Journal of Guidance, Control, and Dynamics
Vol. 40, No. 9, September 2017

Partial-Update Schmidt–Kalman Filter

Kevin M. Brink*
U.S. Air Force Research Laboratory, Eglin Air Force Base, Florida 32542

DOI: 10.2514/1.G002808

The Schmidt–Kalman (or "consider" Kalman filter) has often been used to account for the uncertainty in so-called "nuisance" parameters when they are impactful to filter accuracy and consistency. Usually such nuisance parameters are errors in environment or sensor models or other static biases where actively estimating their value is not required. However, there are times that it is desired or necessary to estimate the nuisance terms themselves. This paper introduces an intermittent form of the Schmidt–Kalman filter, where (within the same filter) nuisance terms are sometimes treated as full filter states and estimated, and other times they are only considered. Similarly, more generic partial-update forms of the Schmidt–Kalman filter are introduced, where only a portion of the traditional full filter update is applied to select states. These modifications extend the Schmidt filter concept for use on problematic static biases and even time-varying states, allowing them to be estimated while still maintaining filter consistency in cases where extended Kalman filter implementations do not. The new filters are shown to be unbiased and consistent analytically, and they are demonstrated in simulation on a classic one-dimensional system and a six-degree-of-freedom inertial measurement unit–camera calibration example.

I. Introduction

THE Kalman filter [1] and subsequent extended Kalman filter (EKF) created an entirely new paradigm in estimation theory, one that has continued to progress for the last 50+ years. There have been a wide range of advancements in Kalman filtering to broaden its applicability and improve the quality of its estimates, from indirect, or error state, implementations [2] to the Schmidt–Kalman filter (SKF) [3], the square-root implementations of both the EKF and Schmidt variations [4], and the unscented and square-root unscented Kalman filter (UKF) [5]. Additional work in interacting multiple models [6], iterated Kalman filters [7], and many more advancements and adaptations have continued to broaden the impact of the original concept over the years.

This paper focuses on the Schmidt–Kalman, or consider Kalman filter, and the innovative way it expanded the envelope of systems that could be well estimated via the Kalman filter. Specifically, the SKF allows for uncertainties in nuisance states, generally static biases, and often referred to as "colored" or time-correlated noise, to be properly accounted for in the filter and specifically in the filter covariance [8]. An excellent discussion of the motivating concepts behind, and benefits of, the SKF can be found in [9], as can relevant derivations and comparisons between several variants of the consider approach. The name Schmidt–Kalman filter, or SKF, will refer to the minimum covariance consider filter from [9], which, up to continuous versus discrete propagation of covariance, is consistent with the same treatment in [8].

The advantage to the SKF (both extended and unscented varieties) is that, in many cases, the nuisance states are only mildly observable, and often estimating them in the traditional sense may cause filter divergence. However, completely ignoring these terms results in an overconfident filter (which also increases the likelihood of eventual divergence). Traditional tuning approaches could be used that account for these nuisance terms in process noise, $\boldsymbol{Q}$, or measurement noise, $\boldsymbol{R}$, but unlike the SKF, this does not properly account for the colored nature of the nuisance states.

The SKF, however, is limited because it only accounts for nuisance parameters, which are constant biases. Also, the filter is unable to take advantage of times when these bias states are, in fact, observable. In this paper, we introduce modifications of the EKF and associated indirect variant, which updates the filter estimates of nuisance parameters when advantageous and leverages the Schmidt approach when observability/information content dictates, effectively switching between full EKF and Schmidt updates within the same filter. This concept is referred to as the intermittent Schmidt–Kalman filter (ISKF).

The paper then investigates a hybrid approach between EKF and SKF, where a partial update is applied to certain states. Here, a partial update refers to only applying a percentage of the update prescribed by the traditional EKF (in the case of the SKF, that percentage is zero). We denote this concept as the partial-update Schmidt–Kalman filter (PSKF).

The PSKF concept is first introduced as a direct extension to the SKF. The consider states from the SKF are all partially updated by the same fixed update weighting, and like the SKF, the traditional "full" filter states are fully updated. We will refer to this specific implementation as the fixed-weight partial-update Schmidt–Kalman filter (FPSKF).

Finally, a fully generalized version of the partial-update Schmidt–Kalman filter concept is introduced (simply referred to as the PSKF) such that the update weight for any state at any update step can be varied based on user/filter needs (note that both the ISKF and FPSKF are implementations of the generalized form). The paper highlights the PSKF's ability to partially update time-varying states (and potential benefits thereof) and not just those traditionally considered in the SKF. These modifications significantly expands the core concept behind the SKF, which is to mitigate the impact of nonlinearities and can be used to improve filter performance for a wide variety of applications.

The rest of the paper is organized into the following sections. Section II provides a description of the example systems and the metrics used throughout the paper. Section III discusses the benefits of the Schmidt approach and compares the nominal performance of the classic EKF and SKF applied to the one-dimensional (1-D) and inertial measurement unit (IMU)–camera calibration examples. Section IV derives the ISKF update approach, including an information metric used to determine if and when to update states. Section V derives the PSKF implementations (both FPSKF and generalized form). Section VI discusses Monte Carlo analyses results for each modification. Section VII discusses use cases and extensions and Sec. VIII provides a conclusion.

II. Example Systems and Evaluation Metrics

This paper employs two example systems to motivate the work and demonstrate the effectiveness of the proposed modifications. The first is the 1-D example from both [9,10] and is a linear system in keeping with the linear system assumptions often employed in

Received 4 February 2017; revision received 18 April 2017; accepted for publication 21 April 2017; published online 18 May 2017.

*Research Engineer, Air Force Research Laboratory, Munitions Directorate. Member AIAA.

Kalman filtering derivations. The second example system is a more complicated, nonlinear example of the IMU–camera calibration filter from [11].

A. One-Dimensional Example

First, we look at the 1-D sample problem used in both [9,10]. The 1-D example consists of a weight in free fall with zero drag and noisy measurements of the object's position; however, the gravity term is not well known and acts as the nuisance parameter/state. The system dynamics are given as

$$z_{k+1} = z_k + v_k(t_{k+1} - t_k) + \frac{1}{2} g_k (t_{k+1} - t_k)^2 \tag{1}$$

$$v_{k+1} = v_k + g_k(t_{k+1} - t_k) \tag{2}$$

$$g_{k+1} = g_k \tag{3}$$

where z is the position, v is the velocity, g is the constant gravity term, t is time, and k is the time step. Following the example in the previous papers, the true initial values for the system are $z_0 = 0.8$ m, $v_0 = 0.3$ m/s, and $g_0 = 9.8$ m/s^2, with initial uncertainties of one unit for each. The time steps are 1 s, and measurements $\tilde{z}$ are available at each time step with an uncertainty of $\boldsymbol{R} = 1$.

B. Inertial Measurement Unit–Camera Calibration Example

The second system used to exercise the modified filters is an IMU–camera calibration example and is a nonlinear system. The measurement model from [11], which was based on [12,13], is reused. An IMU and camera are moved in front of a known target, and the camera measurements are used to determine the system location, IMU–camera lever arms, and attitude offsets. The measurement model used is

$${}^{C}\boldsymbol{p}_{f_i} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \boldsymbol{C}_B^C \left(\boldsymbol{C}_G^B \left({}^{G}\boldsymbol{p}_{fi} - {}^{G}\boldsymbol{p}_{\mathrm{IMU}} \right) - \left({}^{B}\boldsymbol{p}_C \right) \right) \tag{4}$$

The pixel location of the ith known feature is $\boldsymbol{z}_i$, and that feature is located, in the camera frame, at the point ${}^{C}\boldsymbol{p}_{f_i} = [x_i y_i z_i]^T$, where

$$\boldsymbol{z}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = f \begin{bmatrix} x_i/z_i \\ y_i/z_i \end{bmatrix} \tag{5}$$

and f is the camera focal length. Here, $\boldsymbol{C}_a^b$ denotes the rotation from frame a to b, where frames C, B, and G are the camera, body (IMU), and global frames, respectively. The global position of the ith feature is ${}^{G}\boldsymbol{p}_{f_i}$, the global position of the IMU is ${}^{G}\boldsymbol{p}_{\mathrm{IMU}}$, and ${}^{B}\boldsymbol{p}_C$ is the translational offset between the camera and the IMU in the body frame, referred to as the camera lever arm. The basic laboratory setup related to the IMU–camera calibration routine is shown in Fig. 1.

Past experience with EKF implementations of the IMU–camera calibration demonstrated significant sensitivity to these uncertainties, especially in lever arms and IMU–camera attitude offset. In [11], a UKF was used to accommodate these uncertainties but still demonstrated a strong sensitivity to the initial condition, making this system an ideal way to highlight the effects of the proposed modifications on filter consistency.

C. Evaluation Metrics

To analyze the performance of filters in this paper, a Monte Carlo analysis is used. Several runs of each example system with variation on initial conditions, system motion, and system noises are made for each filter. The standard deviation of the estimation errors is then compared to the average covariance/standard deviation modeled in the filter after N runs:

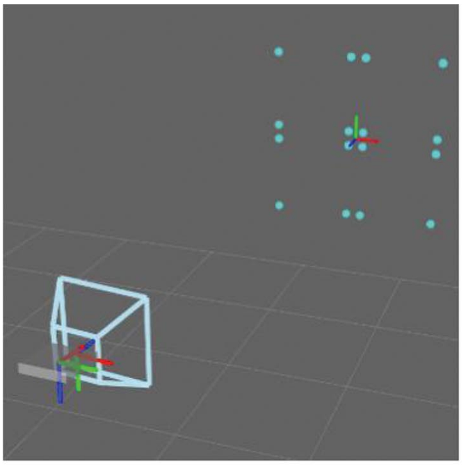
$$\sigma_{i,\mathrm{sampled}} = \mathrm{std}_{\{j=1:N\}}(e_i^j) \tag{6}$$

$$\sigma_{i,\mathrm{ave}} = \frac{1}{N} \sum_{j=1:N} \sqrt{\boldsymbol{P}_{ii}^j} \tag{7}$$

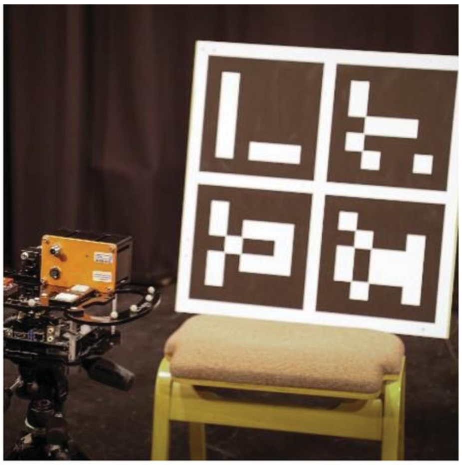
$$e_{i,\mathrm{ave}} = \frac{1}{N} \sum_{j=1:N} e_i^j \tag{8}$$

Here, e_i^j is the error in the ith state of the jth run, and $\sigma_{i,\mathrm{ave}}$ is the average standard deviation derived from the filter covariance (i.e., the average of the square root of the corresponding covariance element from each run). Formally, for a filter to be unbiased and consistent, it requires that $E\{\boldsymbol{e}\} = \boldsymbol{0}$ and $E\{\boldsymbol{e}\boldsymbol{e}^T\} = \boldsymbol{P}$. Therefore, for an unbiased and consistent filter, the values of $\sigma_{i,\mathrm{ave}}$ and $\sigma_{i,\mathrm{sampled}}$ will be equal, suggesting that the errors in a filter are well represented by that particular distribution. Also, the average error $e_{i,\mathrm{ave}}$ should trend to zero as the sampled size increases. These values are displayed within figures through the paper to help visualize filter performance. The normalized estimation error squared (NEES) will also be used to evaluate filter performance in this paper, and a relevant treatment of filter consistency including the NEES metric can be found in [14]. The metric is given as

$$NEES = \frac{1}{N} \sum_{\{j=1:N\}} \boldsymbol{e}^{jT} \boldsymbol{P}^{j-1} \boldsymbol{e}^j \tag{9}$$



a) Simulated IMU-camera calibration with known target



b) IMU-camera hardware with known target

Fig. 1 Flight lab hardware.

Table 1 IMU–camera calibration parameters

State/parameter	Value
IMU position uncertainty	5 cm
IMU attitude uncertainty	0.5 deg
Lever arm uncertainty	5 cm
Camera/IMU attitude uncertainty	0.5 deg
IMU rate	100 Hz
Camera rate	10 Hz
Camera angular accuracy	0.1 deg

where consistent filters will have a NEES value of, or at least near, the number of filter states. For the 1-D example, the ideal NEES will be 3. In the case of the IMU–camera calibration example, a subset of filter states is evaluated. The states included in the NEES calculation are the IMU and camera position and attitude states; therefore, the desired NEES value will be 12.

III. Full Filter Divergence and the Benefits of Schmidt Filtering

The concept behind Schmidt–Kalman filtering is to take systems with nuisance parameters, which cannot be easily or accurately estimated and, instead of neglecting them completely, consider them and their uncertainty when estimating the filter's full states. Here, we look at the performance difference between the EKF and SKF implementations.

For the 1-D example, it is clear that a user cannot safely ignore the error in $\hat{g}$ (this results in an inconstant filter, $\sigma_{\text{sampled}} \gg \sigma_{\text{ave}}$), but both the EKF and SKF implementations are consistent ($\sigma_{\text{sampled}} = \sigma_{\text{ave}}$) with NEES values very near 3. The plots are omitted here because the reader can refer to [9].

