$$

A measurement is assumed to be available at time $k = 0$, as per Eq. (15b):

$$\boldsymbol{y}_0 = H_{f0}\boldsymbol{x}_0 + H_{c0}\boldsymbol{x}_{\boldsymbol{c}0} + \boldsymbol{b}_{y0} \tag{21}$$

Per standard square-root information algorithms, the measurement update proceeds by rearranging and stacking Eqs. (20) and (21).

$$\begin{bmatrix} \bar{\mathcal{R}}_{fxx0} \\ H_{f0} \end{bmatrix} \boldsymbol{x}_0 = \begin{bmatrix} \bar{\boldsymbol{z}}_0 \\ \boldsymbol{y}_0 \end{bmatrix} - \begin{bmatrix} \bar{S}_{xc0} \\ H_{c0} \end{bmatrix} \boldsymbol{x}_{\boldsymbol{c}0} - \begin{bmatrix} \bar{\boldsymbol{b}}_{c0} \\ \boldsymbol{b}_{y0} \end{bmatrix} \tag{22}$$

Standard orthonormal/upper-triangular (QR) factorization is used to compute an orthonormal transformation matrix $\widehat{T}_0$ such that:

$$\widehat{T}_0 \begin{bmatrix} \bar{\mathcal{R}}_{fxx0} \\ H_{f0} \end{bmatrix} = \begin{bmatrix} \widehat{\mathcal{R}}_{xx0} \\ 0 \end{bmatrix} \tag{23}$$

where $\widehat{\mathcal{R}}_{xx0}$ is a square, upper-triangular matrix. The matrices $\widehat{T}_0$ and $\widehat{\mathcal{R}}_{xx0}$ can be computed using the "qr" function that is discussed in the Appendix. The input to this function will be the block matrix on the left-hand side of Eq. (23), and the $\widehat{T}_0$ matrix will be the transpose of the function's "Q" matrix output. The function's "R" output will be the entire block matrix on the right-hand side of Eq. (23).

The matrix $\widehat{T}_0$ effectively compresses all the information about the state $\boldsymbol{x}_0$, both the *a priori* information and measurement information, into the block $\widehat{\mathcal{R}}_{xx0}$. After computing $\widehat{T}_0$, every term in Eq. (22) is premultiplied by this transformation to accomplish the measurement update. The resulting block matrices and vectors are renamed for convenience to obtain the new, updated information equation:

$$\begin{bmatrix} \widehat{\mathcal{R}}_{xx0} \\ 0 \end{bmatrix} \boldsymbol{x}_0 = \begin{bmatrix} \hat{\boldsymbol{z}}_0 \\ \boldsymbol{z}_{r0} \end{bmatrix} - \begin{bmatrix} \widehat{S}_{xc0} \\ S_{rc0} \end{bmatrix} \boldsymbol{x}_{\boldsymbol{c}0} - \begin{bmatrix} \hat{\boldsymbol{b}}_{c0} \\ \boldsymbol{b}_{r0} \end{bmatrix} \tag{24}$$

The new vectors $\hat{\boldsymbol{z}}_0$, $\boldsymbol{z}_{r0}$, $\hat{\boldsymbol{b}}_{c0}$, and $\boldsymbol{b}_{r0}$ and the new matrices $\widehat{S}_{xc0}$ and S_{rc0} are the results of this premultiplication. Note that $\hat{\boldsymbol{z}}_0$, $\widehat{S}_{xc0}$, and $\hat{\boldsymbol{b}}_{c0}$ all have row dimensions of n_x, the number of elements in $\boldsymbol{x}$. The bottom rows of this block equation contain information about the measurement residuals. For purposes of this derivation, only the top rows are necessary to continue the analysis. They comprise the *a posteriori* state information equation at $k = 0$:

$$\widehat{\mathcal{R}}_{xx0}\boldsymbol{x}_0 = \hat{\boldsymbol{z}}_0 - \widehat{S}_{xc0}\boldsymbol{x}_{\boldsymbol{c}0} - \hat{\boldsymbol{b}}_{c0} \tag{25}$$

So far, the analysis looks exactly like the measurement update for a square-root information filter, except that the term $\left(\widehat{S}_{xc0}\boldsymbol{x}_{\boldsymbol{c}0} + \hat{\boldsymbol{b}}_{c0}\right)$ takes the place of the previous zero-mean, identity covariance error term $\boldsymbol{\nu}_{x0}$. A naïve approach might continue along these lines, performing dynamic propagation and measurement update steps that keep track of the sensitivity matrices that multiply the consider state $\boldsymbol{x}_{\boldsymbol{c}k}$.

This approach is not sustainable, however. Successive dynamics and measurement equations introduce a dependence on the consider state vector at times $k = 1, 2, 3, \ldots$ without eliminating the dependence on the consider states at previous times. Neglecting any deterministic biases $\boldsymbol{b}_{ck}$, the error term would expand so

that the information equation at time k would become

$$\widehat{\mathcal{R}}_{xxk}\boldsymbol{x}_k = \hat{\boldsymbol{z}}_k - \begin{bmatrix}\widehat{S}_{xc0,k} & \widehat{S}_{xc1,k} & \cdots & \widehat{S}_{xck}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{\boldsymbol{c}0}\\ \boldsymbol{x}_{\boldsymbol{c}1}\\ \vdots \\ \boldsymbol{x}_{\boldsymbol{c}k}\end{bmatrix} \tag{26}$$

The sensitivity matrix blocks $\widehat{S}_{xcj,k}$ would be recursively computed at each stage of the analysis.

There are two problems associated with this result. First, the large block sensitivity matrix that multiplies the augmented vector of consider states keeps growing as k increases. Eventually, this may cause computer storage problems. This long vector of consider states is unnecessary, because only a subspace of that vector has any effect on the estimation covariance. The dimension of this subspace is less than or equal to n_x, the dimension of the state $\boldsymbol{x}$. In other words, it is theoretically possible to represent all the parts of the error term that actually affect uncertainty with a vector that is the same length as $\boldsymbol{x}_k$.

A second difficulty with Eq. (26) is that the consider analysis would require one to compute the covariance of the vector containing $\boldsymbol{x}_{\boldsymbol{c}0}$, $\boldsymbol{x}_{\boldsymbol{c}1}$, …, $\boldsymbol{x}_{\boldsymbol{c}k}$. This computation is not easy. The initial covariance of $\boldsymbol{x}_{\boldsymbol{c}0}$ is constrained to be the identity matrix, but the covariance of each successive $\boldsymbol{x}_{\boldsymbol{c}k}$ must be found by propagating the previous covariance through the consider dynamics equation. Additionally, the cross-terms of the covariance matrix, of the form $\mathrm{E}\left[\boldsymbol{x}_{\boldsymbol{c}k}\boldsymbol{x}_{\boldsymbol{c}j}^{\mathrm{T}}\right]$, would need to be computed at each step. It is preferable to represent the error term in the information equation in a way that makes these covariance calculations trivial.

After recognizing the weaknesses of the preceding naïve approach, one might be tempted to try to reduce the size of the error term by employing the consider dynamics equation, Eq. (16), to write each consider state $\boldsymbol{x}_{\boldsymbol{c}k}$ in terms of the consider state at the initial time, $\boldsymbol{x}_{\boldsymbol{c}0}$. This new strategy, however, introduces the consider process noise $\boldsymbol{w}_{\boldsymbol{c}k}$ at successive times, so that the information equation at sample time k would become

$$\widehat{\mathcal{R}}_{xxk}\boldsymbol{x}_k = \hat{\boldsymbol{z}}_k - \begin{bmatrix}\widehat{S}_{xc0,k} & \widehat{S}_{xwc0,k} & \cdots & \widehat{S}_{xwck-1,k}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{\boldsymbol{c}0}\\ \boldsymbol{w}_{\boldsymbol{c}0}\\ \vdots \\ \boldsymbol{w}_{\boldsymbol{c}k-1}\end{bmatrix} \tag{27}$$

This result is superior to Eq. (26) when it comes to covariance calculations: Each of the elements of the right-most vector has zero mean and identity covariance, and its elements are uncorrelated:

$$\begin{bmatrix}\boldsymbol{x}_{\boldsymbol{c}0}^{\mathrm{T}} & \boldsymbol{w}_{\boldsymbol{c}0}^{\mathrm{T}} & \cdots & \boldsymbol{w}_{\boldsymbol{c}k-1}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \sim \mathcal{N}\left(0, \mathrm{I}\right) \tag{28}$$

In other words, the covariance of the entire vector is the identity matrix. However, this form of the analysis still has the problem of the growing dimension of its block sensitivity matrix. Furthermore, the calculations necessary to map the consider state back to sample $k = 0$ are nontrivial; they take the form

$$\boldsymbol{x}_{\boldsymbol{c}k+1} = \left(\prod_{j=0}^{k}\Phi_{ck-j}\right)\boldsymbol{x}_{\boldsymbol{c}0} + \sum_{j=0}^{k-1}\left(\prod_{i=0}^{k-j-1}\Phi_{ck-i}\right)\Gamma_{ccj}\boldsymbol{w}_{\boldsymbol{c}j} + \Gamma_{cck}\boldsymbol{w}_{\boldsymbol{c}k} \tag{29}$$

In light of the difficulties associated with these two naïve approaches, a transformation has been introduced to achieve two important goals: A minimal-length error effects vector in the information equation and simple statistical properties of that vector. The operations described are computationally similar to some of the orthogonal transformations performed in Ref. 11 to reduce the dimensions of some sensitivity matrix blocks, but with additional statistical benefits in the present formulation.

