Cascaded Filtering Using the Sigma Point Transformation

Mohammed Shalaby, Charles Champagne Cossette, Jerome Le Ny, and James Richard Forbes

***Abstract*—It is often convenient to separate a state estimation task into smaller "local" tasks, where each local estimator estimates a subset of the overall system state. However, neglecting cross-covariance terms between state estimates can result in overconfident estimates, which can ultimately degrade the accuracy of the estimator. Common cascaded filtering techniques focus on the problem of modelling cross-covariances when the local estimators share a common state vector. This letter introduces a novel cascaded and decentralized filtering approach that approximates the cross-covariances when the local estimators consider distinct state vectors. The proposed estimator is validated in simulations and in experiments on a three-dimensional attitude and position estimation problem. The proposed approach is compared to a naive cascaded filtering approach that neglects cross-covariance terms, a sigma point-based Covariance Intersection filter, and a full-state filter. In both simulations and experiments, the proposed filter outperforms the naive and the Covariance Intersection filters, while performing comparatively to the full-state filter.**

***Index Terms*—Sensor fusion, distributed robot systems.**

I. Introduction

State estimation is an integral part of technologies that use measured data to make decisions. For example, state estimation is used in time synchronization of clocks at the nanosecond level, and autonomous spatial navigation of quadcopters. For complex systems, augmenting all the required states into one monolithic state estimator can become laborious, inconvenient, or even infeasible due to computational or bandwidth limitations. Therefore, the ability to split the state estimation task into many different estimators is desirable [1]–[6]. For example, multi-robot systems would benefit from each robot having a local estimator, and large complex systems would ideally have a set of interconnected and specialized estimators, also known as cascaded estimators. Such modularity allows the independent design, analysis, tuning, debugging, and testing of each state estimator, in addition to providing architectural clarity. More efficient computing capability is also possible as smaller state vectors can be considered in a parallel framework.

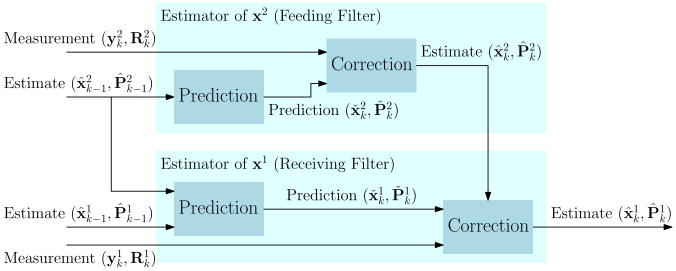
Estimator of $\mathbf{x}^2$ (Feeding Filter)
Measurement $(\mathbf{y}_k^2, \mathbf{R}_k^2)$
Estimate $(\hat{\mathbf{x}}_{k-1}^2, \hat{\mathbf{P}}_{k-1}^2)$
Prediction
Correction
Estimate $(\hat{\mathbf{x}}_k^2, \hat{\mathbf{P}}_k^2)$
Prediction $(\check{\mathbf{x}}_k^2, \check{\mathbf{P}}_k^2)$
Estimator of $\mathbf{x}^1$ (Receiving Filter)
Prediction
Prediction $(\check{\mathbf{x}}_k^1, \check{\mathbf{P}}_k^1)$
Estimate $(\hat{\mathbf{x}}_{k-1}^1, \hat{\mathbf{P}}_{k-1}^1)$
Correction
Estimate $(\hat{\mathbf{x}}_k^1, \hat{\mathbf{P}}_k^1)$
Measurement $(\mathbf{y}_k^1, \mathbf{R}_k^1)$

Fig. 1. A block diagram of the architecture of a cascaded filter at time-step k. The estimator of $\mathbf{x}^2$ is independent from the estimator of $\mathbf{x}^1$, while the estimator of $\mathbf{x}^1$ uses $\hat{\mathbf{x}}_{k-1}^2$ and $\hat{\mathbf{x}}_k^2$ as a measurement.

To be able to use cascaded filtering approaches, the problem of modelling the cross-covariance between the estimates of the different filters must be addressed [1], [7]–[9]. Consider an attitude and heading reference system (AHRS) providing an attitude estimate to a position estimator, where the AHRS is the *feeding filter* and the position estimator is the *receiving filter*, as depicted in Fig. 1. Any error in the AHRS results in an error in the position estimator, because sensors such as an accelerometer rely on attitude information when used for position estimation. Therefore, the estimation error of the attitude estimate and the position estimate are correlated. Assuming the attitude estimation error is uncorrelated with the position estimation error is equivalent to neglecting the fact that the attitude information is not entirely new, thus resulting in an overconfident estimate [4], [10]. In this letter, a state estimator is said to be *consistent* if its calculated error covariance does not underestimate the true error covariance. The approach of neglecting cross-covariances is common [2], [5], [6], [11], but can lead to divergence of the state estimate as a consequence of the filter inconsistency [1], [7], [9].

To accommodate cross-covariances, a master filter is proposed in [7], [8] named the *Federated filter*, that takes as input the output of all the local, sensor-specific filters, and fuses the estimates using a least squares algorithm. This is extended in [12] to nonlinear systems, by designing the master filter to be an unscented Kalman filter (UKF) [13, Section 5.6]. A more recent modification of the Federated filter is an optimal sample-based fusion algorithm discussed in [14]. Another popular fusion algorithm is *Covariance Intersection* (CI), introduced in [9], where a convex combination of the means and covariances of different estimates of the same unknown produces a consistent fused mean and covariance. This involves the selection of weighing parameters, and in [15], the authors address the optimal choice of weights in relation to state estimation of dynamical systems.

Manuscript received October 15, 2020; accepted March 1, 2021. Date of publication March 24, 2021; date of current version April 13, 2021. This letter was recommended for publication by Associate Editor S. Julier and Editor E. Marchand upon evaluation of the reviewers' comments. This work was supported in part by FRQNT under Grant 2018-PR-253646, in part by the William Dawson Scholar Program, in part by the NSERC Discovery Grant Program, and in part by CFI JELF Program. *(Corresponding author: Mohammed Shalaby.)*

Mohammed Shalaby, Charles Champagne Cossette, and James Richard Forbes are with the Department of Mechanical Engineering, McGill University, Montreal, QC H3A 0C3, Canada (e-mail: mohammed.shalaby@mail.mcgill.ca; charles.cossette@mail.mcgill.ca; james.richard.forbes@mcgill.ca).

Jerome Le Ny is with the Department of Electrical Engineering, Polytechnique Montreal, Montreal, QC H3T 1J4, Canada (e-mail: jerome.le-ny@polymtl.ca).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3068694, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3068694

2377-3766 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

A major limitation of all the discussed fusion algorithms is the focus on scenarios where the local filters share a set of common states. In [3], a method for computing pose estimates for a decentralized multi-robot system is presented. However, it is assumed that relative pose measurements between the robots are available to reconstruct a common shared state. Lastly, in [16], two approaches are introduced, one based on the CI method and another that hinges on solving a linear matrix inequality problem.

The presentation of a probabilistic approximation of the propagated cross-covariance terms when the feeding and receiving filters do not share a common set of states is a novel contribution of this letter. In particular, a novel cascaded and decentralized state estimation approach that approximates the cross-covariance terms using a sigma point transformation is introduced. Another contribution of this letter is demonstrating improved performance and consistency in simulation and experiments when compared to both an estimator that neglects cross-covariances and a sigma point-based Covariance Intersection (SPCI) estimator. The simulation and experiments consider attitude and position estimation but in a decoupled and decentralized manner.

The remainder of this letter is organized as follows. Preliminaries are presented in Section II. The cascaded filtering problem is formulated in Section III, where a probabilistic analysis of the approximate prior and posterior distributions of the receiving filter is also given. The proposed algorithm is presented in Section IV, and is then evaluated in simulations in Section V and in experiments in Section VI.