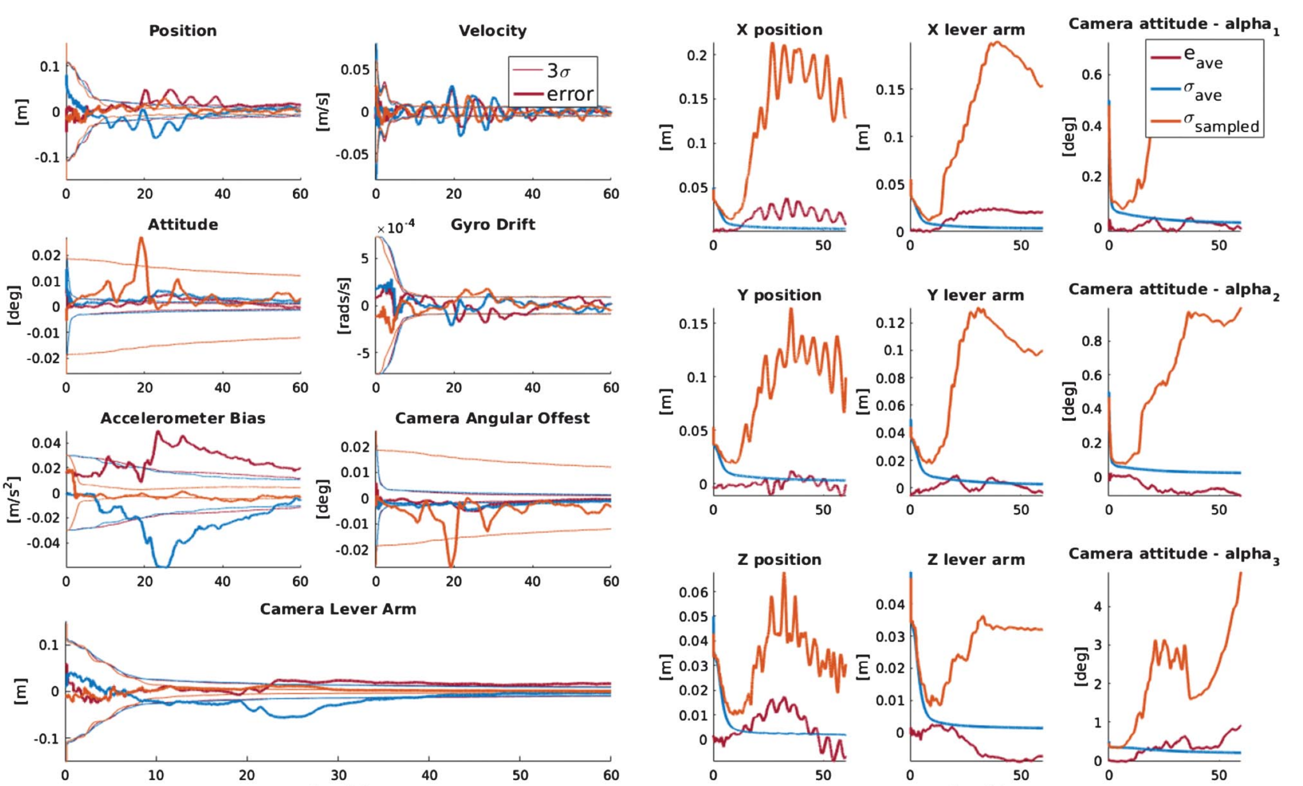
In the second example system, the filter is localizing an IMU using a known target while simultaneously estimating the attitude offset and lever arms between the IMU and camera. Table 1 displays pertinent simulation parameter values, and more detailed descriptions are available in [11]. The EKF would be unable to completely ignore the nuisance states (i.e., treat them as true), as one would expect, but unlike the 1-D case, the full EKF is also not able to effectively estimate these states either.

Results of the full EKF can be seen in Fig. 2, and the single run results do not completely diverge (Fig. 2a) but are well outside of the 3σ bounds at times. If consistent, the Monte Carlo plots in Fig. 2b would show σ_{ave} (blue) and σ_{sampled} (orange) at the same value, and e_{ave} (red) would be near zero. It is obvious that the EKF results are not consistent. To provide a clearer picture of the variation of filter performance across all 100 runs, the x state error for each run in the Monte Carlo set is shown in Fig. 3.

Note that, in Fig. 2a, and in all subsequent "single run" plots, the legend only includes two values: "3σ" and "error." However, there are nine lines plotted in each subfigure; the thick lines are the errors in states (i.e., position, velocity, attitude, etc.), and the thin lines (of matching color) are the associated, estimated 3σ bounds. Additionally, single run plots throughout the paper are the result of the final Monte Carlo run for the given filter and not hand-picked examples.

When the IMU–camera calibration is run as a Schmidt filter, the consistency is dramatically improved compared to the EKF; the results can be seen in Fig. 4. For the SKF, both lever arms and relative attitude states are held constant and only considered, and the result implies that nonlinearities in the system, especially pertaining to the static biases, are causing the full filter to diverge. The error in the x state is shown in Fig. 5, and although not necessarily accurate, the estimates no longer diverge as they did in Fig. 3.

The NEES values associated with the full EKF and SKF runs are listed in Table 2. Recall that the NEES was calculated for the IMU position and attitude and the camera lever arm and attitude, and it has a desired value of 12. The Schmidt approach works, providing more consistent results; however, it does not estimate the nuisance terms. Although the SKF estimates are sound, the system is forced to maintain whatever the initial uncertainty was in the considered states. This dramatically limits the estimation accuracy of the other INS states and the value of the filter; the limitation is clearly seen in Fig. 4, where the IMU position state uncertainty is unable to drop below the associated lever arm uncertainty.



a) Results from a single run b) Results from 100 Monte Carlo runs

Fig. 2 EKF IMU–camera calibration.

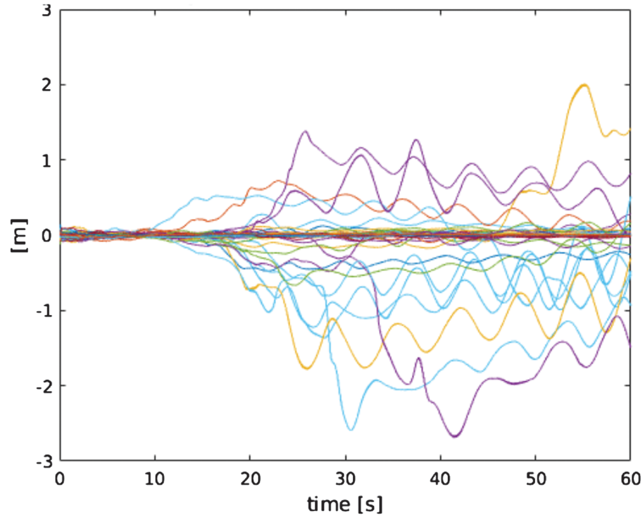


Fig. 3 EKF IMU x state error per run.

IV. Intermittent Schmidt–Kalman Filter

As just seen in the previous section, the same thing that makes the SKF work, namely preventing updates in certain nuisance states, can severely limits its usefulness. The ISKF concept addresses this in a straightforward way, by updating estimates of nuisance states when (functionally) observable and only considering them otherwise. The IMU–camera calibration problem is well suited for the ISKF concept because the IMU–camera lever arms and attitude offsets are static biases that are not always observable to the system; yet, to effectively leverage the camera data, those transforms are needed.

In [11], part of the concept for the ISKF was demonstrated for a multiple-camera system where camera lever arms were initially only considered; then, after a certain point, the lever arms were released and allowed to be updated in a UKF implementation. The paper showed improved consistency when initially considering the lever arm states before allowing estimates to be updated. However, the underlying update approach was never formalized, and the switch from only considering to fully updating the lever arm states was made just once (the filter never switched back to considering the lever arms once they were released for updates) and the timing was user-defined. We now formalize this concept.

A. Intermittent Schmidt–Kalman Filter Update Equations

Similar to [11], the full filter updates (in this case EKF) are applied, and then the nuisance states and their associated covariance values are reset as needed. This method simplifies the system implementation (but does negate any computational savings that the original SKF could provide; a short discussion of computational considerations is included in Sec. VII). If the appropriate states and covariance values

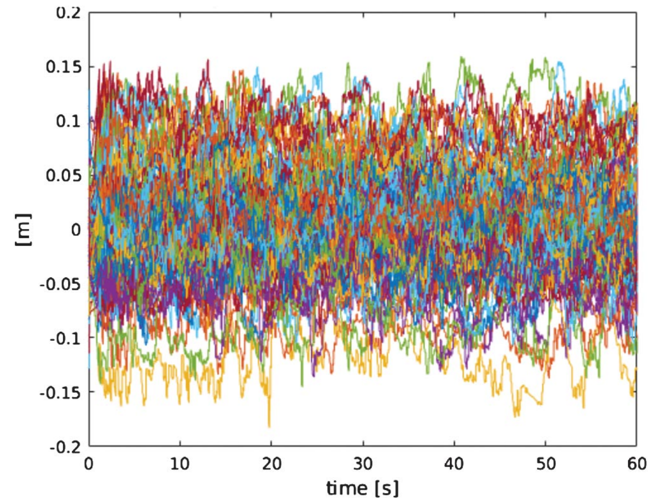
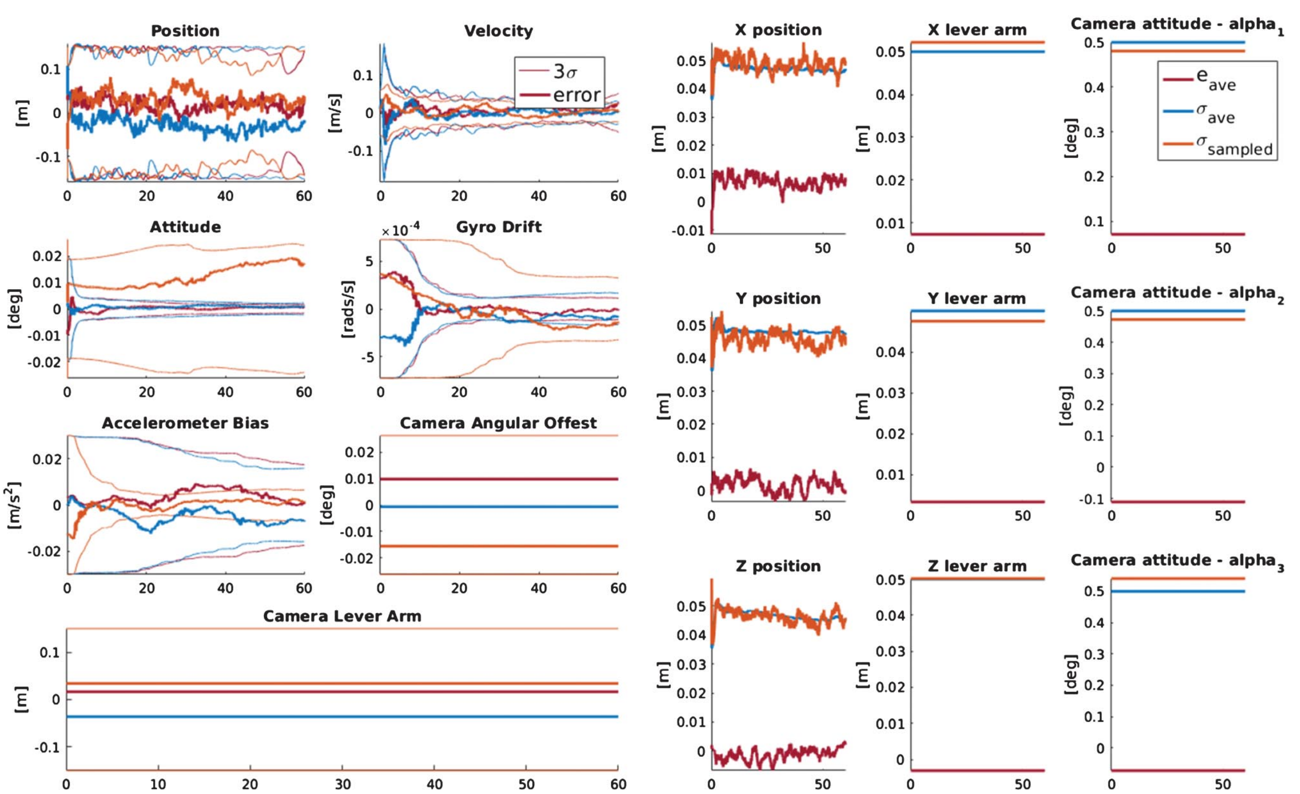


Fig. 5 SKF IMU x state error per run.



a) Results from a single run b) Results from 100 Monte Carlo runs

Fig. 4 SKF IMU–camera calibration.

Table 2 NEES for EKF and SKF

t, s	EKF	SKF
5	101.5	12.1
15	3336.6	12.4
60	$1.6004e+05$	14.1

are always reset, the results are equivalent to the original SKF update [9]. The ISKF update equations are

$$\boldsymbol{K} = \boldsymbol{P}^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}^-\boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{10}$$

$$\hat{\boldsymbol{X}}^+ = \hat{\boldsymbol{X}}^- + \boldsymbol{K}\left(\tilde{z} - \boldsymbol{H}\hat{\boldsymbol{X}}^-\right) \tag{11}$$

$$\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^- \tag{12}$$

$$\cdots\cdots\cdots$$

$$\text{if consider} \begin{cases} \hat{\boldsymbol{y}}^+ = \hat{\boldsymbol{y}}^- \\ \boldsymbol{P}_{yy}^+ = \boldsymbol{P}_{yy}^- \end{cases} \tag{13}$$

where $\boldsymbol{P}$ is the covariance; $\boldsymbol{X}$ is the state vector (see notation comments in the next subsection); $\boldsymbol{H}$ is the Jacobian; $\boldsymbol{R}$ is the measurement covariance; and $\tilde{z}$ is the measurement. The vector $\boldsymbol{y}$ contains the considered states, $\boldsymbol{P}_{yy}$ is the related portion of the covariance, and $[\cdot]^-$ and $[\cdot]^+$ are the propagated and updated values, respectively. Note that $\boldsymbol{y}$ is a subset of $\boldsymbol{X}$, and unlike the traditional SKF, the states accounted for in $\boldsymbol{y}$ may differ from update to update. It is important to note that the final $\hat{\boldsymbol{X}}^+$ and $\boldsymbol{P}^+$ values reflect the resets in Eq. (13).