One version of this transformation applies only at sample $k = 0$. It starts with the information equation error term $\widehat{S}_{xc0}\boldsymbol{x}_{\boldsymbol{c}0}$, where $\boldsymbol{x}_{\boldsymbol{c}0} \sim \mathcal{N}\left(0, \mathrm{I}\right)$ by construction. Recall that the entire error term can be represented by a vector with dimension less than or equal to n_x, whereas $\boldsymbol{x}_{\boldsymbol{c}0}$ has dimension n_{xc0}, which may already exceed this minimum necessary length. An orthonormal transformation of the vector $\boldsymbol{x}_{\boldsymbol{c}0}$ is performed to separate it into two parts: One part, $\boldsymbol{\alpha}_0$, that has an effect on the information equation for $\boldsymbol{x}_0$, and one part, $\boldsymbol{\beta}_0$, that has no effect on the information equation at the current sample time. Conceptually, these newly-defined "error effects variables" bear some resemblance to those of Refs. 12, 17, although the exact

definitions and implementation details are distinct. The present transformation is accomplished by means of LQ factorization, which is analogous to QR factorization except that it decomposes a given matrix into a product of a lower-triangular matrix with possible columns of zeros and an orthonormal matrix. More information about the LQ factorization is given in the Appendix. The LQ factorization is applied here to the matrix $\widehat{S}_{xc0}$:

$$\widehat{S}_{xc0} = \begin{bmatrix} S_{x\alpha 0} & 0 \end{bmatrix} C_0 \tag{30}$$

in order to compute the right-hand-side terms as per the Appendix. In this equation, the computed $S_{x\alpha 0}$ is lower-triangular with at most n_x columns. The computed matrix C_0 is orthonormal with n_{xc0} columns and rows. The matrix C_0 transforms $\boldsymbol{x}_{\boldsymbol{c}0}$ to yield

$$\begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\beta}_0 \end{bmatrix} = C_0 \boldsymbol{x}_{\boldsymbol{c}0} \tag{31}$$

Note, if $n_{xc0} \leq n_x$, then $\boldsymbol{\beta}_0$ will be an empty vector because there is no way to produce an $\boldsymbol{\alpha}_0$ with fewer elements than $\boldsymbol{x}_{\boldsymbol{c}0}$. It is sometimes convenient to define $\begin{bmatrix} L_{\alpha 0} & L_{\beta 0} \end{bmatrix} \equiv C_0^{\mathrm{T}}$, so that

$$\boldsymbol{x}_{\boldsymbol{c}0} = C_0^{\mathrm{T}} \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\beta}_0 \end{bmatrix} = \begin{bmatrix} L_{\alpha 0} & L_{\beta 0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\beta}_0 \end{bmatrix} = L_{\alpha 0} \boldsymbol{\alpha}_0 + L_{\beta 0} \boldsymbol{\beta}_0 \tag{32}$$

As C_0 and its inverse C_0^{T} are both orthonormal, transformation of $\boldsymbol{x}_{\boldsymbol{c}0}$ by either of these matrices preserves its property of identity covariance. This result will be important for calculation of the consider analysis covariances. One can now use Eqs. (30) and (31) to rewrite the state information equation, Eq. (25), so that $\widehat{S}_{xc0}$ is replaced by the potentially lower-dimensional sensitivity matrix $S_{x\alpha 0}$.

$$\begin{aligned} \widehat{\mathcal{R}}_{xx0} \boldsymbol{x}_0 &= \hat{\boldsymbol{z}}_0 - \widehat{S}_{xc0} \boldsymbol{x}_{\boldsymbol{c}0} - \hat{\boldsymbol{b}}_{c0} \\ &= \hat{\boldsymbol{z}}_0 - \begin{bmatrix} S_{x\alpha 0} & 0 \end{bmatrix} C_0 \boldsymbol{x}_{\boldsymbol{c}0} - \hat{\boldsymbol{b}}_{c0} \\ &= \hat{\boldsymbol{z}}_0 - \begin{bmatrix} S_{x\alpha 0} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\beta}_0 \end{bmatrix} - \hat{\boldsymbol{b}}_{c0} \\ &= \hat{\boldsymbol{z}}_0 - S_{x\alpha 0} \boldsymbol{\alpha}_0 - \hat{\boldsymbol{b}}_{c0} \end{aligned} \tag{33}$$

The final line of Eq. (33) is the generic consider analysis form of the *a posteriori* SRIF information equation. It will be used throughout the remaining analysis.

B. Main Algorithm

At this point, it is desirable to shift away from derivations specific to a particular sample so that the full consider analysis can be developed in a manner analogous to mathematical induction. To that end, assume that Eq. (33) is available for some generic sample time k:

$$\widehat{\mathcal{R}}_{xxk} \boldsymbol{x}_k = \hat{\boldsymbol{z}}_k - S_{x\alpha k} \boldsymbol{\alpha}_k - \hat{\boldsymbol{b}}_{ck} \tag{34}$$

Note that the previously derived Eq. (33) can be recovered by plugging in $k = 0$. Assume also that $\boldsymbol{x}_{\boldsymbol{c}k}$ is related to $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ by some transformation of the form:

$$\boldsymbol{x}_{\boldsymbol{c}k} = \begin{bmatrix} L_{\alpha k} & L_{\beta k} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \end{bmatrix} = L_{\alpha k} \boldsymbol{\alpha}_k + L_{\beta k} \boldsymbol{\beta}_k \tag{35}$$

where the vector $\begin{bmatrix} \boldsymbol{\alpha}_k^{\mathrm{T}} & \boldsymbol{\beta}_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ has zero mean and identity covariance. Note, however, that $\boldsymbol{x}_{\boldsymbol{c}k}$ is not required to have identity covariance for $k > 0$, and consequently it is no longer required that $\begin{bmatrix} L_{\alpha k} & L_{\beta k} \end{bmatrix}$ be orthonormal.

To complete the consider covariance analysis algorithm, it is necessary to develop dynamic propagation and measurement update steps that transition these equations from sample k to sample $k+1$. Initialization of the mathematical induction occurs at sample $k = 0$ because Eqs. (33) and (32) constitute Eqs. (34) and (35) when $k = 0$ in the latter equations.

The development starts by rewriting the dynamics, measurement, and process-noise information equations in terms of the $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ vectors instead of $\boldsymbol{x}_{\boldsymbol{c}k}$. First, the system dynamics and measurement equations, Eqs. (15a) and (15b), become

$$\boldsymbol{x}_{k+1} = \Phi_{fk}\boldsymbol{x}_k + \Gamma_{fk}\boldsymbol{w}_k + \Gamma_{xck}\begin{bmatrix} L_{\alpha k} & L_{\beta k}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} + \boldsymbol{b}_{xk} \tag{36a}$$

$$\boldsymbol{y}_k = H_{fk}\boldsymbol{x}_k + H_{ck}\begin{bmatrix} L_{\alpha k} & L_{\beta k}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} + \boldsymbol{b}_{yk} \tag{36b}$$

Next, the consider dynamics of Eq. (16) can be written as

$$\boldsymbol{x}_{\boldsymbol{c}k+1} = \Phi_{ck}\begin{bmatrix} L_{\alpha k} & L_{\beta k}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} + \Gamma_{cck}\boldsymbol{w}_{\boldsymbol{c}k} = \begin{bmatrix}\Phi_{ck}L_{\alpha k} & \Phi_{ck}L_{\beta k} & \Gamma_{cck}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k}\end{bmatrix} \tag{37}$$

Because the consider process noise $\boldsymbol{w}_{\boldsymbol{c}k}$ is independent of the consider state at time k, the augmented right-most vector in Eq. (37) is zero-mean, identity-covariance, and Gaussian:

$$\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k}\end{bmatrix} \sim \mathcal{N}(0, \mathrm{I}) \tag{38}$$

Finally, the process noise information equation, Eq. (17), now takes the form

$$\mathcal{R}_{fwwk}\boldsymbol{w}_k = -S_{wck}\begin{bmatrix} L_{\alpha k} & L_{\beta k}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} - \boldsymbol{b}_{wk} \tag{39}$$

Dynamic propagation proceeds as usual for a square-root information filter, except that in place of the information equation error vector $\boldsymbol{\nu}_{xk}$, one manipulates the terms involving $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$, and $\hat{\boldsymbol{b}}_{ck}$. As in a standard SRIF, the propagation solves the dynamics equation, Eq. (36a), for $\boldsymbol{x}_k$ and substitutes this expression into information Eq. (34). The result is stacked together with process noise information Eq. (39) to yield:

$$\begin{aligned}\begin{bmatrix}\mathcal{R}_{fwwk} & 0 \\ -\widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\Gamma_{fk} & \widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_k \\ \boldsymbol{x}_{k+1}\end{bmatrix} &= \begin{bmatrix}0 \\ \hat{\boldsymbol{z}}_k\end{bmatrix} \\ &- \begin{bmatrix}S_{wck}L_{\alpha k} & S_{wck}L_{\beta k} \\ \left(S_{x\alpha k} - \widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\Gamma_{xck}L_{\alpha k}\right) & \left(-\widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\Gamma_{xck}L_{\beta k}\right)\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} \\ &- \begin{bmatrix}\boldsymbol{b}_{wk} \\ \hat{\boldsymbol{b}}_{ck} - \widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\boldsymbol{b}_{xk}\end{bmatrix}\end{aligned} \tag{40}$$

