

# A new framework for nonlinear Kalman filters <sup>★</sup>

Shida Jiang <sup>a</sup>, Junzhe Shi <sup>a</sup>, Scott Moura <sup>a</sup>

<sup>a</sup>Department of Civil and Environmental Engineering, University of California, Berkeley, CA, USA

---

## Abstract

The Kalman filter (KF) is a state estimation algorithm that optimally combines system knowledge and measurements to minimize the mean squared error of the estimated states. While KF was initially designed for linear systems, numerous extensions of it, such as extended Kalman filter (EKF), unscented Kalman filter (UKF), cubature Kalman filter (CKF), etc., have been proposed for nonlinear systems over the last sixty years. Although different types of nonlinear KFs have different pros and cons, they all use the same framework of linear KF. Yet, according to our theoretical and empirical analysis, the framework tends to give overconfident and less accurate state estimations when the measurement functions are nonlinear. Therefore, in this study, we designed a new framework that can be combined with any existing type of nonlinear KFs and showed theoretically and empirically that the new framework estimates the states and covariance more accurately than the old one. The new framework was tested on four different nonlinear KFs and five different tasks, showcasing its ability to reduce estimation errors by several orders of magnitude in low-measurement-noise conditions. The codes are available at <https://github.com/Shida-Jiang/A-new-framework-for-nonlinear-Kalman-filters>

**Key words:** Kalman filtering; nonlinear observer and filter design; extended Kalman filter; unscented Kalman filter; estimation theory; parameter and state estimation

---

## 1 Introduction

In control theory, the Kalman filter (KF) is a state estimation algorithm designed for linear systems. In such a system, a set of linear state transition functions describes how the states evolve, and a set of linear measurement functions describes the measured values of certain linear combinations of states. It is assumed that both the state transition and measurement functions contain some additive noise. The KF optimally combines the system knowledge and measurements to minimize the expected value of the mean-squared error of the state estimation. Additionally, when the process noise, measurement noise, and state estimation errors all follow a multivariate normal (Gaussian) distribution, the state estimation given by the Kalman filter is also the maximum likelihood estimation [1]. Due to these desirable features, KF is often referred to as the optimal linear state estimator. Since its birth in about 1960, KF has achieved massive success in navigation [2], motion control [3], statistics [4], signal processing [5], etc.

The most significant limitation of KF is that it can only be applied to systems with linear state transition functions and linear measurement functions. Unfortunately, many real-life systems are nonlinear [6]. Therefore, some extensions of KF have been developed to solve the state estimation problem for these nonlinear systems, and the most well-known ones are the extended Kalman filter (EKF) and the unscented Kalman filter (UKF).

EKF linearizes the state transition functions and measurement functions locally at each time step [6]. This technique transforms the nonlinear system into a linear system, where KF can be directly applied. While EKF is relatively simple, the linearization performed in each step neglects the higher-order derivatives, making the state estimations biased and overconfident. Specifically, estimations can be biased since  $\mathbb{E}(f(x)) = f(\mathbb{E}(x))$  is not guaranteed for a nonlinear function, and overconfidence arises when the actual variance is much higher than estimated. These problems are partially solved by the second-order extended Kalman filter (EKF2) [7], which approximates the nonlinear system by its second-order Taylor expansion. However, EKF2 requires the calculation of the Hessian matrix of each state transition function and measurement function, which adds non-trivial computational complexity to the algorithm. Additionally, EKF2 still sometimes gives inaccurate and overcon-

---

<sup>★</sup> This paper was not presented at any IFAC meeting. Corresponding author Shida Jiang Tel. +1 510-944-9339.

Email addresses: [shida\\_jiang@berkeley.edu](mailto:shida_jiang@berkeley.edu) (Shida Jiang), [junzhe\\_shi@berkeley.edu](mailto:junzhe_shi@berkeley.edu) (Junzhe Shi), [smoura@berkeley.edu](mailto:smoura@berkeley.edu) (Scott Moura).

fident estimations, so it is generally used less frequently than EKF.

Both EKF and EKF2 require the calculation of the Jacobian matrices of the state transition functions and measurement functions, which may not always be accessible in real-life systems. In 1997, Simon et al. introduced another extension of KF to nonlinear systems that requires no direct calculation of the Jacobian matrix and avoids the accuracy loss caused by linearization [8]. In later literature, this extension is usually referred to as the UKF [9]. UKF approximates the distribution of the states after nonlinear transformations by sampling several points (called sigma points) around the estimated mean of the states. These points go through the same transformation as the mean, and a more accurate estimation of the states' distribution after the transformation can be obtained with the help of these extra sample points. In UKF, the sigma points are selected deterministically based on each state's standard deviation. This technique guarantees that the sigma points are evenly distributed around the central point and can provide a more accurate state estimation. Similar ideas have also been used in several other nonlinear KFs, including cubature Kalman filter (CKF) [10], quadrature Kalman filter [11], square-root quadrature Kalman filter [12], etc. All these filters use deterministic ways to find several additional points around the central point (the estimated states from the previous iteration) and use those points to approximate the first two moments of the states after the nonlinear transformation. All these methods have some links with EKF2. As pointed out by Gustafsson et al. in [13], when the sigma points are infinitely close to the center, and the hyperparameters in the UKF are set to specific values, UKF will give the same state estimation as the EKF2, although the covariance estimation can still be a bit different. A disadvantage of these nonlinear KFs is that the number of sampling points is usually twice the number of states. Therefore, the method can be increasingly more complicated than EKF as the number of states increases.

While different extensions of the KF use various ways to approximate the first two moments of the states after a nonlinear transformation, they all follow the same framework as in the linear KF. In this paper, this framework is referred to as "the conventional framework" or "the old framework". In such a framework, the state estimations in each iteration can be divided into two steps: Predict and Update. In the "predict" step, the state and covariance estimations are predicted using the state transition function and estimates from the previous iteration. Then, in the "update" step, the measurements correct the state predictions with a feedback gain (also known as the Kalman gain), which is designed to minimize the trace of the states' covariance matrix after the update. After the states are updated, the state covariance matrix is also updated, ending the present iteration. Note that the covariance matrix will have the smallest possi-

ble trace after the update since that's how the Kalman gain is selected.

While such a framework can guarantee the optimality of the state estimation for linear systems, it often leads to overconfident covariance estimation and less accurate state estimation when applied to nonlinear systems. The primary reason for such a phenomenon is the assumption that the Kalman gain minimizes the trace of the state covariance matrix, which is generally not guaranteed for systems with nonlinear measurement functions. In fact, as we will show later, the "update" step can sometimes make the state estimation of the nonlinear system less accurate, so the assumption about the optimality of the "update" step can be extremely harmful, as it makes the algorithm become overconfident in an inaccurate estimation.

The problem described above will become increasingly significant as the measurement noise decreases, and the measurements play a more critical role in the state estimation [14]. In some circumstances, it can even lead to a highly undesirable and counterintuitive phenomenon: the actual estimation error increases as the measurement noise decreases. The primary cause of this phenomenon is that the actual estimation errors drop much slower than the estimated standard deviations, and such overconfidence prevents the filter from correcting itself in later iterations. As the development of technology makes all types of sensors more and more accurate, we can reasonably predict that this problem will become increasingly severe if we stick to the conventional framework for nonlinear KFs.

Unfortunately, so far, there is no general solution that can effectively prevent the nonlinear KF from overconfidence. Although it is possible to mitigate the problem by approximating the measurement function at higher orders (e.g., using EKF2 instead of EKF), these algorithms take a significantly longer computational time, and the problem will still, in general, exist. The lack of theoretical analysis for nonlinear KFs makes finding a general solution even harder. For those few existing theoretical analyses [13–15], the results are based on the assumption that the higher-ordered terms in the Taylor expansions of the measurement functions can be ignored. This assumption may not be reasonable because the predicted states can be very far away from the updated states, and ignoring higher-ordered terms can cause significant errors in the "update" step.

In this study, we proposed a new framework that solves the problem of overconfidence. The key idea of the new framework is to add a step that reevaluates the actual effect of the Kalman gain after the state update. With this additional step, the state covariance estimation can be much more accurate, and any unhelpful update can be withdrawn. The main contributions of this paper are:

- To the best of our knowledge, we are the first to give a mathematical proof that the conventional framework for nonlinear KFs, which has been used for around sixty years, generally has the problem of underestimating the trace of the state covariance matrix.
- We proposed a new framework to solve the problem of inaccurate covariance estimation and validated the framework's effectiveness by using both theoretical and empirical analysis.
- The proposed new framework can be combined with almost any type of nonlinear KF and reduce the state estimation error by more than an order of magnitude in various applications when the measurement noise is low.

The rest of the paper is organized as follows. Section 2 conducts a rigorous theoretical analysis of the problem of the conventional framework for nonlinear KFs and introduces a new framework that solves the problem. Section 3 presents a comprehensive comparison between the two frameworks. Specifically, both frameworks are combined with four different types of nonlinear KFs and validated on five distinct applications. The conclusions are drawn in Section 4. The detailed simulation setup and the detailed algorithms we used for validation can be found in the appendix.

## 2 Method

### 2.1 General formulation of nonlinear Kalman filters

Consider a system with the following dynamics:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \quad (1)$$

where  $x_k$  are the actual states at the time step  $k$ ,  $u_k$  are the inputs at the time step  $k$ , and  $w_k$  are the process noises that follow a zero-mean multivariate distribution with covariance  $Q_k$ . The estimations of the states made in the previous step are denoted as  $\hat{x}_{k-1|k-1}$ . Since the actual states are unknown, the KF assumes that the estimation errors  $\hat{x}_{k-1|k-1} - x_{k-1}$  follow a zero-mean multivariate distribution with covariance  $P_{k-1|k-1}$ . Whichever nonlinear KF is used, the first step of each iteration is called “predict”, which predicts the states using (1) and the estimations made in the previous time step. The predicted states are denoted as  $\hat{x}_{k|k-1}$ , and the errors  $\hat{x}_{k|k-1} - x_k$  are assumed to follow a zero-mean multivariate distribution with covariance  $P_{k|k-1}$ .

The second step of each iteration for any nonlinear KF is called “update”, which corrects the predicted states based on the measurements. Suppose that the measurements of the states are as follows:

$$z_k = h(x_k) + v_k \quad (2)$$

where  $z_k$  are the measurements at time step  $k$ , and  $v_k$  are the measurement noises that follow a zero-mean multivariate distribution with covariance  $R_k$ . Given (2) and the predicted states, the measurements can also be predicted using a particular method specific to the type of nonlinear KFs. The predicted measurements are denoted as  $\hat{y}_{k|k-1}$ , whose errors  $\hat{y}_{k|k-1} - h(x_k)$  are assumed to follow a zero-mean multivariate distribution with covariance  $P_{y,k}$ . The actual cross-covariance matrix between the predicted states and the predicted measurements is denoted as  $P_{xy,k}$ . The differences between the predicted and actual measurements are called the measurement residuals, denoted as  $\tilde{y}_k$ . Namely:

$$\tilde{y}_k = z_k - \hat{y}_{k|k-1} \quad (3)$$

If we assume that the measurement noises are independent from the predicted states, then  $\tilde{y}_k$  should have zero mean, and their covariance  $S_k$  can be calculated by:

$$S_k = P_{y,k} + R_k \quad (4)$$

KFs use the following equation to update the state estimation:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K\tilde{y}_k \quad (5)$$

where  $\hat{x}_{k|k}$  are the updated states, and  $K$  is the Kalman gain. With (3–5), the covariance of  $\hat{x}_{k|k}$  can be calculated as [16]:

$$P_{k|k} = P_{k|k-1} + K S_k K^T - P_{xy,k} K^T - K P_{xy,k}^T \quad (6)$$

KFs select the Kalman gain to minimize the trace of  $P_{k|k}$ . By taking the derivative of  $\text{tr}(P_{k|k})$  in (6), we find that the trace is minimized when:

$$K_{op} = P_{xy,k} S_k^{-1} \quad (7)$$

where  $K_{op}$  is the theoretical optimal value of the Kalman gain. When (7) is satisfied, (6) becomes:

$$P_{k|k,op} = P_{k|k-1} - K_{op} S_k K_{op}^T \quad (8)$$

Equations (5–8) illustrate the key idea of KFs, which is to update the states in a way that minimizes the trace of the state covariance matrix after the update. However, it should be noted that these equations cannot be directly applied to systems with nonlinear measurement functions. The primary reason is that  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$  can only be calculated if the actual state and the distribution of the states are known. Since neither of these holds in practice, the actual values for these covariance matrices are also unknown. In fact, as we will show next, some over-simplifications regarding these matrices are the key factors that can make the conventional framework for nonlinear KFs overconfident and inaccurate.