II. Preliminaries

In this letter, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used to denote a Gaussian random variable $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The notation $\check{(\cdot)}$ and $\hat{(\cdot)}$ denotes the predicted and measurement-corrected state estimates, respectively. The superscripts $(\cdot)^i$ are indices, not exponents. Lastly, $\boldsymbol{\Sigma}_{\mathbf{ab}}$ is used to denote the cross-covariance matrix between vectors $\mathbf{a}$ and $\mathbf{b}$, and the following notation is used for special cross-covariance matrices,

$$\check{\mathbf{P}}^i_k \triangleq \boldsymbol{\Sigma}_{\check{\mathbf{x}}^i_k\check{\mathbf{x}}^i_k}, \quad \hat{\mathbf{P}}^i_k \triangleq \boldsymbol{\Sigma}_{\hat{\mathbf{x}}^i_k\hat{\mathbf{x}}^i_k}, \quad \check{\mathbf{P}}^{1,2}_{p,q} \triangleq \boldsymbol{\Sigma}_{\check{\mathbf{x}}^1_p\check{\mathbf{x}}^2_q}, \quad \hat{\mathbf{P}}^{1,2}_{p,q} \triangleq \boldsymbol{\Sigma}_{\hat{\mathbf{x}}^1_p\hat{\mathbf{x}}^2_q}.$$

A. The Sigma Point Transform

The sigma point transform is a method used to approximate the nonlinear transformation of a distribution. By generating a set of sigma points from the *a priori* distribution and passing them through the nonlinear function, the newly generated points are used to approximate the nonlinearly transformed distribution [13, Section 5.5].

Consider a nonlinear function $\mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{\mathbf{P}}_x) \in \mathbb{R}^p$ and $\mathbf{y} \sim \mathcal{N}(\hat{\mathbf{y}}, \hat{\mathbf{P}}_y) \in \mathbb{R}^q$ are mutually uncorrelated random variables. To find the distribution of $\mathbf{z}$ using a sigma point transformation, the random variables are augmented into one vector $\mathbf{v} \sim \mathcal{N}(\hat{\mathbf{v}}, \hat{\mathbf{P}}_v) \in \mathbb{R}^{p+q}$, where

$$\hat{\mathbf{v}} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{bmatrix}, \quad \hat{\mathbf{P}}_v = \begin{bmatrix} \hat{\mathbf{P}}_x & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{P}}_y \end{bmatrix}. \tag{1}$$

The next step involves generating the sigma points from (1), which can be done using different approaches. The spherical cubature rule [13, Section 6.5] will be used throughout this letter. Define $L \triangleq \dim(\hat{\mathbf{v}})$ and the Cholesky decomposition $\mathbf{L}\mathbf{L}^{\mathsf{T}} \triangleq \hat{\mathbf{P}}_v$, where $\mathbf{L}$ is a lower-triangular matrix. The spherical cubature rule results in a total number of L sigma point pairs, where the i^{th} pair is defined as $\mathbf{s}_i \triangleq \hat{\mathbf{v}} + \sqrt{L}\,\mathrm{col}_i\,\mathbf{L}$, $\mathbf{s}_{i+L} \triangleq \hat{\mathbf{v}} - \sqrt{L}\,\mathrm{col}_i\,\mathbf{L}$. By unstacking the i^{th} sigma point into the two components $\mathbf{x}_i$ and $\mathbf{y}_i$, the i^{th} transformed point is $\mathbf{z}_i = \mathbf{h}(\mathbf{x}_i, \mathbf{y}_i)$. The new transformed distribution can then be approximated using the $2L$ transformed points through

$$\hat{\mathbf{z}} = \frac{1}{2L}\sum_{i=1}^{2L} \mathbf{z}_i, \quad \hat{\mathbf{P}}_z = \frac{1}{2L}\sum_{i=1}^{2L} (\mathbf{z}_i - \hat{\mathbf{z}})(\mathbf{z}_i - \hat{\mathbf{z}})^{\mathsf{T}}. \tag{2}$$

B. The Kalman Filter

Consider a discrete-time linear system given by

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}), \tag{3}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \boldsymbol{\nu}_k, \qquad \boldsymbol{\nu}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k). \tag{4}$$

The minimum mean square error (MMSE) estimator of $\mathbf{x}_k$ given past measurements is the Kalman filter [13, Section 4.3], where the prediction and correction steps are given by

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1}, \tag{5}$$

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}^{\mathsf{T}}_{k-1} + \mathbf{Q}_{k-1}, \tag{6}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k), \tag{7}$$

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}^{\mathsf{T}}_k\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}^{\mathsf{T}}_k + \mathbf{R}_k\right)^{-1}, \tag{8}$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\check{\mathbf{P}}_k, \tag{9}$$

with $\mathbf{1}$ being the identity matrix of appropriate dimensions.

III. Cascaded Filtering

Consider two discrete-time processes evolving through

$$\mathbf{x}^1_k = \mathbf{f}^1\left(\mathbf{x}^1_{k-1}, \mathbf{x}^2_{k-1}, \mathbf{w}^1_{k-1}\right), \tag{10}$$

$$\mathbf{x}^2_k = \mathbf{f}^2\left(\mathbf{x}^2_{k-1}, \mathbf{w}^2_{k-1}\right), \tag{11}$$

where $\mathbf{x}^1_k \in \mathbb{R}^{n_1}$ and $\mathbf{x}^2_k \in \mathbb{R}^{n_2}$ are distinct state vectors, and $\mathbf{w}^i_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^i_{k-1}) \in \mathbb{R}^{\ell_i}$ represents the process noise associated with the evolution of $\mathbf{x}^i_k$. Additionally, consider two measurement signals modelled as

$$\mathbf{y}^1_k = \mathbf{g}^1\left(\mathbf{x}^1_k, \mathbf{x}^2_k, \boldsymbol{\nu}^1_k\right), \tag{12}$$

$$\mathbf{y}^2_k = \mathbf{g}^2\left(\mathbf{x}^2_k, \boldsymbol{\nu}^2_k\right), \tag{13}$$

where $\mathbf{y}^1_k \in \mathbb{R}^{m_1}$ and $\mathbf{y}^2_k \in \mathbb{R}^{m_2}$ are distinct measurement vectors, and $\boldsymbol{\nu}^i_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^i_{k-1}) \in \mathbb{R}^{h_i}$ represents the measurement noise associated with $\mathbf{y}^i_k$. All process and measurement noise are assumed to be mutually independent.

In the process models (10) and (11), no inputs are considered to simplify the derivation of the proposed framework. However, this can be extended to systems with known inputs, since inputs in the receiving filter are dealt with in the standard way, and the approximation to be discussed in Section IV-C still holds when the feeding filter has inputs.

The standard, full-state filtering approach involves designing a filter with the augmented state vector $\mathbf{x}_k = \left[(\mathbf{x}^1_k)^{\mathsf{T}}\ (\mathbf{x}^2_k)^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{n_1+n_2}$, using knowledge of (10)–(13). Meanwhile, the cascaded filtering approach separates this problem into two filters, as shown in Fig. 1. The feeding filter outputs an approximate *a*

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

posteriori distribution of

$$\mathbf{x}_k^2 | \mathcal{I}_k^2 \sim \mathcal{N}\left(\hat{\mathbf{x}}_k^2, \hat{\mathbf{P}}_k^2\right), \tag{14}$$

where $\mathcal{I}_k^i = \{\check{\mathbf{x}}_0^i, \mathbf{y}_{0:k}^i\}$ is an "information" set containing a prior and measurements, $a|b$ denotes a random variable a conditioned on b, $\check{\mathbf{x}}_0^2 \in \mathbb{R}^{n_2}$ is the initial prediction of $\mathbf{x}_k^2$, $\hat{\mathbf{x}}_k^2 \in \mathbb{R}^{n_2}$ is its estimate at time-step k based on (11) and (13), and $\hat{\mathbf{P}}_k^2 \in \mathbb{R}^{n_2 \times n_2}$ is the associated covariance matrix. The receiving filter uses this output alongside (10) and (12) to estimate the *a posteriori* distribution of

$$\mathbf{x}_k^1 | \check{\mathbf{x}}_0^1, \mathbf{y}_{0:k}^1, \check{\mathbf{x}}_0^2, \mathbf{y}_{0:k}^2 \equiv \mathbf{x}_k^1 | \mathcal{I}_k^1, \mathcal{I}_k^2 \tag{15}$$

without access to $\mathcal{I}_k^2$. Naive cascaded filtering [2], [5], [6], [11] considers the estimate $\hat{\mathbf{x}}_k^2$ to be a measurement in the receiving filter. This breaks a requirement for consistency of the Kalman filter and its descendents, which is the conditional independence of the measurement from the state and measurement histories [13, Property 4.2].

A. Receiving Filter Approximate Probability Distributions

When the *a posteriori* distribution of (14) is known, the goal is to find a consistent estimator of (15) without access to the feeding filter's process model (11), measurement model (13), or corresponding inputs and measurements. Such a solution allows the design of an $\mathbf{x}^1$ estimator with minimal knowledge of the inner workings of the $\mathbf{x}^2$ estimator.

