Inverse Extended Kalman Filter—Part II: Highly Nonlinear and Uncertain Systems

Himali Singh, Arpan Chattopadhyay, Member, IEEE, and Kumar Vijay Mishra, Senior Member, IEEE

Abstract—Counter-adversarial system design problems have lately motivated the development of inverse Bayesian filters. For example, inverse Kalman filter (I-KF) has been recently formulated to estimate the adversary’s Kalman-filter-tracked estimates and hence, predict the adversary’s future steps. The purpose of this paper and the companion paper (Part I) is to address the inverse filtering problem in non-linear systems by proposing an inverse extended Kalman filter (I-EKF). The companion paper proposed the theory of I-EKF (with and without unknown inputs) and I-KF (with unknown inputs). In this paper, we develop this theory for highly non-linear models, which employ second-order, Gaussian sum, and dithered forward EKFs. In particular, we derive theoretical stability guarantees for the inverse second-order EKF using the bounded non-linearity approach. To address the limitation of the standard I-EKFs that the system model and forward filter are perfectly known to the defender, we propose reproducing kernel Hilbert space-based EKF to learn the unknown system dynamics based on its observations, which can be employed as an inverse filter to infer the adversary’s estimate. Numerical experiments demonstrate the state estimation performance of the proposed filters using recursive Crâmer-Rao lower bound as a benchmark.

Index Terms—Bayesian filtering, counter-adversarial systems, extended Kalman filter, inverse filtering, non-linear processes.

I. INTRODUCTION

In the companion paper (Part I) [1], we introduced inverse cognition [2] in a non-linear setting and developed inverse extended Kalman filters (I-EKFs). Recall that the inverse cognition problem involves two agents: ‘defender’ (e.g., an intelligent target) and an ‘adversary’ (e.g., a radar) equipped with a (forward) Bayesian tracker. The forward filter provides a posterior distribution of the underlying state to the adversary based on its noisy observations. The cognitive adversary further adapts its actions based on this estimate. For instance, a cognitive radar adapts its waveform to enhance target detection and tracking [3], [4]. The defender observes these adversary’s actions to predict its future actions in a Bayesian sense and hence, requires an estimate of the adversary’s estimate. The inverse cognition problem is motivated by the design of counter-adversarial systems, e.g., an intelligent target observing its adversarial radar’s actions to guard against the latter [2]. Other applications include interactive learning [2], fault diagnosis, cyber-physical security [5], and inverse reinforcement learning [6].

An inverse Bayesian filter provides an estimate of the forward Bayesian filter’s posterior distribution given its noisy measurements. In particular, [7] and [2] developed, respectively, the inverse hidden Markov model (I-HMM) filter and inverse Kalman filter (I-KF) for linear Gaussian state-space models. The tracked estimates from these inverse filters were then used to design smart interference and force the adversarial radar to change its waveform in [8], [9]. In practice, many engineering processes are non-linear and employ extended Kalman filter (EKF) based on local linearization at the state estimates [10], [11].

In the companion paper (Part I) [1], we developed inverse EKF (I-EKF) for a non-linear inverse cognition problem. However, the standard EKF has its limitations. Unlike KF, the filter and (filter) gain equations of EKF are not decoupled and hence offline computations of these EKF quantities are not possible. The EKF performs poorly when the dynamic system is significantly non-linear. It is also very sensitive to initialization and may even completely fail [12]. Recently, KF’s initialization has been studied rigorously in [13], while [14] deals with process noise covariance design. However, they do not consider the non-linear EKF case. These EKF drawbacks have led to the development of several advanced variants, each aiming toward improving either the estimation accuracy (higher-order EKFs [15], [16], Gaussian-sum EKF [17]), convergence (iterated EKF [18]), stability (dithered EKF [19]) or practical feasibility (hybrid EKF [11]). The literature also reports grid-based EKF [20] that is a pre-cursor of particle filter. A comparison of non-linear KFs is available in [21], [22]. These filters reduce the linearization errors that are inherent to the EKF and may provide an improved estimation at the cost of higher complexity and computations. In particular, this paper focuses on developing...
the I-EKF theory for highly nonlinear systems that may employ some of the aforementioned EKF variants as the forward filter.

Further, the inverse stochastic filters developed in [2, 7] and the companion paper (Part I) [1] assume perfect system model information on the part of both adversary and defender. However, in practice, the defender may not have information about the adversary’s forward filter and the strategy employed by the adversary to adapt its actions. Similarly, the adversary may also lack complete system information, including the noise distributions. In such situations, the performance and applicability of inverse filters is limited. The uncertainty in system parameters or noise statistics may result in large estimation errors or even filter divergence [23, 24]. Prior works [23, 25, 26, 27, 28] proposed various adaptive filters to reduce the sensitivity of KF to system uncertainties. However, the general non-linear filtering problem is not considered.

In this paper, we also consider the case of unknown forward filter and system model. Among prior works, for a forward non-linear filtering problem with unknown system model, [29] developed extended kernel recursive least squares (Ex-KRLS) method by transforming the input space to a kernel-based feature space. The KRLS [30] was coupled with EKF to learn the unknown non-linear measurement model in [31, 32], but with a known linear state-transition. In [33], the KF algorithm was reformulated for non-linear state transition functions in reproducing kernel Hilbert space (RKHS) using a conditional embedding operator, but linear observations were assumed. Particle filter based approaches for state estimation and system identification in non-linear system setting were considered in [34], [35]. For linear Gaussian state-space models, an iterative expectation maximization (EM)-based parameter learning was proposed in [36]. We adopt this iterative EM for non-linear parameter learning in our RKHS-based EKF.

Our main contributions in this paper (Part II) are:

1) Inverses of EKF variants. We develop inverse second-order EKF (I-SOEkf) and provide sufficient conditions for its stability. In the process, we obtain stability conditions for forward SOEkf as well and derive the one-step prediction formulation of SOEkf. The presence of second-order terms poses additional challenges in deriving the theoretical stability guarantees for both forward and inverse SOEkf. In addition, we obtain the inverse of Gaussian-sum EKF (GS-EKF) which comprises of several individual EKFs [17], [37]. Here, the \( a \) posteriori density function of the state given the observations is approximated by a sum of Gaussian density functions each of which has its own separate EKF. GS-EKF and hence, its inverse, not only provides enhanced estimation accuracy but is also a popular method to handle non-Gaussianity in system models. In situations where the estimation error is small, the \( a \) posteriori density is approximated adequately by one Gaussian density, and the Gaussian sum filter reduces to the EKF; the same is true for their inverses. Finally, we also examine the inverse of dithered EKF (DEKF) [19], which has been shown to perform better when non-linearities are cone-bounded. This filter introduces dither signals prior to the non-linearities to tighten the cone-bounds and hence, improve stability properties.

We validate the estimation errors of all inverse EKFs through extensive numerical experiments with recursive Cramér-Rao lower bound (RCRLB) [38] as the performance metric.

2) RKHS-EKF. We consider the case when both the state-transition and observation models are unknown to the agent employing a stochastic filter in a non-linear system. In this context, we propose RKHS-based EKF which may be employed by both adversary and defender, respectively, to estimate defender’s state and adversary’s state estimate. In particular, the unknown non-linear functions are represented using kernel function approximation. The EKF provides required state estimates while an online approximate EM is used to learn the system parameters. The presence of non-linear transformations in expectation computation makes the filter derivation non-trivial.

3) Mismatched forward and inverse filters. The inverse filter theory formulated in the companion paper (Part I) [1] assumed a known adversary’s forward filter. If the defender is not certain about the adversary’s forward filter, our proposed RKHS-EKF is helpful in estimating the adversary’s state estimate. However, when the defender employs an inverse filter assuming a forward filter that is not the same as the adversary’s true forward filter, our numerical experiments show that the defender is still able to estimate the adversary’s state estimate with reasonable accuracy. Further, a sophisticated inverse filter may even provide better performance at the cost of computational efforts.

The rest of the paper is organized as follows. In the next section, we describe the system model for the inverse cognition problem. We then discuss the inverses of some EKF variants in Section III. The stability conditions for forward and inverse SOEkf are provided in Section IV. In Section V, we relax the perfect system information assumption and derive RKHS-EKF. Finally, in Section VI, we demonstrate the performance of the proposed inverse filters considering various numerical experiments before concluding in Section VII.

Throughout the paper, we reserve boldface lowercase and uppercase letters for vectors (column vectors) and matrices, respectively. The notation used to denote the \( i \)-th component of vector \( a \) and the \((i, j)\)-th component of matrix \( A \) is \( [a]_i \) and \([A]_{i,j}\), respectively, while \([A]_{i_1\cdots i_d;j_1\cdots j_d}\) represents the submatrix of \( A \) consisting of rows \( i_1 \) to \( i_d \) and columns \( j_1 \) to \( j_d \). We use \((\cdot)^T\) for transpose operation while \(|\cdot|_2\) represents the \( l_2 \) norm for a vector. For matrix \( A \), the trace, determinant, rank, spectral norm and maximum row sum norm are denoted by \( Tr(A) \), \(|A|_1 \), \(|A| \), and \(|A|_\infty \), respectively. The inequality \( A \preceq B \) for matrices \( A \) and \( B \) means that \( B - A \) is a positive semidefinite (p.s.d.) matrix. The \( \mathbb{R}^{m \times n} \) Jacobian matrix of a function \( f : \mathbb{R}^n \to \mathbb{R}^m \) is denoted by \( \nabla f \). Similarly, \( \nabla f \) and \( \nabla^2 f \) denote the gradient vector (\( \mathbb{R}^{m \times n} \)) and Hessian matrix (\( \mathbb{R}^{m \times n} \)), respectively, for a function \( f : \mathbb{R}^n \to \mathbb{R} \). \( I_n \) denotes a \( n \times n \) identity matrix while \( 0_{m \times n} \) denotes a \( n \times m \) all zero matrix. The notation \( \{a_l\}_{l \leq i \leq l + 1} \) denotes a set of elements indexed by integer \( i \). The Gaussian random variable is represented as \( x \sim \mathcal{N}(\mu, Q) \) with mean \( \mu \) and covariance matrix \( Q \) while \( x \sim \mathcal{U}[u_l, u_u] \) means a uniform distribution over \([u_l, u_u]\).
The covariance of random variable \( x \) is denoted by \( \text{Cov}(x) \) while \( \text{Cov}(x, y) \) denotes its cross-covariance with random variable \( y \).

II. DESIRERATA FOR INVERSE COGNITION

Throughout the paper, we focus on discrete-time models. Recall from the companion paper (Part I) [1] that the defender’s discrete-time stochastic state process \( \{x_k\}_{k \geq 0} \), where \( x_k \in \mathbb{R}^{n \times 1} \) is the state at the \( k \)-th time instant, evolves as

\[
x_{k+1} = f(x_k) + w_k,
\]

where \( w_k \sim \mathcal{N}(0_{n \times 1}, Q) \) is the process noise with covariance matrix \( Q \in \mathbb{R}^{n \times n} \). The defender knows its own state perfectly, while the adversary observes the defender’s state in noise as \( y_k \in \mathbb{R}^{p \times 1} \) at time \( k \) as

\[
y_k = h(x_k) + v_k,
\]

where \( v_k \sim \mathcal{N}(0_{p \times 1}, R) \) is the adversary’s measurement noise with covariance matrix \( R \in \mathbb{R}^{p \times p} \).