Propagation is completed by using QR factorization to compute an orthonormal transformation $\overline{T}_{k+1}$ that triangularizes the block matrix on the left. This operation yields the relationship

$$\overline{T}_{k+1}\begin{bmatrix}\mathcal{R}_{fwwk} & 0 \\ -\widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\Gamma_{fk} & \widehat{\mathcal{R}}_{xxk}\Phi_{fk}^{-1}\end{bmatrix} = \begin{bmatrix}\bar{\mathcal{R}}_{wwk} & \bar{\mathcal{R}}_{wxk+1} \\ 0 & \bar{\mathcal{R}}_{xxk+1}\end{bmatrix} \tag{41}$$

where the square, upper-triangular matrices $\bar{\mathcal{R}}_{wwk}$ and $\bar{\mathcal{R}}_{xxk+1}$ and the general matrix $\bar{\mathcal{R}}_{wxk+1}$ are also computed by the factorization. Thus, left matrix multiplication of Eq. (40) by $\overline{T}_{k+1}$ yields:

$$\begin{bmatrix}\bar{\mathcal{R}}_{wwk} & \bar{\mathcal{R}}_{wxk+1} \\ 0 & \bar{\mathcal{R}}_{xxk+1}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_k \\ \boldsymbol{x}_{k+1}\end{bmatrix} = \begin{bmatrix}\bar{\boldsymbol{z}}_{wk} \\ \bar{\boldsymbol{z}}_{k+1}\end{bmatrix} - \begin{bmatrix}\bar{S}_{w\alpha k} & \bar{S}_{w\beta k} \\ \bar{S}_{x\alpha k} & \bar{S}_{x\beta k}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} - \begin{bmatrix}\bar{\boldsymbol{b}}_{wk} \\ \bar{\boldsymbol{b}}_{ck+1}\end{bmatrix} \tag{42}$$

The vectors $\bar{\boldsymbol{z}}_{wk}$ and $\bar{\boldsymbol{z}}_{k+1}$ are the standard SRIF *a priori* information vectors that result from transforming the first term on the right-hand side of Eq. (40). The sensitivity matrices $\bar{S}_{w\alpha k}$, $\bar{S}_{w\beta k}$, $\bar{S}_{x\alpha k}$, and $\bar{S}_{x\beta k}$ come from transformation of the large block matrix on the right-hand side of Eq. (40). The vectors $\bar{\boldsymbol{b}}_{wk}$ and $\bar{\boldsymbol{b}}_{ck+1}$ are biases obtained by transforming the right-most term of Eq. (40).

In Eq. (42), the upper blocks relating to process noise are required for consider analysis of a smoother for the given system. Consider analysis of a smoother can be accomplished by extending the present techniques, as is done in another paper by the authors.[24] For the remainder of the filter analysis derived here, only the lower blocks of Eq. (42) are retained:

$$\bar{\mathcal{R}}_{xxk+1}\boldsymbol{x}_{k+1} = \bar{\boldsymbol{z}}_{k+1} - \begin{bmatrix}\bar{S}_{x\alpha k} & \bar{S}_{x\beta k}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k\end{bmatrix} - \bar{\boldsymbol{b}}_{ck+1} \tag{43}$$

This equation is just the consider version of the *a priori* state information equation at sample $k+1$. Note that its error vector is still written in terms of $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$, rather than $\boldsymbol{\alpha}_{k+1}$ and $\boldsymbol{\beta}_{k+1}$.

The measurement update step at sample $k+1$ closely resembles the procedure of Section A, except that the composite random vector $\begin{bmatrix}\boldsymbol{\alpha}_k^{\mathrm{T}} & \boldsymbol{\beta}_k^{\mathrm{T}} & \boldsymbol{w}_{\boldsymbol{c}k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ appears instead of $\boldsymbol{x}_{\boldsymbol{c}k+1}$. After stacking *a priori* information Eq. (43) and the measurement equation, one obtains an analog of Eq. (22):

$$\begin{bmatrix}\bar{\mathcal{R}}_{xxk+1} \\ H_{fk+1}\end{bmatrix}\boldsymbol{x}_{k+1} = \begin{bmatrix}\bar{\boldsymbol{z}}_{k+1} \\ \boldsymbol{y}_{k+1}\end{bmatrix} - \begin{bmatrix}\bar{S}_{x\alpha k} & \bar{S}_{x\beta k} & 0 \\ (H_{ck+1}\Phi_{ck}L_{\alpha k}) & (H_{ck+1}\Phi_{ck}L_{\beta k}) & (H_{ck+1}\Gamma_{cck})\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k}\end{bmatrix} - \begin{bmatrix}\bar{\boldsymbol{b}}_{ck+1} \\ \boldsymbol{b}_{yk+1}\end{bmatrix} \tag{44}$$

The top line of this equation is just Eq. (43). The bottom line is a modified version of measurement Eq. (15b) at sample $k+1$. In this measurement equation, $\boldsymbol{x}_{\boldsymbol{c}k+1}$ is replaced by the expression on the extreme right-hand side of Eq. (37). As in the $k=0$ case, the orthonormal transformation matrix $\widehat{T}_{k+1}$ is computed via QR factorization of the block matrix on the left, such that

$$\widehat{T}_{k+1}\begin{bmatrix}\bar{\mathcal{R}}_{xxk+1} \\ H_{fk+1}\end{bmatrix} = \begin{bmatrix}\widehat{\mathcal{R}}_{xxk+1} \\ 0\end{bmatrix} \tag{45}$$

where the additional factorization output $\widehat{\mathcal{R}}_{xxk+1}$ is square and upper triangular. Left matrix multiplication of Eq. (44) by $\widehat{T}_{k+1}$ yields:

$$\begin{bmatrix}\widehat{\mathcal{R}}_{xxk+1} \\ 0\end{bmatrix}\boldsymbol{x}_{k+1} = \begin{bmatrix}\hat{\boldsymbol{z}}_{k+1} \\ \boldsymbol{z}_{rk+1}\end{bmatrix} - \begin{bmatrix}\tilde{S}_{x\alpha k} & \tilde{S}_{x\beta k} & \tilde{S}_{xwck} \\ \tilde{S}_{r\alpha k} & \tilde{S}_{r\beta k} & \tilde{S}_{rwck}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k}\end{bmatrix} - \begin{bmatrix}\hat{\boldsymbol{b}}_{ck+1} \\ \boldsymbol{b}_{rk+1}\end{bmatrix} \tag{46}$$

Equation (46) is a transformed version of Eq. (44) in the same way as Eq. (42) is a transformed version of Eq. (40). Therefore, the various new vectors and matrices on the right-hand side of Eq. (46) are computed from the corresponding terms in Eq. (44) in a manner analogous to the Eq. (42) derivation. The lower blocks of Eq. (46) are related to measurement residuals. Neglecting these, the *a posteriori* state information equation at sample time $k+1$ is given by

$$\widehat{\mathcal{R}}_{xxk+1}\boldsymbol{x}_{k+1} = \hat{\boldsymbol{z}}_{k+1} - \begin{bmatrix}\tilde{S}_{x\alpha k} & \tilde{S}_{x\beta k} & \tilde{S}_{xwck}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k}\end{bmatrix} - \hat{\boldsymbol{b}}_{ck+1} \tag{47}$$

Recall that the main algorithm derivation began with Eq. (34), the *a posteriori* state information equation at sample k. The current information equation is not yet in the same form, however, because it depends on the three vectors $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$, and $\boldsymbol{w}_{\boldsymbol{c}k}$ rather than the single vector $\boldsymbol{\alpha}_{k+1}$. At this point, an LQ factorization can be applied to transform the composite vector $\begin{bmatrix}\boldsymbol{\alpha}_k^{\mathrm{T}} & \boldsymbol{\beta}_k^{\mathrm{T}} & \boldsymbol{w}_{\boldsymbol{c}k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ and its coefficient matrix in Eq. (47). This process will yield the desired $\boldsymbol{\alpha}_{k+1}$ as a component of the transformed vector and $S_{x\alpha k+1}$ as the only non-zero block of the transformed matrix, similar to the technique used in Eqs. (30)-(33) for sample $k=0$.

The recursion process also requires an updated version of Eq. (35) that applies at sample $k+1$. This new equation will express $\boldsymbol{x}_{\boldsymbol{c}k+1}$ as a linear combination of $\boldsymbol{\alpha}_{k+1}$ and a newly-defined $\boldsymbol{\beta}_{k+1}$. It will be derived from the consider state dynamics using the dynamics form in Eq. (37). The derivation also uses an LQ factorization that transforms the vector $\begin{bmatrix}\boldsymbol{\alpha}_k^{\mathrm{T}} & \boldsymbol{\beta}_k^{\mathrm{T}} & \boldsymbol{w}_{\boldsymbol{c}k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$. It is convenient to LQ factorize the relevant

coefficient matrices from Eqs. (47) and (37) simultaneously. The required LQ factorization computes another orthonormal matrix C_{k+1} such that the following relationship holds:

$$\begin{bmatrix} \tilde{S}_{x\alpha k} & \tilde{S}_{x\beta k} & \tilde{S}_{xwck} \\ \Phi_{ck}L_{\alpha k} & \Phi_{ck}L_{\beta k} & \Gamma_{cck} \end{bmatrix} = \begin{bmatrix} S_{x\alpha k+1} & 0 & 0 \\ L_{\alpha k+1} & L_{\beta k+1} & 0 \end{bmatrix} C_{k+1} \tag{48}$$

The input to this LQ factorization is the block matrix on the left-hand side of Eq. (48). The upper block row is the coefficient of $\begin{bmatrix} \boldsymbol{\alpha}_k^{\mathrm{T}} & \boldsymbol{\beta}_k^{\mathrm{T}} & \boldsymbol{w}_{\boldsymbol{c}k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ from Eq. (47), and the lower row is the coefficient of the same vector in Eq. (37). In addition to C_{k+1}, the square lower-triangular matrices $S_{x\alpha k+1}$ and $L_{\beta k+1}$ and the general matrix $L_{\alpha k+1}$ are outputs of the factorization. The orthonormal matrix C_{k+1} is used to define a transformation of variables from the vector at sample k to a different vector that will apply at sample $k+1$:

$$\begin{bmatrix} \boldsymbol{\alpha}_{k+1} \\ \boldsymbol{\beta}_{k+1} \\ \boldsymbol{\gamma}_{k+1} \end{bmatrix} \equiv C_{k+1} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k} \end{bmatrix} \tag{49}$$

where $\begin{bmatrix} \boldsymbol{\alpha}_{k+1}^{\mathrm{T}} & \boldsymbol{\beta}_{k+1}^{\mathrm{T}} & \boldsymbol{\gamma}_{k+1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \sim \mathcal{N}(0, \mathrm{I})$ because the orthonormal transformation preserves the vector's identity covariance.