The propagation of $\mathbf{x}_k^1$ is dependent on $\mathbf{x}_{k-1}^2$, and mutual independence cannot be assumed. The conditional joint distribution of $\mathbf{x}_k^1$ and $\mathbf{x}_{k-1}^2$ is therefore assumed to be

$$\begin{bmatrix} \mathbf{x}_k^1 \\ \mathbf{x}_{k-1}^2 \end{bmatrix} \Bigg| \mathcal{I}_{k-1}^1, \mathcal{I}_{k-1}^2$$
$$\sim \mathcal{N}\left(\begin{bmatrix} \check{\mathbf{x}}_k^1 \\ \hat{\mathbf{x}}_{k-1}^2 \end{bmatrix}, \begin{bmatrix} \check{\mathbf{P}}_k^1 & \check{\mathbf{P}}_{k,k-1}^{1,2} \\ \left(\check{\mathbf{P}}_{k,k-1}^{1,2}\right)^{\mathsf{T}} & \hat{\mathbf{P}}_{k-1}^2 \end{bmatrix} \right), \tag{16}$$

where $\check{\mathbf{x}}_k^1 = \mathbf{f}^1(\hat{\mathbf{x}}_{k-1}^1, \hat{\mathbf{x}}_{k-1}^2, \mathbf{0})$. This block partitioning follows from an assumption that the conditional joint distribution is Gaussian. Using [13, Lemma A.2], (16) gives

$$\mathbf{x}_k^1 | \mathcal{I}_{k-1}^1, \mathcal{I}_{k-1}^2, \mathbf{x}_{k-1}^2$$
$$\sim \mathcal{N}\left(\check{\mathbf{x}}_k^1 + \check{\mathbf{P}}_{k,k-1}^{1,2} \left(\hat{\mathbf{P}}_{k-1}^2\right)^{-1} \left(\mathbf{x}_{k-1}^2 - \hat{\mathbf{x}}_{k-1}^2\right), \right.$$
$$\left. \check{\mathbf{P}}_k^1 - \check{\mathbf{P}}_{k,k-1}^{1,2} \left(\hat{\mathbf{P}}_{k-1}^2\right)^{-1} \left(\check{\mathbf{P}}_{k,k-1}^{1,2}\right)^{\mathsf{T}} \right). \tag{17}$$

and since $\mathcal{I}_{k-1}^2$ and $\mathbf{x}_{k-1}^2$ are available only through $\hat{\mathbf{x}}_{k-1}^2(\mathcal{I}_{k-1}^2)$ and $\hat{\mathbf{P}}_{k-1}^2(\mathcal{I}_{k-1}^2)$ in the receiving filter, $\mathbf{x}_{k-1}^2$ in particular is replaced in (17) with $\hat{\mathbf{x}}_{k-1}^2$ to obtain

$$\mathbf{x}_k^1 | \mathcal{I}_{k-1}^1, \hat{\mathbf{x}}_{k-1}^2, \hat{\mathbf{P}}_{k-1}^2$$
$$\sim \mathcal{N}\left(\check{\mathbf{x}}_k^1, \check{\mathbf{P}}_k^1 - \check{\mathbf{P}}_{k,k-1}^{1,2} \left(\hat{\mathbf{P}}_{k-1}^2\right)^{-1} \left(\check{\mathbf{P}}_{k,k-1}^{1,2}\right)^{\mathsf{T}} \right). \tag{18}$$

In (18), $\hat{\mathbf{P}}_{k-1}^2$ is known and $\check{\mathbf{P}}_k^1$ can be obtained using common filtering techniques. However, $\check{\mathbf{P}}_{k,k-1}^{1,2}$ is unknown and is required to consistently update the estimate of $\mathbf{x}_k^1$.

Similarly, an approximation of the distribution of (15) is derived in Appendix A yielding the correction step equations

$$\hat{\mathbf{x}}_k^1 = \check{\mathbf{x}}_k^1 + \mathbf{K}_k^1 \left(\mathbf{y}_k^1 - \check{\mathbf{y}}_k^1\right) - \mathbf{K}_k^{1,2} \mathbf{K}_k^2 \left(\mathbf{y}_k^1 - \check{\mathbf{y}}_k^1\right), \tag{19}$$

$$\hat{\mathbf{P}}_k^1 = \check{\mathbf{P}}_k^1 - \mathbf{K}_k^1 \boldsymbol{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}} - \mathbf{K}_k^{1,2} \left(\left(\check{\mathbf{P}}_{k,k}^{1,2}\right)^{\mathsf{T}} - \mathbf{K}_k^2 \boldsymbol{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}} \right), \tag{20}$$

$$\hat{\mathbf{P}}_{k,k}^{1,2} = \check{\mathbf{P}}_{k,k}^{1,2} - \mathbf{K}_k^1 \boldsymbol{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}}, \tag{21}$$

where $\check{\mathbf{y}}_k^1 = \mathbf{g}^1(\check{\mathbf{x}}_k^1, \hat{\mathbf{x}}_k^2, \mathbf{0})$ is the predicted measurement, and

$$\mathbf{K}_k^1 = \boldsymbol{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \boldsymbol{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1}, \qquad \mathbf{K}_k^2 = \boldsymbol{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \boldsymbol{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1}, \tag{22}$$

$$\mathbf{K}_k^{1,2} = \left(\check{\mathbf{P}}_{k,k}^{1,2} - \mathbf{K}_k^1 \boldsymbol{\Sigma}_{\mathbf{x}_k^2 \mathbf{y}_k^1}^{\mathsf{T}}\right) \left(\hat{\mathbf{P}}_k^2 - \mathbf{K}_k^2 \boldsymbol{\Sigma}_{\mathbf{x}_k^2 \mathbf{y}_k^1}^{\mathsf{T}}\right)^{-1}. \tag{23}$$

The form of the update equation for the cross-covariance matrix $\hat{\mathbf{P}}_{k,k}^{1,2}$ is from the joint distribution of $\mathbf{x}^1$ and $\mathbf{x}^2$, given in the second step of Appendix A. As with the prediction step, naive cascaded filtering techniques neglect the cross-covariance components $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}$ and $\check{\mathbf{P}}_{k,k}^{1,2}$, which appear in (19)–(23). Note that (19) and (20) are similar to the standard Kalman filter, except that this new form of the receiving filter also corrects the feeding filter's state $\mathbf{x}^2$ locally before correcting the state $\mathbf{x}^1$. The updated state of the feeding filter is never communicated back to the feeding filter, resulting in a loss of performance as compared to a full estimator.

IV. Proposed Cascaded Receiving Filter

The proposed framework requires passing Gaussian distributions through nonlinear process and measurement models, which is a well-documented problem with a plethora of possible solutions. In this letter, the sigma point transformation is used as it is a concise, general approach that is suitable for handling nonlinearities. Another possible solution is a linearization-based one, which is further discussed in an extended version of this letter [17, Appendix C].

In the prediction step, $\hat{\mathbf{P}}_{k-1,k-1}^{1,2}$ is used to generate sigma points, which are then propagated forward in time to approximate $\check{\mathbf{P}}_{k,k-1}^{1,2}$. The exact change in the cross-covariance matrix going from $\check{\mathbf{P}}_{k,k-1}^{1,2}$ to $\check{\mathbf{P}}_{k,k}^{1,2}$ is derived in Section IV-C, and an approximation is proposed. In the correction step, $\check{\mathbf{P}}_{k,k}^{1,2}$ is then used to generate sigma points, which allows computing $\hat{\mathbf{P}}_{k,k}^{1,2}$.

A. Prediction Step

For prediction, the goal is to propagate the estimate $\hat{\mathbf{x}}_{k-1}^1$ forward in time to get a predicted state $\check{\mathbf{x}}_k^1$. To do so, the process model (10) and the output of the feeding estimator, $\hat{\mathbf{x}}_{k-1}^2$ and $\hat{\mathbf{P}}_{k-1}^2$, are used. Initially, $\hat{\mathbf{P}}_{0,0}^{1,2}$ is set to be the zero matrix, unless there is prior knowledge of a correlation between the initial estimates of the local estimators.