B. Schmidt Filter Equivalences

The first step in validating the ISKF concept is to show that the ISKF update, when states are considered, is equivalent to the SKF. Note that the standard EKF update equations [Eqs. (10–12)] were shown in the ISKF update. The abusive notation for the state vector $\boldsymbol{X}$ is used for convenience because the SKF breaks the full state vector into states to be estimated, $\boldsymbol{x}$, and states to only be considered, $\boldsymbol{y}$. Similarly, $\boldsymbol{H}$ and $\boldsymbol{P}$ are also broken into estimated and considered portions:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \tag{14}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_x & \boldsymbol{H}_y \end{bmatrix} \tag{15}$$

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{xx} & \boldsymbol{P}_{xy} \\ \boldsymbol{P}_{yx} & \boldsymbol{P}_{xy} \end{bmatrix} \tag{16}$$

The SKF update equations are

$$\boldsymbol{K}_x = \boldsymbol{P}_{xx}^- \boldsymbol{H}_x^T (\boldsymbol{H}_x \boldsymbol{P}_{xx}^- \boldsymbol{H}_x^T + \boldsymbol{H}_x \boldsymbol{P}_{xy}^- \boldsymbol{H}_y^T + \boldsymbol{H}_y \boldsymbol{P}_{yx}^- \boldsymbol{H}_x^T + \boldsymbol{H}_y \boldsymbol{P}_{yy}^- \boldsymbol{H}_y^T + \boldsymbol{R})^{-1} \tag{17}$$

$$\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}_x(\tilde{z} - \boldsymbol{H}_x\hat{\boldsymbol{x}}^- - \boldsymbol{H}_y\hat{\boldsymbol{y}}^-) \tag{18}$$

$$\hat{\boldsymbol{y}}^+ = \hat{\boldsymbol{y}}^- \tag{19}$$

$$\boldsymbol{P}_{xx}^+ = (\boldsymbol{I} - \boldsymbol{K}_x\boldsymbol{H}_x)\boldsymbol{P}_{xx}^- - \boldsymbol{K}_x\boldsymbol{H}_y\boldsymbol{P}_{yx}^- \tag{20}$$

$$\boldsymbol{P}_{xy}^+ = (\boldsymbol{I} - \boldsymbol{K}_x\boldsymbol{H}_x)\boldsymbol{P}_{xy}^- - \boldsymbol{K}_x\boldsymbol{H}_y\boldsymbol{P}_{yy}^- \tag{21}$$

$$\boldsymbol{P}_{yx}^+ = \boldsymbol{P}_{xy}^{+T} \tag{22}$$

$$\boldsymbol{P}_{yy}^+ = \boldsymbol{P}_{yy}^- \tag{23}$$

Here, the equivalence between using the full EKF with a consider state and covariance reset [Eqs. (10–13)] and the classic SKF updates [Eqs. (17–23)] is confirmed.

Lemma 1: Implementing a standard EKF and resetting the considered states $\boldsymbol{y}$ and associated covariance values $\boldsymbol{P}_{yy}$ [Eqs. (10–13)] results in estimates and covariance values equivalent to the SKF implementation in Eqs. (17–23).

Proof: Substituting Eqs. (14–16) into Eqs. (10–12) readily results in

$$\boldsymbol{K} = \begin{bmatrix} (\boldsymbol{P}_{xx}^-\boldsymbol{H}_x^T + \boldsymbol{P}_{xy}^-\boldsymbol{H}_y^T)(\boldsymbol{H}_x\boldsymbol{P}_{xx}^-\boldsymbol{H}_x^T + \boldsymbol{H}_x\boldsymbol{P}_{xy}^-\boldsymbol{H}_y^T + \boldsymbol{H}_y\boldsymbol{P}_{yx}^-\boldsymbol{H}_x^T + \boldsymbol{H}_y\boldsymbol{P}_{yy}^-\boldsymbol{H}_y^T + \boldsymbol{R})^{-1} \\ (\boldsymbol{P}_{yx}^-\boldsymbol{H}_y^T + \boldsymbol{P}_{yy}^-\boldsymbol{H}_y^T)(\boldsymbol{H}_x\boldsymbol{P}_{xx}^-\boldsymbol{H}_x^T + \boldsymbol{H}\boldsymbol{x}_x\boldsymbol{P}_{xy}^-\boldsymbol{H}_y^T + \boldsymbol{H}_y\boldsymbol{P}_{yx}^-\boldsymbol{H}_x^T + \boldsymbol{H}_y\boldsymbol{P}_{yy}^-\boldsymbol{H}_y^T + \boldsymbol{R})^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_x \\ \boldsymbol{K}_y \end{bmatrix} \tag{24}$$

$$\hat{\boldsymbol{X}}^+ = \begin{bmatrix} \hat{\boldsymbol{x}}^- \\ \hat{\boldsymbol{y}}^- \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_x \\ \boldsymbol{K}_y \end{bmatrix}\begin{bmatrix} \tilde{z} - \boldsymbol{H}_x\hat{\boldsymbol{x}}^- - \boldsymbol{H}_y\hat{\boldsymbol{y}}^- \end{bmatrix} \tag{25}$$

$$\begin{aligned} \boldsymbol{P}^+ &= \left(\boldsymbol{I} - \begin{bmatrix} \boldsymbol{K}_x \\ \boldsymbol{K}_y \end{bmatrix}\begin{bmatrix} \boldsymbol{H}_x & \boldsymbol{H}_y \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{P}_{xx}^- & \boldsymbol{P}_{xy}^- \\ \boldsymbol{P}_{yx}^- & \boldsymbol{P}_{yy}^- \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{P}_{xx}^- & \boldsymbol{P}_{xy}^- \\ \boldsymbol{P}_{yx}^- & \boldsymbol{P}_{yy}^- \end{bmatrix} - \begin{bmatrix} [\boldsymbol{K}_x\boldsymbol{H}_x] & [\boldsymbol{K}_x\boldsymbol{H}_y] \\ [\boldsymbol{K}_y\boldsymbol{H}_x] & [\boldsymbol{K}_y\boldsymbol{H}_y] \end{bmatrix}\begin{bmatrix} \boldsymbol{P}_{xx}^- & \boldsymbol{P}_{xy}^- \\ \boldsymbol{P}_{yx}^- & \boldsymbol{P}_{yy}^- \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{P}_{xx}^- & \boldsymbol{P}_{xy}^- \\ \boldsymbol{P}_{yx}^- & \boldsymbol{P}_{yy}^- \end{bmatrix} \\ &\quad - \begin{bmatrix} [\boldsymbol{K}_x\boldsymbol{H}_x\boldsymbol{P}_{xx}^- + \boldsymbol{K}_x\boldsymbol{H}_y\boldsymbol{P}_{yx}^-] & [\boldsymbol{K}_x\boldsymbol{H}_x\boldsymbol{P}_{xy}^- + \boldsymbol{K}_x\boldsymbol{H}_y\boldsymbol{P}_{yy}^-] \\ [\boldsymbol{K}_y\boldsymbol{H}_x\boldsymbol{P}_{xx}^- + \boldsymbol{K}_y\boldsymbol{H}_y\boldsymbol{P}_{yx}^-] & [\boldsymbol{K}_y\boldsymbol{H}_x\boldsymbol{P}_{xy}^- + \boldsymbol{K}_y\boldsymbol{H}_y\boldsymbol{P}_{yy}^-] \end{bmatrix} \end{aligned} \tag{26}$$

The $\boldsymbol{K}_y$ equivalent term from Eq. (24) is then substituted back into Eq. (26) and results in $\boldsymbol{P}_{yx}^+ = \boldsymbol{P}_{xy}^{+T}$. Resetting the values of $\hat{\boldsymbol{y}}^+$ and $\boldsymbol{P}_{yy}^+$ to their prior values, $\hat{\boldsymbol{y}}^-$ and $\boldsymbol{P}_{yy}^-$, results in the updates shown in Eqs. (17–23). □

It should also be noted that, although the update equation for the off-diagonal covariance elements ($\boldsymbol{P}_{xy}$ and $\boldsymbol{P}_{yx}$) is not changed between the EKF, SKF, and ISKF, the Schmidt updates are able to maintain consistency because, even though the covariance term $\mathrm{cov}(x, y)$ remains unchanged, the associated correlation coefficient $\mathrm{cor}(x, y) = [\mathrm{cov}(x, y)/\mathrm{std}(x)\mathrm{std}(y)]$ scales with the standard deviation $\mathrm{std}(\cdot)$ of the states. In other words, although the covariance matches between the full and Schmidt updates (assuming the same prior), the actual resulting correlation is appropriately reduced in the Schmidt case because the filter is inherently ignoring information. (Lemmas that are directly related to the covariance between full and considered states are presented in the Appendices and would provide an alternative means to prove the following theorem.)

Theorem 2: For a linear system where $\hat{\boldsymbol{x}}^- \sim N(\boldsymbol{x}, \boldsymbol{P}^-)$, $\tilde{z} \sim N(z, \boldsymbol{R})$, $\hat{x}^+ = \hat{x}^- + \boldsymbol{K}(\tilde{z} - \boldsymbol{H}\hat{x}^-)$, and $\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^-$, the ISFK update [Eqs. (10–13)] is unbiased and consistent.

Proof: The Kalman filter [1] and the minimum covariance consider filter [9] were derived using minimum variance approaches and shown to be unbiased and consistent estimators for linear system under these assumptions. Therefore the ISKF update, which in Lemma 1 was shown to be the equivalent of either the full Kalman update or the Schmidt update, is also an unbiased and consistent estimator. □

C. Information-Based Updates

We are now able to freely alternate between full filter and Schmidt filter updates using the ISKF update modification. We now provide a means for determining if and when to update versus when to only consider the certain states.

There are numerous heuristics or other approaches that could be conceived based on knowledge of the system and an understanding of its observability to determine if and when to update nuisance states. In the GPS/INS case, when calibrating the antenna lever arm with respect to the IMU, specific motions are induced that are known to make those states observable [15]; a simple heuristic would be to only update during times those motions are happening (or soon after, if the flavor of GPS/INS provides prolonged observability). For the 1-D case, because the example is only intended to help demonstrate the soundness of the ISKF modification, we simply alternate every five updates (with one measurement update per second). However, for the IMU–camera calibration, information metrics are introduced that aid in maintaining consistency of the filter by avoiding problematic updates.

The fundamental question that needs to be addressed within the information metrics is whether or not there is sufficient unique information in a given set of measurements such that the update of the nuisance states would not be fully correlated to the other system states. If there is full, or nearly full correlation, then the system should perform a Schmidt update. For example, if there is not enough information unique to the lever arm, then the filter will benefit from only considering the lever arm, updating the IMU position, and waiting until there is more differentiable information in subsequent updates.

Updating (or not) in this manner captures the majority of the information available in the measurements without allowing the system to update both components (IMU position and lever arm) when only the sum of the components is observable, similarly for the attitude states.

To assess the quality and uniqueness of the information available at an update step, a proxy for the information content is used:

$$\boldsymbol{I}_{\text{EKF}} = [\boldsymbol{P}^-]^{-1}\boldsymbol{K}\boldsymbol{R}^{-1}\boldsymbol{K}^T\Big[[\boldsymbol{P}^-]^{-1}\Big]^T \tag{27}$$

This matrix facilitates the evaluation of the uncertainty-weighted (in both covariance and measurement uncertainty) gains assigned to each state based on the current measurements available, current filter states, and covariance. In the ISKF, when a measurement update is available, the values of $\boldsymbol{I}_{\text{EKF}}$ are evaluated. For each new filter application, the specifics for evaluating $\boldsymbol{I}_{\text{EKF}}$ will differ.

Here, we discuss an application of the information metric concept for the IMU–camera calibration case. The two state types we consider are camera lever arm and IMU–camera attitude offsets; indices $\alpha_{\text{imu}}^{1,2,3}$ are defined that represent the entries of $\boldsymbol{I}_{\text{EKF}}$ associated with the IMU attitude states, and the indices $\alpha_{\text{cam}}^{1,2,3}$ are related to the camera attitude states. The indices $p_{\text{imu}}^{1,2,3}$ and $p_{\text{cam}}^{1,2,3}$ are similarly defined for the IMU position states and lever arm states, respectively. The IMU-related entries of $\boldsymbol{I}_{\text{EKF}}$ are mapped into the body frame with the current best estimate of the global to body rotation, $\hat{\boldsymbol{C}}_G^B$. This allows the metric to look at a one-to-one relation between a nuisance state and a core INS state:

$$\bar{\boldsymbol{I}}_{\text{EKF}}\Big(p_{\text{cam}}^{1,2,3}, p_{\text{cam}}^{1,2,3}\Big) = \boldsymbol{I}_{\text{EKF}}\Big(p_{\text{cam}}^{1,2,3}, p_{\text{cam}}^{1,2,3}\Big) \tag{28}$$

$$\bar{\boldsymbol{I}}_{\text{EKF}}\Big(p_{\text{imu}}^{1,2,3}, p_{\text{imu}}^{1,2,3}\Big) = \hat{\boldsymbol{C}}_G^B\boldsymbol{I}_{\text{EKF}}\Big(p_{\text{imu}}^{1,2,3}, p_{\text{imu}}^{1,2,3}\Big) \tag{29}$$

$$\bar{\boldsymbol{I}}_{\text{EKF}}\Big(p_{\text{imu}}^{1,2,3}, p_{\text{cam}}^{1,2,3}\Big) = \hat{\boldsymbol{C}}_G^B\boldsymbol{I}_{\text{EKF}}\Big(p_{\text{imu}}^{1,2,3}, p_{\text{cam}}^{1,2,3}\Big) \tag{30}$$

$$\bar{\boldsymbol{I}}_{\text{EKF}}\Big(p_{\text{cam}}^{1,2,3}, p_{\text{imu}}^{1,2,3}\Big) = \hat{\boldsymbol{C}}_G^B\boldsymbol{I}_{\text{EKF}}\Big(p_{\text{cam}}^{1,2,3}, p_{\text{imu}}^{1,2,3}\Big) \tag{31}$$

The update metrics used in the IMU–camera ISKF example are

$$\begin{cases} \text{if } \dfrac{\boldsymbol{I}_{\text{EKF}}(\alpha_{\text{imu}}^j \alpha_{\text{cam}}^j)}{\sigma_{\alpha_{\text{imu}}^j}\sigma_{\alpha_{\text{cam}}^j}} < T_\alpha, & \text{then update} \\ \text{otherwise consider} \end{cases} \tag{32}$$

and

$$\begin{cases} \text{if } \dfrac{\bar{\boldsymbol{I}}_{\text{EKF}}(p_{\text{imu}}^j p_{\text{cam}}^j)}{\sigma_{p_{\text{imu}}^j}\sigma_{p_{\text{cam}}^j}} < T_p, & \text{then update} \\ \text{otherwise consider} \end{cases} \tag{33}$$

where the σ terms are the square root of the corresponding diagonal element of $\bar{\boldsymbol{I}}_{\text{EKF}}$, and T values are user-defined thresholds. Equations (32) and (33) are used with the ISKF updates [Eqs. (10–13)] for the IMU–camera calibration examples in Sec. VI.