When this transformation is applied to the consider dynamics equation by substituting Eqs. (48) and (49) into Eq. (37), it yields

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{c}k+1} &= \begin{bmatrix} \Phi_{ck}L_{\alpha k} & \Phi_{ck}L_{\beta k} & \Gamma_{cck} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k} \end{bmatrix} = \begin{bmatrix} L_{\alpha k+1} & L_{\beta k+1} & 0 \end{bmatrix} C_{k+1} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k} \end{bmatrix} \\ &= \begin{bmatrix} L_{\alpha k+1} & L_{\beta k+1} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{k+1} \\ \boldsymbol{\beta}_{k+1} \\ \boldsymbol{\gamma}_{k+1} \end{bmatrix} = \begin{bmatrix} L_{\alpha k+1} & L_{\beta k+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{k+1} \\ \boldsymbol{\beta}_{k+1} \end{bmatrix} \end{aligned} \tag{50}$$

The final result is a copy of Eq. (35), referenced to sample $k+1$ instead of sample k. Likewise, one can substitute Eqs. (48) and (49) into Eq. (47) to get

$$\begin{aligned} \widehat{\mathcal{R}}_{xxk+1}\boldsymbol{x}_{k+1} &= \hat{\boldsymbol{z}}_{k+1} - \begin{bmatrix} \tilde{S}_{x\alpha k} & \tilde{S}_{x\beta k} & \tilde{S}_{xwck} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k} \end{bmatrix} - \hat{\boldsymbol{b}}_{ck+1} \\ &= \hat{\boldsymbol{z}}_{k+1} - \begin{bmatrix} S_{x\alpha k+1} & 0 & 0 \end{bmatrix} C_{k+1} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\beta}_k \\ \boldsymbol{w}_{\boldsymbol{c}k} \end{bmatrix} - \hat{\boldsymbol{b}}_{ck+1} \\ &= \hat{\boldsymbol{z}}_{k+1} - \begin{bmatrix} S_{x\alpha k+1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{k+1} \\ \boldsymbol{\beta}_{k+1} \\ \boldsymbol{\gamma}_{k+1} \end{bmatrix} - \hat{\boldsymbol{b}}_{ck+1} \\ &= \hat{\boldsymbol{z}}_{k+1} - S_{x\alpha k+1}\boldsymbol{\alpha}_{k+1} - \hat{\boldsymbol{b}}_{ck+1} \end{aligned} \tag{51}$$

The final line of Eq. (51) is a copy of Eq. (34), referenced to sample $k+1$ rather than sample k. At this point, the algorithms have come full circle and the dynamic propagation from $k+1$ to $k+2$ can commence.

C. Consider Covariance Calculations

One of the distinguishing features of the SRIF as opposed to the Kalman filter is that the filter does not compute its own covariance estimate at each step. Whenever the covariance is needed, however, it can be reconstructed from the square-root information matrices. This paper's consider covariance analysis operates in the same way; at any point the analyst may compute both the filter's presumed error covariance and the true covariance of the estimation error. More generally, the quantity of interest in a consider analysis will be the matrix mean square error (MSE), which consists of a covariance plus a rank-one term due to biases. These calculations can occur at the *a priori* or *a posteriori* stages of the analysis with equal simplicity.

To compute covariance, one starts with the *a priori* or *a posteriori* state information equation, both repeated here for convenience:

$$\bar{\mathcal{R}}_{xxk}\boldsymbol{x}_k = \bar{\boldsymbol{z}}_k - \begin{bmatrix}\bar{S}_{x\alpha k-1} & \bar{S}_{x\beta k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_{k-1}\\ \boldsymbol{\beta}_{k-1}\end{bmatrix} - \bar{\boldsymbol{b}}_{ck} \tag{52a}$$

$$\widehat{\mathcal{R}}_{xxk}\boldsymbol{x}_k = \hat{\boldsymbol{z}}_k - S_{x\alpha k}\boldsymbol{\alpha}_k - \hat{\boldsymbol{b}}_{ck} \tag{52b}$$

The filter's *a priori* and *a posteriori* state estimates are $\bar{\boldsymbol{x}}_k = \bar{\mathcal{R}}_{xxk}^{-1}\bar{\boldsymbol{z}}_k$ and $\hat{\boldsymbol{x}}_k = \widehat{\mathcal{R}}_{xxk}^{-1}\hat{\boldsymbol{z}}_k$, respectively. Errors in the filter's presumed covariances arise from the filter's incorrect assumption that the errors in the information equations have zero mean and identity covariance. Under this assumption, one can compute the filter's presumed covariances. They are

$$\overline{P}_{fxxk} = \mathrm{E}\Big[(\bar{\boldsymbol{x}}_k - \boldsymbol{x}_k)\,(\bar{\boldsymbol{x}}_k - \boldsymbol{x}_k)^{\mathrm{T}}\Big] = \bar{\mathcal{R}}_{xxk}^{-1}\mathrm{E}\big[\bar{\boldsymbol{\nu}}_{xk}\bar{\boldsymbol{\nu}}_{xk}^{\mathrm{T}}\big]\,\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}} = \bar{\mathcal{R}}_{xxk}^{-1}\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}} \tag{53a}$$

$$P_{fxxk} = \mathrm{E}\Big[(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)\,(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^{\mathrm{T}}\Big] = \widehat{\mathcal{R}}_{xxk}^{-1}\mathrm{E}\big[\boldsymbol{\nu}_{xk}\boldsymbol{\nu}_{xk}^{\mathrm{T}}\big]\,\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}} = \widehat{\mathcal{R}}_{xxk}^{-1}\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}} \tag{53b}$$

The true matrix MSEs of the filter's estimates are only slightly more difficult to compute and require no additional matrix inversions. Starting from Eq. (52a) for the *a priori* case, the true matrix MSE is given by

$$\begin{aligned}
\overline{P}_{xxk} &= \mathrm{E}\Big[(\bar{\boldsymbol{x}}_k - \boldsymbol{x}_k)\,(\bar{\boldsymbol{x}}_k - \boldsymbol{x}_k)^{\mathrm{T}}\Big]\\
&= \bar{\mathcal{R}}_{xxk}^{-1}\mathrm{E}\left[\left(\begin{bmatrix}\bar{S}_{x\alpha k-1} & \bar{S}_{x\beta k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_{k-1}\\ \boldsymbol{\beta}_{k-1}\end{bmatrix} + \bar{\boldsymbol{b}}_{ck}\right)\right.\\
&\qquad\left.\times\left(\begin{bmatrix}\bar{S}_{x\alpha k-1} & \bar{S}_{x\beta k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_{k-1}\\ \boldsymbol{\beta}_{k-1}\end{bmatrix} + \bar{\boldsymbol{b}}_{ck}\right)^{\mathrm{T}}\right]\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}}\\
&= \bar{\mathcal{R}}_{xxk}^{-1}\begin{bmatrix}\bar{S}_{x\alpha k-1} & \bar{S}_{x\beta k-1}\end{bmatrix}\mathrm{E}\left\{\begin{bmatrix}\boldsymbol{\alpha}_{k-1}\\ \boldsymbol{\beta}_{k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_{k-1}^{\mathrm{T}} & \boldsymbol{\beta}_{k-1}^{\mathrm{T}}\end{bmatrix}\right\}\begin{bmatrix}\bar{S}_{x\alpha k-1}^{\mathrm{T}}\\ \bar{S}_{x\beta k-1}^{\mathrm{T}}\end{bmatrix}\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}}\\
&\quad + \bar{\mathcal{R}}_{xxk}^{-1}\bar{\boldsymbol{b}}_{ck}\bar{\boldsymbol{b}}_{ck}^{\mathrm{T}}\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}}\\
&= \bar{\mathcal{R}}_{xxk}^{-1}\begin{bmatrix}\bar{S}_{x\alpha k-1} & \bar{S}_{x\beta k-1}\end{bmatrix}\begin{bmatrix}\bar{S}_{x\alpha k-1}^{\mathrm{T}}\\ \bar{S}_{x\beta k-1}^{\mathrm{T}}\end{bmatrix}\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}} + \bar{\mathcal{R}}_{xxk}^{-1}\bar{\boldsymbol{b}}_{ck}\bar{\boldsymbol{b}}_{ck}^{\mathrm{T}}\bar{\mathcal{R}}_{xxk}^{-\mathrm{T}}
\end{aligned} \tag{54}$$