The state vectors $\mathbf{x}_{k-1}^1$ and $\mathbf{x}_{k-1}^2$ and the process noise $\mathbf{w}_{k-1}^1$ are augmented into a vector $\mathbf{v}$, with

$$\hat{\mathbf{v}}_{k-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1}^1 \\ \hat{\mathbf{x}}_{k-1}^2 \\ \mathbf{0} \end{bmatrix}, \tag{24}$$

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

$$\hat{\mathbf{P}}_{\mathbf{v}_{k-1}} = \begin{bmatrix} \hat{\mathbf{P}}^1_{k-1} & \hat{\mathbf{P}}^{1,2}_{k-1,k-1} & \mathbf{0} \\ \left(\hat{\mathbf{P}}^{1,2}_{k-1,k-1}\right)^{\mathsf{T}} & \hat{\mathbf{P}}^2_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}^1_{k-1} \end{bmatrix}. \tag{25}$$

Let $L \triangleq \dim(\hat{\mathbf{v}}_{k-1})$, and $\mathbf{L}\mathbf{L}^{\mathsf{T}} \triangleq \hat{\mathbf{P}}_{\mathbf{v}_{k-1}}$. Using the spherical cubature rule [13], define the $2L$ sigma points $\mathbf{s}$ to be

$$\mathbf{s}^i_{k-1} \triangleq \hat{\mathbf{v}}_{k-1} + \sqrt{L}\operatorname{col}_i \mathbf{L}, \quad \mathbf{s}^{i+L}_{k-1} \triangleq \hat{\mathbf{v}}_{k-1} - \sqrt{L}\operatorname{col}_i \mathbf{L}.$$

By unstacking each sigma point into

$$\mathbf{s}^i_{k-1} = \left[\left(\hat{\mathbf{x}}^1_{k-1,i}\right)^{\mathsf{T}} \; \left(\hat{\mathbf{x}}^2_{k-1,i}\right)^{\mathsf{T}} \; \left(\mathbf{w}^1_{k-1,i}\right)^{\mathsf{T}}\right]^{\mathsf{T}}, \tag{26}$$

the sigma points are propagated through (10),

$$\check{\mathbf{x}}^1_{k,i} = \mathbf{f}^1\left(\hat{\mathbf{x}}^1_{k-1,i}, \hat{\mathbf{x}}^2_{k-1,i}, \mathbf{w}^1_{k-1,i}\right). \tag{27}$$

Hence, the statistics of the propagated sigma points are

$$\check{\mathbf{x}}^1_k = \frac{1}{2L}\sum_{i=1}^{2L} \check{\mathbf{x}}^1_{k,i}, \quad \check{\mathbf{P}}^1_k = \frac{1}{2L}\sum_{i=1}^{2L}\left(\check{\mathbf{x}}^1_{k,i} - \check{\mathbf{x}}^1_k\right)\left(\check{\mathbf{x}}^1_{k,i} - \check{\mathbf{x}}^1_k\right)^{\mathsf{T}},$$

$$\check{\mathbf{P}}^{1,2}_{k,k-1} = \frac{1}{2L}\sum_{i=1}^{2L}\left(\check{\mathbf{x}}^1_{k,i} - \check{\mathbf{x}}^1_k\right)\left(\hat{\mathbf{x}}^2_{k-1,i} - \hat{\mathbf{x}}^2_{k-1}\right)^{\mathsf{T}}. \tag{28}$$

B. Correction Step

For the correction step, the goal is to correct the predicted state $\check{\mathbf{x}}^1_k$ using the measurement $\mathbf{y}^1_k$, to obtain $\hat{\mathbf{x}}^1_k$. As $\mathbf{y}^1_k$ is a function of $\mathbf{x}^2_k$, the output of the feeding estimator $\hat{\mathbf{x}}^2_k$ and $\hat{\mathbf{P}}^2_k$ is required. As before, the state vectors $\mathbf{x}^1_k$ and $\mathbf{x}^2_k$ and the measurement noise $\boldsymbol{\nu}^1_k$ are augmented into a vector $\mathbf{u}$, with

$$\hat{\mathbf{u}}_k = \begin{bmatrix} \check{\mathbf{x}}^1_k \\ \hat{\mathbf{x}}^2_k \\ \mathbf{0} \end{bmatrix}, \quad \hat{\mathbf{P}}_{\mathbf{u}_k} = \begin{bmatrix} \check{\mathbf{P}}^1_k & \check{\mathbf{P}}^{1,2}_{k,k} & \mathbf{0} \\ \left(\check{\mathbf{P}}^{1,2}_{k,k}\right)^{\mathsf{T}} & \hat{\mathbf{P}}^2_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^1_k \end{bmatrix}. \tag{29}$$

The problem of finding $\check{\mathbf{P}}^{1,2}_{k,k}$ in (29) from (28) is addressed in Section IV-C. Let $M \triangleq \dim(\hat{\mathbf{u}}_k)$, and $\mathbf{M}\mathbf{M}^{\mathsf{T}} \triangleq \hat{\mathbf{P}}_{\mathbf{u}_k}$. Consequently, using the spherical cubature rule, define the $2M$ sigma points $\mathbf{p}$ to be

$$\mathbf{p}^i_k \triangleq \hat{\mathbf{u}}_k + \sqrt{M}\operatorname{col}_i \mathbf{M}, \quad \mathbf{p}^{i+L}_k \triangleq \hat{\mathbf{u}}_k - \sqrt{M}\operatorname{col}_i \mathbf{M}.$$

By unstacking each sigma point into

$$\mathbf{p}^i_k = \left[\left(\check{\mathbf{x}}^1_{k,i}\right)^{\mathsf{T}} \; \left(\hat{\mathbf{x}}^2_{k,i}\right)^{\mathsf{T}} \; \left(\boldsymbol{\nu}^1_{k,i}\right)^{\mathsf{T}}\right]^{\mathsf{T}}, \tag{30}$$

the nonlinear transformation of the sigma points as per the measurement model is

$$\check{\mathbf{y}}^1_{k,i} = \mathbf{g}^1\left(\check{\mathbf{x}}^1_{k,i}, \hat{\mathbf{x}}^2_{k,i}, \boldsymbol{\nu}^1_{k,i}\right). \tag{31}$$

Hence, the statistics of the propagated sigma points are

$$\check{\mathbf{y}}^1_k = \frac{1}{2M}\sum_{i=1}^{2M} \check{\mathbf{y}}^1_{k,i}, \tag{32}$$

$$\boldsymbol{\Sigma}_{\check{\mathbf{x}}^1_k\check{\mathbf{y}}^1_k} = \frac{1}{2M}\sum_{i=1}^{2M}\left(\check{\mathbf{x}}^1_{k,i} - \check{\mathbf{x}}^1_k\right)\left(\check{\mathbf{y}}^1_{k,i} - \check{\mathbf{y}}^1_k\right)^{\mathsf{T}}, \tag{33}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{x}}^2_k\check{\mathbf{y}}^1_k} = \frac{1}{2M}\sum_{i=1}^{2M}\left(\hat{\mathbf{x}}^2_{k,i} - \hat{\mathbf{x}}^2_k\right)\left(\check{\mathbf{y}}^1_{k,i} - \check{\mathbf{y}}^1_k\right)^{\mathsf{T}}, \tag{34}$$

$$\boldsymbol{\Sigma}_{\check{\mathbf{y}}^1_k\check{\mathbf{y}}^1_k} = \frac{1}{2M}\sum_{i=1}^{2M}\left(\check{\mathbf{y}}^1_{k,i} - \check{\mathbf{y}}^1_k\right)\left(\check{\mathbf{y}}^1_{k,i} - \check{\mathbf{y}}^1_k\right)^{\mathsf{T}}. \tag{35}$$

The filter equations given in (19)–(23) can then be used to obtain the approximated distribution of the corrected state $\hat{\mathbf{x}}^1_k$ and its cross-covariance matrix with the state estimate $\hat{\mathbf{x}}^2_k$.

C. Approximating the Effect of the Feeding Filter on the Cross-Covariance

The cross-covariance matrix $\check{\mathbf{P}}^{1,2}_{k,k-1}$ is approximated using sigma points in the prediction step. However, $\check{\mathbf{P}}^{1,2}_{k,k}$ is needed to generate the sigma points for the correction step. If knowledge of the process and measurement models of the feeding filter is available, $\check{\mathbf{P}}^{1,2}_{k,k-1}$ can be propagated to $\check{\mathbf{P}}^{1,2}_{k,k}$ analytically. To elucidate this, consider the linear system

$$\mathbf{x}^1_k = \mathbf{A}^1_{k-1}\mathbf{x}^1_{k-1} + \mathbf{B}^1_{k-1}\mathbf{x}^2_{k-1} + \mathbf{w}^1_{k-1}, \tag{36}$$

$$\mathbf{x}^2_k = \mathbf{A}^2_{k-1}\mathbf{x}^2_{k-1} + \mathbf{w}^2_{k-1}, \tag{37}$$

$$\mathbf{y}^1_k = \mathbf{C}^1_k\mathbf{x}^1_k + \mathbf{D}^1_{k-1}\mathbf{x}^2_k + \boldsymbol{\nu}^1_k, \quad \mathbf{y}^2_k = \mathbf{C}^2_k\mathbf{x}^2_k + \boldsymbol{\nu}^2_k, \tag{38}$$