The adversary’s (forward) stochastic filter computes the estimate \( \hat{x}_k \) of the defender’s state \( x_k \) using \( \{y_j\}_{1 \leq j \leq k} \). The adversary then takes an action \( g(\hat{x}_k) \in \mathbb{R}^{n \times 1} \) using this estimate, whose noisy observation made by the defender is

\[
a_k = g(\hat{x}_k) + \epsilon_k,
\]

with the true state \( x_{k+1} \) and \( v_{k+1} \), respectively, being a known exogenous input and process noise term for the defender’s I-EKF. Here, \( K_{k+1} \) is the forward EKF’s gain matrix, which the I-EKF treats as a time-varying parameter of state transition (4), approximating it using its own estimates as described in the companion paper (Part I) [1]. Finally, using state transition (4) and observation (3), I-EKF follows from the EKF recursions as

**Prediction:** \( \hat{x}_{k+1|k} = f_k(\hat{x}_k, x_{k+1}, 0_{p \times 1}), \quad \Sigma_{k+1|k} = P_k \Sigma (P_k)^T + Q_k \),

**Update:** \( \Sigma_{k+1} = \Sigma_{k+1|k} - \Sigma_{k+1|k} G_{k+1}^T \Sigma_{k+1|k}^{-1} G_{k+1} \),

where \( G_{k+1} = \nabla f_k(x, x_{k+1}, 0_{p \times 1}) \mid_{x = x_k, \hat{x}_k + \epsilon_k} \),

with \( F_k = \nabla^2 f_k(x, x_{k+1}, 0_{p \times 1}) \mid_{x = x_k, \hat{x}_k + \epsilon_k} \).

III. INVERSES FOR HIGHLY NON-LINEAR SYSTEMS

Among various highly nonlinear cases discussed earlier, we focus on the inverses of two-step SOEKF, GS-EKF, and DEKF. In the process, we also derive the inverse of one-step prediction formulation of SOEKF. The one-step prediction formulation is about predicting the next state and analytically more useful while deriving the stability conditions of the filter. Although the two-step prediction-update recursion differs from the one-step prediction formulation in performance and transient behaviour, both have similar convergence properties [43, Section III].

A. Inverse SOEKF

While the EKF formulations are limited to only first-order Taylor series expansion, the forward and inverse filters of SOEKF also include second-order terms. Here, we define the Jacobians as \( F_k = \nabla f(x) \mid_{x = x_k} \) and \( H_{k+1} = \nabla^2 f(x) \mid_{x = x_k} \).

1) **Forward Filter:** At each time step, the SOEKF first predicts the current state \( x_{k+1} \) as \( \hat{x}_{k+1|k} \) with the associated prediction error covariance matrix \( \Sigma_{k+1|k} \) using its previous state estimate. In the update step, the predicted observation estimate \( \hat{y}_{k+1|k} \) and associated error covariance matrix \( \Sigma_{k+1|k} \) are computed based on the predicted state estimate. Finally, the current updated state estimate \( \hat{x}_{k+1} \) and its error covariance matrix \( \Sigma_{k+1} \) are obtained considering the current observation \( y_{k+1} \) and the predicted estimates. Denoting the \( i \)-th Euclidean basis vectors in \( \mathbb{R}^{n \times 1} \) and \( \mathbb{R}^{p \times 1} \) by \( a_i \) and \( b_i \), respectively, the forward SOEKF recursions are [44]

**Prediction:** \( \hat{x}_{k+1|k} = f(\hat{x}_k) + \frac{1}{2} \sum_{i=1}^{n} a_i^T \text{Tr} \left( \nabla^2 f(\hat{x}_k) \right) a_i \Sigma_k \),

\[
\Sigma_{k+1|k} = F_k \Sigma_k F_k^T + Q + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i b_j^T \text{Tr} \left( \nabla^2 f(\hat{x}_k) \right) a_j \Sigma_k \nabla^2 \left( f(\hat{x}_k) \right) a_i \Sigma_k,
\]
Update:
\[
\hat{y}_{k+1|k} = h(\hat{x}_{k+1|k}) + \frac{1}{2} \sum_{i=1}^{p} b_i \text{Tr} \left( \nabla^2 \left[ h(\hat{x}_{k+1|k}) \right] \right)_{i} \Sigma_{k+1|k},
\]
\[
\Sigma_{k+1} = \Sigma_{k+1|k} + \left. H_{k+1} \right| \Sigma_{k+1|k},
\]
\[
+ \frac{1}{2} \sum_{i=1}^{p} b_i b_i^T \text{Tr} \left( \nabla^2 \left[ h(\hat{x}_{k+1|k}) \right] \right)_{i} \Sigma_{k+1|k},
\]
\[
\hat{x}_{k+1} = \hat{x}_{k+1|k} + \Omega,\Sigma_{k+1|k} - \Sigma_{k+1|k},
\]
\[
\hat{x}_{k+1} = \hat{x}_{k+1|k} - \Sigma_{k+1|k} H_{k+1} \Sigma_{k+1|k}.
\]
where \( \Sigma_{k+1} \) is SOEFK’s gain matrix. Unlike EKF, the SOEFK includes Hessian terms in the prediction equations.

2) Inverse Filter: Define \( f(\hat{x}, \Sigma) = f(\hat{x}) + \frac{1}{2} \sum_{i=1}^{n} \tilde{a}_i \text{Tr} \left( \nabla^2 \left[ f(\hat{x}) \right] \Sigma \right)_i \) and \( \hat{h}(\hat{x}, \Sigma) = \hat{h}(\hat{x}) + \frac{1}{2} \sum_{i=1}^{n} b_i \text{Tr} \left( \nabla^2 \left[ h(\hat{x}) \right] \Sigma \right)_i \).

Using (2) and (7)–(9), the state transition equation for the inverse SOEFK (I-SOEFK) is
\[
\hat{x}_{k+1} = f(\hat{x}_k, \Sigma_k) + \Sigma_k h(\hat{x}_k) + K_{k+1} v_{k+1},
\]
The I-SOEFK recursions now follow directly from forward SOEFK’s recursions treating \( a_k \) from (3) as the observation. Denote the Jacobians with respect to the state estimate \( \hat{x}_k \) as \( \tilde{F}_k = \nabla_x f(\hat{x}, x_{k+1}, 0) \), while the Hessians are denoted as \( \nabla^2 \left[ f(\hat{x}) \right] \Sigma \Sigma \). Similar to I-EKF [1], I-SOEFK also follows
\[
\Omega_{k} = \nabla_x R \nabla_x^T \nabla_k \nabla_{k+1},
\]
with \( \nabla_{k+1} = \nabla_x g(x) \big|_{x = \hat{x}_{k+1}} \), while the Hessians are
denoted as \( \nabla^2 \left[ f(\hat{x}) \right] \Sigma \Sigma \). The process noise covariance matrix is \( \Sigma_k = \nabla_k \Sigma \nabla_{k+1} \) with \( \nabla_{k+1} = \nabla_x \nabla_k \nabla_{k+1} \) from (3).

Remark 1: Here, besides the gain matrix \( \Sigma_{k+1} \), the Hessian terms are also treated as time-varying parameters of the state transition and approximated by evaluating them using the previous I-SOEFK’s estimate \( \hat{x}_k \).

Remark 2: When the second-order terms are neglected, the forward SOEFK and I-SOEFK reduce to, respectively, forward EKF and I-EKF.

Remark 3: Similar to I-EKF [1], I-SOEFK also follows from general SOEFK recursions and hence, has similar computational complexity, i.e., \( O(n^5) \) where \( n \) is the state dimension. Note that a derivative-free efficient implementation [45] of SOEFK yields a \( O(n^4) \) complexity.

3) One-Step Prediction Formulation: The one-step prediction formulation of SOEFK for the considered system model can be derived following the derivation of two-step recursion formulation of SOEFK as outlined in [44]. Denoting the Jacobians as \( \tilde{F}_k = \nabla_x f(\hat{x}) \big|_{x = \hat{x}_k} = \hat{h}(\hat{x}_k), \) the one-step prediction SOEFK recursions are
\[
\tilde{x}_k = f(\hat{x}_k) + \frac{1}{2} \sum_{i=1}^{n} \tilde{a}_i \text{Tr} \left( \nabla^2 \left[ f(\hat{x}_k) \right] \Sigma_k \right),
\]
\[
\tilde{\Sigma}_k = \tilde{F}_k \Sigma_k \tilde{F}_k^T + \Omega,
\]
\[
\tilde{x}_{k+1} = \tilde{x}_k + \tilde{h}(\tilde{x}_k) + \frac{1}{2} \sum_{i=1}^{n} \tilde{b}_i \text{Tr} \left( \nabla^2 \left[ h(\tilde{x}_k) \right] \Sigma_k \right) \tilde{x}_k,
\]
\[
\tilde{\Sigma}_{k+1} = \Sigma_k + \tilde{F}_k \Sigma_k \tilde{F}_k^T + \Sigma_k,
\]
where \( \Sigma_k = \Sigma_k \) from (3) and \( \tilde{F}_k = \nabla_x f(\hat{x}, \Sigma) \big|_{x = \hat{x}_k} \). After these prediction steps, the update steps to compute estimate \( \hat{x}_{k+1} \) and \( \Sigma_{k+1} \) are same as (5) and (6), respectively.

Considering the aforementioned state transition and observation (3), the inverse filter follows, \( \text{mutatis mutandis} \), from SOEFK’s one-step prediction formulation. The Jacobians with respect to the state estimate are \( \tilde{F}_k = \nabla_x f(\hat{x}) \big|_{x = \hat{x}_k} = \hat{h}(\hat{x}_k) \)
\[ K_k H_k \text{ and } G_k = \nabla_x g(x) |_{x=\hat{x}_k} \text{, with the process noise covariance matrix } \bar{Q}_k = K_k R_k K_k^T. \]

### B. Inverse GS-EKF

The GS-EKF \[46\] assumes the posterior distribution of the state estimate to be a weighted sum of Gaussian densities, which are updated recursively based on the current observation. This multiple-model filtering technique is also widely used to handle non-Gaussian process and measurement noises in the system model \[47\]. Hence, the I-GS-EKF developed in the following is also applicable to non-Gaussian inverse filtering problems.