## 2.2 The conventional framework and its problem

While calculating  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$  at the actual state is impossible, it is possible to approximate their values based on the predicted states. Different nonlinear filters (e.g. UKF, EKF2, CKF) take different approaches to approximating these covariance matrices. In the subsequent analysis, we denote these approximated covariance matrices as  $\mathbf{P}_{y,k|k-1}$ ,  $\mathbf{S}_{k|k-1}$ , and  $\mathbf{P}_{xy,k|k-1}$  since they are all estimated based on the prediction. Although the detailed process of calculating  $\mathbf{P}_{y,k|k-1}$ ,  $\mathbf{S}_{k|k-1}$ , and  $\mathbf{P}_{xy,k|k-1}$  differs, all existing types of nonlinear KFs directly replace  $S_k$  and  $P_{xy,k}$  in (7) and (8) with the approximated values. Namely, the Kalman gain is selected as:

$$K_{est} = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1} \quad (9)$$

and the state covariance matrix is updated by:

$$\begin{aligned} P_{k|k,est} &= P_{k|k-1} - K_{est} S_{k|k-1} K_{est}^T \\ &= P_{k|k-1} - \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1} \mathbf{P}_{xy,k|k-1}^T \end{aligned} \quad (10)$$

On the other hand, the actual value of the updated state covariance matrix  $\mathbf{P}_{k|k,ac}$  given the selected Kalman gain in (9) can be calculated by substituting (9) into (6). Namely:

$$\begin{aligned} \mathbf{P}_{k|k,ac} &= P_{k|k-1} + \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1} S_k S_{k|k-1}^{-1} \mathbf{P}_{xy,k|k-1}^T \\ &\quad - \mathbf{P}_{xy,k} \mathbf{S}_{k|k-1}^{-1} \mathbf{P}_{xy,k|k-1}^T - \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1} \mathbf{P}_{xy,k}^T \end{aligned} \quad (11)$$

In other words,  $\mathbf{P}_{k|k,ac}$  plays the role of  $P_{k|k}$  in (6), and  $\mathbf{P}_{k|k,est}$  plays the role of  $P_{k|k,op}$  in (8). Intuitively,  $\text{tr}(\mathbf{P}_{k|k,ac})$  tends to be larger than  $\text{tr}(\mathbf{P}_{k|k,est})$  since  $\text{tr}(P_{k|k,op}) = \min \text{tr}(P_{k|k}) \leq \text{tr}(P_{k|k})$ . However, this relationship of size is not always preserved after the approximations. For example, according to (10) and (11), when  $P_{xy,k} = S_k = 1$  and  $\mathbf{P}_{xy,k|k-1} = \mathbf{S}_{k|k-1} = 0.9$ , we have  $\mathbf{P}_{k|k,est} - \mathbf{P}_{k|k,ac} = 0.1 > 0$ . Nevertheless, in most cases in practice [17, 18], we do find that  $\text{tr}(\mathbf{P}_{k|k,ac}) > \text{tr}(\mathbf{P}_{k|k,est})$ , meaning that using (10) will often (although not always) underestimate the actual covariance.

In order to make this statement more rigorous in math, we consider the approximated covariance matrices  $\mathbf{P}_{y,k|k-1}$ ,  $\mathbf{S}_{k|k-1}$ , and  $\mathbf{P}_{xy,k|k-1}$  as noisy versions of  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$ . Specifically, we assume that  $\mathbf{P}_{xy,k|k-1}$  is a random matrix,  $\mathbf{P}_{y,k|k-1}$  is a random positive semidefinite matrix, and  $\mathbf{S}_{k|k-1}$  is a random positive definite matrix. Note that we mark all the random covariance matrices in red in this section for clarity. Additionally, we assume that the “noises” that arise in the approximations are unbiased, so these random matrices

should satisfy:

$$\begin{cases} \mathbb{E}[\mathbf{P}_{y,k|k-1}] = P_{y,k} \\ \mathbb{E}[\mathbf{S}_{k|k-1}] = S_k \\ \mathbb{E}[\mathbf{P}_{xy,k|k-1}] = P_{xy,k} \end{cases} \quad (12)$$

These additional definitions and assumptions are made so that we can consider the “average” case and give a more mathematically rigorous definition of the overconfidence caused by (10). Namely, we have the following theorem.

**Theorem 1** Suppose that  $\mathbf{P}_{xy,k|k-1}$  and  $\mathbf{S}_{k|k-1}$  are random matrices that satisfy (12).  $\mathbf{S}_{k|k-1}$  is symmetric and positive definite. Then,  $P_{k|k,est}$  calculated from (10) satisfies:

$$\mathbb{E}[\mathbf{P}_{k|k,ac} - \mathbf{P}_{k|k,est}] \succeq 0 \quad (13)$$

where “ $\succeq$ ” represents positive semidefiniteness, and the equality only holds when  $\mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1}$  is constant.

**PROOF.** Let  $\mathbf{P} = \mathbf{P}_{xy,k|k-1} S_k^{-\frac{1}{2}}$ ,  $\mathbf{S} = S_k^{-\frac{1}{2}} \mathbf{S}_{k|k-1} S_k^{-\frac{1}{2}}$ . According to (12),

$$\begin{cases} \mathbb{E}[\mathbf{P}] = \bar{\mathbf{P}} = \mathbf{P}_{xy,k} S_k^{-\frac{1}{2}} \\ \mathbb{E}[\mathbf{S}] = I \end{cases} \quad (14)$$

According to (10, 11),

$$\begin{aligned} &\mathbf{P}_{k|k,ac} - \mathbf{P}_{k|k,est} \\ &= \mathbf{P} \mathbf{S}^{-2} \mathbf{P}^T - \bar{\mathbf{P}} \mathbf{S}^{-1} \mathbf{P}^T - \mathbf{P} \mathbf{S}^{-1} \bar{\mathbf{P}}^T + \mathbf{P} \mathbf{S}^{-1} \mathbf{P}^T \\ &= (\mathbf{P} \mathbf{S}^{-1} - \bar{\mathbf{P}})(\mathbf{P} \mathbf{S}^{-1} - \bar{\mathbf{P}})^T + \mathbf{P} \mathbf{S}^{-1} \mathbf{P}^T - \bar{\mathbf{P}} \bar{\mathbf{P}}^T \end{aligned} \quad (15)$$

For any fixed unit vector  $\mathbf{v}$ ,  $\mathbf{P}^T \mathbf{v}$  is a random vector with a mean of  $\bar{\mathbf{P}}^T \mathbf{v}$ . According to Lemma 4 in Appendix A,

$$\mathbb{E}[\mathbf{v}^T \mathbf{P} \mathbf{S}^{-1} \mathbf{P}^T \mathbf{v}] \geq \mathbf{v}^T \bar{\mathbf{P}} \bar{\mathbf{P}}^T \mathbf{v}. \quad (16)$$

Moving the right hand side to the left, the following must hold true:

$$\mathbb{E}[\mathbf{P} \mathbf{S}^{-1} \mathbf{P}^T - \bar{\mathbf{P}} \bar{\mathbf{P}}^T] \succeq 0 \quad (17)$$

Therefore, the expectation of (15) is always p.s.d. due to (17) and the fact that the first term in (15) is p.s.d. The equality only holds when  $\mathbf{P} \mathbf{S}^{-1}$  is constant, which is equivalent to  $K_{est} = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1}$  being constant (i.e., free from approximation errors).

Theorem 1 suggests that using (10) to estimate the state covariance matrix tends to lead to overconfident covariance estimation when the estimated  $S_k$  and  $P_{xy,k}$  are not the same as their actual values. Admittedly, in practice, the errors of the approximated covariance matrices

$P_{y,k|k-1}$ ,  $S_{k|k-1}$ , and  $P_{xy,k|k-1}$  are not as simple as zero-mean random matrices. However, Theorem 1 does give us a structured explanation as to why nonlinear KFs often underestimate the state covariance matrix. Note that an overconfident state covariance estimation can be very detrimental since it prevents the filter from correcting itself in subsequent iterations, thereby affecting the state estimation accuracy. Again, this problem is unavoidable because the actual values of  $S_k$  and  $P_{xy,k}$  are related to the actual state, which is unknown. Therefore, this problem exists for all types of nonlinear KFs as long as (10) is used to update the state covariance estimation.

In plain language, the problem of the conventional framework originates from the assumption that the selected Kalman gain is optimal. However, the Kalman gain, which was optimized based on the approximated measurement function near the predicted states, generally will not minimize the trace of the covariance matrix after the update. The reason is that the states will move to a new location after the update, and the previous approximations may be inaccurate. An example of such a problem is shown in Fig. 1, where we used an EKF to estimate a plane's position. In the example, the state is the plane's position along the x-axis, and the measurement function is the altitude of the ground beneath the plane, which is a nonlinear function of the state. Due to the inaccurate approximation of the measurement function around the predicted state (the dashed purple line, “Approx. 1”, in Fig. 1), the EKF overestimates the position of the plane, and the absolute value of the state estimation error  $|\hat{x}_{k|k} - x_k|$  becomes higher after the “update” step. Meanwhile, the estimated standard deviation of the state,  $\sigma_{k|k}$ , becomes smaller after the “update” step. Such a contradiction suggests that the effectiveness of the Kalman gain is overestimated. This is visualized by the narrow yellow distribution in the “update” step in Fig. 1, which does not cover the true state.

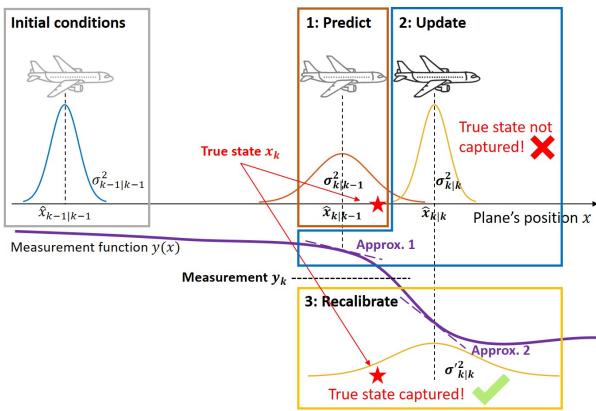


Fig. 1. The basic idea of the new framework for nonlinear KFs. The additional step we propose re-approximates the nonlinear measurement function around the updated states, making the final covariance estimation more accurate.

### 2.3 The proposed new framework for nonlinear Kalman filters

Once we fundamentally understand the problem of the conventional framework, the solution is simple. We can redo the approximation after the update and use the newly approximated measurement function to update the covariance estimation. In our new framework, this new step is called “recalibrate”. As shown in Fig. 1, with this additional step, the covariance matrix may not have the smallest possible trace after the update; instead, it will honestly reflect the effect of the Kalman gain on the state covariance matrix. This is visualized by the wider yellow distribution in the “recalibrate” step, which now covers the true state.