V. Partial-Update Schmidt–Kalman Filter

With the ISKF update established and producing a "bang–bang" switch between full EKF and SKF updates, we turn our attention to applying partial updates of "considered" states. The ISKF tries to prevent any update that would cause divergence in the system (at least with targeted heuristics or an information-based approach for update decisions). Here, the PSKF takes a slightly different approach and limits updates (i.e., only a percentage of the EKF-prescribed update is applied), with the hope that the cumulative effect of the partial updates is beneficial as long as the least accurate updates are not applied at full strength.

The PSKF is introduced in a fixed-weight form (which is denoted as FPSKF) and has close ties to the traditional Schmidt approach; there are full and considered states, where considered states now receive a constant and equally weighted partial update at each update step, and full states receive the standard full EKF update (and if partial update weights are set to zero, this results in the SKF).

Afterward, the generalized form of PSKF is introduced, which allows for any update weighting of any filter states at each update step. This is a significant departure from the traditional Schmidt approach but allows for the "Schmidt treatment" of both traditional nuisance terms and problematic time-varying states, expanding the range of systems, or uncertainty levels, for which the EKF-based filters remain effective.

A. Fixed-Weight Partial-Update Schmidt–Kalman Filter Update Equations

To incorporate a partial update of considered states, the FPSKF update equations are

$$K = P^- H^T (H P^- H^T + R)^{-1} \tag{34}$$

$$\hat{X}^+ = \hat{X}^- + K\left(\tilde{z} - H\hat{X}^-\right) \tag{35}$$

$$P^+ = (I - KH)P^- \tag{36}$$

$$\cdots\cdots\cdots$$

$$\hat{y}^{++} = \gamma \hat{y}^- + (1-\gamma)\hat{y}^+ \tag{37}$$

$$P_{yy}^{++} = \gamma^2 P_{yy}^- + (1-\gamma^2) P_{yy}^+ \tag{38}$$

where $\hat{y}^+$ and P_{yy}^+ are subsets of $\hat{X}^+$ and P^+, respectively. The $[\cdot]^{++}$ notation denotes the FPSKF update values that are used to overwrite their respective full state estimates and covariance values. They, along with the unmodified state and covariance terms, are then used at the next propagation step.

The weighting function $\gamma \in [0,1]$ is responsible for setting the update somewhere between the full EKF update and the SKF update. At the extremes, $\gamma = 0$, we have a full update, and at $\gamma = 1$, we have the Schmidt update. For analysis, it is favorable to have $(1-\gamma)$ be the weight associated with the standard full EKF update; however, intuitively, it is beneficial to think of the update weight in terms of how much of the full update is applied as opposed to how much is neglected. Therefore, we define the update parameter $\beta \in [0,1]$, where β is the percentage of the update applied to the state of interest; now, γ can be defined by the more intuitive β term as $\gamma = 1 - \beta$. Additionally, instead of setting the weighting term with respect to the state update, it can also be thought of in terms of the desired percentage of covariance update, and $(1-\gamma^2)$ is then used to solve for the required γ term.

We now verify that the FPSKF modification is probabilistically sound; however, several lemmas are first required.

Lemma 3: For a linear system where $\hat{x}^- \sim N(x, P^-)$, $\tilde{z} \sim N(z, R)$, $\hat{x}^+ = \hat{x}^- + K(\tilde{z} - H\hat{x}^-)$, and $P^+ = (I - KH)P^-$, then $\mathrm{cov}(\hat{x}_i^+, \hat{x}_j^+) = \mathrm{cov}(\hat{x}_i^+, \hat{x}_j^-)$.

Proof: From Theorem 2, we know that the ISKF updates are consistent; additionally, the cross-correlation terms in the ISKF update equations [Eqs. (17–23)] do not differ for two fully updated states compared to one fully updated state and one consider state. Now, suppose that $\mathrm{cov}(\hat{x}_i^+, \hat{x}_j^+) \neq \mathrm{cov}(\hat{x}_i^+, \hat{x}_j^-)$; because the choice of which states to include in $\hat{y}$ is (theoretically) arbitrary, this implies that either the covariance update for two fully updated states (EKF update) or the covariance update for a full state and a consider state (SKF update), or both, will not be consistent. This contradicts Theorem 2; therefore, $\mathrm{cov}(\hat{x}_i^+, \hat{x}_j^+) = \mathrm{cov}(\hat{x}_i^+, \hat{x}_j^-)$. □

This proof relies heavily on the fact that the SKF is consistent under the system assumptions (shown through the minimum variance derivation in [9]); however, the Appendices provide additional lemmas and an alternative proof of Lemma 3, which only rely on the fact that the EKF is a consistent filter (under the same system assumptions) and provide some insight into why Lemma 3 is true.

Lemma 4: For a linear system where $\hat{x}^- \sim N(x, P^-)$, $\tilde{z} \sim N(z, R)$, $\hat{x}^+ = \hat{x}^- + K(\tilde{z} - H\hat{x}^-)$, and $P^+ = (I - KH)P^-$, the correlation between a state's preupdate and postupdate values is $\mathrm{cor}(\hat{x}_i^-, \hat{x}_i^+) = (\sigma_i^+/\sigma_i^-)$, where $\hat{x}_i^-$ and $\hat{x}_i^+$ are the ith element of $\hat{x}^-$ and $\hat{x}^+$, respectively; $\sigma_i^- = \sqrt{P_{ii}^-}$ and $\sigma_i^+ = \sqrt{P_{ii}^+}$ are the associated standard deviations.

Proof: Assuming n states and m measurements, we start with the term of interest and rewrite x_i^+ with respect to its prior and associated update:

$$\mathrm{cov}(\hat{x}_i^-, \hat{x}_i^+) = \mathrm{cov}(\hat{x}_i^-, \hat{x}_i^- + K_{i,1:m}(\tilde{z} - H\hat{x}^-)) \tag{39}$$

$$= \mathrm{cov}(\hat{x}_i^-, \hat{x}_i^-) + \mathrm{cov}(\hat{x}_i^-, K_{i,1:m}(\tilde{z} - H_{i,1:n}\hat{x}^-)) \tag{40}$$

Expanding $K_{i,1:m}(\tilde{z} - H_{i,1:n}\hat{x}^-)$, followed by algebraic manipulation using covariance identities, yields

$$\begin{aligned}\mathrm{cov}(\hat{x}_i^-, \hat{x}_i^+) = {} & \mathrm{cov}(\hat{x}_i^-, \hat{x}_i^-) + \mathrm{cov}\Big(\hat{x}_i^-, K_{i1}(\tilde{z}_1 - (H_{11}\hat{x}_i^- + \ldots + H_{1n}\hat{x}_n^-)) \\ & + \ldots + K_{im}(\tilde{z}_m - (H_{m1}\hat{x}_1^- + \ldots + H_{mn}\hat{x}_n^-))\Big)\end{aligned} \tag{41}$$

$$\begin{aligned}= {} & \mathrm{cov}(\hat{x}_i^-, \hat{x}_i^-) + K_{i1}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_1 - H_{11}\hat{x}_1^- - \ldots - H_{1n}\hat{x}_n^-) + \ldots \\ & + K_{im}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_m - H_{m1}\hat{x}_1^- \ldots - H_{mn}\hat{x}_n^-)\end{aligned} \tag{42}$$

$$\begin{aligned}= {} & \mathrm{cov}(\hat{x}_i^-, \hat{x}_i^-) + K_{i1}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_1) - K_{i1}\mathrm{cov}(\hat{x}_i^-, H_{11}\hat{x}_1^- + \ldots + H_{1n}\hat{x}_n^-) \\ & + \ldots + K_{im}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_m) - K_{im}\mathrm{cov}(\hat{x}_i^-, H_{11}\hat{x}_1^- + \ldots + H_{1n}\hat{x}_n^-)\end{aligned} \tag{43}$$

Recognizing, from the linear system assumptions, that $\mathrm{cov}(\hat{x}_i^-, \tilde{z}) = 0$, additional algebraic manipulation results in

$$\begin{aligned}\mathrm{cov}(\hat{x}_i^-, \hat{x}_i^+) = {} & \mathrm{cov}(\hat{x}_i^-, \hat{x}_i^-) - K_{i1}H_{11}\mathrm{cov}(\hat{x}_i^-, \hat{x}_1^-) - \ldots \\ & - K_{i1}H_{1n}\mathrm{cov}(\hat{x}_i^-, \hat{x}_n^-) - \ldots - K_{im}H_{11}\mathrm{cov}(\hat{x}_i^-, \hat{x}_1^-) - \ldots \\ & - K_{im}H_{1n}\mathrm{cov}(\hat{x}_i^-, \hat{x}_n^-)\end{aligned} \tag{44}$$

Recalling that $\mathrm{cov}(\hat{x}_j^-, \hat{x}_k^-) = P_{jk}^-$, terms are regrouped with respect to P^- and the matrix $[KHP^-]$, resulting in

$$\mathrm{cov}(\hat{x}_i^-, \hat{x}_i^+) = P_{ii}^- - [KHP^-]_{ii} \tag{45}$$

which implies

$$\mathrm{cor}(\hat{x}_i^-, \hat{x}_i^+) = \frac{P_{ii}^- - [KHP^-]_{ii}}{\sigma_i^- \sigma_i^+} \tag{46}$$

$$= \frac{P_{ii}^+}{\sigma_i^- \sigma_i^+} \tag{47}$$

$$= \frac{\sigma_i^+}{\sigma_i^-} \tag{48}$$

□

Lemma 5: For a linear system where $\hat{x}^- \sim N(x, P^-)$, $\tilde{z} \sim N(z, R)$, $\hat{x}^+ = \hat{x}^- + K(\tilde{z} - H\hat{x}^-)$, and $P^+ = (I - KH)P^-$, after the FPSKF update is applied [Eqs. (34–38)], the partially updated state $\hat{y}_i^{++}$ will be unbiased and consistent.

Proof: Both $\hat{y}_i^-$ and $\hat{y}_i^+$ are unbiased estimates; therefore, the linear combination will also be unbiased. Additionally, because of the fact that state estimates $\hat{y}_i^-$ and $\hat{y}_i^+$ are related through an EKF update, they are correlated (100% if there was no update, i.e., $K = 0$, or 0% if $KH = I$, but likely somewhere in between), and from Lemma 4, the ratio is known to be $\mathrm{cor}(\hat{y}_i^-, \hat{y}_i^+) = (\sigma_i^+/\sigma_i^-)$, where $\sigma_i^- = \sqrt{P_{yy_{ii}}^-}$ and $\sigma_i^+ = \sqrt{P_{yy_{ii}}^+}$.

The variance of the sum or random variables is $\mathrm{var}(a + b) = \mathrm{var}(a) + \mathrm{var}(b) + \mathrm{cov}(a, b)$, and therefore the variance of the corresponding linear combination of $\hat{y}_i^-$ and $\hat{y}_i^+$ can be written as

$$\begin{aligned}\mathrm{var}(\gamma\hat{y}_i^- + (1-\gamma)\hat{y}_i^+) = {} & \gamma^2(\sigma_i^-)^2 + (1-\gamma)^2(\sigma_i^+)^2 \\ & + 2\mathrm{cov}(\gamma\hat{y}_i^-, (1-\gamma)\hat{y}_i^+)\end{aligned} \tag{49}$$

$$= \gamma^2(\sigma_i^-)^2 + (1-\gamma)^2(\sigma_i^+)^2 + 2\gamma(1-\gamma)\mathrm{cov}(\hat{y}_i^-, \hat{y}_i^+) \tag{50}$$

Because $\mathrm{cov}(a, b) = \mathrm{cor}(a, b)sd(a)sd(a)$, we can rewrite Eq. (50) and apply the results of lemma 4:

$$\operatorname{var}(\gamma\hat{y}_i^- + (1-\gamma)\hat{y}_i^+) = \gamma^2(\sigma_i^-)^2 + (1-\gamma)^2(\sigma_i^+)^2 + 2\gamma(1-\gamma)\operatorname{cor}(\hat{y}_i^-, \hat{y}_i^+)\sigma_i^-\sigma_i^+ \tag{51}$$

$$= \gamma^2(\sigma_i^-)^2 + (1-\gamma)^2(\sigma_i^+)^2 + 2\gamma(1-\gamma)(\sigma_i^+)^2 \tag{52}$$

$$= \gamma^2(\sigma_i^-)^2 + \sigma_2^2 - 2\gamma(\sigma_i^+)^2 + \gamma^2(\sigma_i^+)^2 + 2\gamma(\sigma_i^+)^2 - 2\gamma^2(\sigma_i^+)^2 \tag{53}$$

$$= \gamma^2(\sigma_i^-)^2 + (1-\gamma^2)(\sigma_i^+)^2 \tag{54}$$

□

Theorem 6: For a linear system where $\hat{\boldsymbol{x}}^- \sim \boldsymbol{N}(\boldsymbol{x}, \boldsymbol{P}^-)$, $\tilde{z} \sim \boldsymbol{N}(\boldsymbol{z}, \boldsymbol{R})$, $\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\tilde{z} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)$, and $\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P}^-$, the FPSFK update [Eqs. (34–38)] is unbiased and consistent.

Proof: It is already established that states that are fully updated following the standard EKF update equations are unbiased and consistent among themselves [1], and from Lemma 5, we have that that an individual, partially updated state is unbiased and consistent. What remains to be shown is that partially updated states are consistent with respect to each other and fully updated states.