where the state vectors and noise parameters follow the same notation as in (10)–(13). Moreover, consider for the moment that each of the process and measurement models in (36)–(38) are known. Let $\mathbf{G}^2_k \in \mathbb{R}^{n_2 \times m_2}$ denote the Kalman gain of the estimator of $\mathbf{x}^2_k$ as per Section II-B, then

$$\begin{aligned} \check{\mathbf{P}}^{1,2}_{k,k} &= \mathbb{E}\left[\left(\mathbf{x}^1_k - \check{\mathbf{x}}^1_k\right)\left(\mathbf{x}^2_k - \hat{\mathbf{x}}^2_k\right)^{\mathsf{T}} \,\middle|\, \mathcal{I}^1_{k-1}, \mathcal{I}^2_k\right] \\ &= \mathbb{E}\Big[\left(\mathbf{x}^1_k - \check{\mathbf{x}}^1_k\right)\big(\mathbf{A}^2_{k-1}\mathbf{x}^2_{k-1} + \mathbf{w}^2_{k-1} - \mathbf{A}^2_{k-1}\hat{\mathbf{x}}^2_{k-1} \\ &\qquad - \mathbf{G}^2_k\mathbf{C}^2_k\left(\mathbf{A}^2_{k-1}\mathbf{x}^2_{k-1} + \mathbf{w}^2_{k-1}\right) - \mathbf{G}^2_k\boldsymbol{\nu}^2_k \\ &\qquad + \mathbf{G}^2_k\mathbf{C}^2_k\mathbf{A}^2_{k-1}\hat{\mathbf{x}}^2_{k-1}\big)^{\mathsf{T}} \,\Big|\, \mathcal{I}^1_{k-1}, \mathcal{I}^2_k\Big] \\ &= \mathbb{E}\left[\left(\mathbf{x}^1_k - \check{\mathbf{x}}^1_k\right)\left(\mathbf{x}^2_{k-1} - \hat{\mathbf{x}}^2_{k-1}\right)^{\mathsf{T}} \,\middle|\, \mathcal{I}^1_{k-1}, \mathcal{I}^2_k\right] \\ &\qquad \times \left(\mathbf{A}^2_{k-1} - \mathbf{G}^2_k\mathbf{C}^2_k\mathbf{A}^2_{k-1}\right)^{\mathsf{T}} \\ &= \check{\mathbf{P}}^{1,2}_{k,k-1}\boldsymbol{\Psi}_k, \end{aligned} \tag{39}$$

where the assumptions $\mathbb{E}[(\mathbf{x}^1_k - \check{\mathbf{x}}^1_k)(\mathbf{w}^2_{k-1})^{\mathsf{T}} |\, \mathcal{I}^1_{k-1}, \mathcal{I}^2_k] = \mathbf{0}$ and $\mathbb{E}[(\mathbf{x}^1_k - \check{\mathbf{x}}^1_k)(\boldsymbol{\nu}^2_k)^{\mathsf{T}} |\, \mathcal{I}^1_{k-1}, \mathcal{I}^2_k] = \mathbf{0}$ have been used, and $\boldsymbol{\Psi}_k \triangleq ((\mathbf{1} - \mathbf{G}^2_k\mathbf{C}^2_k)\mathbf{A}^2_{k-1})^{\mathsf{T}}$. Since, however, the estimator of $\mathbf{x}^1$ does not have access to the process and measurement models of the feeding filter, computing (39) exactly is not possible. If cooperation from the feeding filter is at all possible, the most accurate solution is for the feeding filter to share the matrix $\boldsymbol{\Psi}_k$ at every time-step, in addition to the estimated distribution of $\mathbf{x}^2$. When this is not possible, however, an alternative is to approximate $\boldsymbol{\Psi}_k$. Using (9), $\boldsymbol{\Psi}_k$ can be rewritten as

$$\boldsymbol{\Psi}_k = \left(\hat{\mathbf{P}}^2_k\left(\check{\mathbf{P}}^2_k\right)^{-1}\mathbf{A}^2_{k-1}\right)^{\mathsf{T}}. \tag{40}$$

Once the feeding filter reaches a steady state, $\hat{\mathbf{P}}^2_k(\check{\mathbf{P}}^2_k)^{-1} \approx \mathbf{1}$, yielding the approximation $\hat{\boldsymbol{\Psi}}_k = (\mathbf{A}^2_{k-1})^{\mathsf{T}}$. Hence, with knowledge of what $\hat{\boldsymbol{\Psi}}$ should be if access to the process model (11) is available, the user can form an educated guess of the Jacobian $\mathbf{A}^2$. Since the state vector $\mathbf{x}^2$ of the feeding filter is known, the user can reconstruct an approximate (i.e., lower fidelity) process model for the states of the feeding filter, and use the Jacobian of

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

this approximate process model as an approximation of the true Jacobian of the process model (11) used in the feeding filter.

One possible issue with over-estimating $\check{\mathbf{P}}^{1,2}_{k,k}$ using this approximation is that $\hat{\mathbf{P}}_{\mathbf{u}_k}$ in (29) can become indefinite. Therefore, at every iteration, the definiteness of $\hat{\mathbf{P}}_{\mathbf{u}_k}$ must be checked, and the cross-covariances are to be deflated using a scalar parameter until the definiteness test passes.

V. Simulation Results

The main criteria for evaluating the proposed cascaded filter is its consistency, which will be done using the normalized estimation error squared (NEES) test on Monte Carlo runs. The NEES test involves computing a chi-squared statistic $\bar{\epsilon}$ using the error trajectory and corresponding covariance of multiple trials. If the statistic is below a certain threshold, the hypothesis that the estimator is consistent cannot be rejected with 95% confidence [18]. The $\pm 3\sigma$ bound test is also considered, and the root-mean-squared-error (RMSE) is computed to evaluate the performance of the estimators.

A. Linear System

To elucidate the benefits of the proposed approach, consider first a toy problem with two discrete-time linear time-invariant processes evolving through

$$x^1_k = x^1_{k-1} - x^2_{k-1} + w^1_{k-1}, \quad w^1_{k-1} \sim \mathcal{N}\left(0, q^1\right), \tag{41}$$

$$x^2_k = x^2_{k-1} + w^2_{k-1}, \quad w^2_{k-1} \sim \mathcal{N}\left(0, q^2\right), \tag{42}$$

where $x^1_k \in \mathbb{R}$ and $x^2_k \in \mathbb{R}$ are distinct states. Additionally, consider two measurements modelled as

$$y^1_k = x^1_k + x^2_k + \nu^1_k, \quad \nu^1_k \sim \mathcal{N}\left(0, r^1\right), \tag{43}$$

$$y^2_k = x^2_k + \nu^2_k, \quad \nu^2_k \sim \mathcal{N}\left(0, r^2\right), \tag{44}$$

where $y^1_k \in \mathbb{R}$ and $y^2_k \in \mathbb{R}$ are distinct measurements.

As before, assume at every time-step k the estimates $\hat{x}^1_{k-1}$, $\hat{x}^2_{k-1}$, $\hat{x}^2_k$, the covariances $\hat{P}^1_{k-1}$, $\hat{P}^2_{k-1}$, $\hat{P}^2_k$, and the cross-covariance $\hat{P}^{1,2}_{k-1,k-1}$ are known, and the goal is to find the estimate $\hat{x}^1_k$ and its corresponding covariance $\hat{P}^1_k$ using the known values, the process model (41), and the measurement model (43). Using (42), Ψ is set to equal 1, with the definiteness check in place as per Section IV-C.

Deriving the update equations for x^1 yields

$$\check{x}^1_k = \hat{x}^1_{k-1} - \hat{x}^2_{k-1}, \quad \hat{x}^1_k = \check{x}^1_k + K(y^1_k - \check{y}^1_k),$$

$$\check{P}^1_k = \hat{P}^1_{k-1} + 2\hat{P}^{1,2}_{k-1,k-1} + \hat{P}^2_{k-1} + q^1,$$

$$\hat{P}^1_k = (1-K)^2\check{P}^1_k - 2(1-K)K\check{P}^{1,2}_{k,k} + K^2(\hat{P}^2_k + r^1),$$

where the optimal gain K is

$$K = \left(\check{P}^1_k + \check{P}^{1,2}_{k,k}\right)\left(\check{P}^1_k + 2\check{P}^{1,2}_{k,k} + \hat{P}^2_k + r^1\right)^{-1}. \tag{45}$$

The proposed cascaded filter approximates the cross-covariance terms $\hat{P}^{1,2}_{k-1,k-1}$ and $\check{P}^{1,2}_{k,k}$, and is evaluated against the naive approach in [11], to reiterate the importance of modelling cross-covariances.