Mathematically, “recalibrate” means to re-approximate the measurement functions around the updated states  $\hat{x}_{k|k}$  and re-estimate the matrices  $P_{y,k}$ ,  $S_k$  and  $P_{xy}$ . We denote these re-approximated covariance matrices as  $P_{y,k|k}$ ,  $S_{k|k}$ , and  $P_{xy,k|k}$  since they are all estimated based on the updated states. In the new framework, the state covariance matrix is updated by replacing  $S_k$  with  $S_{k|k}$  and  $P_{xy,k}$  with  $P_{xy,k|k}$  in (11):

$$\begin{aligned} P_{k|k,new} &= P_{k|k-1} + K_{est} S_{k|k} K_{est}^T \\ &\quad - P_{xy,k|k} K_{est}^T - K_{est} P_{xy,k|k}^T \\ &= P_{k|k-1} + P_{xy,k|k-1} S_{k|k-1}^{-1} S_{k|k} S_{k|k-1}^{-1} P_{xy,k|k-1}^T \\ &\quad - P_{xy,k|k} S_{k|k-1}^{-1} P_{xy,k|k-1}^T - P_{xy,k|k-1} S_{k|k-1}^{-1} P_{xy,k|k} \end{aligned} \quad (18)$$

Note that the proposed new framework selects the same Kalman gain and does the same state update as the conventional one; the only difference is that the state covariance update equation is now a direct approximation of (11), which is the actual state covariance regardless of the selection of the Kalman gain  $K$ . Intuitively, this can prevent the KF from overconfidence because the equation no longer relies on the problematic assumption that the Kalman gain is optimal. To explain this advantage more rigorously, we consider the newly approximated covariance matrices  $P_{y,k|k}$ ,  $S_{k|k}$ , and  $P_{xy,k|k}$  as noisy versions of  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$ . Similar to our previous assumptions about  $P_{y,k|k-1}$ ,  $S_{k|k-1}$ , and  $P_{xy,k|k-1}$ , we assume that  $P_{xy,k|k}$  is a random matrix,  $P_{y,k|k}$  is a random positive semidefinite matrix, and  $S_{k|k}$  is a random positive definite matrix. Additionally, we assume that the “noises” that arise in these approximations are unbiased, so these new random matrices should also satisfy:

$$\begin{cases} \mathbb{E}[P_{y,k|k}] = P_{y,k} \\ \mathbb{E}[S_{k|k}] = S_k \\ \mathbb{E}[P_{xy,k|k}] = P_{xy,k} \end{cases} \quad (19)$$

With these additional definitions and assumptions, we

are able to give the following theorem about the advantage of the new framework:

**Theorem 2** Assume that  $\mathbb{E}(P_{xy,k|k-1}) = \mathbb{E}(P_{xy,k|k}) = P_{xy,k}$  and  $\mathbb{E}(S_{k|k-1}) = \mathbb{E}(S_{k|k}) = S_k$ . If  $P_{xy,k|k-1}$ ,  $P_{xy,k|k}$ ,  $S_{k|k-1}$ , and  $S_{k|k}$  are all independent from each other,  $P_{k|k,new}$  calculated in (18) is an unbiased estimation of  $P_{k|k,ac}$ . Namely, we have:

$$\mathbb{E}(P_{k|k,new} - P_{k|k,ac}) = 0 \quad (20)$$

**PROOF.** With the assumptions made in Theorem 2, we have:

$$\begin{aligned} & \mathbb{E}(P_{k|k,new} - P_{k|k,ac}) \\ &= \mathbb{E}[P_{xy,k|k-1}S_{k|k-1}^{-1}(S_{k|k} - S_k)S_{k|k-1}^{-1}P_{xy,k|k-1}^T] \\ &\quad - \mathbb{E}[(P_{xy,k|k} - P_{xy,k})S_{k|k-1}^{-1}P_{xy,k|k-1}^T] \\ &\quad - \mathbb{E}[P_{xy,k|k-1}S_{k|k-1}^{-1}(P_{xy,k|k} - P_{xy,k})^T] \\ &= 0 \end{aligned} \quad (21)$$

The last equality equals zero because the terms in parentheses are zero, due to the unbiased (*e.g.*  $\mathbb{E}(S_{k|k}) = S_k$ ) and independence assumptions.

Theorem 2 suggests that the proposed new framework can generally prevent the filter from being overconfident if the approximation errors of the measurement function at the predicted and updated states are independent of each other. Admittedly, this assumption about the independence of the approximation errors is very hard to verify and may not hold in practice. Nonetheless, the theorem does explain why recalibrating the covariance matrices  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$  after the state update is advantageous for accurate state covariance estimation. Specifically, the advantage comes from our replacement of (10) with (18), which highlights the fundamental idea of the new framework. That is, instead of reducing the updated state covariance matrix to the smallest possible value, we re-approximate the system after the state update to estimate the actual effect of the Kalman gain more accurately. In fact, in the new framework, the trace of the state covariance matrix may increase after the state update and recalibration if the system is strongly nonlinear and the predicted states have high variances, just like the example shown in Fig. 1. This phenomenon means that the update fails to make the estimation more accurate and should be withdrawn. To address this situation, we also add a “back out” step to the framework, which allows the state estimator to back out of the update when the update is unhelpful. The pseudo-code of the new framework is shown in Algorithm 1. Note that the new framework degrades to the conventional framework when the “recalibrate” and “back out” steps are replaced with (10).

---

**Algorithm 1** The proposed framework for nonlinear Kalman filters

---

**Input:** Process noise covariance matrix  $\mathbf{Q}_k$ , Measurement noise covariance matrix  $\mathbf{R}_k$ , state transition function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ , measurement function  $\mathbf{h}(\mathbf{x})$ , system inputs  $\mathbf{u}_k$ , measurements  $\mathbf{z}_k$

**Initialization:**

- 1:  $\hat{\mathbf{x}}_{0|0} = \mathbb{E}[\mathbf{x}_0]$
- 2:  $\mathbf{P}_{0|0} = \mathbb{E}[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T]$
- 3: **for** every time step  $k$  **do**

**Predict:**

- 4: Estimate  $\hat{\mathbf{x}}_{k|k-1}$  and  $\mathbf{P}_{k|k-1}$   
(based on  $\hat{\mathbf{x}}_{k-1|k-1}$ ,  $\mathbf{P}_{k-1|k-1}$ ,  $\mathbf{f}$ ,  $\mathbf{u}_k$ , and  $\mathbf{Q}_k$ )

**Update:**

- 5: Estimate  $\hat{\mathbf{y}}_{k|k-1}$ ,  $\mathbf{P}_{xy,k|k-1}$  and  $\mathbf{P}_{y,k|k-1}$   
(based on  $\hat{\mathbf{x}}_{k|k-1}$ ,  $\mathbf{P}_{k|k-1}$ , and  $\mathbf{h}$ )
- 6:  $\mathbf{S}_{k|k-1} = \mathbf{P}_{y,k|k-1} + \mathbf{R}_k$
- 7:  $\mathbf{K} = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1}$
- 8:  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1})$

**Recalibrate:**

- 9: Estimate  $\mathbf{P}_{xy,k|k}$  and  $\mathbf{P}_{y,k|k}$   
(based on  $\hat{\mathbf{x}}_{k|k}$ ,  $\mathbf{P}_{k|k-1}$ , and  $\mathbf{h}$ )
- 10:  $\mathbf{S}_{k|k} = \mathbf{P}_{y,k|k} + \mathbf{R}_k$
- 11:  $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} + \mathbf{K} \mathbf{S}_{k|k} \mathbf{K}^T - \mathbf{P}_{xy,k|k} \mathbf{K}^T - \mathbf{K} \mathbf{P}_{xy,k|k}^T$

**Back out:**

- 12: **if**  $\text{tr}(\mathbf{P}_{k|k}) > \text{tr}(\mathbf{P}_{k|k-1})$  **then**
- 13:      $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}$
- 14:      $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1}$
- 15: **end if**

16: **end for**

where,  $\hat{\mathbf{x}}$  is the estimated states,  $\mathbf{P}$  is the states’ covariance matrix,  $\hat{\mathbf{y}}$  is the estimated measurements,  $\mathbf{P}_{xy}$  is the covariance between the states and the measurements,  $\mathbf{P}_y$  is the covariance of the estimated measurements,  $\mathbf{S}$  is the innovation (or residual) covariance,  $\mathbf{K}$  is the Kalman gain. The subscript  $k|k-1$  represents the estimated value before the state update, and  $k|k$  represents the estimated value after the state update.

---

It is worth mentioning that when the measurement functions are all linear, they would require no approximation. Or, to be more precise, the approximation of the measurement functions would become independent of the states. In this case, the approximations of  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$  will not change before and after the recalibration, and our framework will be equivalent to the conventional framework.

To give readers an intuitive understanding of the effect of the “recalibrate” and “back out” steps, in Fig. 2, we showed an example of using different types of nonlinear KFs to do the state update given the measurement  $y = x^3/3 - x^2/8 - x + 1.5383 = 0$ . The measurement function is represented by the thick gray curve in the figure. The predicted state is  $x_{1|0} = 0$ , its variance is  $1.5^2$ , and the standard deviation of the measurement is

$\sigma_y = 0.01$ . This state estimation problem is challenging because the derivative of the measurement function at the predicted state is negative, while the actual state is to the left of the predicted state. The state and covariance estimations made by different algorithms shown in different rows in the figure. Generally, the conventional framework always underestimates the actual state estimation error, which can be visualized by the narrow standard deviation estimations in Fig. 2 that cannot capture the true state. By contrast, the state covariance estimation made by the new framework is much more accurate. Additionally, the new framework is aware that the state estimation becomes less accurate after the update because the estimated variance increases. Therefore, after the “recalibrate” step, the “back out” step will be triggered, and the new framework will withdraw the update and use the predicted state and the predicted state variance as the final output.

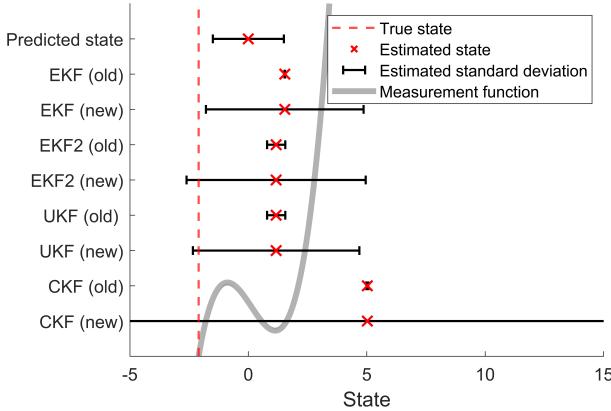


Fig. 2. The comparison of the covariance estimation after the state update and recalibration using different frameworks and different types of nonlinear Kalman filters.

#### 2.4 Relationship to Iterated Kalman Filter

The new framework described above may remind the readers of the iterated Kalman filter (IKF), which also re-approximates the measurement function after calculating the Kalman gain [19–21]. However, IKF is fundamentally different from the new framework in three ways. Firstly, IKF redoing the approximation and recalculates the Kalman gain iteratively until convergence. However, such a convergence is not guaranteed, so the algorithm is less robust and often requires finetuning the step size of each iteration [22]. Even if the iteration converges, the convergence may take a very long time. By contrast, the new framework requires no iteration, and the re-approximation is only done once. Secondly, IKF was initially only used on EKF, and the deduction of iterated EKF (IEKF) requires an explicit approximation of the measurement functions (rows 9 and 10 in Algorithm 2), which is only possible for EKF and EKF2. Although a few attempts have been made to extend IKF

to UKF and CKF [20, 21, 23, 24], a consensus on how to do this robustly has not been reached. Therefore, it is not easy to implement IKF on all types of nonlinear KFs while guaranteeing that the estimator can converge in most cases. Our framework, on the other hand, does not have this problem since there is no convergence issue. Thirdly, and most importantly, the basic idea behind IKF is still to find the optimal Kalman gain and not to estimate the covariance matrix accurately. Namely, IKF always updates the states and covariance matrix simultaneously using (10), resulting in overconfident state estimations. When the algorithm cannot converge properly, the estimation can be even worse because the estimated  $P_{y,k}$ ,  $S_k$ , and  $P_{xy,k}$  can be less accurate, which can lead to a higher degree of overconfidence. In general, in terms of making a more accurate estimation, finding the optimal Kalman gain is not as important as making the covariance estimation accurate, as we will show later in the simulation results.