Starting with partially updated states, we have

$$\operatorname{cov}(\hat{y}_i^{++}, \hat{y}_j^{++}) = \operatorname{cov}((1-\gamma)\hat{y}_i^+ + \gamma\hat{y}_i^-, (1-\gamma)\hat{y}_j^+ + \gamma\hat{y}_j^-) \tag{55}$$

$$= \operatorname{cov}((1-\gamma)\hat{y}_i^+, (1-\gamma)\hat{y}_j^+) + \operatorname{cov}((1-\gamma)\hat{y}_i^+, \gamma\hat{y}_j^-) + \operatorname{cov}(\gamma\hat{y}_i^-, (1-\gamma)\hat{y}_j^+) + \operatorname{cov}(\gamma\hat{y}_i^-, \gamma\hat{y}_j^-) \tag{56}$$

$$= (1-\gamma)^2\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^+) + (1-\gamma)\gamma\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^-) \quad \gamma(1+\gamma)\operatorname{cov}(\hat{y}_i^-, \hat{y}_j^+) + \gamma^2\operatorname{cov}(\hat{y}_i^-, \hat{y}_j^-) \tag{57}$$

From Lemma 3, we can rewrite Eq. (57) with $\operatorname{cov}(\hat{y}_i^-, \hat{y}_j^+)$ and $\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^-)$ replaced by $\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^+)$, and simplify with respect to γ:

$$\operatorname{cov}(\hat{y}_i^{++}, \hat{y}_j^{++}) = (1-\gamma)^2\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^+) + (1-\gamma)\gamma\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^-) + \cdots \quad \gamma(1-\gamma)\operatorname{cov}(\hat{y}_i^-, \hat{y}_j^+) + \gamma^2\operatorname{cov}(\hat{y}_i^-, \hat{y}_j^-) \tag{58}$$

$$= (1-\gamma^2)\operatorname{cov}(\hat{y}_i^+, \hat{y}_j^+) + \gamma^2\operatorname{cov}(\hat{y}_i^-, \hat{y}_j^-) \tag{59}$$

A similar argument is now made for the full and partial state combination:

$$\operatorname{cov}(\hat{y}_i^{++}, \hat{x}_j^{++}) = \operatorname{cov}((1-\gamma)\hat{y}_i^+ + \gamma\hat{y}_i^-, \hat{x}_j^+) \tag{60}$$

$$= \operatorname{cov}((1-\gamma)\hat{y}_i^+, \hat{x}_j^+) + \operatorname{cov}(\gamma\hat{y}_i^-, \hat{x}_j^+) \tag{61}$$

$$= (1-\gamma)\operatorname{cov}(\hat{y}_i^+, \hat{x}_j^+) + \gamma\operatorname{cov}(\hat{y}_i^-, \hat{x}_j^+) \tag{62}$$

Once again, Lemma 3 is leveraged, and we rewrite Eq. (62) replacing $\operatorname{cov}(\hat{y}_i^-, \hat{x}_j^+)$ with $\operatorname{cov}(\hat{y}_i^+, \hat{x}_j^+)$:

$$\operatorname{cov}(\hat{y}_i^{++}, \hat{x}_j^+) = (1-\gamma)\operatorname{cov}(\hat{y}_i^+, \hat{x}_j^+) + \gamma\operatorname{cov}(\hat{y}_i^+, \hat{x}_j^+) \tag{63}$$

$$= \operatorname{cov}(\hat{y}_i^+, \hat{x}_j^+) \tag{64}$$

□

B. Generalized Partial-Update Schmidt–Kalman Filter Update Equations

The FPSKF update equations [Eqs. (34–38)] are now generalized for the case where each state may have its own unique update weight. In this case, $\boldsymbol{\beta} = [\beta_1 \ldots \beta_n]^\top$, with the associated γ vector, $\boldsymbol{\gamma} = [\gamma_1 \ldots \gamma_n]^\top$, corresponding to the full state vector, $\boldsymbol{x} = [x_1 \ldots x_n]^\top$ (the $\boldsymbol{X}$ notation is dropped here because there is no longer a full state/consider state differentiation required because this is now taken into account in $\boldsymbol{\beta}$). Noting that $\boldsymbol{\beta}$ may also be a function of time, the generalized PSKF updates are given as

$$\boldsymbol{K} = \boldsymbol{P}^-\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}^-\boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{65}$$

$$\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\tilde{z} - \boldsymbol{H}\hat{\boldsymbol{x}}^-) \tag{66}$$

$$\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P}^- \tag{67}$$

$$\cdots\cdots\cdots$$

$$\hat{x}_i^{++} = \gamma_i\hat{x}_i^- + (1-\lambda_i)\hat{x}_i^+ \tag{68}$$

$$\boldsymbol{P}_{ij}^{++} = \gamma_i\gamma_j\boldsymbol{P}_{ij}^- + (1-\gamma_i\gamma_j)\boldsymbol{P}_{ij}^- \tag{69}$$

where again the $[\cdot]^{++}$ notation denotes the PSKF update values that are used to overwrite their respective full state estimates and covariance values.

Theorem 7: For a linear system where $\hat{\boldsymbol{x}}^- \sim \boldsymbol{N}(\boldsymbol{x}, \boldsymbol{P}^-)$, $\tilde{z} \sim \boldsymbol{N}(\boldsymbol{z}, \boldsymbol{R})$, $\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\tilde{z} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)$, and $\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P}^-$, the generalized PSFK update [Eqs. (65–69)] is unbiased and consistent.

Proof: It is already shown in Theorem 6 that, for a single nonzero/one γ value, the result remains unbiased and consistent. Therefore, all that must be established is that the covariance between states with arbitrary γ values is consistent. Taking the covariance of two generalized states, we have

$$\operatorname{cov}(\hat{x}_i^{++}, \hat{x}_j^{++}) = \operatorname{cov}(\gamma_i\hat{x}_i^- + (1-\gamma_i)\hat{x}_i^+, \gamma_j\hat{x}_j^- + (1-\gamma_j)\hat{x}_j^+) \tag{70}$$

$$= \operatorname{cov}(\gamma_i\hat{x}_i^-, \gamma_j\hat{x}_j^-) + \operatorname{cov}(\gamma_i\hat{x}_i^-, (1-\gamma_j)\hat{x}_j^+) + \operatorname{cov}((1-\gamma_i)\hat{x}_i^+, \gamma_j\hat{x}_j^-) + \operatorname{cov}((1-\gamma_i)\hat{x}_i^+, (1-\gamma_j)\hat{x}_j^+ \tag{71}$$

Pulling γ terms outside of the covariance operators, applying Lemma 3, and simplifying provides the required result:

$$\operatorname{cov}(\hat{x}_i^{++}, \hat{x}_j^{++}) = \gamma_i\gamma_j\operatorname{cov}(\hat{x}_i^-, \hat{x}_j^-) + \gamma_i(1-\gamma_j)\operatorname{cov}(\hat{x}_i^-, \hat{x}_j^-) + \ldots \quad (1-\gamma_i)\gamma_j\operatorname{cov}(\hat{x}_i^+, \hat{x}_j^-) + (1-\gamma_i)(1-\gamma_j)\operatorname{cov}(\hat{x}_i^+, \hat{x}_j^+) \tag{72}$$

$$= \gamma_i\gamma_j\operatorname{cov}(\hat{x}_i^-, \hat{x}_j^-) + \gamma_i(1-\gamma_j)\operatorname{cov}(\hat{x}_i^+, \hat{x}_j^+) + \ldots \quad (1-\gamma_i)\gamma_j\operatorname{cov}(\hat{x}_i^+, \hat{x}_j^+) + (1-\gamma_i)(1-\gamma_j)\operatorname{cov}(\hat{x}_i^+, \hat{x}_j^+) \tag{73}$$

$$= \gamma_i\gamma_j\operatorname{cov}(\hat{x}_i^-, \hat{x}_j^-) + (1-\gamma_i\gamma_j)\operatorname{cov}(\hat{x}_i^+, \hat{x}_j^+) \tag{74}$$

□

The generalized PSKF framework allows for differing update weights to be applied to any state at any update step while maintaining an unbiased and consistent filter (in the linear case). The values of $\boldsymbol{\beta}$ can easily be set, per update, as 1 or zero resulting in the ISKF, or permanently set at 1 and a single additional value to result in the FPSKF from the previous section (and if that second value is zero, it results in an SKF). Or the update weights can be set based on the unique needs of a particular filter use case.

Recall that the Schmidt filter is effective in some cases where the full EKF diverges because it mitigates the impact of nonlinearities, rendering the system "linear enough," helping to preserve the filter's consistency. This generalized framework provides significantly more flexibility in pursuing the same goal.

Although the PSKF (and ISKF) concept is certainly not optimal (a concept that, in general, is forfeited when working with linearized approximations anyway), it is analytically consistent for linear systems and leverages more of the available information than the SKF otherwise would. Like most EKF applications, we take encouragement from the linear system underpinnings and hope that linearizations and Jacobean evaluations are accurate enough to enable effective filter performance when confronted with nonlinear systems.

VI. Modified Filter Results

The 1-D filter is revisited for both the ISKF and PSKF modifications. The linear example results and associated NEES values support the analysis from earlier sections that the proposed modifications yield an unbiased and consistent estimator. The modifications are then applied to the IMU–camera calibration example, which is used to demonstrate the modifications' significantly improved ability to maintain consistency compared to the full EKF implementation in this challenging scenario.

A. Intermittent Schmidt–Kalman Filter Results

1. One-Dimensional System

The original 1-D example is simulated for 20 s runs to compare the EKF, SKF, and ISKF, and the results can be seen in Fig. 6. The terms σ_{sampled} and σ_{ave} were defined in Eqs. (6) and (7), and position measurements are available every second. The ISKF updates $\hat{g}$ when $t \in [6{:}10, 16{:}20]$ and considers $\hat{g}$ otherwise; this can be seen in Fig. 6b. The performance of the ISKF is not as accurate as the full filter but results in an improvement over the pure Schmidt implementation, as is expected. Most importantly, the ISKF is consistent in this example seen for both position and gravity state estimates and in total NEES (note that position and gravity plots share the same legend).

2. Inertial Measurement Unit–Camera Calibration

The ISKF is now applied to the divergent IMU–camera calibration example. Recall that the full EKF implementation was seen to be inconsistent, and while maintaining consistency, the SKF is unable to improve the estimates of lever arm or camera attitude offset states. The same Monte Carlo analysis that was applied to the EKF and SKF runs is now conducted for the ISKF implementations. Again, the settings from Table 1 are used, and the information metric is applied from Eqs. (32) and (33), where $T_p = 0.9$ and $T_\alpha = 0.9$.

Overall, the ISFK results are dramatic, providing improved consistency while still generating improved estimates of the lever arms attitude offsets. It can be seen in Fig. 7 that one of the camera attitude offset estimates is very restricted in the ISKF, rarely updating and then never updating after about 5 s into the runs. This is due to the information metric not meeting the required threshold. However, improved estimates to the lever arms and the remaining attitude offsets are achieved (and if the goal was to improve IMU position estimation accuracy, the uncertainty was reduced by a factor of 5 compared to the SKF).

The x state errors for each run of the Monte Carlo analysis are shown in Fig. 8 and demonstrate significant reduction in the number and magnitude of outliers compared to the full filter results in Fig. 3. The NEES for the ISKF runs is displayed in Table 3, and it can be seen that the ISKF results in a NEES much closer to the desired value of 12. Overall, the filter performs quite effectively, avoiding the worst of the nonlinearities experienced by the full EKF and providing significant improvement in estimates to full states and of five of the six nuisance states, but the ISKF results are is still (mildly) overconfident.

The frequency of full updates versus consider updates is also shown in Fig. 8, and it is clear that the values of $\boldsymbol{I}_{\text{EKF}}$ and subsequent update rates are highly correlated to the motion of the IMU–camera system (which is sinusoidal in nature). This, combined with the dramatically improved consistency, suggests that the thresholds used do prevent a significant number of inaccurate updates.

Additional Monte Carlo runs with randomized choices of full updates versus consider updates were run, and although omitted here, do improve estimation performance compared to the full EKF, but only nominally, and still displayed several divergent runs. This further suggests that the information metric is effective at identifying if and when to perform updates for this example. Similarly, additional Monte Carlo simulations were run with other values of T_α and T_p. In these cases, larger values of T_α and T_p led to more full EKF updates and reduced filter consistency. Lower threshold values quickly begin to prevent otherwise accurate updates, and results trend toward the SKF results.

B. Fixed-Weight Partial-Update Schmidt–Kalman Filter Results

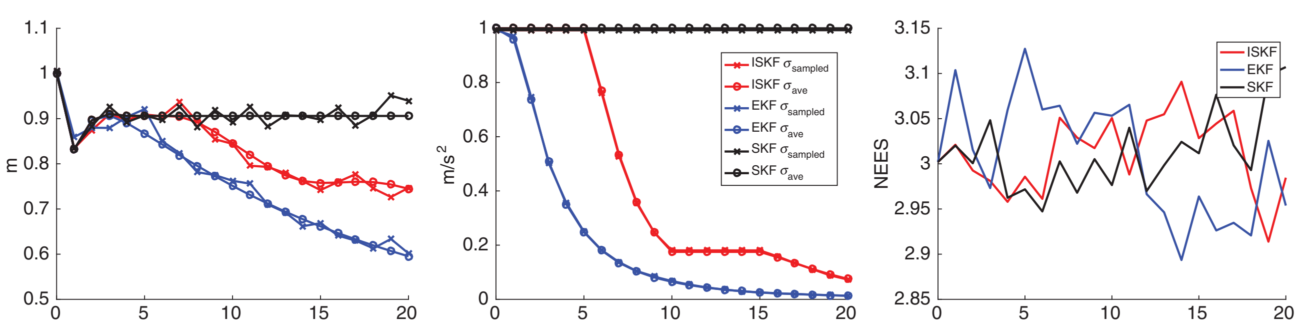
1. One-Dimensional System

Similar to the analysis of the ISKF, we revisit the 1-D case and now apply the FPSKF update equations for update weights of $\beta = 0.1$, 0.25, and then 0.5 (note that this is not the generalized form, and each considered state has the same, fixed update weight for the entire run). The results are displayed in Fig. 9. The terms σ_{sampled} and σ_{ave} were defined in Eqs. (6) and (7). The FPSKF implementations are consistent and maintain a NEES of 3, as expected. It can clearly be seen that, as β is increased, the results shift from the Schmidt results toward full filter results, as expected.

