(Preprint) AAS 25-764

# COUNTER-ADVERSARIAL ESTIMATION FOR SPACE NAVIGATION: THE FUSION REPRODUCING KERNEL HILBERT SPACE EXTENDED KALMAN FILTER

Alberto Zamora[*] and Kyle J. DeMars[†]

Estimating spacecraft states in space is uniquely challenging due to nonlinear and uncertain dynamics, limited or unreliable observations, and the presence of external disturbances, such as signal interference and spoofing. This paper introduces the fusion (inverse) reproducing kernel Hilbert space extended Kalman filter (REKF) as a novel approach for robust and adaptive state estimation in this regime. By leveraging an online expectation-maximization algorithm, the REKF learns unknown system parameters and approximates nonlinear dynamics and measurements through kernel-based function representations. This framework enables accurate threat detection and active sensing, offering a promising approach for resilient spacecraft navigation in uncertain conditions. Simulation results demonstrate that the proposed REKF with online EM significantly outperforms the standard EKF in terms of estimation accuracy, particularly under nonlinear and noisy measurement conditions.

## INTRODUCTION

A wide variety of engineering applications require inferring the parameters of a system, such as its system and observation matrices, as well as its process and measurement noise covariance matrices, by observing its output. Also, prior investigations have proposed adaptive filters to reduce the sensitivity of the Kalman filter to system uncertainties. However, the general case for nonlinear systems filtering is not considered.[1–4] In this context, fusion filtering techniques are key to cognitive and counter-adversarial systems, where nonlinear dynamics and unknown system parameters are unknown to an agent employing the fusion filter.

This work builds on advanced filtering methods, specifically within the framework of the fusion extended Kalman filter (FEKF) paradigm.[5,6] The FEKF is designed to address the challenges of modeling complex, uncertain, and nonlinear dynamics while enabling highly precise state estimation. Applications also include threat detection and active sensing for space navigation operations.

The extended Kalman Filter (EKF) and FEKF are both used for state estimation, but they differ in their approaches. The EKF propagates the state estimate using the nonlinear system model, while the state uncertainty is updated using a linearization of the dynamics around the current estimate.[7–10] In contrast, the FEKF works in refining or correcting an existing state estimate. The FEKF is typically employed in filter fusion systems or adversarial scenarios, where it corrects the output of another filter like the EKF. The key distinction lies in the fact that the EKF provides real-time estimates, while the FEKF works to improve or refine those estimates based on existing estimates.

[*] Researcher, Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843.

[†] Associate Professor, Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843.

The FEKF consists of two stages: the Alpha filter ($A$F) and the Omega filter ($\Omega$F). The $A$F may be any conventional EKF variant, such as the EKF, the second-order EKF, the dithered EKF, or the REKF.[5,6] More sophisticated approaches, such as the Gaussian Sum EKF, may also be incorporated depending on the application.[5] The Omega filter ($\Omega$F) receives the $A$F's estimates and its goal is to compute a more accurate estimate, using the $A$F's state transition equations (STEs) as a reference.

This work considers the case where the system and observation models, as well as the structure of the $A$F, are unknown. To address this, the REKF is proposed as a kernel-based approximation technique within the FEKF framework. The REKF uses kernel function representations to learn unknown nonlinear functions, while an online approximate expectation-maximization (EM) algorithm simultaneously identifies unknown parameters.[11] This enables the $\Omega$-agent to operate even under partial or uncertain knowledge of the$A$-agent's model.

Traditional FEKF theory assumes that the $\Omega$-agent has full access to the $A$F's parameters.[6] However, this assumption is relaxed within the context of this work. The $A$-agent and $\Omega$-agent represent the agents using the $A$F and $\Omega$F, respectively. When the $\Omega$-agent lacks knowledge of the $A$-agent's model, the REKF becomes a valuable tool for estimating the $A$F's outputs indirectly. Although this lack of knowledge about the $A$F introduces a potential model mismatch between the $A$F and $\Omega$F, numerical results demonstrate that the $\Omega$-agent can still estimate the $A$F's state with reasonable accuracy.[5] Furthermore, by increasing the computational effort, a more sophisticated REKF can achieve even better performance despite mismatches.

## THE REPRODUCING KERNEL HILBERT SPACE EXTENDED KALMAN FILTER

The space environment presents unique challenges for state estimation due to nonlinear and uncertain dynamics, limited or unreliable observations, and the potential for external disturbances such as signal interference and spoofing. To address these challenges, the fusion REKF (FREKF) is introduced as an adaptive and robust framework for state estimation. The REKF utilizes kernel-based function representations to approximate unknown nonlinear system dynamics and measurements, leveraging an online expectation-maximization (EM) algorithm to iteratively learn system parameters. This approach makes the REKF highly suitable for real-time, resilient state estimation, particularly in demanding domains such as cislunar operations, where robust threat detection and adaptive sensing are critical for successful operations.

Consider a discrete-time stochastic dynamical system where the $\Omega$-agent's state $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is observed by the $A$-agent as measurements $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$, evolving according to

$$\boldsymbol{x}_k = \boldsymbol{f}_\Omega\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{w}_{k-1} \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_\Omega\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k, \tag{2}$$

where $\boldsymbol{w}_{k-1}$ and $\boldsymbol{v}_k$ are the additive process and measurement noises, respectively. The $A$-agent computes the estimate $\boldsymbol{m}^+_{x,k} \in \mathbb{R}^{n_x}$ of the $\Omega$-agent's state $\boldsymbol{x}_k$ using its observations $\{\boldsymbol{y}_\ell\}_{\ell=1}^k$. The $A$-agent then takes an action $\boldsymbol{h}_A\big(\boldsymbol{m}^+_{x,k}\big)$, whose noisy observation made by the $\Omega$-agent is given by

$$\boldsymbol{z}_k = \boldsymbol{h}_A\big(\boldsymbol{m}^+_{x,k}\big) + \boldsymbol{u}_k, \tag{3}$$

where $\boldsymbol{u}_k$ is the $\Omega$-agent's additive measurement noise. Finally, the $\Omega$-agent computes the estimate $\boldsymbol{m}^{+}_{x\Omega,k} \in \mathbb{R}^{n_x}$ of $\boldsymbol{m}^{+}_{x,k}$ using its observations $\{\boldsymbol{z}_\ell\}_{\ell=1}^{k}$. Note that for this work, the process and measurement noise are taken to be additive, white, zero-mean and mutually uncorrelated. The $A$-agent's noise covariance matrices are known, while the $\Omega$-agent's covariance matrices need to be estimated (see the Expectation-Maximization for Parameter Learning section).

### Unknown $A$F and System Model

In practice, the functions $\boldsymbol{f}_\Omega(\cdot)$ and $\boldsymbol{h}_\Omega(\cdot)$, as well as the corresponding process and measurement noise covariances, may not be known to the agents. This uncertainty arises in scenarios where either the system dynamics or the observation models are partially or entirely unknown, which is common in complex environments like cislunar space where external disturbances such as spoofing can alter the system's behavior.

To overcome this, the REKF uses kernel-based approximations to learn unknown nonlinear dynamics and observations. This enables the $\Omega$-agent to estimate the state of the $A$-agent, even with limited or no prior knowledge about the $A$-agent's system model. This capability is essential for navigating and sensing in cislunar space, where traditional filtering techniques may fail due to the lack of reliable models. The REKF framework combines kernel function representations with an online EM algorithm to iteratively update the system parameters, providing a powerful tool for resilient state estimation and threat detection in uncertain conditions.

An unknown function $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ can be approximated using the reproducing kernel Hilbert space (RKHS) induced by a kernel $\kappa(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.[12] The RKHS-based function approximation has been proposed for nonlinear state-space modeling and recursive least squares algorithms with unknown nonlinear functions.[13–17] The representer theorem from Reference 18 ensures that the optimal approximation in RKHS with respect to an arbitrary loss function takes the form of

$$f(\cdot) = \sum_{\ell=1}^{L} a_\ell \kappa\left(\tilde{\boldsymbol{x}}_\ell, \cdot\right), \tag{4}$$

where $\{\tilde{\boldsymbol{x}}_\ell\}_{\ell=1}^{L}$ are the $L$ input training samples (or dictionary) and $\{a_\ell\}_{\ell=1}^{L}$ are the corresponding mixing parameters that are to be learned. There are many types of kernels used in RKHS, but often the Gaussian kernel, given by

$$\kappa\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\right), \tag{5}$$

with kernel width $\sigma > 0$ controlling the smoothness of the approximation, is an appropriate choice. This is a universal kernel, meaning that any continuous function can be arbitrarily well-approximated by functions from the RKHS, i.e., its induced RKHS is dense in the space of continuous functions.

In the following, a general nonlinear system model is considered, wherein both state transition and observation functions are unknown to the agent employing the filter, and a general REKF is developed. The $\Omega$-agent can employ the REKF as its $\Omega$F without assuming any prior information about the $A$-agent's $A$F. The REKF may be trivially simplified to yield REKF $A$F for the $A$-agent that knows its observation function. In particular, this REKF adopts the EKF to obtain the state estimates while the unknown system parameters are learned using an EM algorithm.[19] The EM algorithm is widely used to compute maximum likelihood estimates in presence of missing data.