Since IKF also uses (10) to update the state covariance matrix, it can also be considered as another framework for nonlinear KFs that is similar to the conventional framework. A natural question is: can we combine the new framework with the IKF to get a higher estimation accuracy? Unfortunately, the answer is no because a prerequisite for the new framework to give accurate covariance estimation is that the approximations made at the predicted states ( $P_{y,k|k-1}$ ,  $S_{k|k-1}$ , and  $P_{xy,k|k-1}$ ) are independent of the approximations made at the updated states ( $P_{y,k|k}$ ,  $S_{k|k}$ , and  $P_{xy,k|k}$ ). However, after IKF converges, these two sets of approximations become the same, which violates this prerequisite in Theorem 2. For example, the IEKF algorithm shown in Algorithm 2 approximates the measurement function iteratively at  $x_{i-1}$  (rows 9–13). When the algorithm converges, in the final iteration, we have  $x_{i-1} \approx x_i = \hat{x}_{k|k}$ , meaning that the approximation of the measurement functions in the final iteration will stay the same after the state update. In fact, to the best of our knowledge, IKF-related algorithms are the only existing types of nonlinear KFs that can not get large benefits from the proposed framework. Nevertheless, for any type of nonlinear KF that can be combined with IKF, its accuracy can be greatly improved when it is combined with our new framework instead. In this sense, we can still say that all existing types of nonlinear KFs can benefit from the proposed framework.

### 3 Results

To demonstrate the effectiveness of the proposed framework, we validated it in five different applications suitable for nonlinear KFs. A brief description of the five applications is presented in Table 1, and a detailed description of the system models and parameter setup can be found in Appendix B. For each scenario, we combined the old and proposed framework with four different

**Algorithm 2** The iterated extended Kalman filter with the old framework

---

**Input:** Process noise covariance matrix  $\mathbf{Q}_k$ , Measurement noise covariance matrix  $\mathbf{R}_k$ , state transition function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ , measurement function  $\mathbf{h}(\mathbf{x})$ , system inputs  $\mathbf{u}_k$ , measurements  $\mathbf{z}_k$

**Initialization:**

- 1:  $\hat{\mathbf{x}}_{0|0} = \mathbb{E}[\mathbf{x}_0]$
- 2:  $\mathbf{P}_{0|0} = \mathbb{E}[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T]$
- 3: **for** every time step  $k$  **do**

**Predict:**

- 4:  $\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}$
- 5:  $\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1})$
- 6:  $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$

**Update:**

- 7:  $\mathbf{x}_0 = \hat{\mathbf{x}}_{k|k-1}$
- 8: **for**  $i = 1$  to 1000 **do**
- 9:      $\mathbf{H}_{k|k-1} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} |_{\mathbf{x}_{i-1}}$
- 10:     $\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}(\mathbf{x}_{i-1}) + \mathbf{H}_{k|k-1}(\mathbf{x}_{i-1} - \hat{\mathbf{x}}_{k|k-1})$
- 11:     $\mathbf{P}_{xy,k|k-1} = \mathbf{P}_{k|k-1} \mathbf{H}_{k|k-1}^T$
- 12:     $\mathbf{P}_{y,k|k-1} = \mathbf{H}_{k|k-1} \mathbf{P}_{k|k-1} \mathbf{H}_{k|k-1}^T$
- 13:     $\mathbf{S}_{k|k-1,i} = \mathbf{P}_{y,k|k-1} + \mathbf{R}_k$
- 14:     $\mathbf{K}_i = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1,i}^{-1}$
- 15:     $\mathbf{x}_i = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_i(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1})$
- 16:    **if**  $i > 1$  and  $\|\mathbf{x}_i - \mathbf{x}_{i-1}\| > \|\mathbf{x}_{i-1} - \mathbf{x}_{i-2}\|$  **then**
- 17:        $\mathbf{S}_{k|k-1,i} = \mathbf{S}_{k|k-1,i-1}$
- 18:        $\mathbf{K}_i = \mathbf{K}_{i-1}$
- 19:        $\mathbf{x}_i = \mathbf{x}_{i-1}$
- 20:       **break**
- 21:    **end if**
- 22:    **if**  $\max_j |1 - \mathbf{x}_i(j)/\mathbf{x}_{i-1}(j)| < 0.001$  **then**
- 23:       **break**
- 24:    **end if**
- 25:   **end for**
- 26:    $\mathbf{S}_{k|k-1} = \mathbf{S}_{k|k-1,i}$
- 27:    $\mathbf{K} = \mathbf{K}_i$
- 28:    $\hat{\mathbf{x}}_{k|k} = \mathbf{x}_i$
- 29:    $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K} \mathbf{S}_{k|k-1} \mathbf{K}^T$
- 30: **end for**

---

Note that rows 16–21 are added to prevent the state estimations from diverging. Namely, when the sign of divergence is detected (row 16), the newest updates of the states, Kalman gain, and measurement residual covariance matrix are withdrawn (rows 17–19). If such a withdrawal is not made, we found in the simulation that the estimation accuracy can be much lower. This characteristic also suggests that the algorithm is not very robust and stable.

types of nonlinear KFs: EKF, EKF2, UKF, and CKF. We also implemented the IEKF for comparison. The detailed EKF, EKF2, UKF, and CKF algorithms used in the paper can be found in Algorithms 3–5 in Appendix C. All these nine ( $4 + 4 + 1$ ) nonlinear KFs are used for state estimation under different measurement noise setups (the covariance of the measurement noises differs in

different setups). To guarantee the fairness of the comparison and reduce the effect of randomness, each KF is simulated 10,000 times for each measurement noise setup, and all the KFs use the same random seed during the simulation. This means that the random initial states' errors, random process noises, and random measurement noises are the same for the  $i^{th}$  simulation of EKF and the  $i^{th}$  simulation of UKF. In other words, all these random noises were generated for the 10,000 runs *a priori* and recalled for simulating each estimation method.

The root mean squared error (RMSE) of the state estimations given by different nonlinear KFs are shown in Figs. 3–7. In all figures, different types of nonlinear KFs are represented by different colors and markers, and the old and new frameworks are represented by dotted lines and solid lines, respectively. The x-axis represents the standard deviations of the measurements. The root mean squared error is calculated from the estimation errors in 10,000 simulations. The estimation error is calculated as the difference between the estimated state and the true state at the end of the final iteration of each simulation. Note that we only showed the RMSE of two representative states for systems with more than two states. For example, in 3D target tracking, the six states are the object's speed and position along the three axes, and Fig. 3 examines the object's speed and position estimation accuracy along the x-axis. As can be seen from the five figures, the simulation results are very similar to each other. Therefore, only the results of the first task, 3D target tracking, are analyzed in depth in this section.

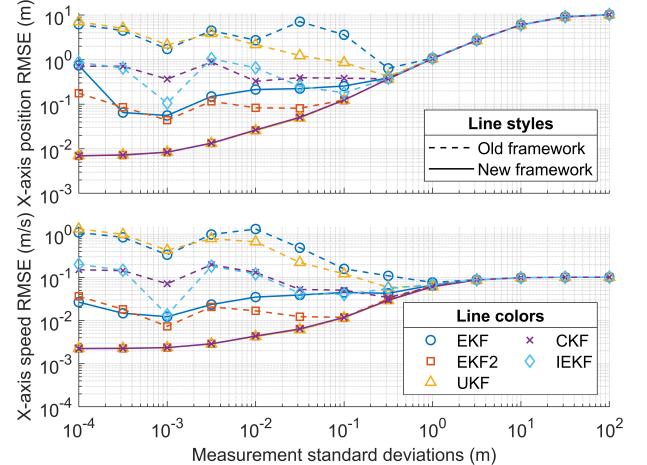


Fig. 3. The root mean squared error of the state estimations under different measurement noise setups (3D target tracking).

As shown in Fig. 3–7, the proposed framework (solid lines) significantly improves the accuracy of all types of nonlinear KFs. The improvement becomes stronger as the measurement noise becomes smaller. This result aligns with our expectations because the proposed framework only changes how the measurements are used

Table 1

A brief description of the five applications of nonlinear Kalman filters investigated in this paper.

Applications	3D target tracking	Terrain-referenced navigation	Synchronous generator state estimation	Pendulum state estimation	Battery state estimation
Number of states	6	2	4	2	3
Linear state transition function?	Yes	Yes	No	No	No
Number of measurements	2	1	1	1	1
Linear measurement functions?	No	No	No	No	No
Number of inputs	3	2	3	1	1
Number of iterations	30	100	100	100	180

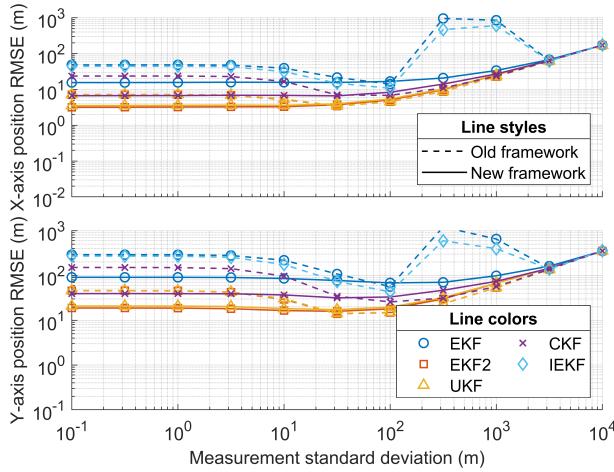


Fig. 4. The root mean squared error of the state estimations under different measurement noise setups (terrain-referenced navigation).

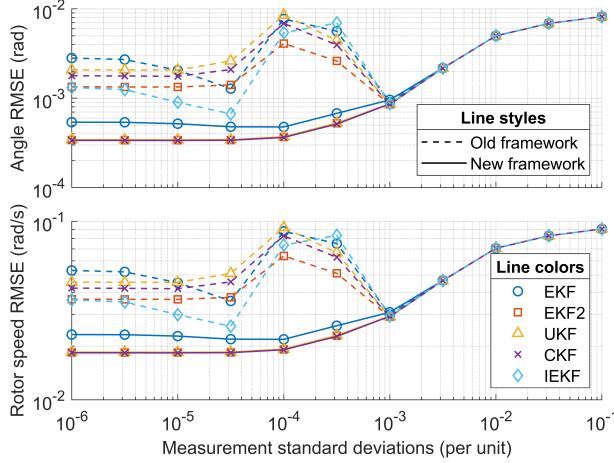


Fig. 5. The root mean squared error of the state estimations under different measurement noise setups (synchronous generator state estimation).

in the nonlinear KF, and the estimations will rely more on the measurements instead of system knowledge as the measurement noise decreases. For all these applications, when the standard deviations of the measurements are small, the RMSE of all four types of nonlinear KFs can

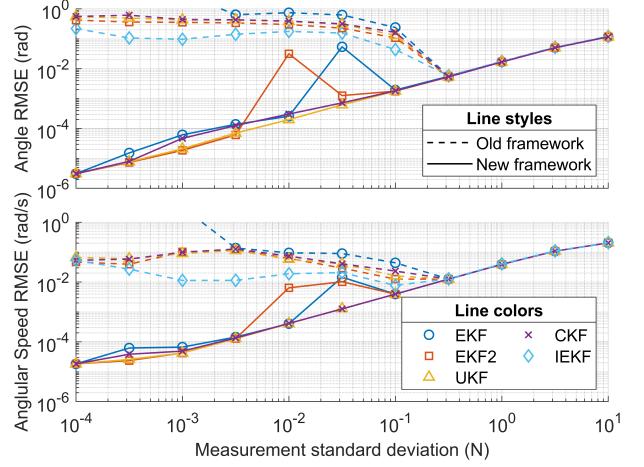


Fig. 6. The root mean squared error of the state estimations under different measurement noise setups (pendulum state estimation).