1) **Forward Filter:** In the forward GS-EKF, we consider \( l \) Gaussians denoting the \( i \)-th Gaussian's probability density function, with mean \( \bar{x}_{i,k} \) and covariance \( \Sigma_{i,k} \) at \( k \)-th time-step, as
\[
\gamma(x - \bar{x}_{i,k}, \Sigma_{i,k}) = \left(2\pi\right)^{-n/2} |\Sigma_{i,k}|^{-1/2} e^{-\frac{1}{2}(x - \bar{x}_{i,k})^T \Sigma_{i,k}^{-1} (x - \bar{x}_{i,k})}
\]

Given observations \( Y_k = \{y_j|1 \leq j \leq k \} \) up to \( k \)-th time instant, the posterior distribution of state \( x_k \), is approximated as
\[
p(x_k|Y_k) = \sum_{i=1}^{l} c_{i,k} \gamma(x_k - \bar{x}_{i,k}, \Sigma_{i,k}), \quad \text{where } c_{i,k} \text{ is } i \text{-th Gaussian's weight at } k \text{-th time step.}
\]

Linearizing the functions as \( F_k x = \nabla_x f(x)|_{x=\bar{x}_{i,k}} \text{ and } H_k h_k = \nabla_x h(x)|_{x=\bar{x}_{i,k}} \) for the \( i \)-th Gaussian, the means \( \{\bar{x}_{i,k}\}_{1 \leq i \leq l} \) and covariance matrices \( \{\Sigma_{i,k}\}_{1 \leq i \leq l} \) are updated based on the current observation \( y_{k+1} \) using independent EKF recursions for each Gaussian. Finally, the weights \( \{c_{i,k}\}_{1 \leq i \leq l} \) are updated as \[46\]:
\[
c_{i,k} = \sum_{i'=1}^{l} c_{i',k} \gamma(y_{k+1} - h(\bar{x}_{i',k+1})), \quad S_{i,k+1} = H_{i,k+1}^T \Sigma_{i,k}^{-1} H_{i,k+1} + R
\]

where \( \Sigma_{i,k} = H_{i,k} \Sigma_{i,k} H_{i,k}^T + R \) and \( \bar{x}_{i,k} \) are the prediction covariance matrix of \( x_{i,k+1} \). With these updated Gaussians, the point-estimate \( \hat{x}_k \) and the associated covariance matrix \( \Sigma_k \) is determined as \( \hat{x}_k = \sum_{i=1}^{l} c_{i,k} \bar{x}_{i,k} \) and \( \Sigma_k = \sum_{i=1}^{l} c_{i,k} \Sigma_{i,k} + \sum_{i=1}^{l} c_{i,k} (\bar{x}_{i,k} - x_{i,k}) (\bar{x}_{i,k} - x_{i,k})^T \).

2) **Inverse Filter:** Consider an augmented state vector \( z_k = \{x_k, \ldots, x_{k+1}\} \) (means and weights of forward GS-EKF). Then, substituting for observation \( y_{k+1} \) from (2) in the forward filter’s updates similar to I-SOEKF and denoting the \( i \)-th EKF’s gain matrix as \( K_{i,k+1} = \Sigma_{i,k+1} H_{i,k+1}^T S_{i,k+1}^{-1} \), yields the state transition equations for inverse GS-EKF (I-GS-EKF) as
\[
\begin{align*}
\bar{x}_{i,k+1} &= f(\bar{x}_{i,k}) + K_{i,k+1} (h(x_{k+1}) + v_{k+1} - h(f(\bar{x}_{i,k}))) \\
c_{i,k+1} &= \sum_{i'=1}^{l} c_{i',k} \gamma(h(x_{k+1}) + v_{k+1} - h(f(\bar{x}_{i',k})).
\end{align*}
\]

(18)

Treating \( \{K_{i,k+1}, S_{i,k+1}\}_{1 \leq i \leq l} \) as time-varying parameters of the state transition equations, which are approximated in a similar way as we approximated the gain matrices for I-EKF and I-SOEKF, the overall state transition in terms of the augmented state is \( z_{k+1} = F_k z_k + v_k \). Similarly, the observation \( a_k \) as a function of augmented state \( z_k \) is \( a_k = g(z_k) + \epsilon_k \).

The I-GS-EKF approximates the posterior distribution of the augmented state as a sum of \( l \) Gaussians with its recursions again following directly from forward GS-EKF’s recursions treating \( a_k \) as the observation. However, the inverse filter estimates an \((n+1)\)-dimensional augmented state \( z_k \) with the Jacobians with respect to the state denoted as \( F_j z_k = \nabla_x f(z_k) |_{z=\bar{z}_{j,k}} \) and \( G_j z_k = \nabla_x g(z) |_{z=\bar{z}_{j,k+1}} \), and the process noise covariance matrix as \( \bar{Q}_k = V_j R V_j^T \).

### C. Inverse DEKF

Consider the adversary employing DEKF \[19\] as its forward filter. In DEKF, the output non-linearities are modified using dither signals so as to tighten the cone-bounds. Dithering tightens this cone such that the non-linearities are smoothed but it may also degrade the near-optimal performance of the EKF after the initial transient phase of estimation. Therefore, dithering is introduced only during the initial transient phase with the aim to improve the filter’s transient performance and avoid divergence. Denote the dither amplitude which controls the tightness of the cone-bounds by \( d \) and its amplitude probability density function by \( p(d) \). The observation function \( h(x) \) is dithered as \( h^*(x) = \int_h^x h(a + x) p(a) da \). If \( d = d_0 e^{-k/\tau} \), where \( d_0 \) and \( \tau \) are constants and \( k \) denotes the time index, then \( h^*(x) \) exponentially as \( k \to \infty \) during the transient phase.

The forward DEKF follows from conventional forward EKF described in Section II (detailed in the companion paper (Part I) \[1\, Section III-C]) by replacing \( h(x) \) with \( h^*(x) \) as the observation function of \( y_k \) during the initial transient phase and hence, the inverse DEKF (I-DEKF) also follows from I-EKF restated in Section II. The dither of the adversary’s filter is assumed to be known to us. Otherwise, the I-DEKF may also proceed with the unmodified observation function. We show in Section VI-B that these two formulations, labeled I-DEKF-1 and I-DEKF-2, respectively, generally vary in their estimation performances especially during the transient phase where the modified observation function is considered.
Remark 6: A DEKF differs from a standard EKF only because of the modified observation function and hence, has the same $O(n^3)$ computational complexity as EKF. The same argument holds for I-DEKF, which also follows from standard I-EKF recursions.

IV. STABILITY ANALYSES

Similar to I-EKF stability analysis in the companion paper (Part I) [1], we first obtain stability conditions of forward SOEKF in the exponential-boundedness-mean-squared sense and then, extend those results to I-SOEKF’s stability. In particular, we adopt the bounded non-linearity approach [43] and consider the one-step prediction formulation of forward and inverse SOEKF derived in Section III.A.3. As shown in [43, Section III], the convergence behaviors of both one- and two-step formulations are similar. In the following, we rely on the definitions of the exponential-boundedness and boundedness-with-probability-one of a stochastic process as well as a useful lemma from [43] that were restated in the companion paper (Part I) [1, Definition 1 and 2, and Lemma 1].

A. Forward SOEKF Stability

Consider the one-step SOEKF’s formulation (10)–(16). Considering second-order terms as well, the Taylor series expansion of functions $f(\cdot)$ and $h(\cdot)$ at the estimate $x_k$ are

$$f(x_k) - f(\hat{x}_k) = F_k(x_k - \hat{x}_k) + \frac{1}{2} \sum_{i=1}^{n} a_i (x_k - \hat{x}_k)^T \nabla^2 f(\hat{x}_k)_{ii} (x_k - \hat{x}_k) + \phi(x_k, \hat{x}_k),$$

$$h(x_k) - h(\hat{x}_k) = H_k (x_k - \hat{x}_k) + \frac{1}{2} \sum_{i=1}^{p} b_i (x_k - \hat{x}_k)^T \nabla^2 h(\hat{x}_k)_{ii} (x_k - \hat{x}_k) + \chi(x_k, \hat{x}_k),$$

where $\phi(\cdot)$ and $\chi(\cdot)$ are suitable non-linear functions to account for third and higher-order terms in the equations. Using these expansions, the error dynamics of the forward filter with $\epsilon_k = x_k - \hat{x}_k$ is

$$\epsilon_{k+1} = (F_k - K_k H_k) \epsilon_k + r_k + q_k + s_k, \quad (19)$$

where

$$r_k = \phi(x_k, \hat{x}_k) - K_k \chi(x_k, \hat{x}_k),$$

$$q_k = \frac{1}{2} \sum_{i=1}^{n} a_i \epsilon_k^T \nabla^2 [f(\hat{x}_k)]_{ii} \epsilon_k - \frac{1}{2} \sum_{i=1}^{n} a_i \text{Tr} (\nabla^2 [f(\hat{x}_k)]_{ii} \Sigma_k) - \frac{1}{2} K_k \sum_{i=1}^{p} b_i \epsilon_k^T \nabla^2 [h(\hat{x}_k)]_{ii} \epsilon_k + \frac{1}{2} K_k \sum_{i=1}^{p} b_i \text{Tr} (\nabla^2 [h(\hat{x}_k)]_{ii} \Sigma_k),$$

$$s_k = w_k - K_k v_k.$$

The following Theorem 1 provides sufficient conditions for the stochastic stability of forward SOEKF.

Theorem 1: (Exponential boundedness of forward SOEKF’s error). Consider the non-linear stochastic system defined by (1) and (2), and SOEKF’s one-step prediction formulation (10)–(16). Let the following assumptions hold true.

1) There exist positive real numbers $\eta, \tau, \sigma, q, q, \tau, \delta$ and real numbers $\sigma, \xi$ (not necessarily positive) such that the following bounds are satisfied for all $k \geq 0$.

$$\sigma I \lessgtr \Sigma_k \lessgtr \tau I,$$

$$\|F_k\| \leq \eta, \quad \|H_k\| \leq \tau,$$

$$\|G_k\| \leq \delta I,$$

$$\|q_k\| \leq K \sigma \leq \delta I,$$

$$\|q_k\| \leq \nabla^2 [f(\hat{x}_k)]_{ii} \leq \tau I \quad \forall i \in \{1, 2, \ldots, n\},$$

$$\|b_k\| \leq \nabla^2 [h(\hat{x}_k)]_{ii} \leq \tau I \quad \forall j \in \{1, 2, \ldots, p\}.$$  

2) $F_k$ is non-singular and $F_k^{-1}$ satisfies the following bound for all $k \geq 0$ for some positive real number $\bar{f}$,

$$\|F_k^{-1}\| \leq \bar{f}.$$  

3) There exist positive real numbers $\kappa_\phi, \kappa_\sigma, \kappa_\chi, \kappa_\epsilon$ such that the non-linear functions $\phi(\cdot)$ and $\chi(\cdot)$ satisfy

$$\|\phi(x, \hat{x})\|_2 \leq \kappa_\phi \|x - \hat{x}\|_2 \quad \text{for} \quad \|x - \hat{x}\|_2 \leq \sigma,$$

$$\|\chi(x, \hat{x})\|_2 \leq \kappa_\chi \|x - \hat{x}\|_2 \quad \text{for} \quad \|x - \hat{x}\|_2 \leq \epsilon.$$  

Then the estimation error given by (19) is exponentially bounded in mean-squared sense if the estimation error is within $\epsilon$ bound for a suitable constant $\epsilon > 0$,

$$\bar{f} < \frac{2\epsilon}{\eta \sigma^2 \sqrt{n \eta p}},$$

$$\bar{q} > \sigma,$$

$$\delta = \frac{1}{\kappa_\text{noise}} \left( \frac{\alpha \epsilon^2}{2 \sigma} - \epsilon_{\text{sec}} \right),$$

for some $\bar{\epsilon} \leq \epsilon$, where $\epsilon, \alpha, \kappa_\text{noise}$ and $\epsilon_{\text{sec}}$ are constants that depend on the bounds assumed on the system.

Proof: See Appendix A.