*System Models for Uncertain Dynamics.* Here, the functions $\boldsymbol{f}_{\Omega}(\cdot)$ and $\boldsymbol{h}_{\Omega}(\cdot)$ and their corresponding process and measurement noise covariances, $\mathbf{P}_{ww,k-1}$ and $\mathbf{P}_{vv,k}$, from the nonlinear state transition and observation model, Eqs. (1) and (2), respectively, are taken to be unknown. Consider a kernel function $\kappa(\cdot,\cdot)$ and a dictionary $\{\tilde{\boldsymbol{x}}_\ell\}_{\ell=1}^{L}$ of size $L$. Denote a kernel vector, such that

$$\boldsymbol{\kappa}(\boldsymbol{x}) = \begin{bmatrix} \kappa(\tilde{\boldsymbol{x}}_1, \boldsymbol{x}) & \cdots & \kappa(\tilde{\boldsymbol{x}}_L, \boldsymbol{x}) \end{bmatrix}^T . \tag{6}$$

Given the unknown dynamics of the system, the kernel approximation is applied to the state transition and observation equations, yielding

$$\boldsymbol{x}_k = \mathbf{A}\boldsymbol{\kappa}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1} \tag{7}$$

$$\boldsymbol{y}_k = \mathbf{B}\boldsymbol{\kappa}(\boldsymbol{x}_k) + \boldsymbol{v}_k, \tag{8}$$

where $\mathbf{A} \in \mathbb{R}^{n_x \times L}$ and $\mathbf{B} \in \mathbb{R}^{n_y \times L}$ include the unknown mixing parameters to be learned. The dictionary $\{\tilde{\boldsymbol{x}}_\ell\}_{\ell=1}^{L}$ can be formed using a sliding window (see Reference 17) or an approximate linear dependency (ALD) criterion (see Reference 15). These approximations allow the REKF to estimate the states as $\boldsymbol{m}_{x,k}^{+}$ and update the system parameters $\boldsymbol{\Theta} = \{\mathbf{A}, \mathbf{B}, \mathbf{P}_{ww,k-1}, \mathbf{P}_{vv,k}\}$ as new data arrives, ensuring that the system adapts in real-time to environmental changes and uncertainties.

*Expectation-Maximization for Parameter Learning.* The EM algorithm is employed to iteratively update the estimates of the unknown parameters. During the E-step, the expected value of the log-likelihood function is computed, and in the M-step, the parameters are updated to maximize the expected log-likelihood.[6] This iterative process enables the REKF to learn the system parameters while refining the state estimates, making it highly effective for real-time operations in unpredictable environments.

Given the unknown parameters $\boldsymbol{\Theta}$, the state estimate $\boldsymbol{m}_{x,k}^{+}$ can be computed using EKF-based recursions.[5] To estimate the unknown parameters, the EM algorithm in Reference 19 is used. Consider the states up to time $k$ as $\boldsymbol{x}_{0:k} = \{\boldsymbol{x}_\ell\}_{\ell=0}^{k}$ and the corresponding observations $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_\ell\}_{\ell=1}^{k}$. The joint conditional probability density function (pdf) given the parameters $\boldsymbol{\Theta}$ is

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k}|\boldsymbol{\Theta}) = p(\boldsymbol{x}_0) \prod_{i=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}, \boldsymbol{\Theta}) \prod_{j=1}^{k} p(\boldsymbol{y}_j|\boldsymbol{x}_j, \boldsymbol{\Theta}), \tag{9}$$

where $p(\boldsymbol{x}_0) = p_g(\boldsymbol{x}_0; \boldsymbol{m}_{x,0}, \mathbf{P}_{xx,0})$ is the initial pdf, $p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}, \boldsymbol{\Theta})$ is the transition density and $p(\boldsymbol{y}_j|\boldsymbol{x}_j, \boldsymbol{\Theta})$ is the measurement likelihood. Under the assumption of additive Gaussian noise, the conditional pdfs become

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}, \boldsymbol{\Theta}) = p_g(\boldsymbol{x}_i; \mathbf{A}\boldsymbol{\kappa}(\boldsymbol{x}_{i-1}), \mathbf{P}_{ww,i-1}) \tag{10}$$

$$p(\boldsymbol{y}_j|\boldsymbol{x}_j, \boldsymbol{\Theta}) = p_g(\boldsymbol{y}_j; \mathbf{B}\boldsymbol{\kappa}(\boldsymbol{x}_j), \mathbf{P}_{vv,j}), \tag{11}$$

where $p_g\left(\cdot;\cdot,\cdot\right)$ denotes the Gaussian pdf. Assume the initial state $\boldsymbol{x}_0 \sim p_g\left(\boldsymbol{x}_0; \boldsymbol{m}_{x,0}, \mathbf{P}_{xx,0}\right)$. Note that this assumption is taken here in order to initialize the EKF recursions. Using this in Eq. (9) along with Eqs. (10) and (11), after some manipulation, yields

$$\begin{aligned}\log\big[p\big(\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k}|\boldsymbol{\Theta}\big)\big] = -\frac{1}{2}\Bigg\{&\log|\mathbf{P}_{xx,0}| + \left(\boldsymbol{x}_0 - \boldsymbol{m}_{x,0}\right)^T \mathbf{P}_{xx,0}^{-1}\left(\boldsymbol{x}_0 - \boldsymbol{m}_{x,0}\right)\\ &+ \sum_{i=1}^{k}\big[\log|\mathbf{P}_{ww,i-1}| + \big(\boldsymbol{x}_i - \mathbf{A}\boldsymbol{\kappa}\left(\boldsymbol{x}_{i-1}\right)\big)^T \mathbf{P}_{ww,i-1}^{-1}\big(\boldsymbol{x}_i - \mathbf{A}\boldsymbol{\kappa}\left(\boldsymbol{x}_{i-1}\right)\big)\big]\\ &+ \sum_{j=1}^{k}\big[\log|\mathbf{P}_{vv,j}| + \big(\boldsymbol{y}_j - \mathbf{B}\boldsymbol{\kappa}\left(\boldsymbol{x}_j\right)\big)^T \mathbf{P}_{vv,j}^{-1}\big(\boldsymbol{y}_j - \mathbf{B}\boldsymbol{\kappa}\left(\boldsymbol{x}_j\right)\big)\big]\Bigg\} + c,\end{aligned} \tag{12}$$

where $c$ denotes the constant terms that do not affect the maximization.

A basic EM algorithm to estimate $\boldsymbol{\Theta}$ based on observations $\boldsymbol{y}_{1:k}$ consists of following two steps that are iterated a fixed number of times or until convergence:

**E-step:** Given the estimate $\hat{\boldsymbol{\Theta}}^-$ of the unknown parameters, the expectation of the joint log-likelihood $\mathbf{Q}$ is computed, where

$$\mathbf{Q}\big(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}^-\big) = \mathbb{E}_{y_{1:k},\Theta^-}\big\{\log\big[p\big(\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k}|\boldsymbol{\Theta}\big)\big]\big\}. \tag{13}$$

**M-step:** The updated parameter estimate $\hat{\boldsymbol{\Theta}}^+ = \arg\max_{\Theta} \mathbf{Q}\big(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}^-\big)$. Note that for online EM, at the $k$-th time step, the current estimate $\hat{\boldsymbol{\Theta}}^-$ is $\hat{\boldsymbol{\Theta}}_{k-1}$ and the updated estimate $\hat{\boldsymbol{\Theta}}^+$ is $\hat{\boldsymbol{\Theta}}_k$.

To ensure that the REKF can operate efficiently in real-time, the EM algorithm is adapted to an online version. This allows the REKF to update the parameter estimates incrementally as new data arrives, minimizing computational overhead while ensuring that the system remains adaptive. The online EM algorithm allows the REKF to improve its estimates over time, even in the presence of incomplete or noisy observations.

For simplicity, denote the conditional expectation operator $\mathbb{E}_{y_{1:k},\hat{\Theta}_{k-1}}\left\{\cdot\right\}$ given $k$ observations by $\mathbb{E}_k\left\{\cdot\right\}$. Now, as derived in Reference 5, analytical solutions to the parameters are obtained through the process described in the *E* and *M* steps by substituting $\boldsymbol{\Theta}$ by each of the parameters. In order to obtain an approximate online estimate at low computational expense, define

$$\mathbf{S}_{x\kappa,k} = \sum_{i=1}^{k}\mathbb{E}_k\left\{\boldsymbol{x}_i\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{i-1}\right)\right\} \tag{14}$$

$$\mathbf{S}_{\kappa 1,k} = \sum_{j=1}^{k}\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_{j-1}\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{j-1}\right)\right\}, \tag{15}$$

such that the estimated $\mathbf{A}$ in Eq. (7), or $\hat{\mathbf{A}}_k$, can be written as

$$\hat{\mathbf{A}}_k = \mathbf{S}_{x\kappa,k}\mathbf{S}_{\kappa 1,k}^{-1}. \tag{16}$$

Note that the computation of the sums requires all $k$ observations to be processed together at time $k$, and the complexity increases as $k$ increases. The sums in Eqs. (14) and (15) can be approximated as

$$\mathbf{S}_{x\kappa,k} \approx \mathbf{S}_{x\kappa,k-1} + \mathbb{E}_k\left\{\boldsymbol{x}_k\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k-1}\right)\right\} \tag{17}$$

$$\mathbf{S}_{\kappa 1,k} \approx \mathbf{S}_{\kappa 1,k-1} + \mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k-1}\right)\right\}. \tag{18}$$