To evaluate consistency, 1000 Monte Carlo trials with varying initial conditions and noise realizations are performed. A naive estimator that assumes the estimate $\hat{x}^2_k$ is independent from the estimate $\hat{x}^1_k$ thinks it has access to more information than it actually does, resulting in an overconfident estimator. Therefore, the naive estimator fails both the $\pm 3\sigma$ test shown in Fig. 2 and the NEES test in Fig. 3, while the proposed estimator that accommodates for cross-covariances passes both consistency checks.

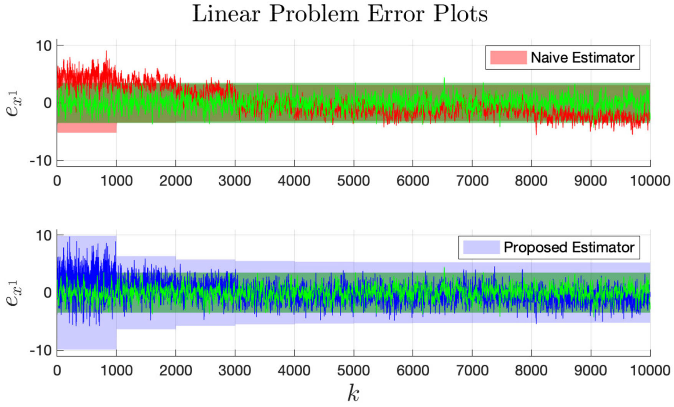

Fig. 2. Single run of the naive and proposed estimators on the linear system. Shaded regions correspond to $\pm 3\sigma$ bounds. Red, blue, and green correspond to naive, proposed, and full estimators, respectively. After $k = 1000$, the $\pm 3\sigma$ bound of the naive and full estimator almost fully overlap.

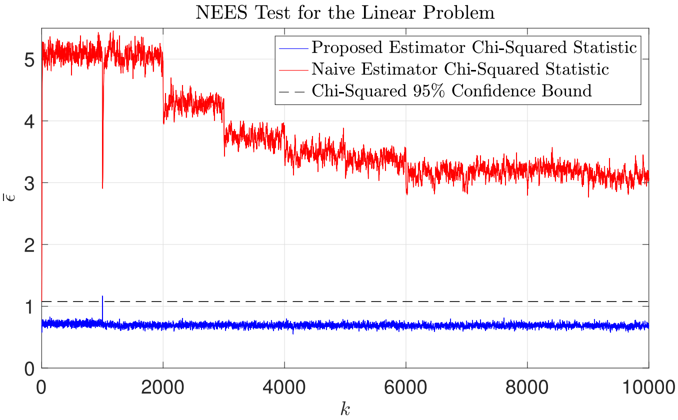

Fig. 3. The NEES test for 1000 Monte Carlo trials on the linear system, showing the consistency evaluation of the cascaded estimators.

The full estimator is clearly the best performer as it can perfectly calculate the cross-covariances and update all states using all measurements. The average RMSE of the proposed framework is 45% worse than the full estimator, while the naive estimator is 90% worse. Therefore, this highlights the importance of addressing cross-covariances to prevent the reuse of old information in the filter.

B. Nonlinear System

Consider a rigid body navigating 3D space with an onboard inertial measurement unit (IMU) and an ultra-wideband (UWB) tag, where the position of the UWB tag p relative to the IMU z resolved in the body frame $\mathcal{F}_b$ is known, and is denoted $\mathbf{r}^{pz}_b \in \mathbb{R}^3$. Additionally, let there be a $\rho \in \mathbb{N}_{>3}$ number of UWB anchors scattered within ranging distance of the UWB tag, thus providing a noisy position measurement of $\mathbf{r}^{p_k w}_a \in \mathbb{R}^3$, which is the position of the UWB tag relative to some arbitrary point w at time-step k, in the absolute frame $\mathcal{F}_a$. The simulation parameters of the system are shown in Table I, and the set-up in Fig. 4.

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SIMULATION PARAMETERS FOR THE NONLINEAR SYSTEM

	Specification	Value
Std. Deviation	Accelerometer (m/s^2)	0.10
	Gyroscope (rad/s)	0.0032
	Magnetometer (μF)	2.00
	Position measurement (m)	0.22
	Initial position (m)	0.45
	Initial velocity (m/s)	0.45
	Initial attitude (rad)	0.22
Rate	IMU (Hz)	100
	Position measurement (Hz)	50

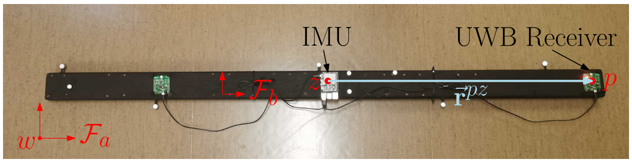

Fig. 4. The prototype used in simulation and for collection of experimental data. The UWB receiver is used to obtain position measurements from a system of UWB anchors.

Using the gyroscope, the magnetometer, and the accelerometer, an AHRS is designed using the invariant extended Kalman filter (IEKF) [19] framework, where accelerometer aiding follows a thresholding rule similar to the one considered in [20]. The output of the AHRS is a direction cosine matrix (DCM) estimate $\hat{\mathbf{C}}_{ab} \in SO(3)$ that gives the relation $\mathbf{r}_a = \hat{\mathbf{C}}_{ab}\mathbf{r}_{\hat{b}}$, where $\mathbf{r}_a$ and $\mathbf{r}_{\hat{b}}$ are the same arbitrary vector resolved in $\mathcal{F}_a$, the absolute frame, and $\mathcal{F}_{\hat{b}}$, the estimated body frame, respectively. The corresponding covariance matrix $\hat{\mathbf{P}}^{\text{AHRS}} \in \mathbb{R}^{3\times 3}$ is also made available by the AHRS.

The position estimator has access to the position measurement $\mathbf{y} \in \mathbb{R}^3$ and the accelerometer measurement $\mathbf{u}_b \in \mathbb{R}^3$ resolved in the body frame $\mathcal{F}_b$. The state vector of the position estimator is

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{r}_a^{zw}(t) \\ \mathbf{v}_a^{zw}(t) \end{bmatrix}, \tag{46}$$

where $\mathbf{r}_a^{zw}(t) \in \mathbb{R}^3$ is the position of the IMU relative to the arbitrary point w resolved in $\mathcal{F}_a$, and $\mathbf{v}_a^{zw}(t) \in \mathbb{R}^3$ is the velocity of the IMU relative to the point w with respect to $\mathcal{F}_a$, resolved in $\mathcal{F}_a$. The corresponding process model is

$$\dot{\mathbf{r}}_a^{zw}(t) = \mathbf{v}_a^{zw}(t), \tag{47}$$

$$\dot{\mathbf{v}}_a^{zw}(t) = \mathbf{C}_{ab}(t)\left(\mathbf{u}_b(t) - \mathbf{w}_b(t)\right) + \mathbf{g}_a, \tag{48}$$

where $\mathbf{w}_b \in \mathbb{R}^3$ denotes white Gaussian process noise, and $\mathbf{g}_a \in \mathbb{R}^3$ is the gravitational acceleration vector resolved in $\mathcal{F}_a$. The discrete-time measurement model is

$$\mathbf{y}_k = \mathbf{r}_a^{z_k w} + \mathbf{C}_{ab_k}\mathbf{r}_b^{p_k z} + \boldsymbol{\nu}_{a_k}, \tag{49}$$

where $\boldsymbol{\nu}_a \in \mathbb{R}^3$ denotes white Gaussian measurement noise. This set-up has inputs in both feeding and receiving filters.

Note that due to the presence of a moment-arm between the IMU and the position sensor, and since the accelerometer measurements are obtained in the body frame $\mathcal{F}_b$, cross-covariances develop between the states of the AHRS and the position estimator. Through knowledge of (47)–(49), $\mathbf{r}_b^{pz}$, and the output of the AHRS, the goal is to design a consistent position estimator. Four approaches are considered. They are

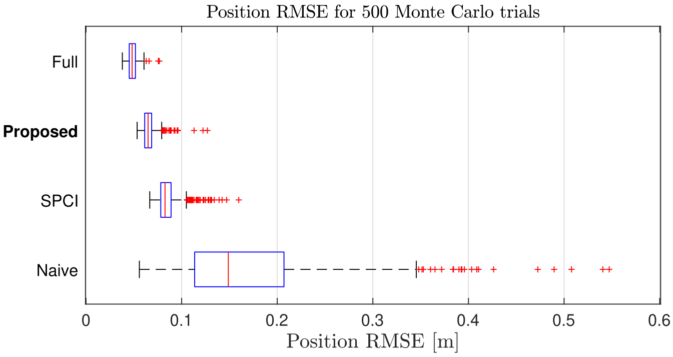

Fig. 5. A box plot showing the median RMSE, outliers, and variation of different estimators over 500 Monte Carlo trials.