Remark 7: It follows from the proof of Lemma 2 (see Appendix A) that the constant $c$ depends on $\delta$. The conditions (21) and (22) suggest that $\delta$ should be chosen appropriately so that both the conditions could be satisfied simultaneously. However, the exact bounds on $\delta$ depend on other bounds assumed on the system matrices in Theorem 1.

Remark 8: Furthermore, as with the bounded non-linearity approach for EKF, these bounds may be conservative and, consequently, the estimation error may remain bounded outside this range [43, Section V].

B. Inverse SOEKF Stability

Considering a suitable non-linear function $\chi(\cdot)$, the Taylor series expansion of $g(\cdot)$ at the estimate $\hat{x}_k$ of I-SOEKF’s one-step prediction formulation is

$$g(\hat{x}_k) - g(\hat{x}_k) = G_k (\hat{x}_k - \hat{x}_k) + \frac{1}{2} \sum_{i=1}^{n_k} d_i (\hat{x}_k - \hat{x}_k)^T \nabla^2 [g(\hat{x}_k)]_{ii} (\hat{x}_k - \hat{x}_k),$$

where $d_i$ is the $i$-th Euclidean basis vector in $\mathbb{R}^{n_k \times 1}$. Finally, the error dynamics of the inverse filter with estimation error
denoted by $\xi_k = \hat{x}_k - \hat{x}_k$ and the inverse filter’s Kalman gain and estimation error covariance matrix by $K_k$ and $\Sigma_k$, respectively, is

$$
\xi_{k+1} = (F_k - K_kG_k)\xi_k + r_k + q_k + \xi_k,
$$

(23)

where

$$
\xi_k = \frac{1}{2} \sum_{i=1}^{n} \tilde{a}_i \xi_k \Sigma_k \left[ f(\hat{x}_k) \right] \xi_k
$$

and estimation error covariance matrix by

$$
\xi_k = \frac{1}{2} \sum_{i=1}^{n} \tilde{a}_i \xi_k \Sigma_k \left[ f(\hat{x}_k) \right] \xi_k
$$

and

$$
\xi_k = \frac{1}{2} \sum_{i=1}^{n} \tilde{a}_i \xi_k \Sigma_k \left[ f(\hat{x}_k) \right] \xi_k
$$

| Remark 9: | As discussed in the companion paper (Part I) [1, Section V-C], the inequality $mI \preceq \Sigma_k \preceq mI$ assumed in Theorem 2 is closely related to the theoretical observability of the non-linear inverse filtering model. Further, the inequality $mI \preceq \Sigma_k \preceq mI$ is a weaker assumption than observability. | □ |

| Remark 10: | Furthermore, the I-EKF’s consistency results of [1, Theorem 6] can be trivially extended to I-SOEKF’s and I-GS-EKF’s state estimates. Note that a GS-EKF’s state estimate is a weighted sum of a finite number of independent EKF recursions, which are consistent as shown in [1, Theorem 6]. |

V. UNKNOWN FORWARD FILTER AND MODEL

In practice, the defender’s state transition function $f(\cdot)$ may not be known to the adversary. Similarly, the defender may not know the specific forward filter employed by the adversary and the adversary’s function $g(\cdot)$ of (3) for computing its actions. In this context, we propose a recursive RKHS-EKF to jointly compute the desired state estimates and learn the unknown functions. In particular, we approximate an unknown function $s(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ by a function in RKHS induced by a kernel $k(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ [49]. The RKHS-based function approximation has been proposed for non-linear state-space modeling [34, 50] and recursive least-squares algorithms with non-linear unknown functions [30, 51, 52]. The representer theorem [53] ensures that the optimal approximation in RKHS with respect to an arbitrary loss function takes the form $s(\cdot) \approx \sum_{i=1}^{M} a_i k(x_i, \cdot)$, where $\{x_i\}_{1 \leq i \leq M}$ are the input training samples or dictionary and $\{a_i\}_{1 \leq i \leq M}$ are the corresponding mixing parameters to be learnt. Often, a Gaussian kernel such as

$$
\kappa(x_i, x_j) = \exp \left( -\frac{|x_i - x_j|^2}{\sigma^2} \right),
$$

with kernel width $\sigma > 0$ controlling the smoothness of the approximation, is commonly used. It is a universal kernel [54], i.e., its induced RKHS is dense in the space of continuous functions.

In the following, we consider a general non-linear system model, wherein both state transition and observation functions are unknown to the agent employing the filter, and develop a general RKHS-EKF. The defender can employ the RKHS-EKF as its inverse filter without assuming any prior information about the adversary’s forward filter. Our RKHS-EKF may be trivially simplified to yield forward RKHS-EKF for the adversary which knows its observation function. In particular, our RKHS-EKF adopts EKF to obtain the state estimates while the unknown system parameters are learnt using EM algorithm [55]. The EM algorithm is widely used to compute maximum likelihood estimates in presence of missing data.

System models for uncertain dynamics: We examine the non-linear state transition (1) and observation (2) with the functions $f(\cdot)$ and $h(\cdot)$ as well as the noise covariances $Q$ and $R$ being unknown. Consider a kernel function $\kappa(\cdot, \cdot)$ and a dictionary $\{x_i\}_{1 \leq i \leq L}$ of size $L$. Denote $\Phi(x) = [\kappa(x_1, x) \ldots \kappa(x_L, x)]^T$. Using the kernel function approximation, the filter’s state transition and observation are approximated, respectively, as

$$
x_{k+1} = \Phi(x_k) + w_k,
$$

(24)

$$
y_k = \Phi(x_k) + v_k,
$$

(25)
where the coefficient matrices \( A \in \mathbb{R}^{n \times L} \) and \( B \in \mathbb{R}^{p \times L} \) consists of the unknown mixing parameters to be learnt. The dictionary \( \{ \{ x_i \} \}_{1 \leq i \leq L} \) can be formed using a sliding window [52] or approximate linear dependency (ALD) [30] criterion. Besides the state estimate \( \hat{x}_k \), the RKHS-EKF needs to estimate the unknown parameters \( \Theta = \{ A, B, Q, R \} \) based on the observations \( \{ y_i \}_{1 \leq i \leq k} \) up to \( k \)-th time step.

In Section V-A, we derive an approximate online EM to obtain the parameter estimates, which is then coupled with EKF to formulate the RKHS-EKF recursions in Section V-B. Finally, the RKHS-EKF to jointly estimate the state and system parameters is summarized in Algorithms 1 and 2.

### A. Parameter Learning

Given \( \Theta \), the state estimate \( \hat{x}_k \) can be computed using EKF-based recursions. To estimate the unknown parameters \( \Theta \), we consider EM algorithm [55].

**EM for parameter learning:** Consider the states up to time \( k \) as \( X^k = \{ x_j \}_{0 \leq j \leq k} \) and the corresponding observations \( Y^k = \{ y_j \}_{1 \leq j \leq k} \). The joint conditional probability density given the parameters \( \Theta \) is

\[
p(X^k, Y^k | \Theta) = p(x_0) \prod_{j=1}^{k} p(x_j | x_{j-1}, \Theta) \prod_{j=1}^{k} p(y_j | x_j, \Theta). \tag{26}
\]

Under the Gaussian noise assumption, the conditional probability densities are

\[
p(x_j | x_{j-1}, \Theta) = \gamma(x_j - A \Phi(x_{j-1}), Q), \tag{27}
\]

\[
p(y_j | x_j, \Theta) = \gamma(y_j - B \Phi(x_j), R), \tag{28}
\]

with the Gaussian probability density function \( \gamma(\cdot, \cdot) \) as defined in Section III-B. Assume \( x_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_0) \). Note that this assumption is also needed to initialize the EKF recursions. Using this in (26) along with (27) and (28), we have

\[
\log p(X^k, Y^k | \Theta) = - \frac{1}{2} \log | \Sigma_0 | - \frac{1}{2} (x_0 - \bar{x}_0)^T \Sigma_0^{-1} (x_0 - \bar{x}_0) + \sum_{j=1}^{k} \left( - \frac{1}{2} \log | Q | - \frac{1}{2} (x_j - A \Phi(x_{j-1}))^T Q^{-1} (x_j - A \Phi(x_{j-1})) \right) + \sum_{j=1}^{k} \left( - \frac{1}{2} \log | R | - \frac{1}{2} (y_j - B \Phi(x_j))^T R^{-1} (y_j - B \Phi(x_j)) \right) + \alpha, \tag{29}
\]

where \( \alpha \) denotes the constant terms which do not affect the maximization.

A basic EM algorithm to estimate \( \Theta \) based on the observations \( Y^k \) consists of following two steps that are iterated a fixed number of times or until convergence: a) **E-step:** Given the estimate \( \hat{\Theta} \) of the unknown parameters \( \Theta \), we compute \( Q(\Theta, \hat{\Theta}) = \mathbb{E}[\log p(X^k, Y^k | \Theta) | Y^k, \hat{\Theta}] \), and b) **M-step:** The updated estimate \( \hat{\Theta}^{(\text{new})} = \arg \max_{\Theta} Q(\Theta, \hat{\Theta}) \). Note that for online EM, at \( k \)-th time step, the current estimate \( \hat{\Theta} \) is \( \hat{\Theta}_{k-1} \) and the updated estimate \( \hat{\Theta}^{(\text{new})} \) is \( \hat{\Theta}_k \).

**Approximate online EM for parameter estimates:** In our RKHS-EKF, we adopt an approximate online version for the EM algorithm such that all the available observations need not be processed at each time step. Further, as we will describe in the E-step, the required conditional expectations can be approximated using the estimates from EKF recursions itself. Hence, we first consider the M-step. Consider the \( k \)-th time step such that \( k \) observations are available to us up to this instant.

**Parameter estimates (M-step):** For simplicity, denote the conditional expectation operator \( \mathbb{E}[|Y_k, \Theta_{k-1}] \) given \( k \) observations by \( \mathbb{E}_k[\cdot] \). Now, from (29), maximizing \( \mathbb{E}_k[\log p(X^k, Y^k | \Theta)] \) with respect to parameter \( A \), we obtain the estimate \( \hat{A}_k \) of \( A \) given \( k \) observations as

\[
\hat{A}_k = \left( \sum_{j=1}^{k} \mathbb{E}_k [x_j \Phi(x_{j-1})^T] \right) \left( \sum_{j=1}^{k} \mathbb{E}_k [\Phi(x_{j-1}) \Phi(x_{j-1})^T] \right)^{-1}. \tag{30}
\]

Note that these expectations are computed using (and hence, functions of) the observations \( Y^k \) and current parameter estimate \( \Theta_{k-1} \) as described later in the E-step. However, this computation requires \( k \) observations to be processed together at time \( k \), and the complexity increases as \( k \) increases. To obtain approximate online estimate at low computations, we define the sum \( S_k^x = \sum_{j=1}^{k} \mathbb{E}_k [x_j \Phi(x_{j-1})^T] \) and \( S_k^1 = \sum_{j=1}^{k} \mathbb{E}_k [\Phi(x_{j-1}) \Phi(x_{j-1})^T] \) such that \( \hat{A}_k = S_k^x (S_k^1)^{-1} \) from (30). We approximate these sums as

\[
S_k^x \approx S_{k-1}^x + \mathbb{E}_k [x_k \Phi(x_{k-1})^T], \tag{31}
\]

\[
S_k^1 \approx S_{k-1}^1 + \mathbb{E}_k [\Phi(x_{k-1}) \Phi(x_{k-1})^T]. \tag{32}
\]

**Remark 11:** This is an approximation because we have not considered an updated parameter estimate \( \hat{\Theta}_{k-1} = \{ A_{k-1}, B_{k-1}, Q_{k-1}, R_{k-1} \} \) (obtained using observations up to time \( k-1 \)) for computing the conditional expectations in \( S_{k-1}^x \) and \( S_{k-1}^1 \). The current parameter estimates are used only in computing expectations \( \mathbb{E}_k [x_k \Phi(x_{k-1})^T] \) and \( \mathbb{E}_k [\Phi(x_{k-1}) \Phi(x_{k-1})^T] \).