Up to this point, Eqs. (17) and (18) seem to be the same as Eqs. (14) and (15). The reality is that Eqs. (17) and (18) are approximations because the prior updated parameter estimates $\boldsymbol{\Theta}_{k-1}\big\{\hat{\mathbf{A}}_{k-1}, \hat{\mathbf{B}}_{k-1}, \hat{\mathbf{P}}_{ww,k-1}, \hat{\mathbf{P}}_{vv,k-1}\big\}$ (which are obtained using observations up to time $k-1$) for computing the expectations in the prior sums $S_{x\kappa,k-1}$ and $S_{\kappa 1,k-1}$, are not considered. Similarly, other approximate parameter updates are obtained by approximating sums as in Eqs. (17) and (18), yielding

$$\mathbf{S}_{y\kappa,k} \approx \mathbf{S}_{y\kappa,k-1} + \mathbb{E}_k\left\{\boldsymbol{y}_k\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\} \tag{19}$$

$$\mathbf{S}_{\kappa 2,k} \approx \mathbf{S}_{\kappa 2,k-1} + \mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_k\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\} \tag{20}$$

$$\hat{\mathbf{B}}_k = \mathbf{S}_{y\kappa,k}\mathbf{S}_{\kappa 2,k}^{-1} \tag{21}$$

$$\begin{aligned}\hat{\mathbf{P}}_{ww,k} = {} & \left(1-\frac{1}{k}\right)\hat{\mathbf{P}}_{ww,k-1} + \frac{1}{k}\big(\mathbb{E}_k\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} - \hat{\mathbf{A}}_k\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right)\boldsymbol{x}_k^T\right\} \\ & - \mathbb{E}_k\left\{\boldsymbol{x}_k\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k-1}\right)\right\}\hat{\mathbf{A}}_k^T + \hat{\mathbf{A}}_k\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k-1}\right)\right\}\hat{\mathbf{A}}_k^T\big)\end{aligned} \tag{22}$$

$$\begin{aligned}\hat{\mathbf{P}}_{vv,k} = {} & \left(1-\frac{1}{k}\right)\hat{\mathbf{P}}_{vv,k-1} + \frac{1}{k}\big(\mathbb{E}_k\left\{\boldsymbol{y}_k\boldsymbol{y}_k^T\right\} - \hat{\mathbf{B}}_k\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_k\right)\boldsymbol{y}_k^T\right\} \\ & - \mathbb{E}_k\left\{\boldsymbol{y}_k\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\}\hat{\mathbf{B}}_k^T + \hat{\mathbf{B}}_k\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_k\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\}\hat{\mathbf{B}}_k^T\big),\end{aligned} \tag{23}$$

where the sums $\mathbf{S}_{y\kappa,k}$ and $\mathbf{S}_{\kappa 2,k}$ are introduced to obtain approximate online estimates, while $\hat{\mathbf{B}}_k$ is the estimate of $\mathbf{B}$ in Eq. (8). Recall that all the results in Eqs. (16) and Eqs. (21) - (23) are the result of substituting each of the parameters to be learned are obtained by following the process described in the E and M steps described earlier in this manuscript by substituting $\boldsymbol{\Theta}$ to each of the parameters $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\hat{\mathbf{P}}_{ww,k}$ and $\hat{\mathbf{P}}_{vv,k}$, one at a time.

*Expectation Computations.* As mentioned earlier, the required expectations are calculated in Eqs. (17) - (23) using the EKF estimates. In this work, it is assumed that the mean and covariance outputs of the EKF recursion represent a Gaussian distribution of the *posterior* distribution of the required states given all the observations available up to the current time instant. However, these expectations involve a nonlinear transformation $\boldsymbol{\kappa}\left(\cdot\right)$. In the EKF, the first-order Taylor series expansion (FOTSE) is used to approximate these nonlinear expectations. The statistics of $\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right)$ given $\boldsymbol{y}_{1:k}$ are also needed. Hence, an augmented state

$$\mathbf{s}_k = \begin{bmatrix}\boldsymbol{x}_k^T & \boldsymbol{x}_{k-1}^T\end{bmatrix}^T, \tag{24}$$

is also considered to obtain a smoothed estimate $\boldsymbol{m}_{x,k-1}^{+}$ of the previous state $\boldsymbol{x}_{k-1}$ given $\boldsymbol{y}_{1:k}$. Using these approximations, the REKF is formulated to jointly compute estimates $\boldsymbol{m}_{x,k}^{+}$ and $\hat{\boldsymbol{\Theta}}_k$. The calculation of the required expectations is detailed in the parameter update step that is presented in the next Section.

### REKF Recursion

In terms of the augmented state $\mathbf{s}_k$, the REKF system model is given by

$$\boldsymbol{s}_k = \tilde{\boldsymbol{f}}_{\Omega}\left(\boldsymbol{s}_{k-1}\right) + \tilde{\boldsymbol{w}}_{k-1} \tag{25}$$

$$\boldsymbol{y}_k = \tilde{\boldsymbol{h}}_{\Omega}\left(\boldsymbol{s}_k\right) + \tilde{\boldsymbol{v}}_k, \tag{26}$$

where $\tilde{\boldsymbol{f}}_{\Omega}\left(\boldsymbol{s}_{k-1}\right) = \left[\left(\mathbf{A}\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right)\right)^T \quad \boldsymbol{x}_{k-1}^T\right]^T$, $\tilde{\boldsymbol{h}}_{\Omega}\left(\boldsymbol{s}_k\right) = \mathbf{B}\boldsymbol{\kappa}\left(\boldsymbol{x}_k\right)$ and $\tilde{\boldsymbol{v}}_k = \boldsymbol{v}_k$. The actual noise covariance matrix of $\tilde{\boldsymbol{w}}_{k-1} = \left[\boldsymbol{w}_{k-1}^T \quad \mathbf{0}_{1\times n_x}\right]^T$ is given by

$$\tilde{\mathbf{P}}_{ww,k-1} = \begin{bmatrix} \hat{\mathbf{P}}_{ww,k-1} & \mathbf{0}_{n_x\times n_x} \\ \mathbf{0}_{n_x\times n_x} & \mathbf{0}_{n_x\times n_x} \end{bmatrix}. \tag{27}$$

At the $k$-th time instant, an estimate of the previous recursion is given by

$$\boldsymbol{m}_{s,k-1}^{+} = \left[\left(\boldsymbol{m}_{x,k-1}^{+}\right)^T \quad \left(\boldsymbol{m}_{x_A,k-2}^{+}\right)^T\right]^T, \tag{28}$$

with respective error covariance matrix $\mathbf{P}_{ss,k-1}^{+}$, estimates of coefficient matrices $\hat{\mathbf{A}}_{k-1}$ and $\hat{\mathbf{B}}_{k-1}$, and estimates of noise covariance matrices $\hat{\mathbf{P}}_{ww,k-1}$ and $\hat{\mathbf{P}}_{vv,k-1}$. The current estimates $\boldsymbol{m}_{s,k}^{+}$, $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\hat{\mathbf{P}}_{ww,k}$ and $\hat{\mathbf{P}}_{vv,k}$ are computed based on the new available observation $\boldsymbol{y}_k$ through the following steps.

*Prediction.* Using $\mathbf{A}_k = \hat{\mathbf{A}}_{k-1}$ and Eq. (27) in Eq. (25), compute the predicted state and the associated prediction error covariance matrix as

$$\boldsymbol{m}_{s,k}^{-} = \tilde{\boldsymbol{f}}_{\Omega}\big(\boldsymbol{m}_{s,k-1}^{+}\big) \tag{29}$$

$$\mathbf{P}_{ss,k}^{-} = \tilde{\mathbf{F}}_{\Omega,k}\mathbf{P}_{ss,k-1}^{+}\tilde{\mathbf{F}}_{\Omega,k}^{T} + \tilde{\mathbf{P}}_{ww,k-1}, \tag{30}$$

where $\tilde{\mathbf{F}}_{\Omega,k} = \nabla_s \tilde{\boldsymbol{f}}_{\Omega}\left(\boldsymbol{s}\right)\Big|_{\boldsymbol{s}=\boldsymbol{m}_{s,k-1}^{+}}$.