2. Inertial Measurement Unit–Camera Calibration

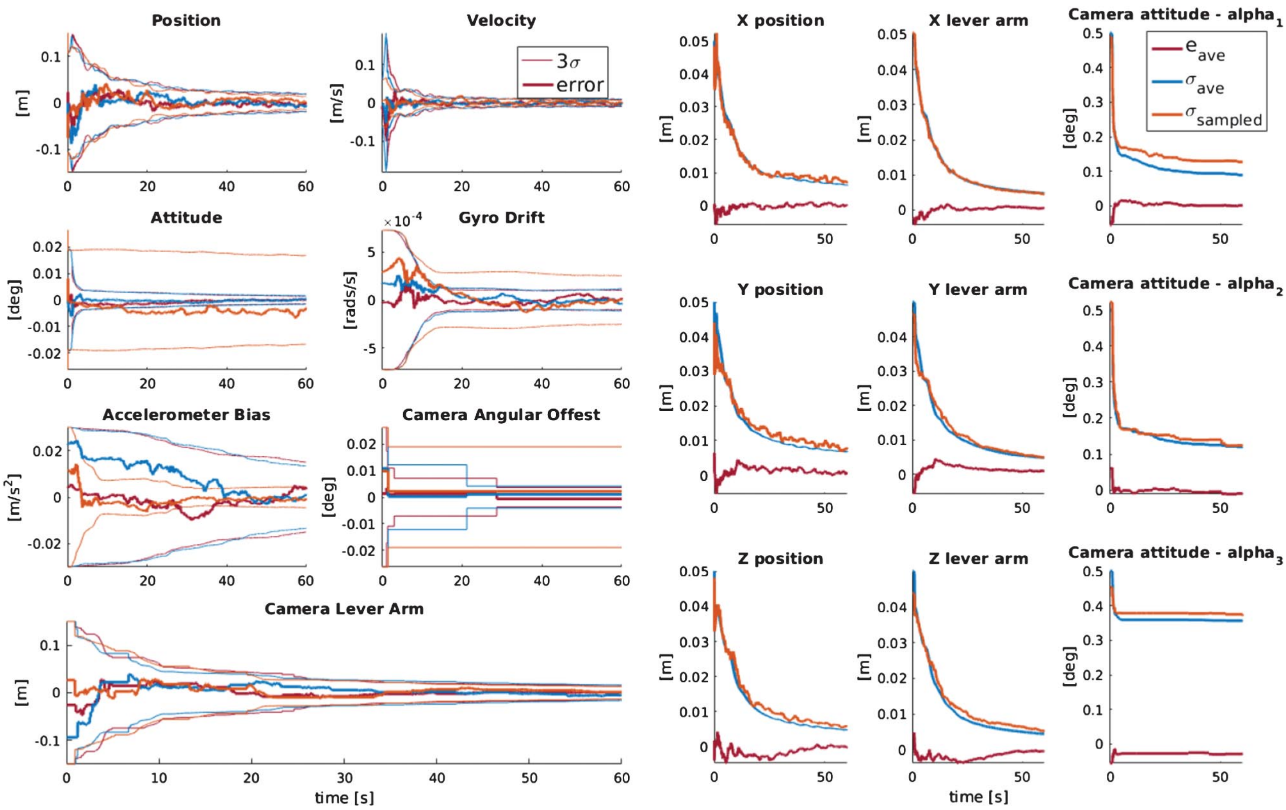
Again, attention is turned to the IMU–camera calibration example, where the full EKF previously failed to maintain consistency. The FPSKF is run in Monte Carlo simulation with an update weight of $\beta = 0.25$ for the lever arm and attitude offset states, which are updated at each time step.

As seen in Fig. 10, the camera attitude offset updates are less restricted without the information threshold used in the ISKF, but the filter is also less consistent in these states as well. The results are less dramatic than



a) Position σ values b) Gravity σ values c) NEES

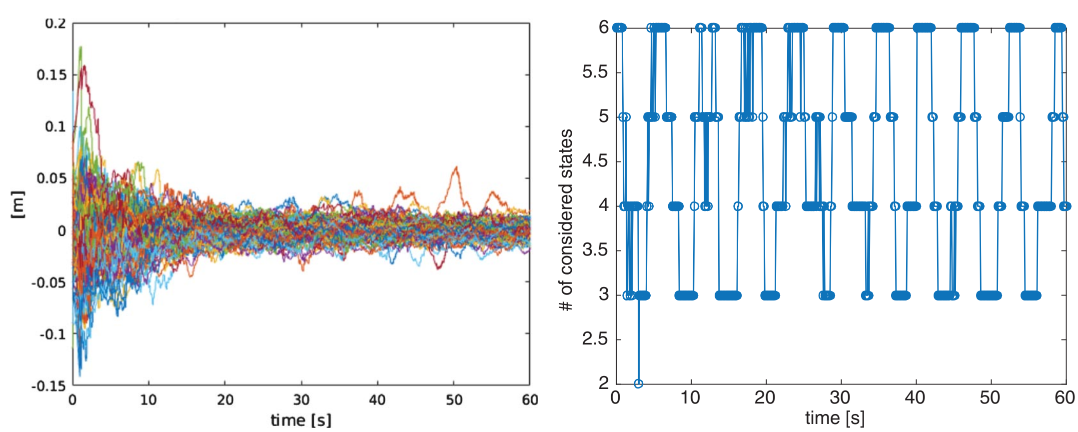
Fig. 6 ISKF Monte Carlo results for the 1-D system.



a) Results from a single run

b) Results from 100 Monte Carlo runs

Fig. 7 ISKF IMU–camera calibration.



a) IMU-camera x state error per run

b) Considred state count at each update for a single run

Fig. 8 ISKF x state and consider count.

the ISKF results but still show significantly diminished divergent behavior compared to the EKF runs (this is also evident in the x state error plot in Fig. 11), even though updates were still applied during the least observable portions of the runs. The NEES for the FPSKF is displayed in Table 4.

C. Generalized Partial-Update Schmidt–Kalman Filter Results

1. One-Dimensional System

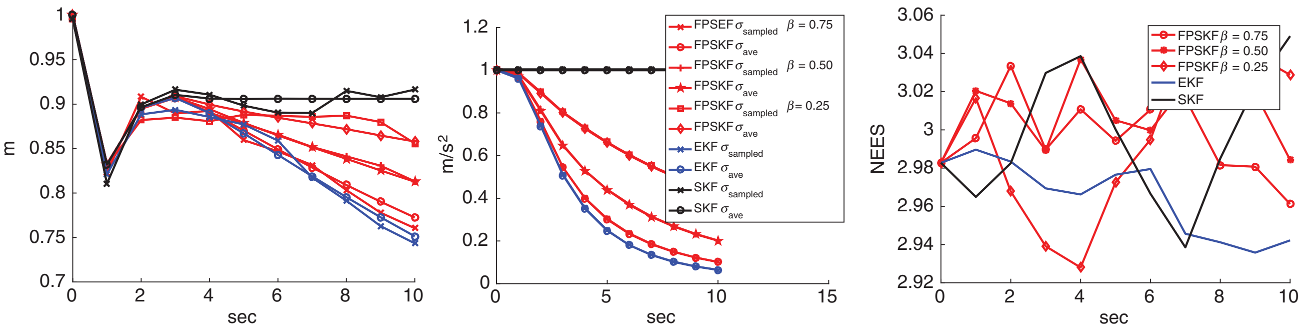
The generalized PSKF is now applied to the 1-D example to demonstrate the filter consistency. Figure 12 shows the results for the case when all three states have differing, but fixed, β values.

Table 3 NEES for the ISKF and EKF

t, s	ISKF	EKF
5	16.5	101.5
15	20.4	3336.6
60	18.0	$1.6004e+05$

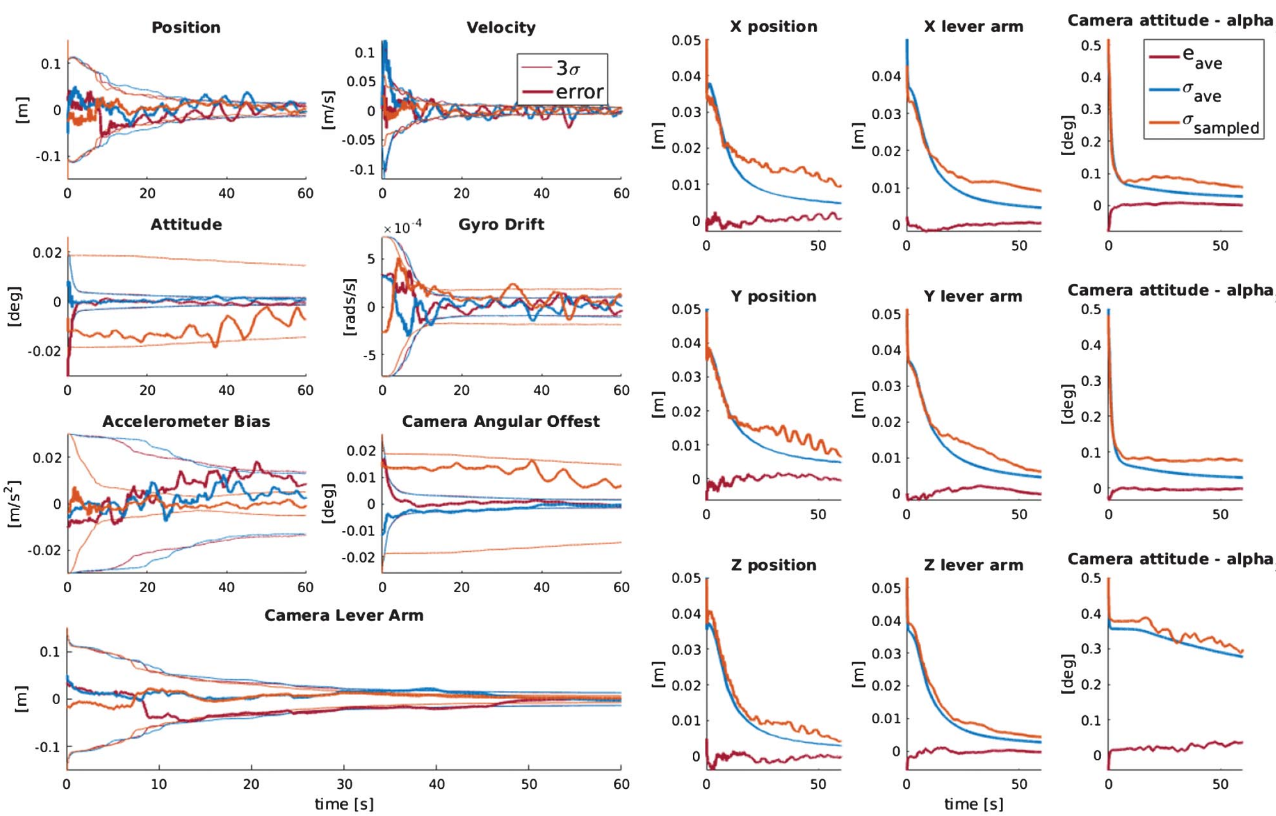
Table 4 NEES for FPSKF and EKF

t, s	FPSKF	EKF
5	26.4	101.5
15	77.1	3336.6
60	170.5	$1.6004e+05$



a) Position σ values b) Gravity σ values c) NEES

Fig. 9 FPSKF Monte Carlo results for the 1-D system.



a) Results from a single run b) Results from 100 Monte Carlo runs

Fig. 10 FPSKF IMU–camera calibration.

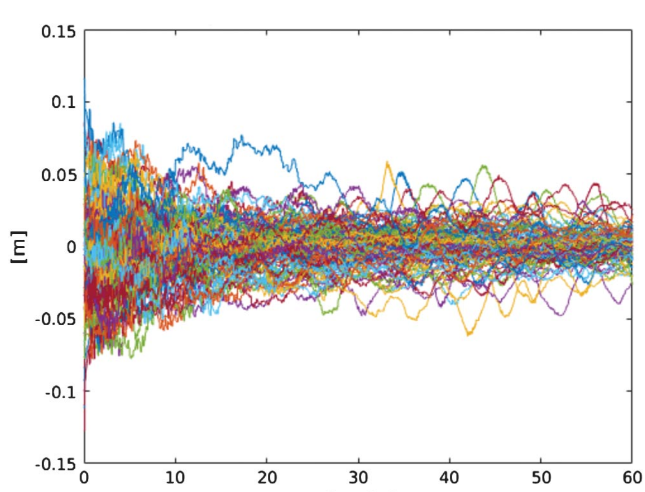
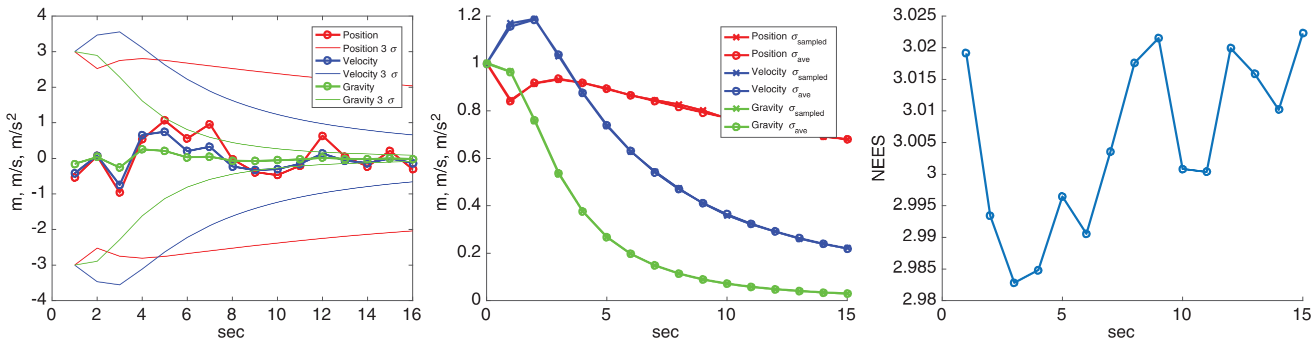


Fig. 11 FPSKF IMU *x* state error per run.

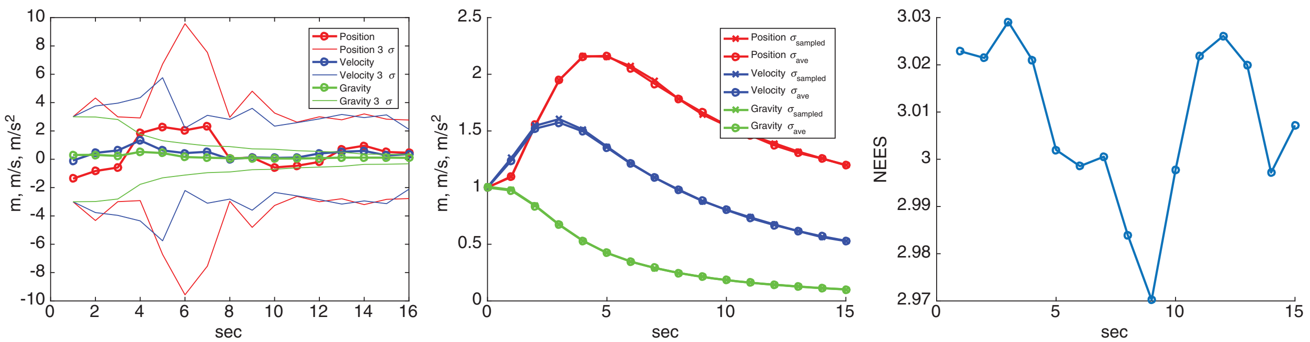
The 1-D example is reused for the case where β values are varied at each step; see Fig. 13. Each individual state's β value is drawn from a uniform distribution between zero and 1 at each time step. Therefore, unlike Figs. 12a and 12b, the sigma bounds for the single run case (displayed in Fig. 13a) and the average sigma bounds over all the MC runs (Fig. 13b) do not match. The sigma values for the randomized weighting are also significantly larger than the previous case because, on average, more information is neglected per updated. The NEES, however, is still appropriate for the system.