Likewise, for the *a posteriori* case, starting from Eq. (52b),

$$\begin{aligned}
P_{xxk} &= \mathrm{E}\Big[(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^{\mathrm{T}}\Big]\\
&= \widehat{\mathcal{R}}_{xxk}^{-1}\mathrm{E}\left[\Big(S_{x\alpha k}\boldsymbol{\alpha}_k + \hat{\boldsymbol{b}}_{ck}\Big)\Big(S_{x\alpha k}\boldsymbol{\alpha}_k + \hat{\boldsymbol{b}}_{ck}\Big)^{\mathrm{T}}\right]\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}}\\
&= \widehat{\mathcal{R}}_{xxk}^{-1}S_{x\alpha k}\mathrm{E}\big[\boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^{\mathrm{T}}\big]\,S_{x\alpha k}^{\mathrm{T}}\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}} + \widehat{\mathcal{R}}_{xxk}^{-1}\hat{\boldsymbol{b}}_{ck}\hat{\boldsymbol{b}}_{ck}^{\mathrm{T}}\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}}\\
&= \widehat{\mathcal{R}}_{xxk}^{-1}S_{x\alpha k}S_{x\alpha k}^{\mathrm{T}}\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}} + \widehat{\mathcal{R}}_{xxk}^{-1}\hat{\boldsymbol{b}}_{ck}\hat{\boldsymbol{b}}_{ck}^{\mathrm{T}}\widehat{\mathcal{R}}_{xxk}^{-\mathrm{T}}
\end{aligned} \tag{55}$$

Thus, the true matrix MSE calculations differ from the filter's covariance calculations in that they are weighted by a product of sensitivity matrices and may have an additional bias term. Note that in the final lines of Eqs. (54) and (55), the first term is the true estimation error covariance, and the second term constitutes the bias effect. The sum of these two terms is designated "P_{xx}" even though it is a matrix MSE rather than a covariance. This non-standard notation emphasizes the fact that the consider quantities $\overline{P}_{xxk}$ and P_{xxk} are most directly comparable to the presumed filter covariances $\overline{P}_{fxxk}$ and P_{fxxk}.

V. Examples

To demonstrate the capabilities of the proposed consider analysis and provide a degree of confidence in the derived algorithms, several concrete examples have been developed. These examples were chosen because they are sufficiently simple to explain, yet they address some realistic engineering problems. One of

the biggest practical challenges of this paper's consider covariance analysis is the process of rewriting each problem in the defined consider form of Section III. Once the examples are in consider form, the analysis algorithms of Section IV can be implemented.

The results of the consider analysis for each example are compared to the results of Monte Carlo simulations. These simulations verify that the true matrix MSEs of the incorrectly-modeled filters are indeed those predicted by the consider covariance analysis. The simulations also prove to be a useful debugging tool for the code that implements the algorithms.

The remainder of this section is organized as follows: Sections A and B lay out the two examples, one of mismodeled noise and the other with incorrect system matrices. Each section begins with a scenario description and then shows step-by-step how to rewrite the given systems in consider form. The Monte Carlo simulations are explained, and the simulation results are presented and discussed.

A. Example: Incorrectly Modeled Noise

1. Scenario Description and Consider Form Setup

The dynamics and measurement models for the first example are given by

$$\begin{bmatrix} r_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k \tag{56a}$$

$$y_k = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} r_k \\ v_k \end{bmatrix} + \nu_k \tag{56b}$$

The states r_k and v_k are thought of as position and velocity for some one-dimensional motion, and the sample interval Δt is assumed to be 0.5 s. The variables w_k and ν_k are scalar process and measurement noise, respectively, and y_k is a scalar measurement. The initial state estimate and initial state error covariance are

$$\bar{\boldsymbol{x}}_0 = \begin{bmatrix} \bar{r}_0 \\ \bar{v}_0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad \overline{P}_{xx0} = \begin{bmatrix} \sigma_{r0}^2 & \sigma_{rv0} \\ \sigma_{rv0} & \sigma_{v0}^2 \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 5 \end{bmatrix} \tag{57}$$

The filter operates on measurements available once per sample interval Δt for a total of 50 s (100 discrete-time samples).

One of the most practical uses for consider covariance analysis is to deal with the situation of incorrect process and measurement noise covariances. Specifically, in this example the filter process noise covariance Q_{fk} and measurement noise covariance R_{fk} are assumed to be

$$Q_{fk} = 1, \quad R_{fk} = 1 \tag{58}$$

The "truth" process noise covariance Q_k and measurement noise covariance R_k are

$$Q_k = 0.25, \quad R_k = 2.25 \tag{59}$$

Note that the "truth" process noise covariance is smaller and the "truth" measurement noise covariance larger than modeled by the filter.

In this example, the system matrices Φ_k, Γ_k, and H_k can be extracted directly from Eqs. (56a) and (56b) for both the filter and the consider form. The initial state square-root information matrix $\bar{\mathcal{R}}_{xx0}$ can be computed from the initial estimation error covariance $\overline{P}_{xx0}$ as in Eq. (14) by using Cholesky factorization or some other standard matrix square-root routine. The initial information state $\bar{\boldsymbol{z}}_0$ can be obtained by substituting the values for $\bar{\mathcal{R}}_{xx0}$ and $\bar{\boldsymbol{x}}_0$ into the inverse of Eq. (13). These calculations yield

$$\bar{\mathcal{R}}_{xx0} = \begin{bmatrix} 1/\sqrt{10} & 0 \\ 0 & 1/\sqrt{5} \end{bmatrix}, \quad \bar{\boldsymbol{z}}_0 = \begin{bmatrix} 3/\sqrt{10} \\ 1/\sqrt{5} \end{bmatrix} \tag{60}$$

The filter's process noise information matrix $\mathcal{R}_{fwwk}$ is the inverse square root of Q_{fk}; it can also be computed by Cholesky factorization, but in this scalar case will just be $1/\sqrt{1} = 1$. Likewise, the "truth" inverse square roots for process and measurement noise, which will be needed later, are just $\mathcal{R}_{wwk} = 1/\sqrt{0.25} = 2$ and $\mathcal{R}_{\nu\nu k} = 1/\sqrt{2.25} = 2/3$.

So far, the quantities defined or computed hold for both the filter's equations and the system's consider form. In order to put each system into consider form, the user must correctly define the matrices Γ_{xck}, Φ_{ck}, Γ_{cck}, S_{wck}, H_{ck}, and $\bar{S}_{xc0}$, which appear in Eqs. (15)-(18). One must also compute any deterministic biases, but for this example there are none. Typically, the design of the matrices is not unique, as it depends on the particular choice and ordering of elements of the consider state vector $\boldsymbol{x_{ck}}$.

A rough procedure for transforming a given system to consider form is as follows: First, one defines $\boldsymbol{x_{ck}}$ so that it contains as elements all the sources of random uncertainty or modeling error in the system, either directly or as some transformed version. Particular care must be taken in defining the initial $\boldsymbol{x_{c0}}$ in a way that also includes initial estimation uncertainty. In addition, it must have zero mean and identity covariance. Next, one writes the equations that describe the dynamics and statistics of the uncertainty components contained in $\boldsymbol{x_{ck}}$. This step typically involves designing the matrices Φ_{ck} and Γ_{cck}, as well as precomputing any deterministic bias disturbances. Third, one specifies how the uncertainty enters the system dynamics and measurements by creating the matrices Γ_{xck} and H_{ck}. Finally, one writes expressions for the sensitivity matrices S_{wck} and $\bar{S}_{xc0}$ that determine how the consider state vector $\boldsymbol{x_{ck}}$ influences the filter's process noise equation and its initial state information equation. Although all of these steps are necessary to define a consider system, it may not always be convenient to perform them in this order. Trial and error may help to determine the most useful of the system description alternatives.

All of the errors in this example are related to noise statistics, and the only additional uncertainty that must be captured by the consider vector $\boldsymbol{x_{ck}}$ is the correctly-modeled uncertainty in the initial estimate. One suitable definition is

$$\boldsymbol{x_{ck}} \equiv \begin{cases} \begin{bmatrix} \boldsymbol{\nu}_{x0} \\ \mathcal{R}_{ww0} w_0 \\ \mathcal{R}_{\nu\nu 0} \nu_0 \end{bmatrix} & k = 0 \\ \begin{bmatrix} w_k \\ \nu_k \end{bmatrix} & k > 0 \end{cases} \tag{61}$$

where the premultiplication at time $k = 0$ by the various square-root information matrices constrains $\boldsymbol{x_{c0}}$ to have identity covariance.