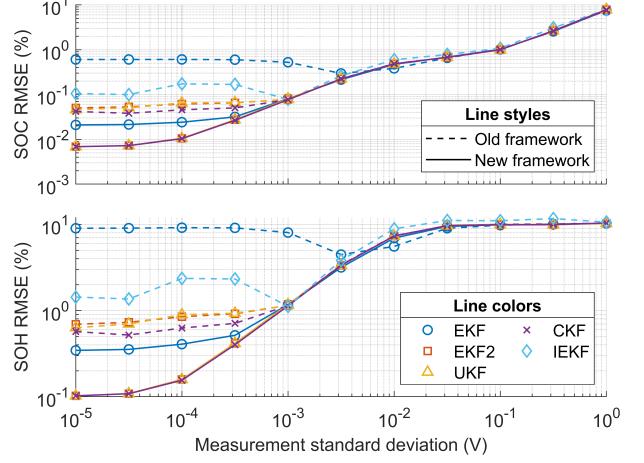


Fig. 7. The root mean squared error of the state estimations under different measurement noise setups (battery state estimation).

usually be reduced by more than an order of magnitude, which is a significant improvement.

Another interesting observation of Figs. 3–7 is that the RMSE of EKF2, UKF, and CKF becomes quite similar

after the new framework is used. This phenomenon can be explained by the fact that all three methods consider the first and second-order terms in the Taylor expansion of the nonlinear state transition and measurement functions in certain ways [13]. On the other hand, EKF is generally slightly worse than the other three types of nonlinear KFs because it only considers the first-order terms in the Taylor expansion.

By comparing the performance of IEKF and EKF (with and without the new framework) in Figs. 3–7, we can see that IEKF does perform better than EKF with the old framework, yet it is almost always worse than EKF with the new framework despite requiring longer computational time. As we mentioned in the Introduction, the main reason for such a difference is that IEKF still gives overconfident covariance estimation, especially when it fails to converge, which happens quite often in practice.

As we mentioned in the Introduction, the fundamental idea of the proposed framework is to make a more accurate covariance estimation by re-approximating the measurement functions after the state update. To validate if the covariance estimation really becomes more accurate in the new framework, we compared the actual and estimated RMSE of the x-axis position (the first state in the first system) in the old and new framework in Fig. 8. In the figure, for each type of nonlinear KF (represented by different line colors) and framework (the old and new frameworks are respectively shown in the top and bottom subfigures), the closeness between the actual RMSE (the solid line) and estimated RMSE (dotted line) represents the accuracy of the state covariance estimation. Apparently, when the standard deviations of the two measurements are lower than 0.1 meters, the conventional framework always significantly underestimates the covariance, sometimes by several orders of magnitude, regardless of the type of nonlinear KFs used. On the other hand, for the new framework, only the RMSE of EKF is underestimated, and the covariance estimation of other types of nonlinear KFs matches the actual value very well. This, again, can be explained by the fact that EKF neglects the second and higher-order information of the measurement functions, so its covariance estimation is less accurate than other methods. Nevertheless, the covariance estimation of EKF is still much more precise when combined with the new framework. Another interesting observation of Fig. 8 is that the estimated RMSE of the x-axis position (the dashed lines) is very similar for different combinations of frameworks and KFs. This suggests that the accuracy of the estimation mainly relates to whether the covariance estimation is accurate or not, which leads to the conclusion mentioned at the end of the Introduction — to achieve a more accurate estimation, finding the optimal Kalman gain is not as important as accurately estimating the covariance.

Another important metric for evaluating the performance of different filters is the convergence character-

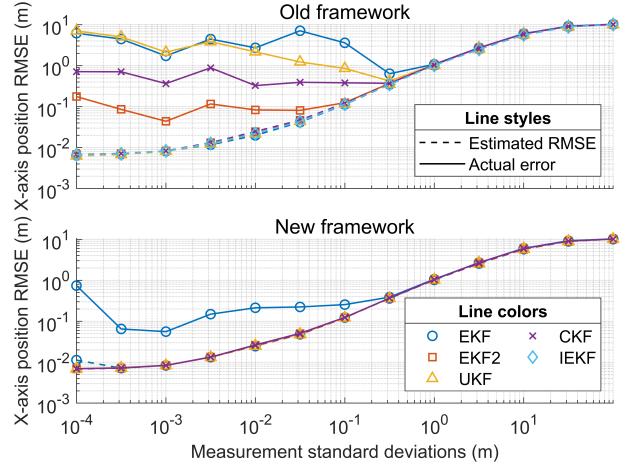


Fig. 8. The actual and estimated root mean squared error of the x-axis position estimation.

istics. We visualized this metric by fixing the standard deviations of the two measurements to 0.01 m and plotted the RMSE of the state estimations after different numbers of iterations. The results are shown in Fig. 9. From the figure, we can see that all types of nonlinear KFs can converge faster after being combined with the new framework. This result suggests that the proposed framework is very suitable for applications that require high convergence speed.

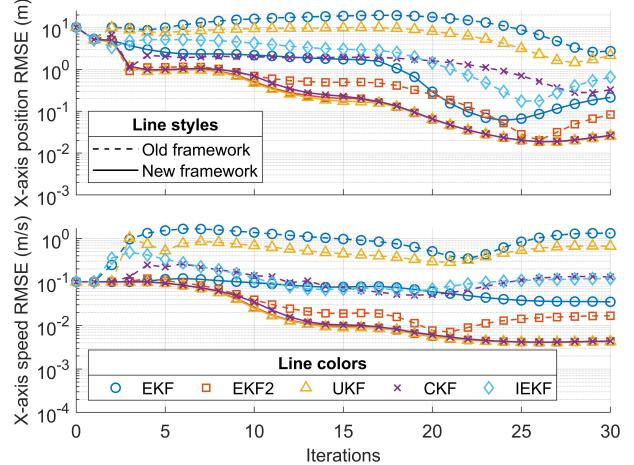


Fig. 9. The root mean squared error of the state estimations after different numbers of iterations.

Since the extra steps introduced in the new framework take time, it is also important to compare the computational time of the two frameworks. The results are summarized in Fig. 10. Note that the average runtime shown in the figure is calculated from over 100,000 simulations run on the same computer. Specifically, for each combination of nonlinear KF and framework, we recorded the method's runtime whenever it was used and calculated the average. After the average runtime is calculated, the value is normalized with respect to the average runtime

of the conventional EKF for the same application. In other words, the normalized average runtime of “EKF (old)” always equals one in Fig. 10.

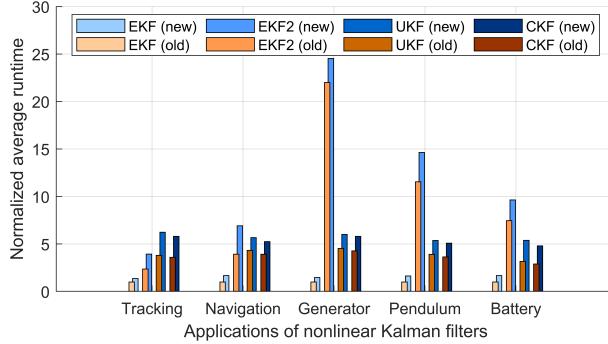


Fig. 10. Comparison of the computational time of different methods and frameworks.

From the result, we can see that the new framework generally increases the computational time by 10–90%, varying for different nonlinear systems. Namely, when the system has linear state transition functions and nonlinear measurement functions (which is true for the first two applications in the table), the “update” step should be accountable for most of the computational time in the conventional framework and re-approximating the measurement functions can almost double the runtime in the worst case. However, noticing that the new framework can reduce the estimation error by more than an order of magnitude when the measurement noise is small, we can say that the extra computational complexity is generally worth it. Another thing to note is that the combination of EKF and the new framework requires significantly less computational time than the combination of any other nonlinear KFs and the old framework. At the same time, the former’s accuracy is generally higher than the latter in the applications we examined in this paper. Therefore, even in situations where the computational time is of major concern, the proposed framework can still be helpful because the new EKF can outperform the old EKF2, UKF, and CKF, in addition to saving a lot of computational time.

## 4 Conclusion

In this paper, we proposed a new framework that can be applied to all types of nonlinear KFs. The new framework re-approximates the measurement functions after the state update, solving the problem of overconfident state covariance estimation, which is an inherent problem in the conventional framework for nonlinear KFs. The new framework was tested in five different applications, reducing the estimation errors by more than an order of magnitude when the measurement noise is low and significantly reducing the convergence time of the algorithm. Considering that all types of sensors will become increasingly more accurate in the future, the pro-

posed framework is expected to bring greater benefits for all kinds of nonlinear KFs over time.

The main limitation of this work is that the proposed framework can only be used in discrete-time nonlinear KFs. For continuous-time nonlinear KFs, the state estimations change continuously over time, so the problem of overconfident covariance estimation does not exist. However, all digital sensors in real life give discrete-time measurements. Even though it is possible to make the signals continuous through interpolation, the continuous-time state estimation problem sometimes includes several partial differential equations, so it is hard and takes a long time to solve [25]. Therefore, most applications of nonlinear KFs in real life use the discrete-time form, and they can all benefit from the new framework proposed in this paper.

## Acknowledgements

This material is based upon work supported, in part, by the National Science Foundation under Grant No. 1847177.

## References

- [1] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2005.
- [2] Dah-Jing Jwo and Ta-Shun Cho. Critical remarks on the linearised and extended kalman filters with geodetic navigation examples. *Measurement*, 43(9):1077–1089, 2010.
- [3] Thao Tran Phuong, Kiyoshi Ohishi, Chowarat Mitsantisuk, Yuki Yokokura, Kouhei Ohnishi, Roberto Oboe, and Asif Sabanovic. Disturbance observer and kalman filter based motion control realization. *IEEJ Journal of Industry Applications*, 7(1):1–14, 2018.
- [4] Emery N Brown and Christopher H Schmid. [9] application of the kalman filter to computational problems in statistics. In *Methods in enzymology*, volume 240, pages 171–181. Elsevier, 1994.
- [5] Michael Roth, Gustaf Hendeby, Carsten Fritzsche, and Fredrik Gustafsson. The ensemble kalman filter: a signal processing perspective. *EURASIP Journal on Advances in Signal Processing*, 2017:1–16, 2017.
- [6] Kumar Pakki Bharani Chandra and Da-Wei Gu. *Nonlinear Filtering: Methods and Applications*. Springer, 2019.
- [7] A Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
- [8] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. Spie, 1997.
- [9] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.
- [10] Ienkaran Arasaratnam and Simon Haykin. Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6):1254–1269, 2009.

- [11] Ienkaran Arasaratnam, Simon Haykin, and Robert J Elliott. Discrete-time nonlinear filtering algorithms using gauss–hermite quadrature. *Proceedings of the IEEE*, 95(5):953–977, 2007.
- [12] Ienkaran Arasaratnam and Simon Haykin. Square-root quadrature kalman filtering. *IEEE Transactions on Signal Processing*, 56(6):2589–2593, 2008.
- [13] Fredrik Gustafsson and Gustaf Hendeby. Some relations between extended and unscented kalman filters. *IEEE Transactions on Signal Processing*, 60(2):545–555, 2011.
- [14] Mark R Morelande and Angel F Garcia-Fernandez. Analysis of kalman filter approximations for nonlinear measurements. *IEEE Transactions on Signal Processing*, 61(22):5477–5484, 2013.
- [15] Shiyuan Wang, Wanli Wang, Badong Chen, and K Tse Chi. Convergence analysis of nonlinear kalman filters with novel innovation-based method. *Neurocomputing*, 289:188–194, 2018.
- [16] Kumar Pakki Bharani Chandra and Da-Wei Gu. Nonlinear filtering. *Cham, Switzerland: Springer*, 2019.
- [17] Daisuke Hotta and Yoichiro Ota. Why does enkf suffer from analysis overconfidence? an insight into exploiting the ever-increasing volume of observations. *Quarterly Journal of the Royal Meteorological Society*, 147(735):1258–1277, 2021.
- [18] Kai Xiong, HY Zhang, and CW Chan. Performance evaluation of ukf-based nonlinear filtering. *Automatica*, 42(2):261–270, 2006.
- [19] Bradley M Bell and Frederick W Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.
- [20] Ronghui Zhan and Jianwei Wan. Iterated unscented kalman filter for passive target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):1155–1163, 2007.
- [21] Jing Mu and Yuan-li Cai. Iterated cubature kalman filter and its application. In *2011 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, pages 33–37. IEEE, 2011.
- [22] Jindřich Havlík and Ondřej Straka. Performance evaluation of iterated extended kalman filter with variable step-length. In *Journal of Physics: Conference Series*, volume 659, page 012022. IOP Publishing, 2015.
- [23] Lubin Chang, Baiqing Hu, Guobin Chang, and An Li. Marginalised iterated unscented kalman filter. *IET Control Theory & Applications*, 6(6):847–854, 2012.
- [24] Guobin Chang, Tianhe Xu, and Qianxin Wang. Alternative framework for the iterated unscented kalman filter. *IET Signal Processing*, 11(3):258–264, 2017.
- [25] Simo Sarkka. On unscented kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Transactions on automatic control*, 52(9):1631–1641, 2007.
- [26] Youngjoo Kim, Hyochoong Bang, et al. Introduction to kalman filter and its applications. *Introduction and Implementations of the Kalman Filter*, 1:1–16, 2018.
- [27] Esmaeil Ghahremani and Innocent Kamwa. Online state estimation of a synchronous generator using unscented kalman filter from phasor measurements units. *IEEE Transactions on Energy Conversion*, 26(4):1099–1108, 2011.
- [28] Shida Jiang, Junzhe Shi, Manashita Borah, and Scott Moura. Weaknesses and improvements of the extended kalman filter for battery state-of-charge and state-of-health estimation. In *2024 American Control Conference (ACC)*, pages 1441–1448. IEEE, 2024.