TABLE II
RMSE OF THE ESTIMATORS AVERAGED 500 TRIALS

	Average RMSE	% Diff. to Full
Full - Position (m)	0.0487	-
Proposed - Position (m)	**0.0662**	**35.9%**
SPCI - Position (m)	0.0862	77.0%
Naive - Position (m)	0.1733	256%
Full - Attitude (rad)	0.0190	-
AHRS - Attitude (rad)	0.0306	61.1%

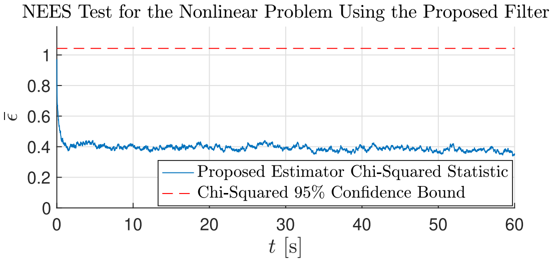

Fig. 6. The NEES test results for the 500 Monte Carlo trials, showing the consistency of the proposed estimator.

1) an AHRS and a **naive** sigma point Kalman filter, where cross-covariances are neglected,
2) an AHRS and the **proposed** sigma point-based filter, where $\hat{\boldsymbol{\Psi}}$ is the identity matrix as per Section IV-C,
3) an AHRS and a sigma point-based Covariance Intersection (**SPCI**) filter, and
4) a **full** sigma point Kalman filter that augments the state vector $\mathbf{x}(t)$ with the attitude states $\mathbf{C}_{ab}(t)$.

The SPCI filter used is a sigma point extension of [16], and is designed in a similar way to the proposed framework. The main difference is that instead of approximating the cross-covariances to compute (25) and (29), the estimated covariance matrices $\check{\mathbf{P}}^1$, $\hat{\mathbf{P}}^1$, and $\hat{\mathbf{P}}^2$ are inflated as per the CI approach, and the cross-covariances are assumed to be zero. This involves tuning an additional scalar parameter. More details on this approach are given in [17, Appendix B].

For the full sigma point Kalman filter, the geodesic L_2-mean discussed in [21] is used for computing means and covariances on $SO(3)$, which are required for sigma point transformations with attitude states. The results of the full sigma point Kalman filter are used as the baseline best possible performance, but in

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

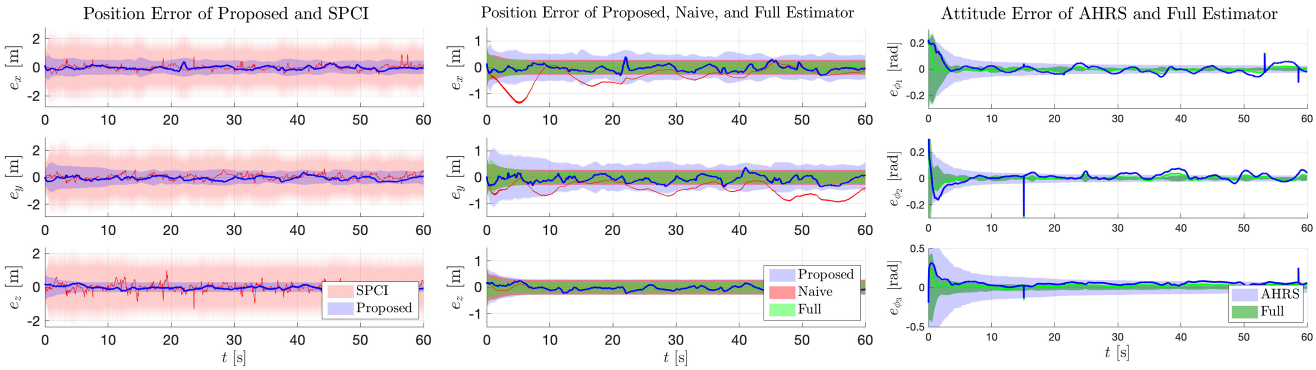

Fig. 7. The error trajectories associated with the slow-pace experimental run for the 4 position estimators, and the AHRS. The AHRS is shared by all 3 cascaded estimators. The shaded regions correspond to the $\pm 3\sigma$ bounds, and the colour of each error trajectory and covariance region are the same.

practice would be computationally heavy, inflexible, and even impossible for larger systems.

The computational overhead of the proposed framework is not significantly different from a naive SPKF or the SPCI filter, as all the considered cascaded estimators use sigma point transformations and a similar set of equations. Additionally, a more computationally efficient linearization-based approach is possible as shown in [17, Appendix C].

To evaluate the 4 estimators, 500 Monte Carlo trials are performed, each 60 seconds long with varying initial conditions and noise realizations. A summary of the position RMSEs is given in Fig. 5 and Table II. The proposed estimator clearly outperforms both the SPCI and naive estimators, even though all 3 estimators share the same AHRS. As expected, the proposed cascaded filter also passes the consistency test, as shown in Fig. 6.

The proposed framework provides a 35.9% worse position estimate than the full estimator even though the provided attitude estimates are 61.1% worse. The worse attitude estimates are due to the UWB measurements correcting the attitude states in the full estimator, but not in cascaded architectures as discussed in Section III. However, the loss of performance is compensated by the modularity, computational gain, and flexibility of the proposed cascaded filter.

TABLE III
RMSE of the Different Estimators on 3 Experimental Runs

	RMSE		
	Slow	Medium	Fast
Full - Position (m)	0.2187	0.19613	0.17518
Proposed - Position (m)	**0.23138**	**0.24818**	**0.28261**
SPCI - Position (m)	0.26690	0.32152	0.32045
Naive - Position (m)	0.69616	0.50807	1.3655
Full - Attitude (rad)	0.04400	0.03225	0.02896
AHRS - Attitude (rad)	0.08934	0.07669	0.13757

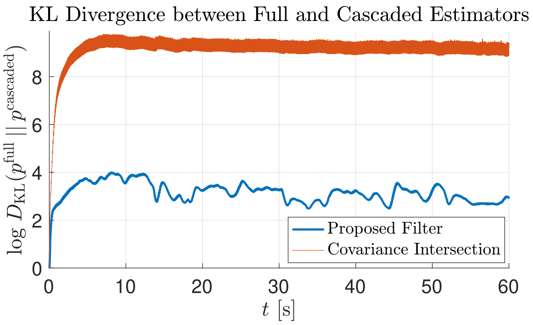

Fig. 8. The logarithm of the KL divergence measure associated with the slow-pace experimental run. The KL divergence is computed between the estimated distribution of each cascaded estimator and the full estimator.

VI. Experimental Results

Experimental data is collected for the nonlinear system discussed in Section V-B using the prototype shown in Fig. 4. The IMU data is collected using a Raspberry Pi Sense HAT at 240 Hz, and the position measurements are collected at 16 Hz using the Pozyx Creator Kit, which is a UWB-based positioning system. Five UWB anchors communicate with a UWB tag placed on the body 84 cm away from the IMU. This is complemented with ground truth data collected using an OptiTrack optical motion capture system at 120 Hz.

Three datasets are tested using the four estimators discussed in Section V-B. Each run involves moving the rigid body randomly in a volume of approximately 5 m × 4 m × 2 m while recording the IMU, UWB, and ground truth data. The main difference between the three datasets is the pace at which the robot is moved around and rotated.

The $\pm 3\sigma$ bound error plots for the 4 estimators on the slow pace experiment are shown in Fig. 7, and an RMSE-based comparison on the three datasets is shown in Table III. As expected, the naive cascaded filter performs poorly, while the performance of the proposed cascaded filter beats the SPCI approach and is comparable to the full estimator. Not only does the proposed framework achieve a lower position RMSE, but it is also less conservative as compared to the SPCI. This is further displayed in Fig. 8, where the KL divergence [22, Chapter 9] measure shows that the estimated distribution of the proposed estimator is closer to that of the full estimator, which is the best available estimate of the true distribution.