Based on this definition, the consider dynamics equation can be constructed. The first component of $\boldsymbol{x_{c0}}$ is related to uncertainty in the initial estimate; it does not enter the system directly at any later times and thus has no dynamic behavior to describe. The process and measurement noise elements, w_k and ν_k, are zero-mean white stochastic random variables. They do not have any true dynamics because samples at different times are uncorrelated. Thus, Φ_{ck} is an appropriately-dimensioned matrix of zeros for all samples k. Note that Φ_{c0} will have fewer rows than columns in order to omit the component of $\boldsymbol{x_{c0}}$ related to initial estimate uncertainty from all later $\boldsymbol{x_{ck}}$ consider states. Even though the dynamics of w_k and ν_k are trivial, the consider dynamics equation is used to describe their "truth" statistical behavior by means of the Γ_{cck} matrix, which applies both at $k = 0$ and later sample times:

$$\Gamma_{cck} = \begin{bmatrix} \mathcal{R}^{-1}_{wwk+1} & 0 \\ 0 & \mathcal{R}^{-1}_{\nu\nu k+1} \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 1.5 \end{bmatrix} \tag{62}$$

No unestimated disturbances enter the state dynamics equation, so Γ_{xck} is a matrix of zeros. In the measurement equation, the matrix H_{ck} is a block matrix which extracts the measurement noise component of $\boldsymbol{x_{ck}}$:

$$H_{ck} = \begin{cases} \begin{bmatrix} 0 & 0 & \mathcal{R}^{-1}_{\nu\nu 0} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.5 \end{bmatrix} & k = 0 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & k > 0 \end{cases} \tag{63}$$

The first component of $\boldsymbol{x_{ck}}$ (or the second component when $k = 0$) is the "truth" system process noise. The consider analysis knows how the "truth" process noise enters the filter's process noise information equation, Eq. (17). It designs S_{wck} to yield the "truth" process noise covariance, taking into account the consider dynamics model:

$$S_{wck} = \begin{cases} \begin{bmatrix} 0 & -\left(\mathcal{R}_{fww0} \mathcal{R}^{-1}_{ww0}\right) & 0 \end{bmatrix} = \begin{bmatrix} 0 & -0.5 & 0 \end{bmatrix} & k = 0 \\ \begin{bmatrix} -\mathcal{R}_{fwwk} & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 \end{bmatrix} & k > 0 \end{cases} \tag{64}$$

Finally, the sensitivity matrix $\bar{S}_{xc0}$ effectively contains the information about the way in which the "truth" initial estimation error covariance differs from what the filter assumes. In this example, of course, there is no difference.

$$\bar{S}_{xc0} = \begin{bmatrix} (\bar{\mathcal{R}}_{fxx0}\bar{\mathcal{R}}_{xx0}^{-1}) & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{65}$$

2. Monte Carlo Simulations

At this point, all the consider form matrices have been defined, and the algorithms of Section IV can be applied to compute both filter and true covariances analytically. True covariance can also be estimated numerically by means of Monte Carlo simulations. This technique computes sample mean and covariance from a large number of simulated estimation errors, which can be thought of as samples of a random process. As the number of trials increases, the estimated covariance approaches the true covariance. This paper's examples each use 5000 trials in their Monte Carlo simulations.

To implement the simulations, a "truth" model is created for the scenario based on the defined system matrices and "truth" noise covariances. The "truth" model takes as inputs the initial state and histories of measurement and process noise, which are constrained to have the statistical properties specified in the example. It outputs a history of the state vector and a series of noisy measurements. None of the "truth" model calculations use the special consider model form developed in the previous subsection. These "truth" measurements are fed into the incorrectly-modeled filter, which outputs histories of state estimates. By differencing the "truth" model's states and the filter's state estimates, a history of estimation error is generated.

This entire procedure is repeated many times, each with different samples from the same "truth" distributions of the initial state, process noise, and measurement noise. The result is a large number of estimation error histories, each of which is an independent instance of the same random process. Sample means, covariances, and standard deviations are computed at each time step, and compared with the analytical results from a single run of the consider analysis.

The Monte-Carlo covariance matrices agree closely with the covariances computed by the consider analysis. This result holds for the elements corresponding to both the position and velocity states, as well as the cross-covariances of position and velocity. Additionally, 99% confidence bounds on the estimated covariances have been computed, and it has been verified that the differences between the Monte Carlo and consider analysis covariances generally fall well within the expected bounds.

To illustrate, Fig. 1 plots three related quantities: The standard deviations of error in the *a posteriori* position estimates as computed by the incorrect filter, by the consider analysis, and by the Monte Carlo simulations. In this figure, the circles representing the Monte Carlo estimates of standard deviations clearly

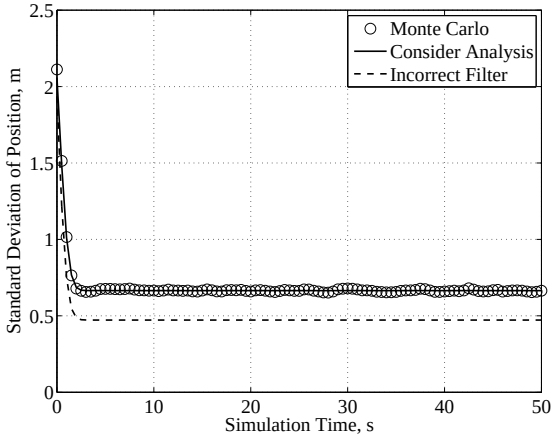

Figure 1. Standard deviations of *a posteriori* position error for incorrect noise example.

follow the solid line of the analytical consider standard deviations. Both are greater than the filter's computed standard deviations by an approximately-constant offset after the filter reaches a steady state. This offset represents the additional error one could expect in the estimates produced by a filter that used incorrect

noise covariances as described. Note that the Monte Carlo 99% confidence bounds are not shown on the plot because they tend to add clutter.

Although filter modeling errors in general are likely to *increase* the estimation error standard deviations as shown in Fig. 1, this is not always the case. In some scenarios, the true covariance could actually be lower than the filter's computed covariance. This is in fact the case for the standard deviations associated with the *a priori* velocity estimates for this example. The reason for this is simple: the "truth" process noise intensity is lower than that assumed by the filter, so immediately after a dynamic propagation step the true estimation error standard deviations are lower than those predicted by the filter. After each measurement update, however, the higher-than-modeled measurement noise intensity reverses the situation so that the true *a posteriori* standard deviations are higher than those of the filter. Even though the filter would have a lower *a priori* velocity estimation error than predicted, its estimates would be computed based on suboptimal filter gains. In other words, the true error would be lower than the filter predicts but not as low as it could be with an optimal filter.

It is also important to note that while Monte Carlo simulations can yield the same covariance results as the consider analysis and are conceptually simpler to program, they incur a far greater computational burden. Each Monte Carlo trial must re-run the entire "truth" model and filter, and a large number of trials may be required for sufficient accuracy. In contrast, the consider analysis runs just once, with only moderate computational increase beyond that of a single filter run. The consider analysis also has no need to compute the specific estimates associated with a particular measurement history realization, and so does not require a "truth"-model run to produce measurements.

B. Example: Incorrect System Matrices

1. Scenario Description and Consider Form Setup

The filter used in the second example is identical to that of the first example. As before, the dynamics and measurement models are given by Eqs. (56a) and (56b), with the same position and velocity states r_k and v_k and time step $\Delta t = 0.5$ s. The initial estimate and covariance of Eq. (57) are also used, and the nominal process and measurement noise covariances are those of Eq. (58). From these equations, the filter matrices Φ_{fk}, Γ_{fk}, H_{fk}, $\overline{\mathcal{R}}_{fxx0}$, and $\mathcal{R}_{fwwk}$ can be extracted.

In contrast to the previous example, the "truth" process and measurement noise covariances are both identical to those of the filter, such that $Q_k = Q_{fk}$ and $R_k = R_{fk}$. Once again, the filter's initial covariance $\overline{P}_{fxx0}$ is assumed to be correct.

The "truth" model system equations, however, have slightly different matrices than those assumed by the filter, such that $\Phi_k \neq \Phi_{fk}$, $\Gamma_k \neq \Gamma_{fk}$, and $H_k \neq H_{fk}$. The "truth" dynamics and measurement models are given by

$$\begin{bmatrix} r_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 0.95 & 1.01\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} w_k \tag{66a}$$

$$y_k = \begin{bmatrix} 0.95 & 1.05 \end{bmatrix} \begin{bmatrix} r_k \\ v_k \end{bmatrix} + \nu_k \tag{66b}$$

The generalized consider system defined by Eqs. (15a) and (15b) writes the "truth" system dynamics as a perturbed version of the filter's dynamics equation, and writes the "truth" system measurement model as a perturbed version of the filter's measurement model. To carry out this procedure for the current example, one need only see what terms must be added to the filter system equations in order to make them equal the "truth" system equations. Specifically,

$$\boldsymbol{x}_{k+1} = \Phi_{fk}\boldsymbol{x}_k + \Gamma_{fk}\boldsymbol{w}_k + (\Phi_k - \Phi_{fk})\,\boldsymbol{x}_k + (\Gamma_k - \Gamma_{fk})\,\boldsymbol{w}_k \tag{67a}$$

$$\boldsymbol{y}_k = H_{fk}\boldsymbol{x}_k + \boldsymbol{\nu}_k + (H_k - H_{fk})\,\boldsymbol{x}_k \tag{67b}$$

The perturbation to the filter's dynamics must next be written as a sum of a zero-mean stochastic part, $\Gamma_{xck}\boldsymbol{x}_{\boldsymbol{c}k}$, and a deterministic time-varying bias part $\boldsymbol{b}_{xk}$. Likewise, the perturbation to the filter's measurement equation must be decomposed into a zero-mean stochastic part, $H_{ck}\boldsymbol{x}_{\boldsymbol{c}k}$, and a deterministic time-varying bias part $\boldsymbol{b}_{yk}$.