## A Supplementary proof for Theorem 1

**Lemma 3** Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be an underlying probability space. Assume the random column vectors

$$x : \Omega \rightarrow \mathbb{R}^n, y : \Omega \rightarrow \mathbb{R}^n$$

satisfy  $\mathbb{E}[||x||^2] < \infty$  and  $\mathbb{E}[||y||^2] < \infty$  (i.e.,  $x, y \in L^2(\Omega; \mathbb{R}^n)$ ). Then, the operation on  $L^2(\Omega; \mathbb{R}^n)$  by

$$f(u, v) = \mathbb{E}[u^T v]$$

is an inner product on  $L^2(\Omega; \mathbb{R}^n)$ .

**PROOF.** We start by proving that  $f$  is well-defined on  $L^2$ . For any  $u, v \in L^2$ , both  $\mathbb{E}[u^T u]$  and  $\mathbb{E}[v^T v]$  are finite. Therefore,  $\mathbb{E}[(u+v)^T (u+v)]$  should also be finite. Notice that

$$\mathbb{E}[(u+v)^T (u+v)] = \mathbb{E}[u^T u] + \mathbb{E}[v^T v] + 2\mathbb{E}[u^T v],$$

we can conclude that  $\mathbb{E}[u^T v]$  should also be finite. Therefore,  $f$  is well-defined on  $L^2$ . Next, to prove that  $f$  is an inner product, we need to show that it satisfies symmetry, linearity in the first argument, and positive-definiteness. Symmetry is satisfied because  $\mathbb{E}[u^T v] = \mathbb{E}[v^T u]$ . Linearity is satisfied because for any scalars  $a, b$  and  $x, y, z \in L^2$ ,

$$\mathbb{E}[(ax+by)^T z] = a\mathbb{E}[x^T z] + b\mathbb{E}[y^T z].$$

Positive-definiteness is satisfied because for any  $x \in L^2$ ,  $\mathbb{E}[x^T x] \geq 0$ , and the equality sign only holds when  $x \equiv 0$ .

**Lemma 4** Let  $\mathbf{S}$  be a random symmetric and positive definite matrix whose mean is the identity matrix  $\mathbf{I}$ . Let  $\mathbf{v}$  be a random vector whose mean is  $\mathbf{v}_0$ . The following inequality holds:

$$\mathbb{E}(\mathbf{v}^T \mathbf{S}^{-1} \mathbf{v}) \geq \mathbf{v}_0^T \mathbf{v}_0 \quad (\text{A.1})$$

Note that the equality sign only holds when  $\mathbf{S}^{-1} \mathbf{v}$  is constant.

**PROOF.** Let  $\mathbf{x} = \mathbf{S}^{-\frac{1}{2}} \mathbf{v}, \mathbf{y} = \mathbf{S}^{\frac{1}{2}} \mathbf{v}_0$ . From the definitions,

$$\begin{cases} \mathbb{E}[\mathbf{x}^T \mathbf{x}] = \mathbb{E}[\mathbf{v}^T \mathbf{S}^{-1} \mathbf{v}] \\ \mathbb{E}[\mathbf{y}^T \mathbf{y}] = \mathbb{E}[\mathbf{v}_0^T \mathbf{S} \mathbf{v}_0] = \mathbf{v}_0^T \mathbf{v}_0 \\ \mathbb{E}[\mathbf{x}^T \mathbf{y}] = \mathbb{E}[\mathbf{v}^T \mathbf{v}_0] = \mathbf{v}_0^T \mathbf{v}_0 \end{cases} \quad (\text{A.2})$$

According to Lemma 3,  $\mathbb{E}[\mathbf{x}^T \mathbf{y}]$  is an inner product on  $L^2(\Omega; \mathbb{R}^n)$ . Applying Cauchy–Schwarz inequality, we have:

$$\mathbb{E}[\mathbf{v}^T \mathbf{S}^{-1} \mathbf{v}] \geq \frac{(\mathbf{v}_0^T \mathbf{v}_0)^2}{\mathbf{v}_0^T \mathbf{v}_0} \quad (\text{A.3})$$

which completes the proof. Note that the equality holds only when  $\mathbf{S}^{-1} \mathbf{v}$  is constant.

## B The detailed simulation setup

This paper uses five different applications of nonlinear KFs to test the effectiveness of the new framework: 3D target tracking, terrain-referenced navigation, synchronous generator state estimation, pendulum state estimation, and battery state estimation.

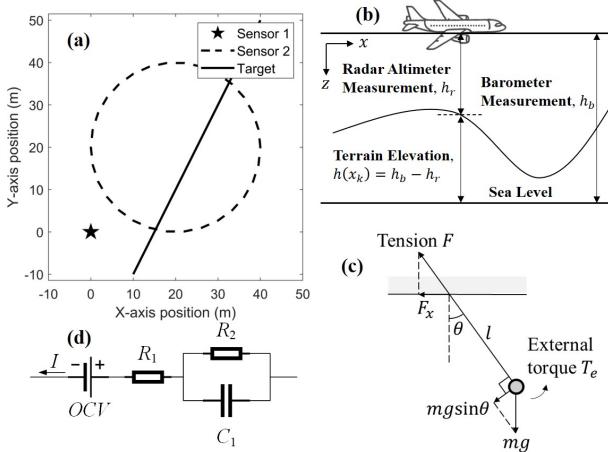


Fig. B.1. Examples of nonlinear systems used in this paper for validation. (a) 3D object tracking. (b) Terrain-referenced navigation. (c) Pendulum state estimation. (d) Battery state estimation.

### B.1 3D target tracking

This example is adapted from Example 3.3.1 in [26]. As shown in Fig. B.1(a), a target moves in 3D space, and two sensors are used to track its location. The first sensor is at the origin, and the second sensor moves in a circle. The second sensor's location  $(x_{s1}, x_{s2}, x_{s3})$  at step  $k = 1, 2, \dots, 30$  is represented by:

$$\begin{cases} x_{s1,k} = 20 + 20 \cos\left(\frac{(k-1)\pi}{15}\right) \\ x_{s2,k} = 20 + 20 \sin\left(\frac{(k-1)\pi}{15}\right) \\ x_{s3,k} = 0 \end{cases} \quad (\text{B.1})$$

The states are the target's positions  $(x_1, x_2, x_3)$  and speeds  $(v_1, v_2, v_3)$  along the three axes. The measurements  $y_1, y_2$  are the distances between the target and the two sensors. The inputs are the target's accelerations along the three axes, which are assumed to be zero in this example. When the time interval is set to  $\Delta t = 1\text{s}$ , the system's state transition and measurement functions can be written as (B.2, B.3). Note that the noises are

omitted in the equations.

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ v_{1,k} \\ v_{2,k} \\ v_{3,k} \end{bmatrix} = \begin{bmatrix} x_{1,k-1} + v_{1,k-1} \\ x_{2,k-1} + v_{2,k-1} \\ x_{3,k-1} + v_{3,k-1} \\ v_{1,k-1} \\ v_{2,k-1} \\ v_{3,k-1} \end{bmatrix} \quad (\text{B.2})$$

$$\begin{bmatrix} y_{1,k} \\ y_{2,k} \end{bmatrix} = \begin{bmatrix} \sqrt{\sum_{i=1}^3 x_{i,k}^2} \\ \sqrt{\sum_{i=1}^3 (x_{i,k} - x_{si,k})^2} \end{bmatrix} \quad (\text{B.3})$$

The true initial states of the target are  $x_{1,0} = 10, x_{2,0} = -10, x_{3,0} = 50, v_{1,0} = 1, v_{2,0} = 2, v_{3,0} = 0$ ; the initial state covariance matrix  $P_0$  is diagonal, and its diagonal elements are 100, 100, 100, 0.01, 0.01, and 0.01. The initial state estimations are generated randomly from the multivariate Gaussian distribution with covariance  $P_0$  and mean equal to the true initial states. The process noise covariance matrix is also diagonal, and its diagonal values are 0, 0, 0,  $10^{-6}$ ,  $10^{-6}$ , and  $10^{-6}$ .

### B.2 Terrain-referenced navigation

This example is adapted from Example 3.3.2 in [26]. As shown in Fig. B.1(b), a plane flies above an uneven landscape. The states are the planes' positions along the two axes, denoted as  $x_1$  and  $x_2$ . The measurement  $y_k$  is the terrain's elevation at the plane's present location, which, in practice, is calculated as the difference between the barometer and radar altimeter measurements. The plane knows the contour map in advance, so the terrain's elevation can tell the aircraft its possible locations. However, this task is difficult because all the points on a contour line have the same elevation, and system knowledge must be combined for accurate state estimations. The system inputs are the plane's speeds along the x and y axes, which are 0.5 km/s and 0, respectively. When the time interval is set to  $\Delta t = 1\text{s}$ , the system's state transition and measurement functions can be written as (B.4, B.5). Note that all the units for distance in this example are kilometers, and the noises are omitted in the equations.

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \begin{bmatrix} x_{1,k-1} + 0.5 \\ x_{2,k-1} \end{bmatrix} \quad (\text{B.4})$$

$$y_k = \sin \sqrt{\left(\frac{x_{1,k}}{40}\right)^2 + \left(\frac{x_{2,k}}{40}\right)^2} \quad (\text{B.5})$$

The true initial states of the target are  $x_{1,0} = 10 \text{ km}, x_{2,0} = 10 \text{ km}$ ; the initial state covariance matrix  $P_0$  is diagonal, and its diagonal elements are both  $1 \text{ km}^2$ . The initial state estimations are generated randomly from the multivariate Gaussian distribution with covariance  $P_0$  and mean equal to the true initial states. The process

noise covariance matrix is also diagonal, and its diagonal values are both  $0.25 \text{ m}^2$ .