Due to imperfectly calibrated hardware, further unmodelled phenomena such as indoor magnetic perturbations, and nonlinearities of the system, the AHRS in the slow-pace experiment takes over 15 seconds to achieve near steady-state conditions, as shown in Fig. 7. Additionally, the AHRS performs poorly particularly in the fast pace experiment, as shown in Table III. However, the proposed framework still achieves a performance comparable to the full estimator, which shows that the approximation

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.

discussed in Section IV-C is reasonable, even for such complex systems. Additionally, the worse attitude estimates as compared to the simulation runs means that the cross-covariances between the AHRS and the position estimator are more significant, which is why the naive estimator performs so poorly. This emphasizes the significance of the proposed estimator, particularly when estimation error is persistent in the feeding filter.

VII. Conclusion

In this letter, the importance of modelling cross-covariances in cascaded estimation is addressed, and a novel approach is introduced using a sigma point transformation. The main contribution is the probabilistic approximation of the propagation of cross-covariance terms when the local filters do not estimate a common set of states. The proposed approach is compared to the naive filtering approach and a Covariance Intersection filter approach both in simulation and in experiment. Superior, consistent results are achieved for both a linear and a nonlinear problem. Comparable performance to a full non-cascaded estimator is also achieved, but with all the advantages of cascaded filtering such as improved flexibility and reduced computational complexity.

Appendix

The conditional joint distribution of $\mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{y}_k^1 \big| \mathcal{I}_{k-1}^1, \mathcal{I}_k^2$ is assumed to be Gaussian according to

$$\mathcal{N}\left(\begin{bmatrix} \check{\mathbf{x}}_k^1 \\ \hat{\mathbf{x}}_k^2 \\ \check{\mathbf{y}}_k^1 \end{bmatrix}, \begin{bmatrix} \check{\mathbf{P}}_k^1 & \check{\mathbf{P}}_{k,k}^{1,2} & \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \\ \left(\check{\mathbf{P}}_{k,k}^{1,2}\right)^{\mathsf{T}} & \hat{\mathbf{P}}_k^2 & \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \\ \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}} & \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}} & \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1} \end{bmatrix}\right).$$

Using [13, Lemma A.2] to condition on $\mathbf{y}_k^1$ as well, the conditional joint distribution of $\mathbf{x}_k^1, \mathbf{x}_k^2 \big| \mathcal{I}_k^1, \mathcal{I}_k^2$ is given by

$$\mathcal{N}\left(\begin{bmatrix} \check{\mathbf{x}}_k^1 + \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \left(\mathbf{y}_k^1 - \check{\mathbf{y}}_k^1\right) \\ \hat{\mathbf{x}}_k^2 + \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \left(\mathbf{y}_k^1 - \check{\mathbf{y}}_k^1\right) \end{bmatrix},\right.$$

$$\left.\begin{bmatrix} \check{\mathbf{P}}_k^1 - \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}} & \check{\mathbf{P}}_{k,k}^{1,2} - \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}} \\ \left(\check{\mathbf{P}}_{k,k}^{1,2}\right)^{\mathsf{T}} - \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}} & \hat{\mathbf{P}}_k^2 - \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}} \end{bmatrix}\right).$$

Using [13, Lemma A.2] again, and by replacing the conditioning on $\mathcal{I}_k^2$ using $\hat{\mathbf{x}}_k^2(\mathcal{I}_k^2)$ and $\hat{\mathbf{P}}_k^2(\mathcal{I}_k^2)$ as was done in Section III, the distribution of $\mathbf{x}_k^1 | \mathcal{I}_k^1, \hat{\mathbf{x}}_k^2, \hat{\mathbf{P}}_k^2$ is given by

$$\begin{aligned} \mathcal{N}\Big(&\check{\mathbf{x}}_k^1 + \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \left(\mathbf{y}_k^1 - \check{\mathbf{y}}_k^1\right) \\ &+ \left(\check{\mathbf{P}}_{k,k}^{1,2} - \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}}\right) \\ &\left(\hat{\mathbf{P}}_k^2 - \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}}\right)^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \left(\mathbf{y}_k^1 - \check{\mathbf{y}}_k^1\right), \\ &\check{\mathbf{P}}_k^1 - \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}} - \left(\check{\mathbf{P}}_{k,k}^{1,2} - \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}}\right) \\ &\left(\hat{\mathbf{P}}_k^2 - \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1}^{\mathsf{T}}\right)^{-1} \\ &\left(\left(\check{\mathbf{P}}_{k,k}^{1,2}\right)^{\mathsf{T}} - \mathbf{\Sigma}_{\hat{\mathbf{x}}_k^2 \check{\mathbf{y}}_k^1} \mathbf{\Sigma}_{\check{\mathbf{y}}_k^1 \check{\mathbf{y}}_k^1}^{-1} \mathbf{\Sigma}_{\check{\mathbf{x}}_k^1 \check{\mathbf{y}}_k^1}^{\mathsf{T}}\right)\Big). \end{aligned}$$

Therefore, the filter equations using a Bayesian approach are given by (19)–(23).

Acknowledgment

The authors would like to thank Natalia Pavlasek for her role in collecting the experimental data.

References

[1] P. O. Arambel, C. Rago, and R. K. Mehra, "Covariance intersection algorithm for distributed spacecraft state estimation," in *Proc. Amer. Control Conf.*, 2001, pp. 4398–4403.

[2] D. M. Bevly and B. Parkinson, "Cascaded Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles," *IEEE Trans. Control Syst. Tech.*, vol. 15, no. 2, pp. 199–208, Mar. 2007.

[3] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *Proc. IEEE Int. Conf. Int. Robots Syst.*, 2013, pp. 1412–1417.

[4] H. Mokhtarzadeh and D. Gebre-Egziabher, "Cooperative inertial navigation," *J. Inst. Navigation*, vol. 61, no. 2, pp. 77–94, 2014.

[5] S. Zihajehzadeh, D. Loh, T. J. Lee, R. Hoskinson, and E. J. Park, "A cascaded kalman filter-based GPS/MEMS-IMU integration for sports applications," *J. Int. Meas. Confed*, vol. 73, pp. 200–210, 2015.

[6] J. W. Song and C. G. Park, "Enhanced pedestrian navigation based on course angle error estimation using cascaded Kalman filters," *Sensors (Switzerland)*, vol. 18, no. 4, 2018, Art. no. 1281.

[7] N. A. Carlson, "Federated square root filter for decentralized parallel processes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 3, pp. 517–525, May 1990.

[8] N. A. Carlson and M. P. Berarducci, "Federated Kalman filter simulation results," *Navigation*, vol. 41, no. 3, pp. 297–322, 1994.

[9] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," *Proc. Amer. Control Conf.*, vol. 4, pp. 2369–2373, 1997.

[10] G. Gobbini, "Relative navigation by means of passive ranging," Ph.D. dissertation, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Aug. 2005.

[11] Z. Lendek, R. Babuška, and B. De Schutter, "Distributed Kalman filtering for cascaded systems," *Eng. Appl. AI*, vol. 21, no. 3, pp. 457–469, 2008.

[12] J. Ali and J. Fang, "Multisensor data synthesis using federated form of unscented Kalman filtering," in *Proc. IEEE Int. Conf. Ind. Tech.*, 2005, pp. 524–529.

[13] S. Sarkka, *Bayesian Filtering and Smoothing*. New York, NY, USA: Cambridge Univ. Press, 2013.

[14] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck, "Optimal sample-based fusion for distributed state estimation," in *Proc. 19th Int. Conf. Inf. Fusion*, 2016, pp. 1600–1607.

[15] M. Reinhardt, B. Noack, and U. D. Hanebeck, "Covariance intersection in state estimation of dynamical systems," in *Proc. Int. Conf. Inf. Fusion*, 2014, pp. 1–7.

[16] J. Zhu and S. S. Kia, "Cooperative localization under limited connectivity," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1523–1530, Dec. 2019.

[17] M. Shalaby, C. C. Cossette, J. Le Ny, and J. R. Forbes, "Cascaded filtering using the sigma point transformation (*extended version*)," 2021, *arXiv: 2103.04249*.

[18] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley Sons, Inc., 2002.

[19] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.

[20] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. New York, NY, USA: McGraw-Hill Educ., 2008.

[21] Y. Dai, J. Trumpf, H. Li, N. Barnes, and R. Hartley, "Rotation averaging with application to camera-rig calibration," *Lecture Notes Comp. Sci.*, vol. 5995*LNCS*, no. PART 2, pp. 335–346, 2010.

[22] S. Kullback, *Information Theory and Statistics*. Gloucester, MA, USA: Dover Punlications, Inc., 1968.

Authorized licensed use limited to: McGill Libraries. Downloaded on September 14,2024 at 02:08:17 UTC from IEEE Xplore. Restrictions apply.