*Measurement Update.* Using $\mathbf{B}_k = \hat{\mathbf{B}}_{k-1}$ and $\mathbf{P}_{vv,k} = \hat{\mathbf{P}}_{vv,k-1}$ in Eq. (26), compute

$$\boldsymbol{m}^{+}_{s,k} = \boldsymbol{m}^{-}_{s,k} + \mathbf{K}_k\big(\boldsymbol{y}_k - \boldsymbol{m}^{-}_{y,k}\big) \tag{31}$$

$$\mathbf{P}^{+}_{ss,k} = \mathbf{P}^{-}_{ss,k} - \mathbf{K}_k\tilde{\mathbf{H}}_{\Omega,k}\mathbf{P}^{-}_{ss,k} \tag{32}$$

$$\boldsymbol{m}^{-}_{y,k} = \tilde{\boldsymbol{h}}_{\Omega}\big(\boldsymbol{m}^{-}_{s,k}\big) \tag{33}$$

$$\mathbf{P}^{-}_{sy,k} = \mathbf{P}^{-}_{ss,k}\tilde{\mathbf{H}}^{T}_{\Omega,k} \tag{34}$$

$$\mathbf{P}^{-}_{yy,k} = \tilde{\mathbf{H}}_{\Omega,k}\mathbf{P}^{-}_{sy,k} + \mathbf{P}_{vv,k-1} \tag{35}$$

$$\mathbf{K}_k = \mathbf{P}^{-}_{sy,k}\big(\mathbf{P}^{-}_{yy,k}\big)^{-1}, \tag{36}$$

where $\tilde{\mathbf{H}}_{\Omega,k} = \nabla_s\tilde{\boldsymbol{h}}_{\Omega}\left(\boldsymbol{s}\right)\Big|_{\boldsymbol{s}=\boldsymbol{m}^{-}_{s,k}}$. In Eq. (30), $\boldsymbol{m}^{+}_{s,k} = \left[\big(\boldsymbol{m}^{+}_{x,k}\big)^{T} \quad \big(\boldsymbol{m}^{+}_{x,k-1}\big)^{T}\right]^{T}$, where $\boldsymbol{m}^{+}_{x,k-1}$ is the REKF's estimate of $\boldsymbol{x}_k$. The prediction and measurement update steps follow from the standard EKF recursions to estimate the augmented state $\boldsymbol{s}_k$ with the system model as given by Eqs. (25) and (26).

*Parameters Update.* The parameter estimates are updated by approximating the required expectations in Eqs. (17) - (23). Consider $\mathbb{E}_k\left\{\boldsymbol{x}_k\boldsymbol{\kappa}^{T}\left(\boldsymbol{x}_{k-1}\right)\right\}$ from Eq. (17). Based on the standard EKF, linearize $\boldsymbol{\kappa}\left(\cdot\right)$ as

$$\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right) \approx \boldsymbol{\kappa}\big(\boldsymbol{m}^{+}_{x,k-1}\big) + \nabla\boldsymbol{\kappa}\big(\boldsymbol{m}^{+}_{x,k-1}\big)\big(\boldsymbol{x}_{k-1} - \boldsymbol{m}^{+}_{x,k-1}\big), \tag{37}$$

where $\nabla\boldsymbol{\kappa}\big(\boldsymbol{m}^{+}_{x,k-1}\big) = \nabla_x\boldsymbol{\kappa}\left(\boldsymbol{x}\right)|_{\boldsymbol{x}=\boldsymbol{m}^{+}_{x,k-1}}$. Also, similar to the EKF, the assumption of negligible error in conditional means is taken. That is, $\mathbb{E}_k\left\{\boldsymbol{x}_k\right\} \approx \boldsymbol{m}^{+}_{x,k}$ and $\mathbb{E}_k\left\{\boldsymbol{x}_{k-1}\right\} \approx \boldsymbol{m}^{+}_{x,k-1}$. Substituting Eq. (37) into the expectations from Eqs. (17) - (23), yields

$$\mathbb{E}_k\left\{\boldsymbol{x}_k\boldsymbol{\kappa}^{T}\left(\boldsymbol{x}_{k-1}\right)\right\} = \boldsymbol{m}^{+}_{x,k}\boldsymbol{\kappa}^{T}\big(\boldsymbol{m}^{+}_{x,k-1}\big) + \mathbf{P}^{*}_{xx,k}\nabla\boldsymbol{\kappa}^{T}\big(\boldsymbol{m}^{+}_{x,k-1}\big), \tag{38}$$

and, by definition,

$$\mathbf{P}^{*}_{xx,k} \approx \big[\mathbf{P}^{+}_{ss,k}\big]_{(1:n_x,n_x+1:2n_x)} \tag{39}$$

$$\mathbf{P}^{+}_{xx,k-1} \approx \big[\mathbf{P}^{+}_{ss,k}\big]_{(n_x+1:2n_x,n_x+1:2n_x)} \tag{40}$$

$$\mathbf{P}^{+}_{xx,k} \approx \big[\mathbf{P}^{+}_{ss,k}\big]_{(1:n_x,1:n_x)}, \tag{41}$$

where the notation $\big[\mathbf{P}^{+}_{ss,k}\big]_{(a:b,c:d)}$ refers to the sub-matrix of the state covariance matrix $\mathbf{P}^{+}_{ss,k}$, where $a : b$ and $c : d$ are the rows and columns that form the sub-matrix, respectively. The remaining expectations are computed as

$$\begin{aligned}\mathbb{E}_k\big\{\boldsymbol{\kappa}\left(\boldsymbol{x}_{k-1}\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k-1}\right)\big\} &= \boldsymbol{\kappa}\big(\boldsymbol{m}^+_{x,k-1}\big)\boldsymbol{\kappa}^T\big(\boldsymbol{m}^+_{x,k-1}\big) \\ &\quad + \nabla\boldsymbol{\kappa}\big(\boldsymbol{m}^+_{x,k-1}\big)\mathbf{P}^+_{xx,k-1}\nabla\boldsymbol{\kappa}^T\big(\boldsymbol{m}^+_{x,k-1}\big) \\ \mathbb{E}_k\big\{\boldsymbol{\kappa}\left(\boldsymbol{x}_{k}\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k}\right)\big\} &= \boldsymbol{\kappa}\big(\boldsymbol{m}^+_{x,k}\big)\boldsymbol{\kappa}^T\big(\boldsymbol{m}^+_{x,k}\big) + \nabla\boldsymbol{\kappa}\big(\boldsymbol{m}^+_{x,k}\big)\mathbf{P}^+_{xx,k}\nabla\boldsymbol{\kappa}^T\big(\boldsymbol{m}^+_{x,k}\big) \\ \mathbb{E}_k\big\{\boldsymbol{x}_k\boldsymbol{\kappa}^T\left(\boldsymbol{x}_{k-1}\right)\big\} &= \boldsymbol{m}^+_{x,k}\big(\boldsymbol{m}^+_{x,k}\big)^T + \mathbf{P}^+_{xx,k},\end{aligned} \tag{42}$$

which are used to compute $\hat{\boldsymbol{\Theta}}_k$ using Eqs. (17) - (23). Further, by substituting Eq. (8) into the measurement expectations from Eq. (23), yields

$$\mathbb{E}_k\left\{\boldsymbol{y}_k\boldsymbol{y}_k^T\right\} = \hat{\mathbf{P}}_{vv,k-1} + \hat{\mathbf{B}}_k\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_k\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\}\hat{\mathbf{B}}_k^T \tag{43}$$

$$\mathbb{E}_k\left\{\boldsymbol{y}_k\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\} = \hat{\mathbf{B}}_k\mathbb{E}_k\left\{\boldsymbol{\kappa}\left(\boldsymbol{x}_k\right)\boldsymbol{\kappa}^T\left(\boldsymbol{x}_k\right)\right\}, \tag{44}$$

which are used to compute Eqs. (19) and (23).

*Dictionary Update.* The dictionary $\{\tilde{\boldsymbol{x}}_\ell\}_{\ell=1}^L$ is updated using the new estimate $\boldsymbol{m}^+_{x,k}$ based on the sliding window or ALD criterion.[15] The sliding window method consists of storing the last $L$ estimates (samples).

The ALD criterion is a process that considers the previous dictionary samples and compares them to the current state estimate. If the sample contains information that is relevant to the algorithm (if it is different to the previous ones by a defined tolerance), then the sample is added to the dictionary. In other words, the ALD criterion builds the dictionary using samples that are not approximately linearly dependent on the dictionary vectors. To build the dictionary via the ALD criterion, the importance of the last estimate is measured by computing the ALD coefficient $\varepsilon$, as follows

$$\varepsilon = \kappa_{s,k} - \boldsymbol{\kappa}_{v,k}\mathbf{K}^{-1}_{G,k}\boldsymbol{\kappa}^T_{v,k} \tag{45}$$

$$\kappa_{s,k} = \kappa\left(\boldsymbol{m}^+_{x,k}, \boldsymbol{m}^+_{x,k}\right) \tag{46}$$

$$\boldsymbol{\kappa}_{v,k} = \boldsymbol{\kappa}\left(\boldsymbol{m}^+_{x,k}\right) \tag{47}$$

$$\left[\mathbf{K}_{G,k}\right]_{ij} = \kappa\left(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j\right), \tag{48}$$

where $\mathbf{K}_{G,k}$ is the Gram matrix. These values are found after solving a minimization problem to find optimal expansion coefficients $\boldsymbol{a}_k = \mathbf{K}^{-1}_{G,k}\boldsymbol{\kappa}^T_{v,k}$, with $\varepsilon = \kappa_{s,k} - \boldsymbol{\kappa}_{v,k}\boldsymbol{a}_k \leq \delta$ for each sample of the dictionary, as shown in Section III-A from Reference 15. Note that the author uses row vectors to formulate the ALD criterion, while in this manuscript the preferred notation uses column vectors. The ALD coefficient is compared to a user-defined threshold $\delta$ (also known as the level of sparsification), such that if $\varepsilon > \delta$, the last sample is added to the dictionary. Otherwise, the dictionary remains intact.