**Remark 12:** Typically, in an EM algorithm, the E and M-steps are iterated a number of times considering the most recent parameter estimates in each E-step. In our RKHS-EKF, because of these approximations, the current observation is considered only once to obtain the current state estimates and update the parameter estimates.

Similarly, other approximate parameter updates (by approximating the summations as in (31) and (32)) are obtained as

\[
\hat{Q}_k = \left( 1 - \frac{1}{k} \right) \hat{Q}_{k-1} + \frac{1}{k} \left[ \mathbb{E}_k [x_k x_k^T] - \hat{A}_k \mathbb{E}_k [\Phi(x_{k-1}) x_k^T] - \mathbb{E}_k [x_k \Phi(x_{k-1})^T] \hat{A}_k^T + \hat{A}_k \mathbb{E}_k [\Phi(x_{k-1}) \Phi(x_{k-1})^T] \right], \tag{33}
\]

\[
\hat{B}_k = S_k^y (S_k^1)^{-1}, \tag{34}
\]
The necessary conditions for the recursive computation of the required expectations in (31)–(35) using the covariance matrix and noise covariance matrices estimates \( \hat{H}_k \), and \( \hat{S}_k \), and \( \hat{y}_k \) are as follows. The computation of the required expectations is detailed in the parameters update step (Section VI.B.3).

\[
\hat{R}_k = \left(1 - \frac{1}{k}\right) \hat{R}_{k-1} + \frac{1}{k} (E_k[y_k y_k^T] - \hat{B}_k E_k[\Phi(x_k) y_k^T]) - E_k[y_k \Phi(x_k)^T] \hat{B}_k^T + \hat{B}_k E_k[\Phi(x_k) \Phi(x_k)^T] \hat{B}_k^T,
\]
where the sums \( S_{y}^k = \sum_{j=1}^{k} E_k[y_j \Phi(x_j)^T] \) and \( S_{\Phi}^k = \sum_{j=1}^{k} E_k[\Phi(x_j) \Phi(x_j)^T] \) are introduced to obtain approximate online estimates and are evaluated as \( S_{y}^k = S_{y}^{k-1} + E_k[y_k \Phi(x_k)^T] \) and \( S_{\Phi}^k = S_{\Phi}^{k-1} + E_k[\Phi(x_k) \Phi(x_k)^T] \). Further, using (25), \( E_k[y_k y_k^T] = \hat{B}_k E_k[\Phi(x_k) \Phi(x_k)^T] \hat{B}_k^T + \hat{R}_{k-1} \) and \( E_k[y_k \Phi(x_k)^T] = \hat{B}_k E_k[\Phi(x_k) \Phi(x_k)^T] \hat{B}_k^T \).

**Expectation computations (E-step):** As mentioned earlier, we compute the required expectations in (31)–(35) using the EKF estimates. A standard EKF recursion provides a Gaussian approximation of the conditional posterior distribution of the required states given all the observations available up to current time instant. However, the expectations here involve a non-linear transformation \( \Phi(\cdot) \). In EKF, first-order Taylor series expansion approximates these non-linear expectations. We also need the statistics of \( \Phi(x_{k-1}) \) given \( Y_k \). Hence, we consider an augmented state \( \tilde{x}_{k-1} = [x_{k-1}^T \, x_{k-1}^T]^T \) to obtain a smoothed estimate \( \hat{x}_{k-1} \) of the previous state \( x_{k-1} \) given \( Y_k \). Using these approximations, we formulate the RKHS-EKF to jointly compute estimates \( \hat{x}_k \) and \( \hat{S}_k \) as follows. The computation of the required expectations is detailed in the parameters update step (Section VI.B.3).

**B. Recursion**

In terms of the augmented state \( z_k \), the RKHS-EKF system model is

\[
\begin{align*}
\hat{z}_k &= \hat{f}(\hat{z}_{k-1}) + \tilde{w}_{k-1}, \\
y_k &= h(\hat{z}_k) + v_k,
\end{align*}
\]
where \( \hat{f}(\hat{z}_{k-1}) = [(A \Phi(x_{k-1})^T \, x_{k-1}^T - \hat{A} \hat{x}_{k-1})^T \, \hat{S}_{k-1} \hat{x}_{k-1}^T]^T \) and \( \tilde{w}_{k-1} = \Phi(\hat{x}_{k-1}) \). The actual noise covariance matrix of \( \tilde{w}_{k-1} = [w_{k-1}^T \, 0_{1 \times n}]^T \) is \( \tilde{Q} = \begin{bmatrix} Q & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix} \). At \( k \)-th time instant, we have from the previous recursion, an estimate \( \hat{z}_{k-1} = [\hat{x}_{k-1}^T - \hat{S}_{k-2 \times k-2}^T x_{k-2}^T]^T \) with the associated error covariance matrix \( \hat{S}_{k-1}^2 \), coefficient matrices estimates \( \hat{A}_{k-1} \) and \( \hat{B}_{k-1} \), and noise covariance matrices estimates \( \hat{Q}_{k-1} \) and \( \hat{R}_{k-1} \). We compute estimates \( \hat{z}_k, \hat{A}_k, \hat{B}_k, \hat{Q}_k, \) and \( \hat{R}_k \) based on the new observation \( y_k \) through the procedure summarized below.

1) **Prediction:** Using \( A = \hat{A}_{k-1} \) and \( \hat{Q}_{k-1} = \begin{bmatrix} Q_{k-1} & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix} \) in (36), compute the predicted state and associated prediction error covariance matrix as

\[
\hat{z}_{k|k-1} = \hat{f}(\hat{z}_{k-1}), \quad \hat{S}_{k|k-1} = \hat{F}_{k-1} \hat{S}_{k-1} \hat{F}_{k-1}^T + \hat{Q}_{k-1},
\]

where \( \hat{F}_{k-1} = \nabla \hat{f}(\hat{z}_{k-1}) \) at \( z = \hat{z}_{k-1} \).

2) **Measurement update:** Using \( B = \hat{B}_{k-1} \) and \( R = \hat{R}_{k-1} \) in (37), compute

\[
\hat{S}_k = \hat{H}_k \hat{S}_{k|k-1} \hat{H}_k^T + R_k,
\]
where

\[
\hat{z}_k = \hat{z}_{k|k-1} + \hat{S}_{k|k-1} \hat{H}_k^T (y_k - \hat{h}(\hat{z}_{k|k-1})),
\]

\[
\hat{S}_k = \hat{S}_{k|k-1} - \hat{S}_{k|k-1} \hat{H}_k^T \hat{S}_k^{-1} \hat{H}_k \hat{S}_{k|k-1},
\]

where \( \hat{H}_k = \nabla \hat{h}(\hat{z}_k) \) at \( z = \hat{z}_{k|k-1} \). Here, \( \hat{z}_k = [\hat{x}_{k|k}^T \, \hat{x}_{k|k}^T]^T \) where \( \hat{x}_{k|k} \) is the RKHS-EKF’s estimate of state \( x_k \). The prediction and measurement update steps follow from the standard EKF recursions to estimate the augmented state \( z_k \) with the considered system model (36) and (37).

3) **Parameters update:** Update the parameter estimates by approximating the required expectations in (31)–(35). Consider \( E_k[x_k \Phi(x_{k-1})^T] \) from (31). Based on the standard EKF, linearize \( \Phi(\cdot) \) as

\[
\Phi(x_{k-1}) \approx \Phi(\hat{x}_{k-1}) + \nabla \Phi(\hat{x}_{k-1}) (x_{k-1} - \hat{x}_{k-1}),
\]
where \( \nabla \Phi(\hat{x}_{k-1}) \) is obtained by similar means. Thus, \( E_k[x_k \Phi(x_{k-1})^T] \) can be computed using the measurement update. Similarly, we have

\[
E_k[\Phi(x_{k-1}) \Phi(x_{k-1})^T] = \Phi(\hat{x}_{k-1}) \Phi(\hat{x}_{k-1})^T + \nabla \Phi(\hat{x}_{k-1}) \text{Cov}(x_{k-1} - \hat{x}_{k-1}) \nabla \Phi(\hat{x}_{k-1})^T,
\]

\[
E_k[\Phi(\hat{x}_{k|k}) \Phi(\hat{x}_{k|k})^T] = \Phi(\hat{x}_{k|k}) \Phi(\hat{x}_{k|k})^T + \nabla \Phi(\hat{x}_{k|k}) \text{Cov}(x_{k-1} - \hat{x}_{k|k}) \nabla \Phi(\hat{x}_{k|k})^T,
\]

\[
E_k[\Phi(\hat{x}_{k|k}) \Phi(x_{k-1})] = \text{Cov}(x_{k-1} - \hat{x}_{k|k}) + \hat{x}_{k|k} \nabla \Phi(\hat{x}_{k|k})^T,
\]

where \( \text{Cov}(x_{k-1} - \hat{x}_{k|k}) \approx [\hat{S}_{k|k}]_{n+1 \times n+1} \) and \( \text{Cov}(x_{k-1} - \hat{x}_{k|k}) \approx [\hat{S}_{k|k}]_{l \underline{n} \times l \underline{n}} \). With these expectations, the updated parameter estimates \( \hat{\Theta}_k \) can be computed using (31)–(35).

Finally, the dictionary \( \{\hat{x}_l\}_{1 \leq l \leq L} \) is updated using the new estimate \( \hat{x}_{k|k} \) based on the sliding window or ALD criterion. The following Algorithms 1 and 2 summarize the RKHS-EKF’s initialization and recursions, respectively. Note that the dictionary size increases when \( x_{k|k} \) is added to the dictionary based on the ALD criterion or when the current size \( L \) is less than the considered window length (initial transient phase) in the sliding window criterion. For simplicity, we initialize all mixing parameters (elements of \( A_0 \) and \( B_0 \)) with one in Algorithm 1. However, the mixing parameters can be initialized with arbitrary values.