These results support the unbiased and consistent nature of the generalized PSKF updates when applied to linear systems, even in the case when update weights are varied for each state at each measurement update. Clearly, partially updating fully observable states, especially with a randomized weighting approach, is ineffective in terms of estimation accuracy of the linear system. Worse, for some (nonlinear) systems, randomized weighting would also likely cause divergence. However, the generalized form provides the framework to tailor the update weights as needed, allowing for more targeted weighting approaches, and ideally improved filter consistency, and it does so in a mathematically sound way.



a) Single Run b) Monte Carlo results c) NEES

Fig. 12 1-D PSKF results when each state has a unique update weight, $\beta = [0.9, 0.8, 0.7]^{\top}$.



a) Single Run b) Monte Carlo results c) NEES

Fig. 13 1-D PSKF results for $\beta(t) = rand_{1\times 3}$.

2. *Inertial Measurement Unit–Camera Calibration*

Returning to the IMU–camera calibration example, recall that the ISKF, with proper heuristics, or in this case an information metric, managed to avoid the worst of the minimally observable updates. The FPSKF, on the other hand, allows for partial updates to avoid a filter overreaction to particularly large measurement errors or hazardous linearization errors, even when notionally they are observable. Both were fairly effective at mitigating nonlinear effects, which cause filter divergence in the full EKF case.

Here, we use the PSKF framework for two filter implementations. The first (PSKF 1) combines the two update approaches from the ISFK and FPSKF. The second (PSKF 2) combines these previous filters but also restricts the IMU position and attitude updates.

The Monte Carlo runs continue to use the parameters from Table 1, and the information metric thresholds of $T_p = 0.9$ and $T_\alpha = 0.9$ are used with Eqs. (32) and (33) to determine if/when to update the lever arms and camera attitude offset states. The associated update weights for both PKSF implementations are listed in Table 5.

The generalized results for PSKF 2 can be seen in Fig. 14 and shows increased robustness beyond any of the previous filters (PSKF 1 displayed very similar but slightly less consistent results, and plots are omitted). The individual errors in the x state are plotted in Fig. 15, as are the number of considered states per update. Again, it is clear that the camera attitude states reach their final estimate values fairly quickly in the PSKF runs. Similar to the ISKF, this limits the final accuracy of the estimates, but it also results in a consistent filter that leverages most, if not all, available information possible without generating divergent results.

Finally, the NEES values for both the PSKF runs are almost perfect and shown in Table 6. This stands in stark contrast to the EKF NEES values reprinted in the same table. Note that the partial updates (95%) applied to the IMU position and attitude states in PSKF 2 did, in this case, have a mildly beneficial impact on filter consistency compared to PSKF 1.

Table 5 PSKF IMU–camera calibration update weights

State	β values, PSKF 1	β values, PSKF 2
IMU position	1.0	0.95
IMU velocity	1.0	1.0
IMU attitude	1.0	0.95
Gyro bias	1.0	1.0
Accelerometer bias	1.0	1.0
Lever arm	0.25 or 0.0	0.25 or 0.0
Camera angular offset	0.25 or 0.0	0.25 or 0.0

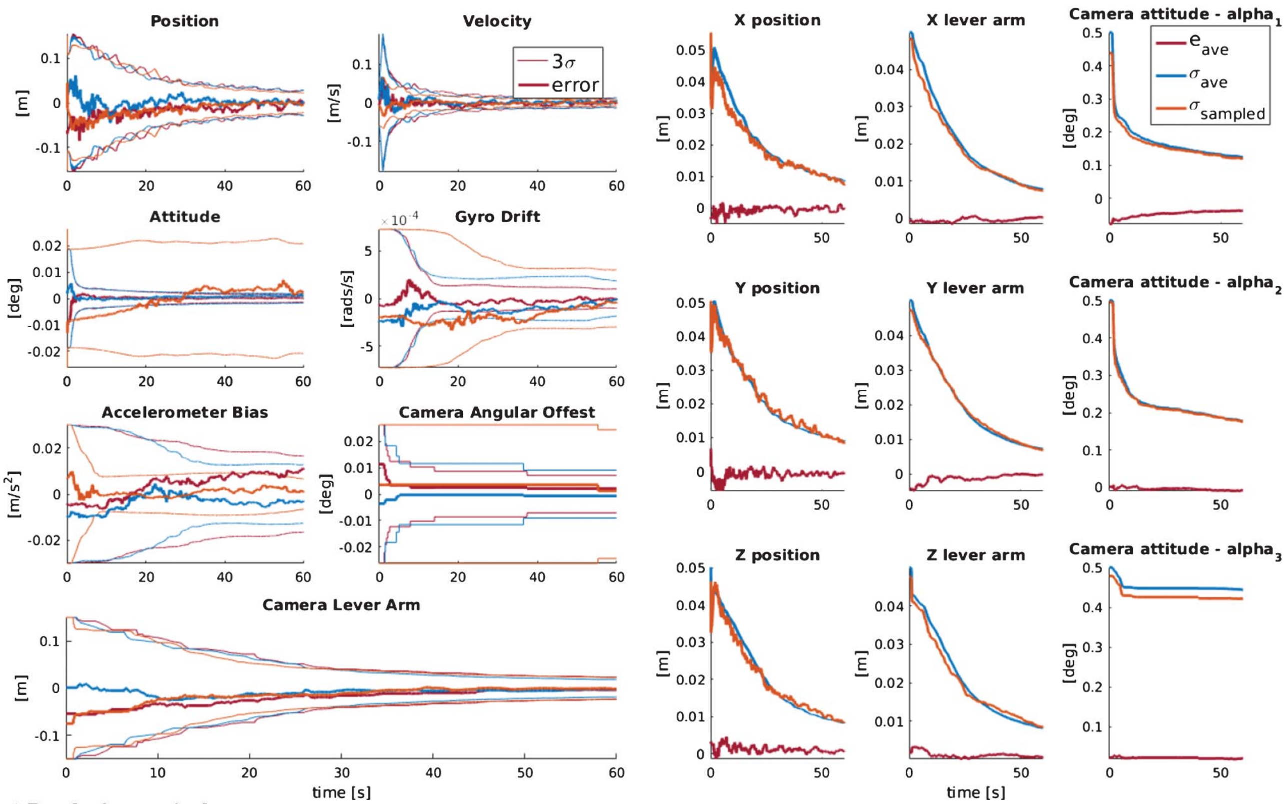
D. Results Discussion

The concepts of the PSKF (as well as ISKF and FPSKF variants) were shown to be sound and to have the potential to significantly improve filter consistency for many filtering applications through easy-to-implement modifications to the standard EKF update. We now share some relevant information about the results in this paper.

1) The estimation performance of the IMU–camera example (for any of the filters) can be affected by the amounts and types of motion; however, the intent of the example was not to demonstrate the accuracy of the calibration process but instead showcase the effectiveness of the differing filter modifications, in comparison to the EKF, with the information content at hand. Therefore, no special effort was made to select the most effective trajectories.

2) In the IMU–camera examples, both the ISKF and PSKF modifications produce consistent results if their respective initial uncertainties are decreased, yet still at a level where the EKF is not adequate. Additionally, the PSKF demonstrated very similar results with a larger initial uncertainty of 1.5 deg in all attitude states; the results were more consistent than either the ISKF or FPSKF, which performed at just 0.5 deg.

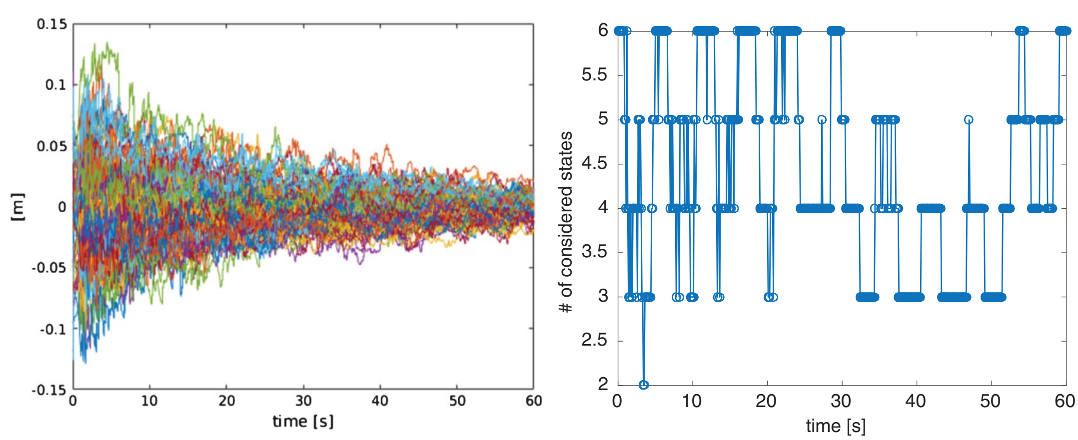
3) Filter results, across the board, are improved (some to a larger extent) when the first update in the IMU–camera calibration is only considered. The very first measurement update is made based on a covariance with no cross-correlation terms between the nuisance states and core INS states; therefore, the first update is made with both the largest uncertainty and the least information. When just starting a filter run, it may be advisable to perform an ISKF (non)update or



a) Results from a single run

b) Results from 100 Monte Carlo runs

Fig. 14 PSKF IMU–camera calibration.



a) IMU x state error per run

b) State count with zero update weight at each update for a single run

Fig. 15 PSKF x state and consider count.

limited PSKF update, regardless of the filter approach leveraged afterward.

4) In the PSKF example (PSKF 2), the IMU position and attitude updates where chosen to be limited (partially updated) because they, like the lever arm and camera attitude offset, appear directly in the measurement model. However, other states could also (or alternatively) be partially updated. No thorough examination was made as to which β_i values would be most effective as initial uncertainties were further increased.

5) The modifications shown in this paper are also appropriate for the UKF. The same 1-D and IMU–camera calibration examples have been run on a standard UKF [5]. There were nuanced differences in simulation setup, information metric, and results, which are outside of the scope of this paper. However, it is fair to say that the UKF-based runs displayed results similar to those seen in this paper for the EKF, in that the ISKF, FPSKF, and PSKF modifications extended the range of uncertainty in which the UKF-based filters were able to maintain consistency.

Table 6 NEES for PSKF and EKF

t, s	PSKF 1	PSKF 2	EKF
5	12.5	12.1	101.5
15	13.1	11.8	3336.6
60	12.6	12.3	$1.6004e+05$

VII. Extensions and Use Cases

The filter modifications shared here may be useful for many estimation applications to provide the filter developer with additional tools for handling static biases, things like the lever arm on GPS receivers [15] or the lever arm and orientation of cameras for systems that cannot be easily calibrated in a lab setting [16]. Other various applications currently leveraging the SKF may also benefit [17–19].

It was also shown that the PSKF modifications can be applied to time-varying states, which opens up an entirely new set of filtering applications where the Schmidt concept may be brought to bear. Nominally, time-varying states/biases are required to be estimated; otherwise, their uncertainty grows (potentially without bound, but certainly to unacceptable size), or they go unmodeled and are artificially accounted for with process noise applied on other filter states. For states with larger process noises and/or less accurate propagation, giving up information within an update cycle may be ill advised, but for questionably observable or significantly nonlinear states or measurements, there may be a benefit of reducing the update weight slightly, as seen in the PSKF results.

Another area where the PSKF approach has immediate applicability would be the case of a user wanting to artificially limit a particular state or its covariance. There are times when, practically speaking, a user knows that a state cannot be known to better than a certain level, or it is preferable to avoid "too tight" of a covariance, even when there is process noise, but the EKF does not necessarily agree and will diligently reduce the covariance as long as the associated gains are nonzero. Cases with flicker noise [20] or where otherwise mild biases suddenly jump in practice can benefit from a "not too tight" covariance. There are several ad hoc ways to limit the covariance from going below some minimum desired value, but they are often just that, ad hoc, and not surprisingly, they are also not well documented. The PSKF approach offers a simple mapping between the user-defined, minimal desired covariance value and the necessary β_i term [Eq. (69)] to generate the correct partial state update to coincide with the assigned covariance.

A similar approach can be used for constraining a state's value. As an example, the inverse depth parameterization of unknown features [21] for simultaneous localization and mapping (SLAM) artificially resets the state and its covariance when an update causes an inverse depth state to become negative (or fall below some $\rho_{\min}$ value) [22]. Although the approach is innovative, practical, and generally functional, it resets the covariance based on how far the inverse depth estimate went below $\rho_{\min}$, i.e., a function of $\boldsymbol{K}(\tilde{\boldsymbol{z}} - \boldsymbol{H}\hat{\boldsymbol{x}})$, but the covariance reset should be a function of $\boldsymbol{KH}$, and it is suspected that this discrepancy adversely affects filter consistency, which is already a notorious issue for SLAM systems [14]. Here, Eq. (68) could be used to solve for the proper value of β_i based on the full update and $\rho_{\min}$ and used to correct the covariance accordingly.

The PSKF approach could be combined with the classic SKF implementation to maintain some of the computational saving normally afforded by the SKF, especially in cases with large numbers of considered parameters. The traditional SKF could be applied to true consider states (which are never updated), whereas the PSKF updates would target key nuisance (or certain time-varying) states. And finally, the PSKF modification, or at least the concept, should be extensible to the square root and other filter forms (at varying levels of computational impact).