### REKF Algorithm

The algorithms in Tables 1 and 2 summarize the initialization and recursion of the REKF, respectively. Note that the size of the dictionary increases when the current size $L$ is less than the window length considered (initial transient phase) in the sliding window criterion or when $\boldsymbol{m}_{x,k}^{+}$ is added to the dictionary based on the ALD criterion.

The REKF initialization reshapes the initial means and covariances to fit the sizes needed by the algorithm, i.e., according to the augmented states and measurements. The unknown system matrices, $\hat{\mathbf{A}}_0$ and $\hat{\mathbf{B}}_0$ are initialized with ones for simplicity (not identity matrices), but they could take any other suitable value as desired by the user.[6]

As was described in the Expectation-Maximization for Parameter Learning Section, the REKF adopts EKF recursions to obtain state estimates, while the unknown system parameters are learned using an online EM algorithm. Here, the *a priori* state estimate $\boldsymbol{m}_{s,k}^{-}$ and its respective measurement update $\boldsymbol{m}_{y,k}^{-}$ are generated using kernel function approximations with the previously learned system matrices, $\hat{\mathbf{A}}_{k-1}$ and $\hat{\mathbf{B}}_{k-1}$, respectively.

Note that the size of the dictionary increases when the current size $L$ is less than the window length considered (initial transient phase) in the sliding window criterion or when $\boldsymbol{m}_{x,k}^{+}$ is added to the dictionary based on the ALD criterion.

**Table 1. REKF initialization**

```
Input: $\boldsymbol{m}_{x,0}$, $\mathbf{P}_{xx,0}$
Output: $\boldsymbol{m}_{s,0}$, $\mathbf{P}_{ss,0}$, $L$, $\{\hat{\boldsymbol{x}}_\ell\}_{\ell=1}^{L}$, $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$, $\hat{\mathbf{P}}_{ww,0}$, $\hat{\mathbf{P}}_{vv,0}$, $\mathbf{S}_{x\kappa,0}$, $\mathbf{S}_{\kappa 1,0}$, $\mathbf{S}_{y\kappa,0}$, $\mathbf{S}_{\kappa 2,0}$
1: $\boldsymbol{m}_{s,0} \leftarrow \left[\boldsymbol{m}_{x,0}^{T} \quad \boldsymbol{m}_{x,0}^{T}\right]^{T}$ and $\mathbf{P}_{ss,0} \leftarrow \text{blkdiag}\big([\mathbf{P}_{xx,0} \quad \mathbf{P}_{xx,0}]\big)$.
2: Set $L = 1$ and $\hat{\boldsymbol{x}}_1 = \boldsymbol{m}_{x,0}$.
3: $\hat{\mathbf{A}}_0 \leftarrow \mathbf{1}_{n_x \times L}$ and $\hat{\mathbf{B}}_0 \leftarrow \mathbf{1}_{n_y \times L}$.
4: Initialize $\hat{\mathbf{P}}_{ww,0}$ and $\hat{\mathbf{P}}_{vv,0}$ with some suitable p.d.[*] noise covariance matrices.
5: Set $\mathbf{S}_{x\kappa,0} = \mathbf{0}_{n_x \times L}$, $\mathbf{S}_{\kappa 1,0} = \mathbf{0}_{L \times L}$, $\mathbf{S}_{y\kappa,0} = \mathbf{0}_{n_y \times L}$ and $\mathbf{S}_{\kappa 2,0} = \mathbf{0}_{L \times L}$.
 Return $\boldsymbol{m}_{s,0}$, $\mathbf{P}_{ss,0}$, $L$, $\{\hat{\boldsymbol{x}}_\ell\}_{\ell=1}^{L}$, $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$, $\hat{\mathbf{P}}_{ww,0}$, $\hat{\mathbf{P}}_{vv,0}$, $\mathbf{S}_{x\kappa,0}$, $\mathbf{S}_{\kappa 1,0}$, $\mathbf{S}_{y\kappa,0}$, $\mathbf{S}_{\kappa 2,0}$.
```

**Table 2. REKF recursion**

```
Input: $\boldsymbol{y}_k$, $\boldsymbol{m}_{s,k-1}^{+}$, $\mathbf{P}_{ss,k-1}^{+}$, $\hat{\mathbf{A}}_{k-1}$, $\hat{\mathbf{B}}_{k-1}$, $\hat{\mathbf{P}}_{ww,k-1}$, $\hat{\mathbf{P}}_{vv,k-1}$, $\mathbf{S}_{x\kappa,k-1}$, $\mathbf{S}_{\kappa 1,k-1}$, $\mathbf{S}_{y\kappa,k-1}$, $\mathbf{S}_{\kappa 2,k-1}$
Output: $\boldsymbol{m}_{x,k}^{+}$, $\boldsymbol{m}_{s,k}^{+}$, $\mathbf{P}_{ss,k}^{+}$, $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\hat{\mathbf{P}}_{ww,k}$, $\hat{\mathbf{P}}_{vv,k}$, $\mathbf{S}_{x\kappa,k}$, $\mathbf{S}_{\kappa 1,k}$, $\mathbf{S}_{y\kappa,k}$, $\mathbf{S}_{\kappa 2,k}$
1: Compute $\boldsymbol{m}_{s,k}^{-}$ and $\mathbf{P}_{ss,k}^{-}$ using Eqs. (29) and (30).
2: Compute $\boldsymbol{m}_{s,k}^{+}$ and $\mathbf{P}_{ss,k}^{+}$ using Eqs. (31) and (32).
3: $\boldsymbol{m}_{x,k}^{+} \leftarrow \left[\boldsymbol{m}_{s,k}^{+}\right]_{(1:n_x)}$.
4: $\mathbf{P}_{xx,k}^{*} \leftarrow \left[\mathbf{P}_{ss,k}^{+}\right]_{(1:n_x,n_x+1:2n_x)}$, $\mathbf{P}_{xx,k-1}^{+} \leftarrow \left[\mathbf{P}_{ss,k}^{+}\right]_{(n_x+1:2n_x,n_x+1:2n_x)}$ and $\mathbf{P}_{xx,k}^{+} \leftarrow \left[\mathbf{P}_{ss,k}^{+}\right]_{(1:n_x,1:n_x)}$.
5: Compute the expectations using Eq. (38) and Eqs. (42) - (44).
6: Compute $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\hat{\mathbf{P}}_{ww,k}$ and $\hat{\mathbf{P}}_{vv,k}$ using Eqs. (16) - (23).
7: Update the dictionary $\{\hat{\boldsymbol{x}}_\ell\}_{\ell=1}^{L}$ using $\boldsymbol{m}_{x,k}^{+}$ based on the sliding window or ALD criterion.
8: If dictionary size increases then augment $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\mathbf{S}_{x\kappa,k}$, $\mathbf{S}_{\kappa 1,k}$, $\mathbf{S}_{y\kappa,k}$ and $\mathbf{S}_{\kappa 2,k}$ with suitable initial values to take into account the updated dictionary size.
 Return $\boldsymbol{m}_{x,k}^{+}$, $\boldsymbol{m}_{s,k}^{+}$, $\mathbf{P}_{ss,k}^{+}$, $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\hat{\mathbf{P}}_{ww,k}$, $\hat{\mathbf{P}}_{vv,k}$, $\mathbf{S}_{x\kappa,k}$, $\mathbf{S}_{\kappa 1,k}$, $\mathbf{S}_{y\kappa,k}$, $\mathbf{S}_{\kappa 2,k}$.
```

## EXAMPLE: IN-PLANE CLOHESSY-WILTSHIRE EQUATIONS

This section presents simulation results designed to evaluate the performance of the REKF in a scenario where direct access to measurements is not available to the filter. To illustrate this, the in-

[*] Positive definite.

plane Clohessy-Wiltshire (CW) equations are used to model the motion of a *chaser* particle that is in close proximity to a *target*, with a mismatched filtering setup: the $A$-agent employs a conventional EKF, while the $\Omega$-agent uses the proposed REKF. Two scenarios are investigated to test the FEKF performance. The first case uses linear $A$-agent's observations, while nonlinear range observations are used for the second. Note that with linear system dynamics and measurements, the EKF becomes a Kalman filter (KF).

### Dyanmics Modeling

The CW equations are chosen for the simulations because they provide a well-known linearized representation of relative motion in a circular reference orbit, commonly used for spacecraft proximity operations and navigation.[20] The simulated system follows the model

$$\boldsymbol{x}_k = \begin{bmatrix} 4-3c & 0 & s/n & 2(1-c)/n \\ 6(s-n\Delta t) & 1 & 2(c-1)/n & (4s-3n\Delta t)/n \\ 3ns & 0 & c & 2s \\ 6n(c-1) & 0 & -2s & 4c-3 \end{bmatrix} \boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \tag{49}$$

where $n$ is the mean motion of the target, $\Delta t = t_k - t_{k-1}$, $s = \sin(n\Delta t)$, $c = \cos(n\Delta t)$, the in-plane state vector is given by the in-plane positions and velocities $\boldsymbol{x}_k = \begin{bmatrix} x_{1,k} & x_{2,k} & \dot{x}_{1,k} & \dot{x}_{2,k} \end{bmatrix}^T$ and $\boldsymbol{s}_k = \begin{bmatrix} \boldsymbol{x}_k^T & \boldsymbol{x}_{k-1}^T \end{bmatrix}^T$ is the augmented state.