**Remark 13:** The computational complexity of RKHS-EKF depends on the state dimension \( n \) as well as the size of the dictionary \( L \). In particular, the prediction and measurement update steps of RKHS-EKF follow from standard EKF recursion and hence, have a computational complexity of \( O(l^2) \), where \( l = 2n \) is the dimension of augmented state \( z_k \).
Algorithm 1 RKHS-EKF initialization

\textbf{Input:} $\mathbf{x}_0$, $\Sigma_0$

\textbf{Output:} $\hat{\mathbf{x}}_0$, $\hat{\Sigma}_0$, $L$, $\{\tilde{\mathbf{x}}_l\}_{1 \leq l \leq L}$, $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$, $\hat{\mathbf{Q}}_0$, $\hat{\mathbf{R}}_0$, $\mathbf{S}^0_{\phi\phi}$, $\mathbf{S}^0_{\phi \mathbf{x}}$, $\mathbf{S}^0_{\mathbf{x} \phi}$, $\mathbf{S}^0_{\mathbf{x} \mathbf{x}}$

1. $\tilde{\mathbf{z}}_0 \leftarrow [\mathbf{x}^T_0 \mathbf{x}^T_0]^T$, and $\hat{\Sigma}_0 \leftarrow \begin{bmatrix} \Sigma_0 & 0_{n \times n} \\ 0_{n \times n} & \Sigma_0 \end{bmatrix}$.
2. Set $L = 1$ and $\tilde{\mathbf{x}}_1 \leftarrow \mathbf{x}_0$.
3. $\mathbf{A} = I_{n \times L}$ and $\mathbf{B} = I_{p \times L}$.
4. Initialize $\mathbf{Q}_0$ and $\mathbf{R}_0$ with some suitable p.d. noise covariance matrices.
5. Set $\mathbf{S}^0_{\phi \phi} = 0_{n \times L}$, $\mathbf{S}^0_{\phi \mathbf{x}} = 0_{p \times L}$, and $\mathbf{S}^0_{\mathbf{x} \phi} = 0_{L \times n}$.

return $\mathbf{z}_0$, $\hat{\Sigma}_0$, $L$, $\{\tilde{\mathbf{x}}_l\}_{1 \leq l \leq L}$, $\hat{\mathbf{A}}_0$, $\hat{\mathbf{B}}_0$, $\hat{\mathbf{Q}}_0$, $\hat{\mathbf{R}}_0$, $\mathbf{S}^0_{\phi\phi}$, $\mathbf{S}^0_{\phi \mathbf{x}}$, $\mathbf{S}^0_{\mathbf{x} \phi}$, $\mathbf{S}^0_{\mathbf{x} \mathbf{x}}$.

Algorithm 2 RKHS-EKF recursion

\textbf{Input:} $\mathbf{z}_{k-1}$, $\hat{\Sigma}_{k-1}$, $\mathbf{A}_{k-1}$, $\mathbf{B}_{k-1}$, $\mathbf{Q}_{k-1}$, $\hat{\mathbf{R}}_{k-1}$, $\mathbf{S}^{0\phi}_{k-1}$, $\mathbf{S}^{0\mathbf{x}}_{k-1}$, $\mathbf{S}^{\mathbf{x}\phi}_{k-1}$, $\mathbf{S}^{\mathbf{x}\mathbf{x}}_{k-1}$, and $\mathbf{y}_k$

\textbf{Output:} $\hat{\mathbf{x}}_{k|k}$, $\hat{\Sigma}_{k|k}$ using (38)-(41).

1. Compute $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\Sigma}_{k|k-1}$ using (38).
2. Compute $\mathbf{x}_{k|k}$ and $\hat{\Sigma}_{k|k}$ using (39)-(41).
3. $\hat{\mathbf{x}}_{k|k} \leftarrow [\hat{\mathbf{z}}_k]_n$.
4. $\text{Cov}([\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}]_n, [\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}]_n) \leftarrow [\hat{\Sigma}^z_k]_{1\times 1}$.

return $\hat{\mathbf{x}}_{k|k}$, $\hat{\Sigma}_{k|k}$, $\hat{\mathbf{A}}_k$, $\hat{\mathbf{B}}_k$, $\hat{\mathbf{Q}}_k$, $\hat{\mathbf{R}}_k$, $\mathbf{S}^{0\phi}_k$, $\mathbf{S}^{0\mathbf{x}}_k$, $\mathbf{S}^{\mathbf{x}\phi}_k$, $\mathbf{S}^{\mathbf{x}\mathbf{x}}_k$, and $\mathbf{y}_k$

The parameters update step involves matrix multiplications of complexity $O(n^3)$ and matrix inversions of complexity $O(L^3)$. The updated parameters $\mathbf{A}_k$ and $\mathbf{B}_k$ require inversion of $L \times L$ matrices $\mathbf{S}^{0\mathbf{x}}_k$ and $\mathbf{S}^{\mathbf{x}\mathbf{x}}_k$, respectively. Note that the dictionary size $L$ is a user-defined constant under the sliding window criterion. The conditions for a finite $L$ under the ALD criterion follow from [30, Theorem 3.1].

VI. NUMERICAL EXPERIMENTS

We considered two illustrative examples to demonstrate the performance of our proposed inverse filters. We compared their estimation errors with RCRNL [38] as a benchmark. The RCRNL $\mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \geq \mathbf{J}^{-1}_k$, where $\mathbf{J}_k$ is the Fisher information matrix, provides a lower bound on mean-squared error (MSE). For the non-linear system given by (1) and (2), the forward information matrix $\mathbf{J}_k$ recurses reduces to [56]

\[ J_{k+1} = Q_k^{-1} + H_k^{T} R_k^{-1} H_k + \frac{1}{\beta^2} F_k^{-1} (J_k + F_k^{T} Q_k^{-1} F_k)^{-1} F_k^{T} Q_k^{-1}, \]

where $F_k = \nabla_{\mathbf{x}} f(x_k | x_{k-1})$ and $H_k = \nabla_{\mathbf{x}} h(x_k | x_{k-1})$. These recursions can be trivially extended to other system models considered in this paper as well as to compute the posterior information matrix $\mathbf{J}_k$ for inverse filter’s estimate $\hat{\mathbf{x}}_k$. We refer the reader to Section VI of the companion paper (Part I) [1] for further details on computation of RCRNL for forward and inverse filters’ state estimates. Some recent studies on cognitive radar target tracking instead consider posterior CRLB [32] as a metric to tune tracking filters.

In the following, we also consider the proposed inverse filters’ performance under incorrect forward filter assumption such that the misspecified CRB (MCRB) [57], [58] may provide further insights. However, the MCRB as it exists in the literature, cannot be directly applied to the mismatched forward and inverse filter case. Firstly, MCRB’s existence can be guaranteed only under certain regularity conditions, like the non-singularity of a generalized Fisher information matrix and the existence of a unique pseudo-true parameter vector minimizing the Kullback-Leibler divergence (KLD) between the true and assumed densities [57]. These regularity conditions cannot be trivially verified for our inverse filtering problem because of the non-linear transformation of the Gaussian noise terms. Furthermore, discrete-time filtering problems require a recursively computed lower bound as the observations $\mathbf{y}^k = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ and the states $\mathbf{x}^k = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$ up to $k$-th time step cannot be processed jointly. To the best of our knowledge, such recursive bounds for the misspecified non-linear filtering problems have not been proposed so far. Hence, in our experiments, we only consider RCRNL computing assuming a perfectly specified model.

A. Inverse Filters for SOEKF and GS-EKF

Recall from Section VI-B of the companion paper (Part I) [1], the discrete-time non-linear system model of FM demodulator [46, Section VIII-B]. The state-transition and observation equations are

\begin{align*}
\mathbf{x}_{k+1} &\equiv \begin{bmatrix} \lambda_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} \exp(-T/\beta) & 0 \\ -\beta \exp(-T/\beta) & -1 \end{bmatrix} \begin{bmatrix} \lambda_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} 1 \\ -\beta \end{bmatrix} \omega_k, \\
y_k = \sqrt{2} \sin \theta_k \cos \theta_k + v_k, \quad \omega_k \sim \mathcal{N}(0, 0.01), \quad v_k \sim \mathcal{N}(0, 0), \quad \epsilon_k \sim \mathcal{N}(0, 5), \quad T = 2\pi/16, \quad \beta = 100.
\end{align*}

with $\omega_k \sim \mathcal{N}(0, 0.01)$, $\mathbf{v}_k \sim \mathcal{N}(0, 0)$, $\epsilon_k \sim \mathcal{N}(0, 0.01)$, $\lambda_k$ is the forward filter’s estimate of $\lambda_k$.

The initial state $\mathbf{x}_0 = [\lambda_0, \theta_0]^T$, and its estimates for all forward and inverse filters including mean estimates for GS-EKF were set randomly with $\lambda_0 \sim \mathcal{N}(0, 1)$ and $\theta_0 \sim \mathcal{U}[-\pi, \pi]$. The initial covariances were set to $\Sigma_0 = 10 I_2$ and $\hat{\Sigma}_0 = 5 I_2$ for forward and inverse SOEKF. In the case of GS-EKF, we considered 5 Gaussians for the forward filter with the initial covariances and weights set to $10 I_2$ and $1/5$, respectively. For
I-GS-EKF, an augmented state $\mathbf{z}_k$ of $\{\mathbf{x}_{i,k}, c_{i,k}\}_{1 \leq i \leq 5}$ was considered resulting in a 15-dimensional state vector with the initial weight estimates set to $1/5$ and the initial covariance estimates $\{\Sigma_{i,0}\}_{1 \leq i \leq 5}$ as $5\mathbf{I}_{15}$. All other parameters of the system including the initial information matrix estimates were identical to those in Section VI-B in the companion paper (Part I) [1]. The Gaussian noise term $\mathbf{v}_{k+1}$ in I-GS-EKF’s state transition equation (18) is transformed through a non-linear function $\gamma(\cdot, \cdot)$ such that (46) is not applicable. The RCRLB in this case is derived using the general $\mathbf{J}_k$ recursions given by [38, eq. (21)], which we omit here.

Fig. 1 shows the time-averaged RMSE (AMSE) $\sqrt{(\sum_{i=1}^{k} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2)/nk}$ at $k$-th time step with $n$-dimensional actual state $\mathbf{x}_i$ and its estimate $\hat{\mathbf{x}}_i$, and RCRLB for both forward and inverse SOEKF and GS-EKF, averaged over 500 runs. Fig. 1 shows that the forward GS-EKF with $l = 5$ performs better than both forward EKF and forward SOEKF. Considering second-order terms of the Taylor series expansion, in addition to the first-order terms, does not improve the estimation performance for this system as observed in Fig. 1(a). When including second-order terms in the forward SOEKF yields smaller gains, then this is also reflected in the RCRLB of the inverse filters, i.e., I-EKF and I-SOEKF have similar lower bound values. Both I-EKF and I-SOEKF converge to the same steady-state estimation error values, which is also higher than that of the corresponding forward filters. Being suboptimal filters, the forward as well as inverse EKF and SOEKF do not achieve the RCRLB on the estimation error. However, the difference between AMSE and RCRLB for the inverse filters is less than that for the forward filters. We conclude that I-EKF and I-SOEKF are more efficient here.

For GS-EKF in Fig. 1(b), the estimation error of the inverse filter is same as that of the forward filter when $l = 2$ but improves significantly when $l = 5$. Note that this improvement in performance comes at the expense of increased computational complexity because the inverse filter estimates an augmented state of dimension $l(n + 1)'$, which is larger than the forward filter’s state dimension $n'$.