A significant difficulty for the current example arises because both perturbations depend on $\boldsymbol{x}_k$. This state vector is driven by the stochastic noise $\boldsymbol{w}_k$, but has a time-varying, deterministic, non-zero mean due to non-zero $\bar{\boldsymbol{x}}_0$. The effect of the non-zero mean can be resolved by recognizing that $\boldsymbol{x}_k$ can be written as

$$\boldsymbol{x}_k = \tilde{\boldsymbol{x}}_k + \boldsymbol{x}_{bk} \tag{68}$$

where $\tilde{\boldsymbol{x}}_k$ is the zero-mean, stochastic part of $\boldsymbol{x}_k$ driven by $\boldsymbol{w}_k$, and $\boldsymbol{x}_{bk}$ is the deterministic part of $\boldsymbol{x}_k$. The component $\boldsymbol{x}_{bk}$ can be pre-computed for any sample k by propagating the filter's initial estimate (which can be thought of as $\boldsymbol{x}_{b0} = \bar{\boldsymbol{x}}_0$) through the "truth" dynamics equation while setting the process noise to zero. In other words,

$$\boldsymbol{x}_{bk} = \Phi_{k-1}\Phi_{k-2}\cdots\Phi_0\boldsymbol{x}_{b0} \tag{69}$$

The stochastic component $\tilde{\boldsymbol{x}}_k$, on the other hand, is initially drawn from a zero-mean distribution with covariance specified by $\overline{P}_{xx0}$. It has dynamics modeled by the system's "truth" dynamics equation:

$$\tilde{\boldsymbol{x}}_{k+1} = \Phi_k\tilde{\boldsymbol{x}}_k + \Gamma_k\boldsymbol{w}_k \tag{70}$$

It is now possible to write the perturbation to the filter's dynamics as a random part, $(\Phi_k - \Phi_{fk})\,\tilde{\boldsymbol{x}}_k + (\Gamma_k - \Gamma_{fk})\,\boldsymbol{w}_k$, and a deterministic part, $(\Phi_k - \Phi_{fk})\,\boldsymbol{x}_{bk}$. The corresponding random part of the measurement model's perturbation is $(H_k - H_{fk})\,\tilde{\boldsymbol{x}}_k$, and its deterministic part is $(H_k - H_{fk})\,\boldsymbol{x}_{bk}$.

The consider vector $\boldsymbol{x}_{\boldsymbol{c}k}$ must capture the dynamical and statistical behavior of the random parts of the system. In this example, those random parts include process noise, measurement noise, initial uncertainty, and a copy of the random part of $\boldsymbol{x}_k$ itself. A suitable choice for the consider vector is therefore:

$$\boldsymbol{x}_{\boldsymbol{c}k} \equiv \begin{cases} \begin{bmatrix} \bar{\mathcal{R}}_{xx0}\tilde{\boldsymbol{x}}_0 \\ \mathcal{R}_{ww0}w_0 \\ \mathcal{R}_{\nu\nu0}\nu_0 \end{bmatrix} & k = 0 \\ \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ w_k \\ \nu_k \end{bmatrix} & k > 0 \end{cases} \tag{71}$$

As before, the consider vector at sample $k = 0$ is defined such that it has identity covariance. In contrast to the preceding example, the consider state at sample $k = 0$ has no more elements than the consider states at all samples $k > 0$. With this definition of $\boldsymbol{x}_{\boldsymbol{c}k}$, one can write the matrices for the consider dynamics equation for $k > 0$:

$$\Phi_{ck} = \begin{bmatrix} \Phi_k & \Gamma_k & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.95 & 1.01\Delta t & 0.1 & 0 \\ 0 & 1 & 0.9 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{72a}$$

$$\Gamma_{cck} = \begin{bmatrix} 0 & 0 \\ \mathcal{R}^{-1}_{wwk+1} & 0 \\ 0 & \mathcal{R}^{-1}_{\nu\nu k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{72b}$$

Note how the consider dynamics matrices model the dynamic behavior of $\tilde{\boldsymbol{x}}_k$, the statistical relationship between $\tilde{\boldsymbol{x}}_k$ and w_k, and the covariances of w_k and ν_k. This definition of the consider vector allows the consider dynamics equation to model the "truth" system dynamics, which the filter cannot do with its incorrect system matrices Φ_{fk} and Γ_{fk}. At $k = 0$, the Γ_{cck} matrix is unchanged, but Φ_{ck} becomes

$$\Phi_{c0} = \begin{bmatrix} \Phi_0\bar{\mathcal{R}}^{-1}_{xx0} & \Gamma_0\mathcal{R}^{-1}_{ww0} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.95\sqrt{10} & 1.01\sqrt{5}\Delta t & 0.1 & 0 \\ 0 & \sqrt{5} & 0.9 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{73}$$

in order to transition from the original identity-covariance consider state vector at $k = 0$ to its new, non-identity-covariance form for $k > 0$.

In contrast to the previous example, Γ_{xck} is not a matrix of zeros, and $\boldsymbol{b}_{xk}$ is not a vector of zeros. The matrix Γ_{xck}, in combination with the defined $\boldsymbol{x}_{\boldsymbol{c}k}$, specifies the random part of the perturbation to the filter's dynamics equation. This matrix is given by

$$\Gamma_{xck} = \begin{bmatrix} (\Phi_k - \Phi_{fk}) & (\Gamma_k - \Gamma_{fk}) & 0 \end{bmatrix} = \begin{bmatrix} -0.05 & 0.01\Delta t & 0.1 & 0 \\ 0 & 0 & -0.1 & 0 \end{bmatrix} \tag{74}$$

for $k > 0$ and by

$$\Gamma_{xc0} = \begin{bmatrix} (\Phi_0 - \Phi_{f0})\,\bar{\mathcal{R}}_{xx0}^{-1} & (\Gamma_0 - \Gamma_{f0})\,\mathcal{R}_{ww0}^{-1} & 0 \end{bmatrix} = \begin{bmatrix} -0.05\sqrt{10} & 0.01\sqrt{5}\Delta t & 0.1 & 0 \\ 0 & 0 & -0.1 & 0 \end{bmatrix} \tag{75}$$

for the special case of $k = 0$.

The deterministic disturbance $\boldsymbol{b}_{xk}$ can be computed in advance by employing the formula

$$\boldsymbol{b}_{xk} = (\Phi_k - \Phi_{fk})\,\boldsymbol{x}_{bk} = (\Phi_k - \Phi_{fk})\,\Phi_{k-1}\Phi_{k-2}\cdots\Phi_0\boldsymbol{x}_{b0} \tag{76}$$

where $\boldsymbol{x}_{b0}$ is the initial estimate $[3 \quad 1]^{\mathrm{T}}$.

Just as Γ_{xck} models the sensitivity of the filter's dynamics equation to $\boldsymbol{x}_{\boldsymbol{c}k}$, the consider measurement sensitivity matrix H_{ck} models the way in which $\boldsymbol{x}_{\boldsymbol{c}k}$ perturbs the filter's measurements. Additionally, it must include the effects of the correctly-modeled measurement noise ν_k. This matrix is given by

$$H_{ck} = \begin{cases} \begin{bmatrix} (H_0 - H_{f0})\,\bar{\mathcal{R}}_{xx0}^{-1} & 0 & \mathcal{R}_{\nu\nu0}^{-1} \end{bmatrix} = \begin{bmatrix} -0.05\sqrt{10} & 0.05\sqrt{5} & 0 & 1 \end{bmatrix} & k = 0 \\ \begin{bmatrix} (H_k - H_{fk}) & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.05 & 0.05 & 0 & 1 \end{bmatrix} & k > 0 \end{cases} \tag{77}$$

The effect of the deterministic part of $\boldsymbol{x}_k$ on the measurements is the consider measurement bias $\boldsymbol{b}_{yk}$, which is just

$$\boldsymbol{b}_{yk} = (H_k - H_{fk})\,\boldsymbol{x}_{bk} = (H_k - H_{fk})\,\Phi_{k-1}\Phi_{k-2}\cdots\Phi_0\boldsymbol{x}_{b0} \tag{78}$$

The matrix S_{wck} selects the components of $\boldsymbol{x}_{\boldsymbol{c}k}$ related to process noise. Because the "truth" and modeled process noise are identical in this example, it is relatively simple:

$$S_{wck} = \begin{cases} \begin{bmatrix} 0 & -\left(\mathcal{R}_{fww0}\mathcal{R}_{ww0}^{-1}\right) & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \end{bmatrix} & k = 0 \\ \begin{bmatrix} 0 & -\mathcal{R}_{fwwk} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \end{bmatrix} & k > 0 \end{cases} \tag{79}$$

Finally, the matrix $\bar{S}_{xc0}$ specifies that the initial estimation error covariance is correct.

$$\bar{S}_{xc0} = \begin{bmatrix} -\mathrm{I} & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{80}$$

Likewise, the initial estimation error has a mean of zero, so the bias $\bar{\boldsymbol{b}}_{c0}$ is zero. At this point, the system has been written in consider form, and the analysis algorithms can proceed in a standard fashion.

2. Monte Carlo Simulations

Once again, all elements of the Monte Carlo matrix MSEs agree closely with the analytical matrix MSEs found by the consider analysis, and they fall within the computed confidence bounds. Figure 2, the Example 2 equivalent of Fig. 1, shows the root mean square (RMS) estimation errors for the *a posteriori* position estimates. The close agreement between the solid curve and the circles in Fig. 2 shows the correctness of the consider analysis.

Because this example includes biases, i.e. $\boldsymbol{b}_{xk}$ and $\boldsymbol{b}_{yk}$, the matrix MSE is the consider analysis quantity of interest. True error covariances could be computed, but they are less interesting than the matrix MSE because they only constitute a portion of the total error.