For the simulations, the target is taken to be orbiting around Earth in a circular orbit with a semimajor axis of $6775$ $km$, and the gravitational parameter of the Earth given by $\mu = 398600.442$ $km^3/s^2$.

### Initial Conditions

The initial estimates for the mean and covariance of the object are given by

$$\mathbf{m}_{x,0} = \begin{bmatrix} 1000 & 0 & -1.23 & -1.73 \end{bmatrix}^T \tag{50}$$

$$\mathbf{P}_{xx,0} = \operatorname{diag}\left(\begin{bmatrix} 1\times10^2 & 1\times10^2 & 4\times10^{-6} & 4\times10^{-6} \end{bmatrix}\right), \tag{51}$$

in units for positions and velocities of $m$ and $m/s$, respectively. The initial state is assumed to be drawn from a Gaussian distribution with mean and covariance as shown in Eqs. (50) and (51), respectively.

### Scenario #1: Linear Measurements

The first scenario uses linear measurements of position in the $A$F, as given by

$$\boldsymbol{y}_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \boldsymbol{v}_k \tag{52}$$

$$\boldsymbol{z}_k = \mathbf{B}\boldsymbol{\kappa}\left(\boldsymbol{m}_{s,k}\right) + \boldsymbol{u}_k, \tag{53}$$

where $\boldsymbol{y}_k$ are the $A$-agent measurements and $\boldsymbol{z}_k$ the $\Omega$-agent observations. Recall that the REKF does not have access to the system model in Eqs. (49) and (52). Instead, the REKF uses approximations of these as in Eqs. (25) and (53). The REKF is tasked with estimating the state indirectly, using noisy observations $\boldsymbol{z}_k$ rather than raw measurements.

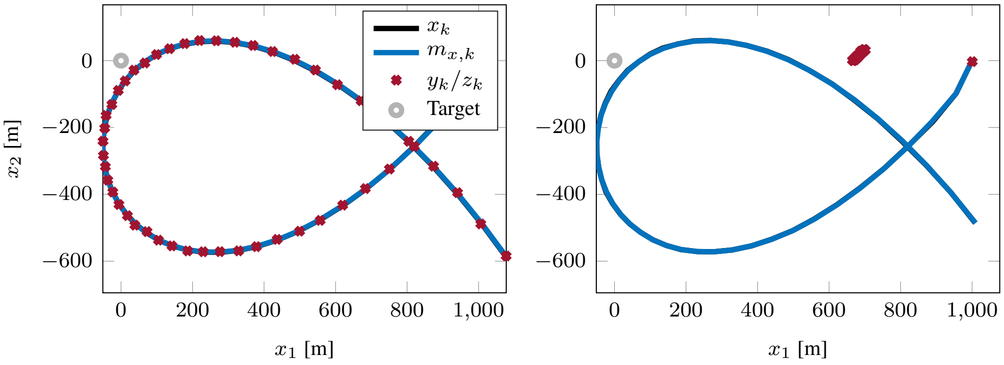
The initial process noise covariance is taken to have values of

$$\mathbf{P}_{ww,k-1} = \operatorname{diag}\big(\begin{bmatrix} 1\times10^{-6} & 1\times10^{-6} & 1\times10^{-9} & 1\times10^{-9}\end{bmatrix}\big), \tag{54}$$

in units of $m^2$ and $m^2/s^2$. Samples of the process noise are drawn from a zero-mean Gaussian distribution to simulate noise effects on the evolution of the true states. The initial measurement noise covariances use values of $\mathbf{P}_{vv,k-1} = 4\mathbf{I}_2$ and $\mathbf{P}_{vv,k-1} = 4$, respectively. For the $\Omega$F, the initial process noise covariance is taken to have values of $\tilde{\mathbf{P}}_{ww,k-1} = \big(1\times10^{6}\big)\mathbf{P}_{ww,k-1}$, while its measurement noise covariance was initialized as $\mathbf{P}_{uu,k-1} = \big(0.1\big)\mathbf{P}_{vv,k-1}$. This choice reflects the need for the $\Omega$F to initially explore a wider range of dynamic behaviors (via a larger process noise) while placing greater confidence in early measurements (via a smaller measurement noise), thereby accelerating the learning of unknown model components.

The sliding window was chosen to build the dictionary in with $L = 50$, the kernel approximation uses a smoothing coefficient of $\sigma = \sqrt{1\times10^{9}}$ and the online approximate EM algorithm uses a tolerance of $1\times10^{-3}$ to estimate the parameters. The simulation uses a fixed time-step and a final time $t_f = 2800\ s$.

A Monte Carlo analysis comprising 1000 trials over 50 time steps is conducted to statistically characterize the performance of the filter. The average mean squared error (AMSE) is used to assess estimation accuracy. Figure 1 demonstrates the state trajectories, filter estimates, and observations for both the $A$F and $\Omega$F in a representative trial. An interesting finding is that the $\Omega$-agent's measurements don't track the $A$-agent's observations. The REKF uses the available information contained in the dictionary and the current posterior $A$F estimate to produce new observations. The REKF tracks the true state effectively despite the system model being unknown.



(a)

(b)

**Figure 1. Illustration of the true and estimated states, as well as the observations of the (a) $A$F and (b) $\Omega$F for one of the trials, with linear $A$F observations.**

Figures 2 and 3 show the ensemble average errors and standard deviations for both the $A$F and the $\Omega$F across all trials, revealing that the filter produces unbiased estimates, with error envelopes consistently contained within the Monte Carlo intervals.

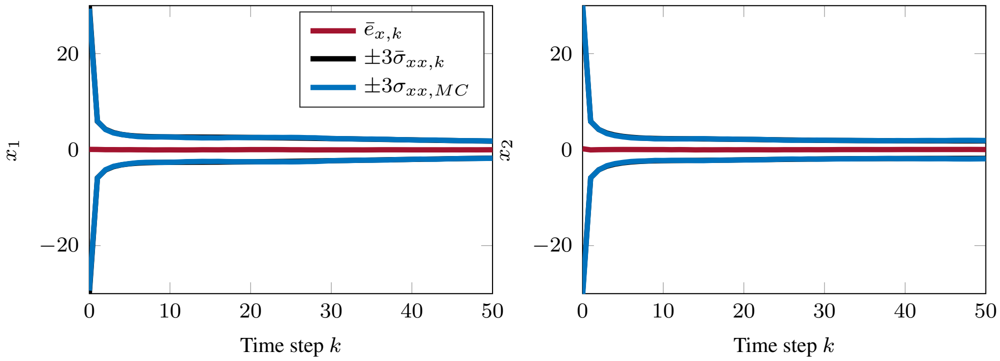


**Figure 2. Illustration of the average and trial error accompanied with their respective average and Monte Carlo intervals for the $A$F with linear $A$F observations.**

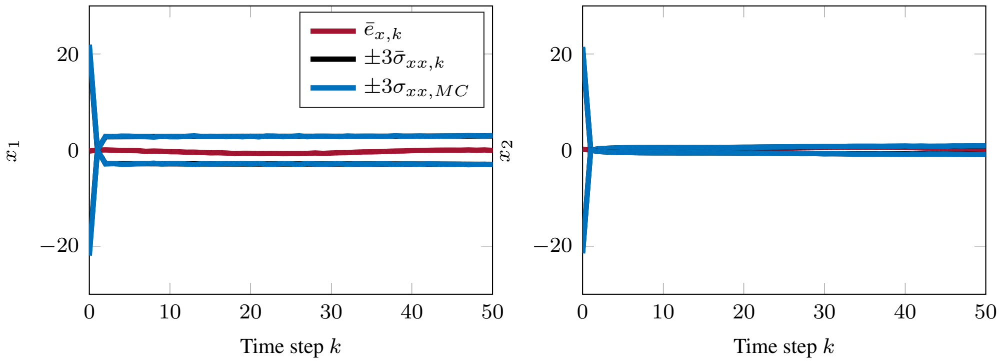


**Figure 3. Illustration of the average and trial error accompanied with their respective average and Monte Carlo intervals for the $\Omega$F with linear $A$F observations.**

On average, the total run time of one trial for the given simulation settings is approximately $10.43\ s$. Note that the maximum window length is used to test the time-performance of the filter, but smaller window length could be used to considerably reduce the run time.

Most notably, the REKF ($\Omega$F) consistently outperforms the EKF ($A$F) in terms of estimation accuracy and uncertainty. As illustrated in Figure 4, the REKF achieves a lower AMSE than the EKF, despite relying on approximate dynamics and indirect measurements. This improvement highlights the REKF's ability to learn and compensate for unmodeled dynamics through its kernel-based structure and online parameter adaptation. In particular, the REKF maintains reliable estimation performance even when the assumed system model is incomplete, indicating its potential for scenarios where accurate analytical models are unavailable or measurements are highly indirect.