The I-SOEKF assume initial covariance $\Sigma_0$ as $5\mathbf{I}_2$ (the true $\Sigma_0$ of forward SOEKF is $10\mathbf{I}_2$) and a random initial state for these recursions. In spite of this difference in the initial estimates, I-SOEKF’s error performance is comparable to that of the forward SOEKF. Interestingly, despite similar differences in the initial estimates, I-GS-EKF with $l = 5$ outperforms the forward GS-EKF.

In Fig. 2, we consider the case when the defender assumes a different forward filter from the actual adversary’s filter. In Fig. 2(b), the I-GS-EKF with $l = 5$ is considered. The adversary’s true forward filter and that assumed by the defender for different inverse filters of Fig. 2 are provided in Table I. The forward as well as inverse filters (with perfect information of forward filter) are also included for comparison.

From Fig. 2, we observe that the I-SOEKF and I-EKF have similar performance regardless of the true forward filter employed by the adversary because both the forward EKF and SOEKF also have same estimation accuracy. On the other hand, forward GS-EKF performs better than EKF for the considered system. Hence, I-EKF-2 based on observations from forward GS-EKF has lower estimation error than I-EKF-1 (with EKF as true forward filter) even though our assumption about forward filter is not true. However, I-EKF-2’s estimation error is still more than I-GS-EKF-1 which knows the true forward filter. Furthermore, I-GS-EKF assuming forward GS-EKF even though the true forward filter is EKF (I-GS-EKF-2 case) also has lower estimation error than I-EKF-1, which uses perfect forward filter information. Hence, we conclude that assuming a more sophisticated forward GS-EKF and using I-GS-EKF provides higher estimation accuracy than I-EKF at the expense of

Table I: Summary of Forward-Inverse Filters in Fig. 2

| Fig. | True Forward Filter | Assumed Forward Filter | Inverse Filter |
|------|---------------------|------------------------|---------------|
| 2a   | EKF                 | EKF                    | I-EKF-1       |
| 2a   | SOEKF               | SOEKF                  | I-SOEKF-1     |
| 2b   | GS-EKF              | EKF                    | I-GS-EKF-2    |
| 2b   | GS-EKF              | GS-EKF                 | I-GS-EKF-1    |
| 2b   | EKF                 | GS-EKF                 | I-GS-EKF-2    |

Note: For the inverse filters, the true forward filter is assumed.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
B. Inverse DEKF

We consider the application of coordinate estimation of a stationary target from bearing observations taken by a moving sensor [19]. The actual coordinates of the stationary target are \((X, Y)\) and that of the sensor at \(k\)-th time instant are \((p_k^X, p_k^Y)\).

The constant velocity of sensor is \(s\). The forward and inverse EKF as well as DEKF were implemented in a modified coordinate basis with the state estimate \(x_k = [p_k^X/Y, s/Y, s, X/Y]^T\) and system model

\[
x_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ \Delta t/Y \\ \Delta t \\ 0 \end{bmatrix} w_k,
\]

\[
y_k = \arctan([x_k^4]_1 - [x_k^4]_1) + v_k, \; \alpha_k = ([x_k^4]_4)^2 + \epsilon_k,
\]

where \(w_k \sim \mathcal{N}(0, \sigma^2)\), \(v_k \sim \mathcal{N}(0, 2^2)\), \(\epsilon_k \sim \mathcal{N}(0, 1.5^2)\) and \(\Delta t = 20\). The initial covariance estimates were set to \(\Sigma_0 = \text{diag}(4.44 \times 10^{-7}, 0.5 \times 10^{-6}, 1, 0.1)\) and \(\Sigma_0 = \text{diag}(10^{-6}, 6 \times 10^{-7}, 5, 0.5)\), respectively, for the forward and inverse filters. The initial state estimate for inverse filters were \((0, 0.002, 200, 2)^T\). The initial information matrices estimates \(\mathbf{J}_0\) and \(\mathbf{J}_0\) were set to \(\Sigma_0\) and \(\Sigma_0\), respectively. All other parameters of the system including the dither and the estimates were identical to those in [19].

With 200 time-steps, the modified observation function \(h^*(\cdot)\) in the forward DEKF replaced \(h(\cdot)\) up to 80 time-steps. Fig. 3(a) shows the absolute error and RCRCLB, averaged over 400 runs, for estimation of \(X/Y\) whose estimate at the \(k\)-th time instant are given by \([x_k^4]_1\) and \([x_k^4]_4\) of the forward and inverse filters. The RCRCLB at \(k\)-th time instant is \(\sqrt{\frac{\mathbf{J}_k^{-1}}{4^4}}\). The inverse filters’ estimation errors were significantly lower than that of the forward filters. While I-DEKF-1 and I-DEKF-2 differ in their transient performance, they converge to the same steady-state error as I-EKF.

C. Inverse RKHS-EKF

Consider the same FM-demodulator system as in Section VI-A. The defender employs RKHS-EKF to infer the adversary’s state estimate while the forward filter is a simplified RKHS-EKF for known observation function. The initial state estimates for both filters were drawn at random similar to Section VI-A. For both forward and inverse RKHS-EKF, we considered the Gaussian kernel with kernel width \(\sigma\) set to \(\sigma^2 = 30\) and 50, respectively. The initial covariance estimates \((\Sigma_0)\) were \(10\mathbf{I}_1\) and \(5\mathbf{I}_1\), respectively, for forward and inverse RKHS-EKF.

To construct the forward RKHS-EKF’s dictionary, we used the sliding window criterion [52] with window length 2. On the other hand, the dictionary for inverse RKHS-EKF (I-RKHS-EKF) was constructed using ALD criterion [30]. All the initial coefficient estimates \((\mathbf{A}_0, \mathbf{B}_0)\) were set to 1. The noise covariance estimates \(\mathbf{Q}_0\) and \(\mathbf{R}_0\) were \(\text{diag}(1, 10)\) and 5, respectively. All other system parameters were same as in Section VI-A.

Fig. 3(b) shows the AMSE for forward and inverse RKHS-EKF, compared to EKF and I-EKF, respectively. We also include the case when the inverse filter assumes an incorrect forward filter. In particular, I-EKF assumes a forward EKF employed by the adversary. While I-EKF considers the correct forward filter, the true forward filter is RKHS-EKF in case of I-EKF-2. On the other hand, I-RKHS-EKF does not make any such assumption. The true forward filters are RKHS-EKF and EKF for I-RKHS-EKF-1 and I-RKHS-EKF-2, respectively.

From Fig. 3(b), we observe that forward RKHS-EKF has higher estimation error than forward EKF because it does not have any prior information about the evolution of state \(x_k\). The I-EKF, with perfect knowledge of the system model, performs worse than the forward EKF. On the contrary, I-RKHS-EKF (I-RKHS-EKF-1 and I-RKHS-EKF-2) without any prior information, has smaller error than the corresponding forward filter. The improved performance of RKHS-based inverse filter is explained by the fact that it is a kernel function approximation and learns the unknown system dynamics completely based on its observations. On the other hand, I-EKF uses a first-order Taylor approximation for the non-linear functions. Here, the Gaussian kernel provides a better function approximation than linearization. Interestingly, with the true forward filter as RKHS-EKF, both I-RKHS-EKF-1 and I-RKHS-EKF-2 have higher estimation accuracy than the forward filter, which is better than forward EKF and I-EKF as well. This suggests that an inverse filter may have a lower estimation error even though the forward filter is inaccurate in its estimates.

VII. Summary

We developed the I-EKF theory of the companion paper (Part I) [1] to highly non-linear systems that employ inverses of SOEKF, GS-EKF, and DEKF. We showed that the stability of forward SOEKF is sufficient for I-SOEKF to be stable provided the system model satisfies certain additional assumptions. The inverse filters developed in the companion paper (Part I) [1] and those based on EKF variants are limited in the sense that
they assume a known forward filter. The prior work [2] also assumes perfect system information for inverse filters. To this end, we proposed RKHS-EKF for such uncertain systems. The unknown functions are approximated using a kernel while the unknown parameters are learnt online using an approximate EM algorithm coupled with EKF recursions. Numerical examples demonstrate the estimation efficacy of the proposed filters with RCR-LB as a performance measure. Our experiments with mismatched forward filters suggest that the inverse filter that assumes a similar advanced forward filter may provide better state estimates but at a high computational cost. While [3], [4] suggest adapting a radar’s waveform based on its observations, the developed inverse filters consider observing such a cognitive radar’s adaptations and estimating its information for a counter-adversarial defender to guard against it.

**APPENDIX A**

PROOF OF THEOREM 1

**Lemma 1:** Under the assumptions of Theorem 1, the following bounds hold true for all \( k \geq 0 \).

1. \( \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{a}_i \bar{a}_j^T \text{Tr} \left( \nabla^2 [f(\hat{x}_k)], \Sigma_k \nabla^2 [f(\hat{x}_k)], \Sigma_k \right) \) is a p.s.d. matrix and satisfies the upper bound
   \[
   \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{a}_i \bar{a}_j^T \text{Tr} \left( \nabla^2 [f(\hat{x}_k)], \Sigma_k \nabla^2 [f(\hat{x}_k)], \Sigma_k \right) \leq \frac{\sigma^2}{\sigma^2} n^2 I.
   \]

2. \( \sum_{i=1}^{p} \sum_{j=1}^{p} b_i b_j^T \text{Tr} \left( \nabla^2 [h(\hat{x}_k)], \Sigma_k \nabla^2 [h(\hat{x}_k)], \Sigma_k \right) \) is a p.s.d. matrix and satisfies the upper bound
   \[
   \sum_{i=1}^{p} \sum_{j=1}^{p} b_i b_j^T \text{Tr} \left( \nabla^2 [h(\hat{x}_k)], \Sigma_k \nabla^2 [h(\hat{x}_k)], \Sigma_k \right) \leq \frac{b^2}{\sigma^2} np I.
   \]

3. \( \|M_k\| \leq \beta \) with \( \beta > 0 \).