### B.3 Synchronous generator state estimation

This example is adapted from a research paper [27]. The system model is precisely the same as in the paper. The states of the system are the rotor angle  $\delta$ , rotor speed  $\Delta\omega$ , the q-axis component of the voltage  $e'_q$ , and the d-axis component of the voltage  $e'_d$ . For simplicity, these states are denoted as  $x_1$  to  $x_4$ . The measurement  $y_k$  is electrical output power. The system's inputs include the mechanical input torque  $T_m$ , the steady-state internal voltage of the armature  $E_{fd}$ , and the terminal bus voltage  $V_t$ . For simplicity, these inputs are denoted as  $u_1$  to  $u_3$ . The time interval is set to  $\Delta t = 0.1 \text{ ms}$ . The input profiles are the same as in the paper, which are:

$$\begin{cases} u_{1,k} = 0.8 \\ u_{2,k} = 2.11 + 0.0002k \\ u_{3,k} = 1.002 \end{cases} \quad (\text{B.6})$$

All the model parameters are selected to be the same as in [27]. The system's state transition and measurement functions can be written as (B.7, B.8). Note that the noises are omitted from the equations.

$$\begin{aligned} \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ x_{4,k} \end{bmatrix} &= \begin{bmatrix} x_{1,k-1} \\ x_{2,k-1} \\ x_{3,k-1} \\ x_{4,k-1} \end{bmatrix} \\ &+ \begin{bmatrix} 377x_{2,k-1}\Delta t \\ \frac{\Delta t}{13} \left[ u_{1,k-1} - \frac{u_{3,k-1}x_{3,k-1} \sin x_{1,k-1}}{0.375} \right] \\ \frac{\Delta t}{0.131} \left[ u_{2,k-1} - x_{3,k-1} \right] \\ \frac{-x_{4,k-1}\Delta t}{0.0131} \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ \frac{\Delta t}{13} \left[ 0.9215u_{3,k-1}^2 \sin(2x_{1,k-1}) - 0.05x_{2,k-1} \right] \\ \frac{-4.4933\Delta t}{0.131} \left( x_{3,k-1} - u_{3,k-1} \cos x_{1,k-1} \right) \\ \frac{0.6911u_{3,k-1} \sin x_{1,k-1}\Delta t}{0.0131} \end{bmatrix} \quad (\text{B.7}) \\ y_k &= \frac{u_{3,k-1}x_{3,k} \sin x_{1,k}}{0.375} + 0.9215u_{3,k-1}^2 \sin(2x_{1,k}) \end{aligned}$$

The true initial states of the system are  $x_{1,0} = 0.4$ ,  $x_{2,0} = x_{3,0} = x_{4,0} = 0$ . The initial state covariance matrix  $P_0$  is diagonal, and its diagonal elements are  $10^{-4}$ ,  $10^{-10}$ ,  $10^{-4}$ , and  $10^{-4}$ . The initial state estimations are generated randomly from the multivariate Gaussian distribution with covariance  $P_0$  and mean equal to the true initial states. The process noise covariance matrix is also diagonal; its diagonal values are  $10^{-10}$ ,  $10^{-16}$ ,  $10^{-10}$ , and  $10^{-10}$ .

### B.4 Pendulum state estimation

As shown in Fig. B.1(c), one end of the pendulum is fixed to the ceiling with a thin rope. The system's states are the pendulum's angular position  $\theta$  and angular speed  $\omega$ . The measurement  $y_k$  is the horizontal component of tension in the rope. The system input is the external torque exerted on the pendulum, which is zero in this example. The mass of the ball is  $m = 1 \text{ kg}$ , the length of the rope is  $l = 1 \text{ m}$ , and the gravity acceleration is  $g = 9.8 \text{ m/s}^2$ . The time interval is set to  $\Delta t = 0.01 \text{ s}$ . According to Newton's law, when the time interval is small, the system's state transition and measurement functions can be written as (B.9, B.10). Note that the noises are omitted in the equations.

$$\begin{bmatrix} \omega_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \omega_{k-1} - \frac{g}{l} \sin(\theta_{k-1})\Delta t \\ \theta_{k-1} + \omega_{k-1}\Delta t \end{bmatrix} \quad (\text{B.9})$$

$$y_k = mg \cos \theta_k \sin \theta_k + ml\omega_k^2 \sin \theta_k \quad (\text{B.10})$$

The true initial states of the system are  $\omega_0 = 0$ ,  $\theta_0 = \pi/4$ . The initial state covariance matrix  $P_0$  is diagonal, and its diagonal elements are both  $(\pi/18)^2$ . The initial state estimations are generated randomly from the multivariate Gaussian distribution with covariance  $P_0$  and mean equal to the true initial states. The process noise covariance matrix is also diagonal; its diagonal values are  $10^{-10}$  and 0.

### B.5 Battery state estimation

This example is adapted from a research paper [28]. As shown in Fig. B.1(d), a battery can be modeled as several circuit components, including a voltage source OCV (stands for the cell's open-circuit voltage), an internal resistor  $R_1$ , a transfer resistor  $R_2$ , and a capacitor  $C_1$ . A battery's two most important states are its state of charge (SOC) and state of health (SOH). The SOC is the ratio of the battery's remaining capacity to its present maximum capacity, and the SOH is the ratio of the battery's present maximum capacity to its initial maximum capacity. In general, SOC describes the battery's energy level, while SOH describes the battery's aging level. The states of this system are the SOC, SOH, and capacitor voltage  $U_c$ . The measurement  $y_k$  is the battery's terminal voltage. The input is the charging current, which, in this example, is a three-level square wave shown below:

$$I_k = \begin{cases} -2, & 75 \geq k > 15 \\ 2, & 165 \geq k > 105 \\ 0, & \text{else} \end{cases} \quad (\text{B.11})$$

The battery's initial maximum capacity is  $Q_0 = 1$  Ah. The parameters in the equivalent circuit model are chosen to be  $R_1 = 0.01$ ,  $R_2 = 0.05$ ,  $1/(R_2C_1) = 0.008$ . The OCV is a function of SOC and SOH, and it can be written as:

$$OCV = \sum_{i=0}^9 \left( \frac{SOH - 0.8}{0.2} a_{100,i+1} + \frac{1 - SOH}{0.2} a_{80,i+1} \right) \quad (\text{B.12})$$

where  $a_{100} = [1390.38, -6961.31, 14760.31, -17230.92, 12055.71, -5162.75, 1330.60, -196.37, 15.60, 2.96]$ ,  $a_{80} = [813.94, -4229.96, 9345.49, -11415.38, 8396.15, -3801.07, 1043.09, -165.29, 14.28, 2.96]$ . The time interval is set to  $\Delta t = 1$  s, and the total simulation time is 180 seconds. The system's state transition and measurement functions can be written as (B.13,B.14). Note that the noises are omitted in the equations.

$$\begin{bmatrix} SOC_k \\ U_{c,k} \\ SOH_k \end{bmatrix} = \begin{bmatrix} SOC_{k-1} \\ U_{c,k-1} \exp\left(\frac{-\Delta t}{R_2 C_1}\right) \\ SOH_{k-1} \end{bmatrix} + \begin{bmatrix} -\frac{g}{l} \sin(\theta_{k-1}) \Delta t \\ [1 - \exp\left(\frac{-\Delta t}{R_2 C_1}\right)] R_2 I_{k-1} \\ 0 \end{bmatrix} \quad (\text{B.13})$$

$$y_k = OCV(SOC_k, SOH_k) + U_{c,k} + R_1 I_k \quad (\text{B.14})$$

The true initial states of the system are  $SOC_0 = 60\%$ ,  $U_{c,0} = 0$ ,  $SOH_0 = 90\%$ . The initial state covariance matrix  $P_0$  is diagonal; its diagonal elements are  $0.04$ ,  $10^{-10}$ , and  $0.01$ . The initial state estimations are  $80\%$ ,  $0$ , and  $100\%$ , respectively. The input current is noisy, and its standard deviation is  $1$  mA. This input noise is the only source of the process noises. Since the input current also appears in the measurement function, the actual measurement noise is the combination of the measurement noise of the voltmeter and the noise caused by the term  $R_1 I_k$  in the measurement function.

### C The detailed algorithm of various nonlinear Kalman filters used in the paper

The detailed algorithms of EKF, EKF2, UKF, and CKF with the new framework are shown in Algorithms 3–5. Note that all these algorithms degrade to the conventional framework when the “Recalibrate” and “Back out” steps are replaced by (10).

---

**Algorithm 3** The extended Kalman filter with the new framework

**Input:** Process noise covariance matrix  $Q_k$ , Measurement noise covariance matrix  $R_k$ , state transition function  $f(\mathbf{x}, \mathbf{u})$ , measurement function  $h(\mathbf{x})$ , system inputs  $\mathbf{u}_k$ , measurements  $\mathbf{z}_k$

**Initialization:**

- 1:  $\hat{\mathbf{x}}_{0|0} = \mathbb{E}[\mathbf{x}_0]$
- 2:  $P_{0|0} = \mathbb{E}[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T]$
- 3: **for** every time step  $k$  **do**

**Predict:**

- 4:  $\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}$
- 5:  $\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1})$
- 6:  $P_{k|k-1} = \mathbf{F}_k P_{k-1|k-1} \mathbf{F}_k^T + Q_k$

**Update:**

- 7:  $\mathbf{H}_{k|k-1} = \frac{\partial h}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k|k-1}}$
- 8:  $\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$
- 9:  $P_{xy,k|k-1} = P_{k|k-1} \mathbf{H}_{k|k-1}^T$
- 10:  $P_{y,k|k-1} = \mathbf{H}_{k|k-1} P_{k|k-1} \mathbf{H}_{k|k-1}^T$
- 11:  $S_{k|k-1} = P_{y,k|k-1} + R_k$
- 12:  $\mathbf{K} = P_{xy,k|k-1} S_{k|k-1}^{-1}$
- 13:  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1})$

**Recalibrate:**

- 14:  $\mathbf{H}_{k|k} = \frac{\partial h}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k|k}}$
- 15:  $P_{xy,k|k} = P_{k|k-1} \mathbf{H}_{k|k}^T$
- 16:  $P_{y,k|k} = \mathbf{H}_{k|k} P_{k|k-1} \mathbf{H}_{k|k}^T$
- 17:  $S_{k|k} = P_{y,k|k} + R_k$
- 18:  $P_{k|k} = P_{k|k-1} + \mathbf{K} S_{k|k} \mathbf{K}^T - P_{xy,k|k} \mathbf{K}^T - \mathbf{K} P_{xy,k|k}^T$

**Back out:**

- 19: **if**  $\text{tr}(P_{k|k}) > \text{tr}(P_{k|k-1})$  **then**
- 20:      $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}$
- 21:      $P_{k|k} = P_{k|k-1}$
- 22: **end if**
- 23: **end for**

---

---

**Algorithm 4** The second-order extended Kalman filter with the new framework

---

**Input:** Process noise covariance matrix  $\mathbf{Q}_k$ , Measurement noise covariance matrix  $\mathbf{R}_k$ , states number  $n_x$ , measurements number  $n_m$ , state transition function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ , measurement function  $\mathbf{h}(\mathbf{x})$ , system inputs  $\mathbf{u}_k$ , measurements  $\mathbf{z}_k$

**Initialization:**

- 1:  $\hat{\mathbf{x}}_{0|0} = \mathbb{E}[\mathbf{x}_0]$
- 2:  $\mathbf{P}_{0|0} = \mathbb{E}[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T]$
- 3: **for** every time step  $k$  **do**
- Predict:**
- 4:  $\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}$
- 5:  $\mathbf{F}_{i,k}^* = [\frac{\partial^2 f_i}{\partial x_m \partial x_n} |_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}]_{mn}, i = 1, \dots, n_x$
- 6:  $\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) + \frac{1}{2} [\text{tr}(\mathbf{F}_{i,k}^* \mathbf{P}_{k-1|k-1})]_i$
- 7:  $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \frac{1}{2} [\text{tr}(\mathbf{F}_{i,k}^* \mathbf{P}_{k-1|k-1} \mathbf{F}_{j,k}^* \mathbf{P}_{k-1|k-1})]_{ij} + \mathbf{Q}_k$

**Update:**

- 8:  $\mathbf{H}_{k|k-1} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k|k-1}}$
- 9:  $\mathbf{H}_{i,k|k-1}^* = [\frac{\partial^2 h_i}{\partial x_m \partial x_n} |_{\hat{\mathbf{x}}_{k|k-1}}]_{mn}, i = 1, \dots, n_m$
- 10:  $\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) + \frac{1}{2} [\text{tr}(\mathbf{H}_{i,k|k-1}^* \mathbf{P}_{k|k-1})]_i$
- 11:  $\mathbf{P}_{xy,k|k-1} = \mathbf{P}_{k|k-1} \mathbf{H}_{k|k-1}$
- 12:  $\mathbf{P}_{y,k|k-1} = \mathbf{H}_{k|k-1} \mathbf{P}_{k|k-1} \mathbf{H}_{k|k-1}^T + \frac{1}{2} [\text{tr}(\mathbf{H}_{i,k|k-1}^* \mathbf{P}_{k|k-1} \mathbf{H}_{j,k|k-1}^* \mathbf{P}_{k|k-1})]_{ij}$
- 13:  $\mathbf{S}_{k|k-1} = \mathbf{P}_{y,k|k-1} + \mathbf{R}_k$
- 14:  $\mathbf{K} = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1}$
- 15:  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1})$