### Scenario #2: Nonlinear Measurements

This scenario uses nonlinear range measurements of position in the $A$F, as given by

$$\boldsymbol{y}_k = \sqrt{x_{1,k}^2 + x_{2,k}^2} + \boldsymbol{v}_k, \tag{55}$$

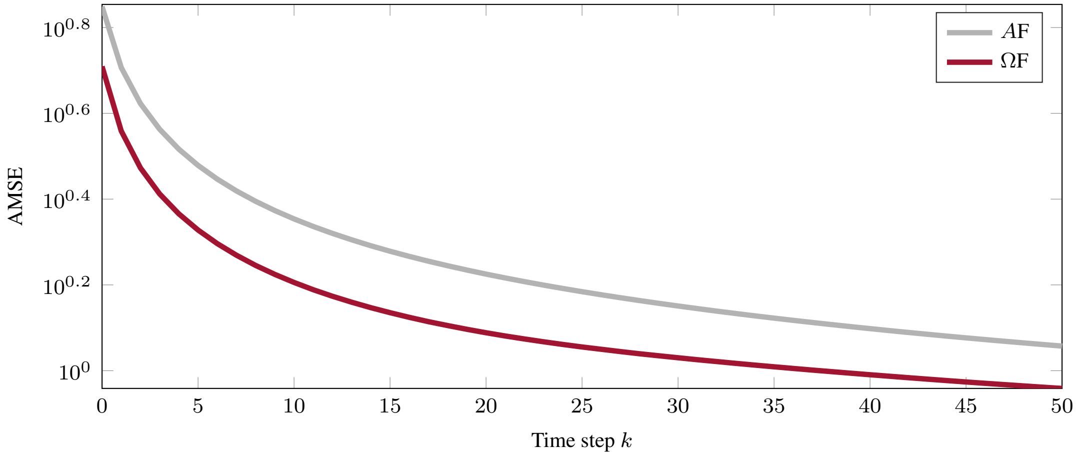


**Figure 4. Illustration of the AMSE for the $A$ and $\Omega$ filters, with linear $A$F observations.**

where $\boldsymbol{y}_k$ are the $A$-agent measurements and $\boldsymbol{z}_k$ the $\Omega$-agent observations, as given by Eq. (53).

The initial process noise covariance was taken to have values of

$$\mathbf{P}_{ww,k-1} = \text{diag}\left(\begin{bmatrix} 1\times10^{-6} & 1\times10^{-6} & 1\times10^{-9} & 1\times10^{-9}\end{bmatrix}\right), \tag{56}$$

in units of $m^2$ and $m^2/s^2$. Samples of the process noise are drawn from a zero-mean Gaussian distribution to simulate noise effects on the evolution of the true states. The initial measurement noise covariances use values of $\mathbf{P}_{vv,k-1} = 4\mathbf{I}_2$ and $\mathbf{P}_{vv,k-1} = 4$, respectively. For the $\Omega$F, the initial process noise covariance is taken to have values of $\tilde{\mathbf{P}}_{ww,k-1} = \left(1\times10^{6}\right)\mathbf{P}_{ww,k-1}$, while its measurement noise covariance was initialized as $\mathbf{P}_{uu,k-1} = \left(0.1\right)\mathbf{P}_{vv,k-1}$.

Again, the sliding window was chosen to build the dictionary with $L = 50$, the kernel approximation uses a smoothing coefficient of $\sigma = \sqrt{1\times10^{9}}$ and the online approximate EM algorithm uses a tolerance of $1\times10^{-3}$ to estimate the parameters. The simulation uses a fixed time-step and a final time $t_f = 2800\ s$.

Similarly as in the first scenario, a Monte Carlo analysis comprising 1000 trials over 50 time steps is conducted to statistically characterize the performance. Figures 5 and 6 demonstrate the state trajectories, filter estimates, and observations for both filters in a representative trial.

Figures 7 and 8 show the ensemble average errors and standard deviations of both filters across all trials, revealing that the filter produces unbiased estimates, with error envelopes consistently contained within the Monte Carlo intervals.

On average, the total run time of one trial for the given simulation settings is approximately $10.45\ s$. The REKF ($\Omega$F) consistently outperforms the EKF ($A$F) in terms of estimation accuracy and uncertainty.

As illustrated in Figure 9, the REKF achieves a lower AMSE than the EKF, despite relying on approximate dynamics and indirect measurements. These findings highlight the potential of the REKF to provide accurate state estimates under limited knowledge, setting a strong foundation for

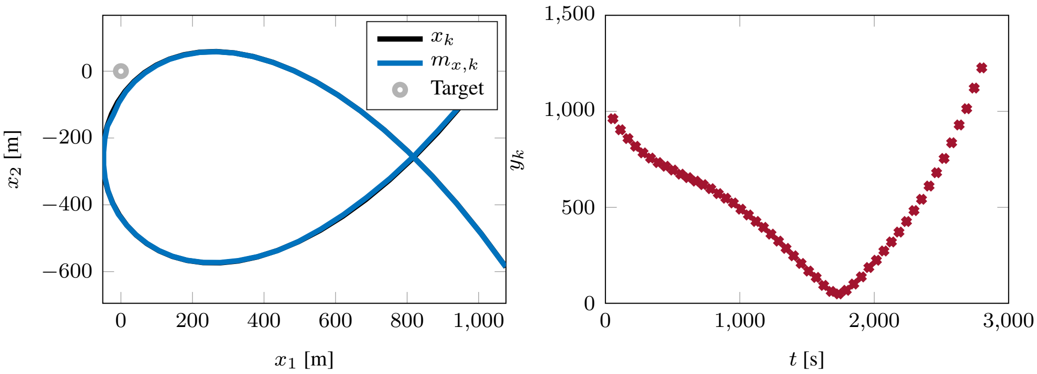


**Figure 5. Illustration of the true and estimated states, as well as the observations of the $A$F for one of the trials with nonlinear $A$F observations.**

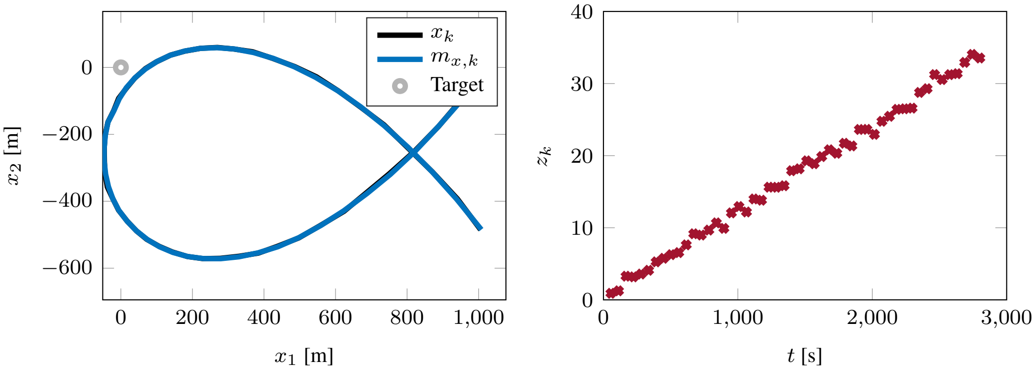


**Figure 6. Illustration of the true and estimated states, as well as the observations of the $\Omega$F for one of the trials with nonlinear $A$F observations.**

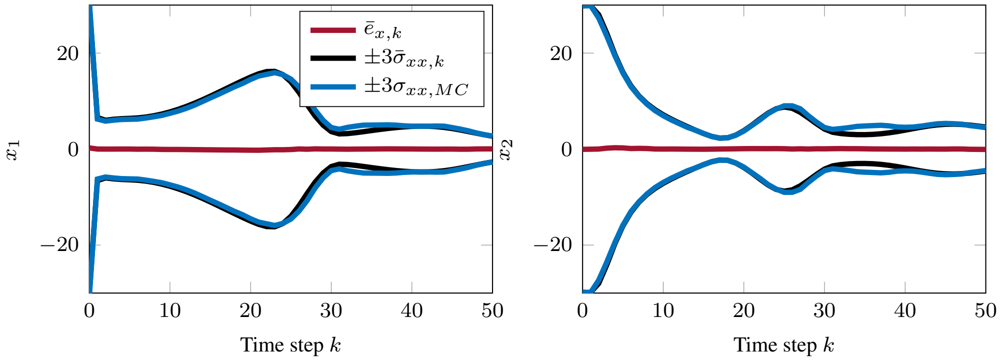


**Figure 7. Illustration of the average and trial error accompanied with their respective average and Monte Carlo intervals for both the (a) $A$F and (b) $\Omega$F, with nonlinear $A$F observations.**

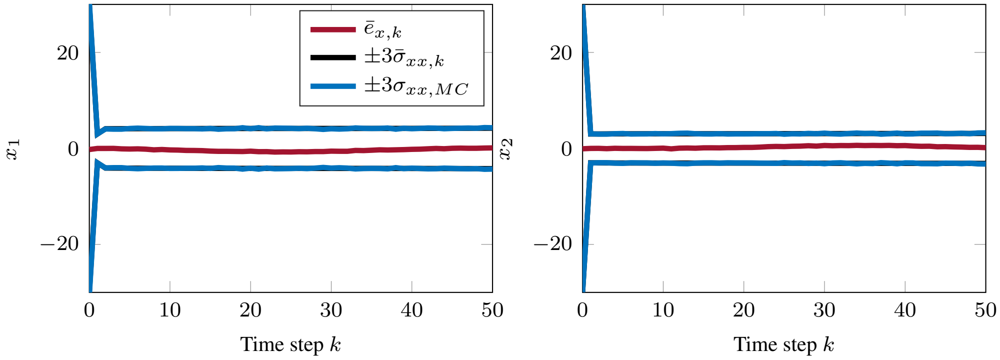


**Figure 8. Illustration of the average and trial error accompanied with their respective average and Monte Carlo intervals for both the (a) $A$F and (b) $\Omega$F, with nonlinear $A$F observations.**

future applications in nonlinear and adversarial scenarios.