VIII. Conclusions

This paper has introduced the intermittent Schmidt–Kalman filter (ISKF), the fixed-weight partial-update Schmidt–Kalman filter (FPSKF), and the partial-update Schmidt–Kalman filter (PSKF). The modifications were shown to be easily implementable through small modifications of the classic EKF implementation and were shown to be sound in analysis and demonstration of a classic 1-D simulation. An information-based metric was developed for the IMU–camera calibration example and applied to ISKF and PSKF modifications to determine if and when to update the nuisance terms. When applied to the same IMU–camera calibration example, the ISKF, FPSKF, and especially the PSKF were shown to be effective and consistent, whereas the classic, full EKF implementation was not.

Given the wide array of systems that continue to leverage the EKF or SKF and the wide variety of biases, nonlinearities, and numerical instabilities that may be encountered, it is hoped that the modifications introduced in this paper will provide a new and flexible set of tools to better accommodate the real-world challenges of Kalman filtering.

Appendix A: Supporting Lemmas

Here, an alternative proof of Lemma 3 is presented, one that only relies on the consistency of the EKF under the linear system assumptions. To simplify the presentation, some additional lemmas (which also provide useful insight) are first needed.

Lemma 8: For a linear system where $\hat{\boldsymbol{x}}^- \sim N(\boldsymbol{x}, \boldsymbol{P}^-)$, $\tilde{\boldsymbol{z}} \sim N(\boldsymbol{z}, \boldsymbol{R})$, $\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\tilde{\boldsymbol{z}} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)$, and $\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P}^-$, then $\mathrm{cov}(\hat{x}_\mathrm{i}^-, \delta\hat{x}_\mathrm{j}) = -[\boldsymbol{KHP}^-]_{ij}$, where $\delta\hat{x}_i = \boldsymbol{K}_{i,1:m}(\tilde{\boldsymbol{z}} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)$ is the update applied to the ith element of $\hat{\boldsymbol{x}}$.

Proof: Assuming n states and m measurements and starting with the covariance between updated state values, we have

$$\mathrm{cov}(\hat{x}_i^-, \delta\hat{x}_j) = \mathrm{cov}(\hat{x}_i^-, \boldsymbol{K}_{j,\mathbf{1}:m}(\tilde{\boldsymbol{z}} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)) \tag{A1}$$

$$\begin{aligned} &= \mathrm{cov}(\hat{x}_i^-, \boldsymbol{K}_{j,1}(\tilde{z}_1 - (\boldsymbol{H}_{11}\hat{x}_1^- + \ldots + \boldsymbol{H}_{1n}\hat{x}_n^-)) + \ldots \\ &\quad + \boldsymbol{K}_{jm}(\tilde{z}_m - (\boldsymbol{H}_{m1}\hat{x}_1^- + \ldots + \boldsymbol{H}_{m,n}\hat{x}_n^-))) \end{aligned} \tag{A2}$$

$$\begin{aligned} &= \boldsymbol{K}_{j1}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_1 - \boldsymbol{H}_{11}\hat{x}_1^- - \ldots - \boldsymbol{H}_{1n}\hat{x}_n^-) + \ldots \\ &\quad + \boldsymbol{K}_{jm}\mathrm{cov}(\hat{x}_i, \tilde{z}_m - \boldsymbol{H}_{m1}\hat{x}_1^- - \ldots - \boldsymbol{H}_{mn}\hat{x}_n^-) \end{aligned} \tag{A3}$$

$$\begin{aligned} &= \boldsymbol{K}_{j1}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_1) - \boldsymbol{K}_{j1}\mathrm{cov}(\hat{x}_i^-, \boldsymbol{H}_{11}x_1^- + \ldots + \boldsymbol{H}_{1n}\hat{x}_n^-) + \ldots \\ &\quad + \boldsymbol{K}_{jm}\mathrm{cov}(\hat{x}_i^-, \tilde{z}_m) - \boldsymbol{K}_{jm}\mathrm{cov}(\hat{x}_i^-, \boldsymbol{H}_{11}\hat{x}_1^- + \ldots + \boldsymbol{H}_{1n}x_n^-) \end{aligned} \tag{A4}$$

Recognizing, from the linear system assumptions, that $\mathrm{cov}(\hat{x}_i^-, \tilde{z}) = 0$, additional algebraic manipulation results in

$$\begin{aligned} \mathrm{cov}(\hat{x}_i^-, \delta\hat{x}_j) &= -\boldsymbol{K}_{j1}\boldsymbol{H}_{11}\mathrm{cov}(\hat{x}_i^-, \hat{x}_1^-) - \ldots - \boldsymbol{K}_{j1}\boldsymbol{H}_{1n}\mathrm{cov}(\hat{x}_i^-, \hat{x}_n^-) - \ldots \\ &\quad - \boldsymbol{K}_{jm}\boldsymbol{H}_{11}\mathrm{cov}(\hat{x}_i^-, \hat{x}_1^-) - \ldots - \boldsymbol{K}_{jm}\boldsymbol{H}_{1n}\mathrm{cov}(\hat{x}_i^-, \hat{x}_n^-) \end{aligned} \tag{A5}$$

$$\begin{aligned} &= -\boldsymbol{K}_{j1}\boldsymbol{H}_{11}\boldsymbol{P}_{i1}^- - \ldots - \boldsymbol{K}_{j1}\boldsymbol{H}_{1n}\boldsymbol{P}_{in}^- - \ldots \\ &\quad - \boldsymbol{K}_{jm}\boldsymbol{H}_{11}\boldsymbol{P}_{i1}^- - \ldots - \boldsymbol{K}_{jm}\boldsymbol{H}_{1n}\boldsymbol{P}_{in}^- \end{aligned} \tag{A6}$$

$$= -[\boldsymbol{KHP}^-]_{ij} \tag{A7}$$

□

Lemma 9: For a linear system where $\hat{\boldsymbol{x}}^- \sim N(\boldsymbol{x}, \boldsymbol{P}^-)$, $\tilde{\boldsymbol{z}} \sim N(\boldsymbol{z}, \boldsymbol{R})$, $\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\tilde{\boldsymbol{z}} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)$, and $\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P}^-$, then $\mathrm{cov}(\delta\hat{x}_i, \delta\hat{x}_j) = [\boldsymbol{KHP}^-]_{ij}$, where $\delta\hat{x}_i = \boldsymbol{K}_{i,1:m}(\tilde{z} - \boldsymbol{H}\hat{x}^-)$ is the update applied to the ith element of $\hat{\boldsymbol{x}}$.

Proof: We start with

$$\mathrm{cov}(\hat{x}_i^+, \hat{x}_j^+) = \mathrm{cov}(\hat{x}_i^- + \delta\hat{x}_i, \hat{x}_j^- + \delta\hat{x}_j) \tag{A8}$$

$$= \mathrm{cov}(\hat{x}_i^-, \hat{x}_j^-) + \mathrm{cov}(\hat{x}_i^-, \delta\hat{x}_j) + \mathrm{cov}(\delta\hat{x}_i, \hat{x}_j^-) + \mathrm{cov}(\delta\hat{x}_i, \delta\hat{x}_j) \tag{A9}$$

From Lemma 8, we have that $\mathrm{cov}(\hat{x}_i^-, \delta\hat{x}_j) = -[\boldsymbol{KHP}^-]_{ij}$, and we know that $[\boldsymbol{KHP}^-]$ is symmetric, and so although $[\boldsymbol{KHP}^-]_{ij}$ and $[\boldsymbol{KHP}^-]_{ji}$ are formed with different elements of $\boldsymbol{K}$, $\boldsymbol{H}$, and $\boldsymbol{P}^-$, they are equal. Because $\mathrm{cov}(\hat{x}_i^+, \hat{x}_j^+) = \boldsymbol{P}_{ij}^- - [\boldsymbol{KHP}^-]_{ij}$, Eq. (A9) then implies that $\mathrm{cov}(\delta\hat{x}_\mathrm{i}, \delta\hat{x}_\mathrm{j}) = [\boldsymbol{KHP}^-]_{ij}$. □

This result intuitively makes sense because the covariance of the updates should be the negative of the change in covariance of the estimates (for an unbiased system).

Appendix B Alternative Proof of Lemma 3

Lemma 3: For a linear system where $\hat{x}^- \sim N(x, \boldsymbol{P}^-)$, $\tilde{z} \sim \boldsymbol{N}(z, \boldsymbol{R})$, $\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\tilde{z} - \boldsymbol{H}\hat{\boldsymbol{x}}^-)$, and $\boldsymbol{P}^+ = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^-$, then cov $(\hat{x}_i^+, \hat{x}_j^+) = \mathrm{cov}(\hat{x}_i^+, \hat{x}_j^-)$.

Proof: Using Lemmas 8 and 9, we have

$$\mathrm{cov}(\hat{x}_i^+, \hat{x}_j^+) = \mathrm{cov}(\hat{x}_i^- + \delta\hat{x}_i, \hat{x}_j^- + \delta\hat{x}_j) \tag{B1}$$

$$= \mathrm{cov}(\hat{x}_i^-, \hat{x}_j^-) + \mathrm{cov}(\hat{x}_i^-, \delta\hat{x}_j) + \mathrm{cov}(\delta\hat{x}_i, \hat{x}_j^-) + \mathrm{cov}(\delta\hat{x}_i, \delta\hat{x}_j) \tag{B2}$$

$$= \boldsymbol{P}_{ij} - [\boldsymbol{KHP}^-]_{ij} - [\boldsymbol{KHP}^-]_{ij} + [\boldsymbol{KHP}^-]_{ij} \tag{B3}$$

$$= \boldsymbol{P}_{ij}^- - [\boldsymbol{KHP}^-]_{ij} \tag{B4}$$

$$= \mathrm{cov}(\hat{x}_i^-, \hat{x}_j^-) + \mathrm{cov}(\delta\hat{x}_i, \hat{x}_j^-) \tag{B5}$$

$$= \mathrm{cov}(\hat{x}_i^+, \hat{x}_j^-) \tag{B6}$$

□

References

[1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *ASME Journal of Basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.
doi:10.1115/1.3662552

[2] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Academic Press, New York, 1979, pp. 296–305, Chap. 6.

[3] Schmidt, S. F., "Application of State Space Methods to Navigation Problems," *Advanced Control Systems*, *Advances in Control Systems*, Vol. 3, 1966, pp. 293–340.

[4] Bellantoni, J. F., and Dodge, K. W., "A Square Root Formulation of the Kalman–Schmidt Filter," *AIAA Journal*, Vol. 5, No. 7, 1967, pp. 1309–1314.
doi:10.2514/3.4189

[5] Julier, S. J., and Uhlmann, J. K., "New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, Vol. 182, July 1997.
doi:10.1117/12.280797

[6] Zarchan, P., and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, 4th ed., AIAA, Reston, VA, 2015, Chap. 20, pp. 777–815.
doi:10.2514/5.9781624102776.0727.0776

[7] Hu, X., Bao, M., Zhang, X. P., Guan, L., and Hu, Y. H., "Generalized Iterated Kalman Filter and Its Performance Evaluation," *IEEE Transactions on Signal Processing*, Vol. 63, No. 12, June 2015, pp. 3204–3217.
doi:10.1109/TSP.2015.2423266

[8] Brown, R., and Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed., Wiley, New York, 1997, pp. 361–367.

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

[10] Stauch, J., and Jah, M., "Unscented Schmidt–Kalman Filter Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 1, 2015, pp. 117–123.
doi:10.2514/1.G000467

[11] Brink, K., and Soloviev, A., "Filter-Based Calibration for an IMU and Multi-Camera System," *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, April 2012, pp. 730–739.
doi:10.1109/PLANS.2012.6236950

[12] Mirzaei, F., and Roumeliotis, S., "A Kalman Filter-Based Algorithm for IMU–Camera Calibration: Observability Analysis and Performance Evaluation," *IEEE Transaction on Robotics*, Vol. 24, No. 5, 2008, 1143–1156.
doi:10.1109/TRO.2008.2004486

[13] Kelly, J., and Sukhatme, G., "Visual-Inertial Simultaneous Localization, Mapping and Sensor-to-Sensor Self-Calibration," *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'09)*, IEEE Publ., Piscataway, NJ, 2009, pp. 360–368.
doi:10.1109/CIRA.2009.5423178

[14] Bailey, T., Nieto, J., Guivant, J., Stevens, M., and Nebot, E., "Consistency of the EKF-SLAM Algorithm," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE Publ., Piscataway, NJ, Oct. 2006, pp. 3562–3568.
doi:10.1109/IROS.2006.281644

[15] Hong, S., Lee, M. H., Chun, H.-H., Hong Kwon, S., and Speyer, J. L., "Experimental Study on the Estimation of Lever Arm in GPS/INS," *IEEE Transactions on Vehicular Technology*, Vol. 55, No. 2, March 2006, pp. 431–448.
doi:10.1109/TVT.2005.863411

[16] Venable, D. T., "Improving Real World Performance for Vision Navigation in a Flight Environment," Ph.D. Thesis, U.S. Air Force Inst. of Technology, Wright-Patterson AFB, OH, 2016.

[17] Novoselov, R. Y., Herman, S. M., Gadaleta, S. M., and Poore, A. B., "Mitigating the Effects of Residual Biases with Schmidt–Kalman Filtering," *Proceedings of the 7th International Conference on Information Fusion*, IEEE Publ., Piscataway, NJ, July 2005, pp. 356–358.
doi:10.1109/ICIF.2005.1591877

[18] Bar-Itzhack, I., and Nathan, J., "Consider-Filters for Recursive Attitude Determination," *Guidance, Navigation, and Control Conference*, AIAA Paper 1994-3675, 1994.
doi:10.2514/6.1994-3675

[19] Yang, Y., Yue, X., and Dempster, A. G., "GPS-Based Onboard Real-Time Orbit Determination for Leo Satellites Using Consider Kalman Filter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 52, No. 2, April 2016, pp. 769–777.
doi:10.1109/TAES.2015.140758

[20] Pittelkau, M. E., "Attitude Determination Kalman Filter with a $1/f$ Flicker Noise Gyro Model," *Proceedings of the 26th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+*, ION, Sept. 2013, pp. 2143–2160.

[21] Solà, J., Vidal-Calleja, T., Civera, J., and Montiel, J., "Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines," *International Journal of Computer Vision*, Vol. 97, No. 3, 2012, pp. 339–368.
doi:10.1007/s11263-011-0492-5

[22] Parsley, M. P., and Julier, S. J., "Avoiding Negative Depth in Inverse Depth Bearing-Only SLAM," *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE Publ., Piscataway, NJ, Sept. 2008, pp. 2066–2071.
doi:10.1109/IROS.2008.4651118