**Recalibrate:**

- 16:  $\mathbf{H}_{k|k} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_{k|k}}$
  - 17:  $\mathbf{H}_{i,k|k}^* = [\frac{\partial^2 h_i}{\partial x_m \partial x_n} |_{\hat{\mathbf{x}}_{k|k}}]_{mn}, i = 1, \dots, n_m$
  - 18:  $\mathbf{P}_{xy,k|k} = \mathbf{P}_{k|k-1} \mathbf{H}_{k|k}^T$
  - 19:  $\mathbf{P}_{y,k|k} = \mathbf{H}_{k|k} \mathbf{P}_{k|k-1} \mathbf{H}_{k|k}^T + \frac{1}{2} [\text{tr}(\mathbf{H}_{i,k|k}^* \mathbf{P}_{k|k-1} \mathbf{H}_{j,k|k}^* \mathbf{P}_{k|k-1})]_{ij}$
  - 20:  $\mathbf{S}_{k|k} = \mathbf{P}_{y,k|k} + \mathbf{R}_k$
  - 21:  $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} + \mathbf{K} \mathbf{S}_{k|k} \mathbf{K}^T - \mathbf{P}_{xy,k|k} \mathbf{K}^T - \mathbf{K} \mathbf{P}_{xy,k|k}^T$
- Back out:**
- 22: **if**  $\text{tr}(\mathbf{P}_{k|k}) > \text{tr}(\mathbf{P}_{k|k-1})$  **then**
  - 23:      $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}$
  - 24:      $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1}$
  - 25:     **end if**
  - 26: **end for**
- 

---

**Algorithm 5** The cubature Kalman filter with the new framework

---

**Input:** Process noise covariance matrix  $\mathbf{Q}_k$ , Measurement noise covariance matrix  $\mathbf{R}_k$ , states number  $n_x$ , state transition function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ , measurement function  $\mathbf{h}(\mathbf{x})$ , system inputs  $\mathbf{u}_k$ , measurements  $\mathbf{z}_k$

**Initialization:**

- 1:  $\hat{\mathbf{x}}_{0|0} = \mathbb{E}[\mathbf{x}_0]$
- 2:  $\mathbf{P}_{0|0} = \mathbb{E}[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T]$
- 3: **for** every time step  $k$  **do**

**Predict:**

- 4:  $\mathbf{L} = [\mathbf{L}_1, \dots, \mathbf{L}_{n_x}] = \text{chol}(\mathbf{P}_{k-1|k-1})$ ,  
where  $\mathbf{L} \mathbf{L}^T = \mathbf{P}_{k-1|k-1}$
- 5:  $\mathcal{X}_{i,k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{n_x} \mathbf{L}_i, i = 1, \dots, n_x$
- 6:  $\mathcal{X}_{i+n_x,k-1} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{n_x} \mathbf{L}_i, i = 1, \dots, n_x$
- 7:  $\mathcal{X}_{i,k|k-1}^* = \mathbf{f}(\mathcal{X}_{i,k-1}, \mathbf{u}_{k-1}), i = 1, \dots, 2n_x$
- 8:  $\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} \mathcal{X}_{i,k|k-1}^*$
- 9:  $\mathbf{P}_{k|k-1} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} [\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}] [\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}]^T + \mathbf{Q}_k$

**Update:**

- 10:  $\mathbf{L} = [\mathbf{L}_1, \dots, \mathbf{L}_{n_x}] = \text{chol}(\mathbf{P}_{k|k-1})$
  - 11:  $\mathcal{X}_{i,k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \sqrt{n_x} \mathbf{L}_i, i = 1, \dots, n_x$
  - 12:  $\mathcal{X}_{i+n_x,k|k-1} = \hat{\mathbf{x}}_{k|k-1} - \sqrt{n_x} \mathbf{L}_i, i = 1, \dots, n_x$
  - 13:  $\mathcal{Y}_{i,k|k-1} = \mathbf{h}(\mathcal{X}_{i,k|k-1}), i = 1, \dots, 2n_x$
  - 14:  $\hat{\mathbf{y}}_{k|k-1} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} \mathcal{Y}_{i,k|k-1}$
  - 15:  $\mathbf{P}_{xy,k|k-1} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}] [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}]^T$
  - 16:  $\mathbf{P}_{y,k|k-1} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}] [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}]^T$
  - 17:  $\mathbf{S}_{k|k-1} = \mathbf{P}_{y,k|k-1} + \mathbf{R}_k$
  - 18:  $\mathbf{K} = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1}$
  - 19:  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1})$
- Recalibrate:**
- 20:  $\mathcal{X}_{i,k|k} = \hat{\mathbf{x}}_{k|k} + \sqrt{n_x} \mathbf{L}_i, i = 1, \dots, n_x$
  - 21:  $\mathcal{X}_{i+n_x,k|k} = \hat{\mathbf{x}}_{k|k} - \sqrt{n_x} \mathbf{L}_i, i = 1, \dots, n_x$
  - 22:  $\mathcal{Y}_{i,k|k} = \mathbf{h}(\mathcal{X}_{i,k|k}), i = 1, \dots, 2n_x$
  - 23:  $\hat{\mathbf{y}}_{k|k} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} \mathcal{Y}_{i,k|k}$
  - 24:  $\mathbf{P}_{xy,k|k} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} [\mathcal{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}] [\mathcal{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}]^T$
  - 25:  $\mathbf{P}_{y,k|k} = \sum_{i=1}^{2n_x} \frac{1}{2n_x} [\mathcal{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}] [\mathcal{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}]^T$
  - 26:  $\mathbf{S}_{k|k} = \mathbf{P}_{y,k|k} + \mathbf{R}_k$
  - 27:  $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} + \mathbf{K} \mathbf{S}_{k|k} \mathbf{K}^T - \mathbf{P}_{xy,k|k} \mathbf{K}^T - \mathbf{K} \mathbf{P}_{xy,k|k}^T$
- Back out:**
- 28: **if**  $\text{tr}(\mathbf{P}_{k|k}) > \text{tr}(\mathbf{P}_{k|k-1})$  **then**
  - 29:      $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}$
  - 30:      $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1}$
  - 31:     **end if**
  - 32: **end for**
-

---

**Algorithm 6** The unscented Kalman filter with the new framework

---

**Input:** Process noise covariance matrix  $\mathbf{Q}_k$ , Measurement noise covariance matrix  $\mathbf{R}_k$ , states number  $n_x$ , state transition function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ , measurement function  $\mathbf{h}(\mathbf{x})$ , system inputs  $\mathbf{u}_k$ , measurements  $\mathbf{z}_k$ , hyperparameters  $\alpha = 10^{-3}$ ,  $\beta = 2$ ,  $\kappa = 0$

**Initialization:**

- 1:  $\hat{\mathbf{x}}_{0|0} = \mathbb{E}[\mathbf{x}_0]$
- 2:  $\mathbf{P}_{0|0} = \mathbb{E}[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T]$
- 3:  $\lambda = \alpha^2(n_x + \kappa) - n_x$
- 4:  $W_i^{(c)} = W_i^{(m)} = 1/(2n_x + 2\lambda), i = 1, \dots, n_x$
- 5:  $W_0^{(c)} = \lambda/(n_x + \lambda) + 1 - \alpha^2 + \beta$
- 6:  $W_0^{(m)} = \lambda/(n_x + \lambda)$
- 7: **for** every time step  $k$  **do**

**Predict:**

- 8:  $\mathbf{L} = [\mathbf{L}_1, \dots, \mathbf{L}_{n_x}] = \text{chol}(\mathbf{P}_{k-1|k-1})$ , where  $\mathbf{L}\mathbf{L}^T = \mathbf{P}_{k-1|k-1}$
- 9:  $[\mathbf{L}_1, \dots, \mathbf{L}_{n_x}] = i^{\text{th}}$  column of  $\mathbf{L}, i = 1, \dots, n_x$
- 10:  $\mathbf{X}_{0,k-1} = \hat{\mathbf{x}}_{k-1|k-1}$
- 11:  $\mathbf{X}_{i,k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{\lambda + n_x} \mathbf{L}_i, i = 1, \dots, n_x$
- 12:  $\mathbf{X}_{i+n_x,k-1} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{\lambda + n_x} \mathbf{L}_i, i = 1, \dots, n_x$
- 13:  $\mathbf{X}_{i,k|k-1} = \mathbf{f}(\mathbf{X}_{i,k-1}, \mathbf{u}_{k-1}), i = 0, 1, \dots, 2n_x$
- 14:  $\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathbf{X}_{i,k|k-1}$
- 15:  $\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n_x} W_i^{(c)} [\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}] [\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}]^T + \mathbf{Q}_k$

**Update:**

- 16:  $\mathbf{L} = [\mathbf{L}_1, \dots, \mathbf{L}_{n_x}] = \text{chol}(\mathbf{P}_{k-1|k})$
- 17:  $\mathbf{X}_{0,k|k-1} = \hat{\mathbf{x}}_{k|k-1}$
- 18:  $\mathbf{X}_{i,k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \sqrt{\lambda + n_x} \mathbf{L}_i, i = 1, \dots, n_x$
- 19:  $\mathbf{X}_{i+n_x,k|k-1} = \hat{\mathbf{x}}_{k|k-1} - \sqrt{\lambda + n_x} \mathbf{L}_i, i = 1, \dots, n_x$
- 20:  $\mathbf{Y}_{i,k|k-1} = \mathbf{h}(\mathbf{X}_{i,k|k-1}), i = 0, \dots, 2n_x$
- 21:  $\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathbf{Y}_{i,k|k-1}$
- 22:  $\mathbf{P}_{xy,k|k-1} = \sum_{i=0}^{2n_x} W_i^{(c)} [\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}] [\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}]^T$
- 23:  $\mathbf{P}_{y,y,k|k-1} = \sum_{i=0}^{2n_x} W_i^{(c)} [\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}] [\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}]^T$
- 24:  $\mathbf{S}_{k|k-1} = \mathbf{P}_{y,y,k|k-1} + \mathbf{R}_k$
- 25:  $\mathbf{K} = \mathbf{P}_{xy,k|k-1} \mathbf{S}_{k|k-1}^{-1}$
- 26:  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1})$

**Recalibrate:**

- 27:  $\mathbf{X}_{i,k|k} = \mathbf{X}_{i,k|k-1} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{y}}_{k|k-1}), i = 0, \dots, 2n_x$
- 28:  $\mathbf{Y}_{i,k|k} = \mathbf{h}(\mathbf{X}_{i,k|k}), i = 0, \dots, 2n_x$
- 29:  $\hat{\mathbf{y}}_{k|k} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathbf{Y}_{i,k|k}$
- 30:  $\mathbf{P}_{xy,k|k} = \sum_{i=0}^{2n_x} W_i^{(c)} [\mathbf{X}_{i,k|k} - \hat{\mathbf{x}}_{k|k}] [\mathbf{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}]^T$
- 31:  $\mathbf{P}_{y,y,k|k} = \sum_{i=0}^{2n_x} W_i^{(c)} [\mathbf{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}] [\mathbf{Y}_{i,k|k} - \hat{\mathbf{y}}_{k|k}]^T$
- 32:  $\mathbf{S}_{k|k} = \mathbf{P}_{y,y,k|k} + \mathbf{R}_k$
- 33:  $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} + \mathbf{K} \mathbf{S}_{k|k} \mathbf{K}^T - \mathbf{P}_{xy,k|k} \mathbf{K}^T - \mathbf{K} \mathbf{P}_{xy,k|k}^T$

**Back out:**

- 34: **if**  $\text{tr}(\mathbf{P}_{k|k}) > \text{tr}(\mathbf{P}_{k|k-1})$  **then**
- 35:      $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}$
- 36:      $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1}$
- 37: **end if**
- 38: **end for**

---