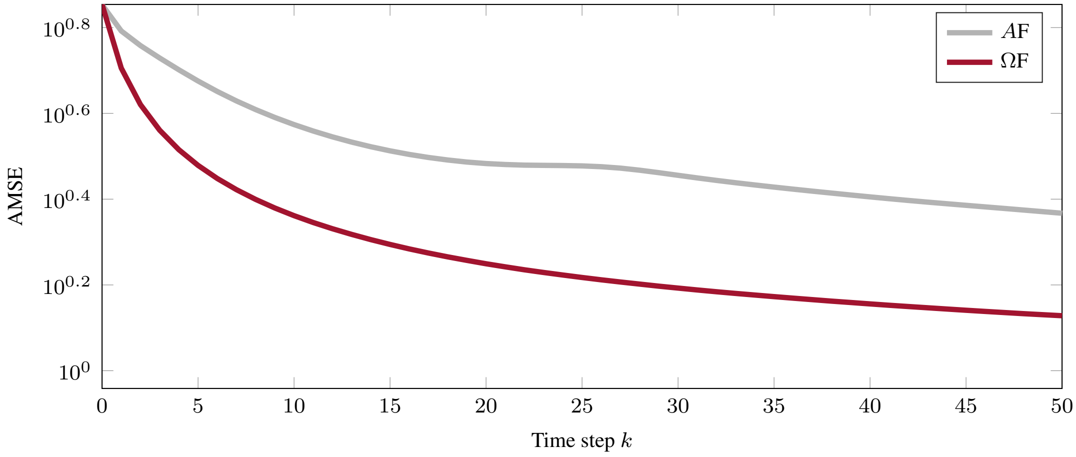


**Figure 9. Illustration of the AMSE for the $A$ and $\Omega$ filters, with nonlinear $A$F observations.**

The results reveal several key differences between the two measurement scenarios. In Scenario #1 (linear measurements), both the $A$F and the $\Omega$F exhibit well-behaved estimation errors. The REKF achieves lower AMSE compared to the EKF even though it operates with approximate dynamics and indirect measurements. The error envelopes remain tight and unbiased, with most of the $\Omega$F error trajectories staying within the standard deviation bounds computed from Monte Carlo trials.

In Scenario #2 (nonlinear measurements), the differences become more pronounced. The nonlinear range measurements introduce additional state-observation coupling, which generally increases the estimation difficulty for the EKF. As shown in Figure 9, the EKF AMSE remains consistently higher compared to Scenario #1, reflecting its sensitivity to unmodeled nonlinearities. In contrast, the REKF adapts its kernel-based dynamics and measurement models online, maintaining error bounds comparable to those in the linear case. This suggests that the REKF effectively learns and compensates for nonlinear observation effects.

Another noticeable distinction is the transient behavior is that the REKF shows a short initial learning phase where errors momentarily increase, particularly in Scenario #2, before converging to a steady-state accuracy that surpasses that of the EKF. This behavior is consistent with the REKF's reliance on data-driven dictionary updates and parameter adaptation. Once sufficient training data have been processed, the REKF stabilizes and provides robust performance.

Overall, these comparisons indicate that the REKF's advantage grows as measurement complexity increases. While both filters perform adequately under simple, linear measurements, the REKF shows clear robustness and adaptability under nonlinear measurement conditions where traditional EKF performance degrades.

## CONCLUSION

The results of this work demonstrate that the reproducing kernel Hilbert space extended Kalman filter (REKF) is capable of producing unbiased estimates, with error envelopes consistently contained within the Monte Carlo intervals. On the other hand, the REKF achieves a low average mean squared error (AMSE) despite relying on approximate dynamics and observations. In other words, the REKF is a practical tool to provide accurate estimates under limited knowledge.

The REKF maintained consistent performance across both linear and nonlinear measurement scenarios, whereas the EKF exhibited degraded accuracy under nonlinear range measurements. The adaptive, kernel-based modeling of the REKF enabled effective compensation for unmodeled dynamics and nonlinear observation effects, resulting in reduced steady-state estimation errors relative to the EKF.

A brief transient learning phase was observed in the REKF due to online parameter adaptation, after which the filter converged to reliable and robust estimation performance. These results demonstrate that the REKF provides a flexible and robust alternative to conventional model-based filters, particularly in scenarios where the measurement model is nonlinear or only partially known.

The REKF framework offers a robust solution for state estimation in cases where highly nonlinear dynamics, limited observations, and disturbances present significant challenges. By leveraging kernel-based approximations and an approximate online expectation-maximization (EM) algorithm, the REKF is able to adapt to changing system dynamics and provide accurate state estimates despite uncertain conditions. This makes it an ideal tool for resilient spacecraft navigation, threat detection, and active sensing in uncertain environments, such as cislunar space.

## REFERENCES

[1] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, "A Novel Adaptive Kalman Filter With Inaccurate Process and Measurement Noise Covariance Matrices," *IEEE Transactions on Automatic Control*, Vol. 63, No. 2, 2018, pp. 594–601, 10.1109/TAC.2017.2730480.

[2] H. Zhu, G. Zhang, Y. Li, and H. Leung, "An Adaptive Kalman Filter With Inaccurate Noise Covariances in the Presence of Outliers," *IEEE Transactions on Automatic Control*, Vol. 67, No. 1, 2022, pp. 374–381, 10.1109/TAC.2021.3056343.

[3] S. Yi and M. Zorzi, "Robust Kalman Filtering Under Model Uncertainty: The Case of Degenerate Densities," *IEEE Transactions on Automatic Control*, Vol. 67, No. 7, 2022, pp. 3458–3471, 10.1109/TAC.2021.3106861.

[4] S. M. K. Mohamed and S. Nahavandi, "Robust Finite-Horizon Kalman Filtering for Uncertain Discrete-Time Systems," *IEEE Transactions on Automatic Control*, Vol. 57, No. 6, 2012, pp. 1548–1552, 10.1109/TAC.2011.2174697.

[5] H. Singh, A. Chattopadhyay, and K. V. Mishra, "Inverse Extended Kalman Filter—Part I: Fundamentals," *IEEE Transactions on Signal Processing*, Vol. 71, 2023, pp. 2936–2951, 10.1109/TSP.2023.3304756.

[6] H. Singh, A. Chattopadhyay, and K. V. Mishra, "Inverse Extended Kalman Filter—Part II: Highly Nonlinear and Uncertain Systems," *IEEE Transactions on Signal Processing*, Vol. 71, 2023, pp. 2952–2967, 10.1109/TSP.2023.3304761.

[7] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*. Prentice Hall, 2001.

[8] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. CRC Press, 2004.

[9] A. Taghvaei, J. d. Wiljes, P. G. Mehta, and S. Reich, "Robust Finite-Horizon Kalman Filtering for Uncertain Discrete-Time Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 140, No. 3, 2017, p. 030904, 10.1115/1.4037780.

[10] T. Lange and W. Stannat, "On the continuous time limit of the ensemble Kalman filters," *Mathematics of Computation*, Vol. 90, No. 327, 2020, pp. 233–265, 10.1090/mcom/3588.

[11] Z. Zhang, J. Zhu, S. Zhang, and F. Gao, "Process monitoring using recurrent Kalman variational auto-encoder for general complex dynamic processes," *Engineering Applications of Artificial Intelligence*, Vol. 123, 2023, p. 106424, 10.1016/j.engappai.2023.106424.

[12] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, Vol. 68, 1950, pp. 337—404, 10.1090/S0002-9947-1950-0051437-7.

[13] F. Tobar, P. M. Djurić, and D. P. Mandic, "Unsupervised State-Space Modeling Using Reproducing Kernels," *IEEE Transactions on Signal Processing*, Vol. 63, No. 19, 2015, pp. 5210–5221, 10.1109/TSP.2015.2448527.

[14] L. Ralaivola and F. d'Alché Buc, "Dynamical modeling with kernels for nonlinear time series prediction," *Advances in Neural Information Processing Systems*, 2003, 10.1109/TSP.2015.2448527.

[15] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, Vol. 52, No. 8, 2004, pp. 2275–2285, 10.1109/TSP.2004.830985.

[16] W. Liu, I. Park, Y. Wang, and J. C. Principe, "Extended Kernel Recursive Least Squares Algorithm," *IEEE Transactions on Signal Processing*, Vol. 57, No. 10, 2009, pp. 3801–3814, 10.1109/TSP.2009.2022007.

[17] S. Van Vaerenbergh, J. Via, and I. Santamaria, "A Sliding-Window Kernel RLS Algorithm and Its Application to Nonlinear Channel Identification," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006, 10.1109/TSP.2009.2022007.

[18] B. Schölkopf, R. Herbrich, and A. J. Smola, "A Generalized Representer Theorem," *Computational Learning Theory*, 2001, 10.1007/3-540-44581-1_27.

[19] B. Hajek, *Random Processes for Engineers*. Cambridge University Press, 2015.

[20] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Elsevier Butterworth-Heinemann, 2005.