In the incorrect noise example of Fig. 1, the estimation error standard deviations leveled off to a constant value after some transient behavior. For this example, in contrast, the true RMS errors diverge steadily from

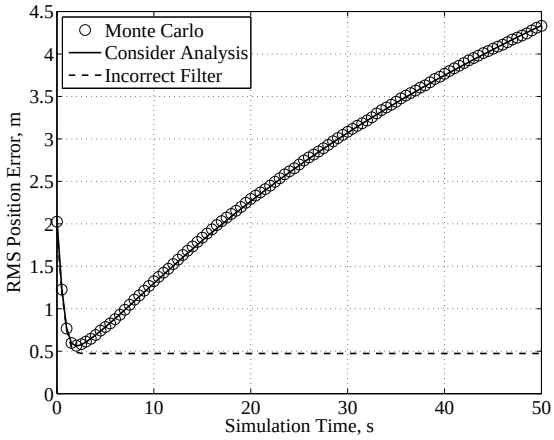

Figure 2. RMS *a posteriori* position errors for incorrect system matrices example.

those reported by the filter. Because the filter is not modeling the correct system dynamics or measurements, it is unable to maintain trustworthy state estimates.

This second example examined filter performance for a particular set of incorrect system matrices. Given the possibility of uncertainty in the "truth" system matrices, a more valuable analysis might compute the distribution of RMS estimation errors resulting from a chosen statistical distribution of system matrix errors. Such results can be obtained by wrapping an outer Monte Carlo analysis around a consider covariance analysis. Many different particular "truth" models would be generated for the given filter, and a consider analysis of each case would produce the corresponding RMS estimation error history. The result would be a Monte Carlo distribution of RMS estimation error histories. This distribution would characterize the performance range of a given filter design. Significant computational savings can be realized from this analysis approach. The consider covariance analysis that computes the true matrix MSE for each "truth"/filter combination replaces a burdensome inner Monte Carlo analysis. Computational efficiency is thus increased by two or three orders of magnitude. Additional savings are possible by optimizing the consider analysis algorithms for this application. Many of the calculations are specific to the filter being analyzed and are independent of the "truth" model. By performing such calculations only once rather than repeating them for each Monte Carlo "truth" model trial, approximately 25% of the remaining computational burden can be eliminated.

Note how this example and the preceding example are both linear, consistent with all of this paper's consider derivations. It should be possible, however, to extend these techniques to some nonlinear systems, similar to what is done in Ref. 25. The obvious extension would use linearizations of the system dynamics and measurement models. As in the case of an Extended Kalman Filter, such an approach should succeed if neglected higher-order terms are sufficiently small.

VI. Conclusions

This paper presents a new form of consider covariance analysis that can be applied to square-root information filters with many different forms of filter modeling errors. The analysis computes the true estimation error covariance or the true matrix mean square error, whichever is most relevant. It is suitable for studying filter design choices, for analyzing potential filter performance, or for calculating the uncertainty of previously computed estimates after a filter model error has been discovered. The new consider analysis begins by casting each problem into a predefined general form. This problem form makes it possible for the key algorithms to operate on a wide class of systems without customization.

Two specific examples demonstrate the power of the new method: One with incorrectly modeled noise and the other with erroneous system matrices. For these cases, Monte Carlo simulations provide independent verification of the consider analysis equations. Comparison of the resulting true and mismodeled root mean square estimation errors illustrates several of the types of degraded filter behavior that can be studied in this manner.

Appendix: QR and LQ Factorization Calculations

One well-known technique for matrix calculations is QR factorization, which decomposes any given matrix into a product of an orthonormal matrix "Q" on the left and an upper triangular matrix "R" on the right. Both orthonormal and triangular matrices have convenient theoretical and computational properties. Among other applications, QR factorization is a central feature of the standard SRIF algorithms.

The LQ factorization is also exploited in this paper's consider covariance analysis. It decomposes a given matrix into a lower triangular matrix "L" on the left and an orthonormal matrix "Q" on the right.

The computational routines necessary to perform QR factorization are available in many software packages including MATLAB; they will not be discussed here. The less-common LQ factorization can be obtained from a QR-factorization routine by implementing the following pseudocode, written with MATLAB-style syntax:

$$
\begin{aligned}
&\texttt{function}\,[L, Q] = \texttt{lq}\,(A)\\
&\qquad [\tilde{Q}, \tilde{R}] = \texttt{qr}\left(A^{\mathrm{T}}\right)\\
&\qquad Q = \tilde{Q}^{\mathrm{T}}\\
&\qquad L = \tilde{R}^{\mathrm{T}}\\
&\texttt{end function}
\end{aligned}
$$

This algorithm yields a lower-triangular matrix L and an orthonormal matrix Q such that $LQ = A$ for any square or rectangular matrix A. Of course, this pseudocode assumes that the statement $[Q, R] = \texttt{qr}(A)$ produces an orthonormal matrix Q and an upper-triangular matrix R such that $QR = A$. The function "qr" is standard in MATLAB.

References

[1]D'Appolito, J. and Hutchinson, C., "Low Sensitivity Filters for State Estimation in the Presence of Large Parameter Uncertainties," *IEEE Transactions on Automatic Control*, Vol. 14, No. 3, 1969, pp. 310–312.

[2]Gelb, A., editor, *Applied Optimal Estimation*, The Analytical Sciences Corporation, The M.I.T. Press, 1974.

[3]Lichten, S. M. and Border, J. S., "Strategies for High-Precision Global Positioning System Orbit Determination," *Journal of Geophysical Research*, Vol. 92, No. B12, Nov. 1987, pp. 12751–12762.

[4]Schutz, B. E., Cheng, M. K., Shum, C. K., Eanes, R. J., and Tapley, B. D., "Analysis of Earth Rotation Solution from Starlette," *Journal of Geophysical Research*, Vol. 94, No. B8, Aug. 1989, pp. 10167–10174.

[5]Bierman, G. J., "The Treatment of Bias in the Square-Root Information Filter/Smoother," *Journal of Optimization Theory and Applications*, Vol. 16, No. 1, 1975, pp. 165–178.

[6]Montenbruck, O. and Gill, E., *Satellite Orbits: Models, Methods, and Applications*, Springer-Verlag, 2000.

[7]Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Academic Press, 2004.

[8]Nishimura, T., "On the a priori Information in Sequential Estimation Problems," *IEEE Transactions on Automatic Control*, Vol. 11, No. 2, 1966, pp. 197–204.

[9]Heffes, H., "The Effect of Erroneous Models on the Kalman Filter Response," *IEEE Transactions on Automatic Control*, Vol. 11, No. 3, July 1966, pp. 541–543.

[10]Huddle, J. and Wismer, D., "Degradation of Linear Filter Performance Due to Modeling Error," *IEEE Transactions on Automatic Control*, Vol. 13, No. 4, 1968, pp. 421–423.

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

[12]Curkendall, D. W., *Problems in Estimation Theory with Applications to Orbit Determination*, Ph.D. thesis, UCLA School of Engineering and Applied Science, Sept. 1972.

[13]Fagin, S. L., "Recursive Linear Regression Theory, Optimal Filter Theory and Error Analysis of Optimal Systems," *IEEE international convention record*, Vol. 12, 1964, pp. 216–245.

[14]Griffin, R. E. and Sage, A. P., "Large and Small Scale Sensitivity Analysis of Optimum Estimation Algorithms," *IEEE Transactions on Automatic Control*, Vol. 13, No. 4, Aug. 1968, pp. 320–329.

[15]Griffin, R. E. and Sage, A. P., "Sensitivity Analysis of Discrete Filtering and Smoothing Algorithms," *AIAA Journal*, Vol. 7, No. 10, Oct. 1969, pp. 1890–1897.

[16]Sage, A. P. and Melsa, J. L., *Estimation Theory with Applications to Communications and Control*, McGraw-Hill Book Company, 1971.

[17]Morf, M., Verriest, E., Dobbins, J., and Kailath, T., "Square-Root Algorithms for Model Sensitivity Analysis," *Proc. of the 1977 Conf. on Information Sciences and Systems*, Vol. 11, Mar. 30-Apr. 1, 1977, pp. 64–69.

[18]Lewis, F. L., *Optimal Estimation*, John Wiley & Sons, 1986.

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

[20]Schmidt, S. F., "Applications of State-Space Methods to Navigation Problems," *Advances in Control Systems*, edited by C. T. Leondes, Vol. 3, Academic Press, New York, 1966, pp. 293–340.

[21]Woodbury, D. P. and Junkins, J. L., "On the Consider Kalman Filter," *AIAA Guidance, Navigation, and Control Conference*, 2-5 Aug. 2010, AIAA-2010-7752.

[22]Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises and Solutions*, John Wiley & Sons, 3rd ed., 1997.

[23]Dyer, P. and McReynolds, S., "Extension of Square-Root Filtering to Include Process Noise," *Journal of Optimization Theory and Applications*, Vol. 3, No. 6, 1969, pp. 444–458.

[24]Hinks, J. C. and Psiaki, M. L., "A Multipurpose Square-Root Consider Covariance Analysis for Linear Smoothers," *AIAA Guidance, Navigation, and Control Conference*, 13-16 Aug. 2012.

[25]Hinks, J. C. and Psiaki, M. L., "Estimation of Atmospheric and Ionospheric Effects in Multi-Satellite Orbit Determination Using Crosslinks," *AAS/AIAA Astrodynamics Specialist Conference*, 1-4 Aug. 2011, pp. 2117–2135.