**Proof:** Using the bounds from the assumptions of Theorem 1, we have for all \( i, j \in \{1, 2, \ldots, n\} \)
\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \bar{a}_i \bar{a}_j^T \text{Tr} \left( \nabla^2 [f(\hat{x}_k)], \Sigma_k \nabla^2 [f(\hat{x}_k)], \Sigma_k \right) \leq \frac{\sigma^2}{\sigma^2} n^2 I,
\]
which implies
\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \bar{a}_i \bar{a}_j^T \text{Tr} \left( \nabla^2 [f(\hat{x}_k)], \Sigma_k \nabla^2 [f(\hat{x}_k)], \Sigma_k \right) \leq \frac{\sigma^2}{\sigma^2} n^2 I.
\]

**Lemma 2:** Under the assumptions of Theorem 1, there exists a real number \( \alpha \) with \( 0 < \alpha < 1 \) such that
\[
(F_k - K_k H_k)^T \Sigma_k^{-1} (F_k - K_k H_k) \leq (1 - \alpha) \Sigma_k^{-1}.
\]

**Proof:** Using (11) and (16), we have
\[
\Sigma_k^{-1} = F_k \Sigma_k F_k^T + Q_k - K_k S_k K_k^T + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \bar{a}_i \bar{a}_j^T \text{Tr} \left( \nabla^2 [f(\hat{x}_k)], \Sigma_k \nabla^2 [f(\hat{x}_k)], \Sigma_k \right) - Q_k - (F_k \Sigma_k H_k^T + M_k) K_k^T.
\]
Using the positive semi-definiteness of the last term (as proved in Lemma 1) and substituting for \( K_k S_k \) using (14), we have
\[
\Sigma_k^{-1} \geq F_k \Sigma_k F_k^T + Q_k - (F_k \Sigma_k H_k^T + M_k) K_k^T.
\]
Rearranging the terms,
\[
\Sigma_k^{-1} \geq (F_k - K_k H_k) \Sigma_k (F_k - K_k H_k)^T + Q_k + K_k H_k \Sigma_k (F_k - K_k H_k)^T - M_k K_k^T.
\]
First, we consider the last two terms. Putting \( M_k = K_k S_k - F_k \Sigma_k H_k^T \), the last terms can be expressed as
\[
K_k H_k \Sigma_k (F_k - K_k H_k)^T + M_k K_k^T.
\]
Let \( A = K_k H_k \Sigma_k F_k^T \). Using the bounds of Theorem 1 and Lemma 1, we can show that \( \|A\| \leq \frac{7\sigma^2}{7\sigma^2 + \beta} \). Also, \( A + A^T \) is a symmetric \( n \times n \) matrix with
\[
\|A + A^T\| \leq \frac{2\sqrt{7} \sigma^2}{7\sigma^2 + \beta} I.
\]
This implies
\[
\frac{2\sqrt{7} \sigma^2}{7\sigma^2 + \beta} I \leq A + A^T \leq \frac{2\sqrt{7} \sigma^2}{7\sigma^2 + \beta} I.
\]
Using this and other bounds, we have
\[
K_k H_k \Sigma_k (F_k - K_k H_k)^T - M_k K_k^T \geq -c I,
\]
where \( c = \frac{2\sqrt{7} \sigma^2}{7\sigma^2 + \beta} + \left( 2\sigma^2 + \frac{\sigma^2}{\sigma^2} np \right) \left( \frac{7\sigma^2 + \beta}{7\sigma^2} \right)^2 \) is a positive constant. Hence,
\[
K_k H_k \Sigma_k (F_k - K_k H_k)^T + Q_k - c I.
\]
Similar to the proof of [43, Lemma 3.1], we can prove \( F_k - K_k H_k \) to be invertible using matrix inversion lemma. Here,
\[
F_k^{-1} (F_k - K_k H_k) \Sigma_k
\]
\[
= \Sigma_k - \Sigma_k (H_k^T + \Sigma_k^{-1} F_k^{-1} M_k) S_k^{-1} H_k \Sigma_k.
\]
The R.H.S. is in the form of the matrix inversion lemma:
\[
(A + UCV)^{-1} = A^{-1} - A^{-1} U (C^{-1} + VA^{-1} U)^{-1} VA^{-1},
\]
where \( A \) and \( C \) are invertible matrices. Hence, the matrix inversion lemma is applicable if \( R_k = \frac{1}{4} \sum_{i=1}^{p} \sum_{j=1}^{p} b_i b_j^T \text{Tr} \left( \nabla^2 [h(\hat{x}_k)], \Sigma_k \nabla^2 [h(\hat{x}_k)], \Sigma_k \right) - H_k F_k K_k^T \) is invertible (because \( C \) needs to be invertible). Since the difference of two matrices \( X - Y \)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
is invertible if maximum singular value of $Y$ is strictly less than the minimum singular value of $X,$ the required difference is invertible if $\|H_k F_k^{-1} M_k\| < \frac{2\epsilon}{\hbar n \sigma^2 n^2}.$

The condition (20) is sufficient for this to be satisfied. Note that this condition is a sufficient but not necessary condition for invertibility of $F_k - K_k H_k.$ With this condition, Lemma 2 can be proved using similar approach as in [43, Lemma 3.1] with

$$1 - \alpha = \left(1 + \frac{q - c}{\sigma(f + 2\sigma^2 + \frac{\rho}{\epsilon})^2}\right)^{-1}.$$ 

The condition (21) is sufficient for $0 < \alpha < 1.$

Substituting for constant $c$ and using $q \leq \delta$ and $r \leq \delta,$ the inequality (21) requires the following inequality to be satisfied

$$2\sigma^2 \delta + \left(2\sigma^2 + \frac{1}{2} \sigma^2 n \rho\right) \left(\frac{\delta}{\sigma} + \beta^2\right) < \delta.$$ 

Rearranging the terms, we have

$$\delta^3 - (3\sigma^2 + \beta^2) \delta - (\delta + \beta^2) (2\sigma^2 n \rho) > 0.$$ 

It can be observed that this inequality requires the noise bound $\delta$ to be large enough and other bounds on matrices like $F, H, \sigma$ etc. to be small.

**Lemma 3:** Under the assumptions of Theorem 1, there exists positive real constants $\kappa_{\text{mon}}, \epsilon'$ such that

$$r_k^T \Sigma_{k+1}^{-1} (2(F_k - K_k H_k) e_k + r_k) \leq \kappa_{\text{mon}} \|e_k\|^2,$$

for $\|e_k\|_2 \leq \epsilon'$.

**Proof:** The proof follows from [43, Lemma 3.2] with $\kappa_{\text{mon}} = \frac{\epsilon'}{2} \left(\frac{\delta}{\sigma} + \frac{\sigma \delta}{\epsilon} + \epsilon' \epsilon''\right),$ where $\kappa' = \kappa_{\sigma} + \frac{\sigma \delta}{\epsilon \epsilon''} \epsilon'$ and $\epsilon'' = \min(\epsilon', \epsilon'').$

**Lemma 4:** Under the assumptions of Theorem 1, there exists positive real constant $\kappa_{\text{noise}}$ independent of $\delta$ such that

$$E[s_k^T \Sigma_{k+1}^{-1} s_k] \leq \kappa_{\text{noise}} \delta.$$ 

**Proof:** The proof follows from [43, Lemma 3.3] with $\kappa_{\text{noise}} = \frac{n}{2} + \frac{\sigma^2 + \sigma \delta}{\sigma \epsilon}$.

**Lemma 5:** Under the assumptions of Theorem 1, there exist positive real constants $\kappa_{\text{sec}}, \kappa_{\text{sec}}$ such that

$$q_k^T \Sigma_{k+1}^{-1} (2(F_k - K_k H_k) e_k + 2r_k + q_k) \leq \kappa_{\text{sec}} \|e_k\|^2 + \kappa_{\text{sec}} \|e_k\|^2,$$

for $\|e_k\|_2 \leq \epsilon'$ where $\epsilon'$ is same as in Lemma 3.

**Proof:** Using the upper bound on $\nabla^2[f(\hat{x}_k)]_i$ from the assumptions of Theorem 1, we have

$$e_k^T \nabla^2 [f(\hat{x}_k)]_i e_k \leq \pi ||e_k||^2,$$

which implies

$$\|\sum_{i=1}^{n} \tilde{a}_i e_k^T \nabla^2 [f(\hat{x}_k)]_i e_k\|_2 \leq \pi n ||e_k||^2.$$ 

Similarly, using other bounds, it is trivial to show that for all $k \geq 0,$ the following bounds are satisfied

$$\|\sum_{i=1}^{n} \tilde{a}_i \text{Tr} \left(\nabla^2 [f(\hat{x}_k)]_i \Sigma_k\right)\|_2 \leq \pi n ||e_k||^2,$$

$$\|K_k \sum_{i=1}^{p} b_{i} e_k^T \nabla^2 [h(\hat{x}_k)]_i e_k\|_2 \leq \frac{\beta \rho (\|f\| + \beta)}{\epsilon \epsilon'} ||e_k||^2,$$

$$\|K_k \sum_{i=1}^{p} b_{i} \text{Tr} \left(\nabla^2 [h(\hat{x}_k)]_i \Sigma_k\right)\|_2 \leq \frac{\beta \rho (\|f\| + \beta)}{\epsilon \epsilon'}.$$ 

Using these bounds, we have

$$\|q_k\|_2 \leq \kappa_q ||e_k||^2 + c_q,$$

where $\kappa_q = \frac{1}{2} \left(\pi n + \frac{\beta \rho (\|f\| + \beta)}{\epsilon \epsilon'}\right)$ and $c_q = \frac{1}{2} \left(\pi n ||e_k||^2 + \frac{\beta \rho (\|f\| + \beta)}{\epsilon \epsilon'}\right).$

Using the bounds on $\|q_k\|_2$ and $\|r_k\|_2$ as used in [43, Lemma 3.2], the lemma can be proved with

$$\kappa_{\text{sec}} = \frac{\epsilon'}{2} \left(\frac{\delta}{\sigma} + \frac{\sigma \delta}{\epsilon} + \epsilon' \epsilon''\right) + \frac{c_q}{\sigma} \left(\frac{\delta}{\sigma} + \frac{\sigma \delta}{\epsilon} + \epsilon' \epsilon''\right) + 2\epsilon' \epsilon'' + \kappa_q \epsilon'',$$

and

$$c_{\text{sec}} = \frac{c_q}{\sigma} \left(\frac{\delta}{\sigma} + \frac{\sigma \delta}{\epsilon} + \epsilon' \epsilon''\right) + \frac{c_q}{\sigma} \left(\frac{\delta}{\sigma} + \frac{\sigma \delta}{\epsilon} + \epsilon' \epsilon''\right) + 2\epsilon' \epsilon'' + \kappa_q \epsilon''.$$

**B. Proof of the Theorem**

Consider $V_k(e_k) = e_k^T \Sigma_k^{-1} e_k.$ Using the bounds on $\Sigma_k$ from assumptions of Theorem 1, we have

$$\frac{1}{\sigma} \|e_k\|^2 \leq V_k(e_k) \leq \frac{1}{\sigma} \|e_k\|^2.$$ 

Also, substituting for $e_{k+1}$ using (19), we have

$$V_{k+1}(e_{k+1}) = e_{k+1}^T (F_k - K_k H_k) e_{k+1} + r_{k+1}^T \Sigma_{k+1}^{-1} (2(F_k - K_k H_k) e_k + r_k + q_k) + s_{k+1}^T \Sigma_{k+1}^{-1} s_{k+1} + 2s_{k+1}^T \Sigma_{k+1}^{-1} ((F_k - K_k H_k) e_k + r_k + q_k).$$

Using Lemmata 2, 3, and 5, we have for $\|e_k\|_2 \leq \epsilon'$

$$V_{k+1}(e_{k+1}) \leq (1 - \alpha) V_k(e_k) + \kappa_{\text{mon}} \|e_k\|^2 + \kappa_{\text{sec}} \|e_k\|^2 + \kappa_{\text{sec}} \|e_k\|^2 + c_{\text{sec}} + s_{k+1}^T \Sigma_{k+1}^{-1} s_{k+1} + 2s_{k+1}^T \Sigma_{k+1}^{-1} ((F_k - K_k H_k) e_k + r_k + q_k).$$
The last term $s_2^k \sum_{j=1}^{n} [(F_k - K_k H_k) e_k + r_k + q_k]$ vanishes on taking expectation conditioned on $e_k$ and hence, for $\|e_k\|_2 \leq \epsilon'$,
\[
E[V_{k+1}(e_{k+1}) | e_k] \leq (1-\alpha) V_k(e_k) + \kappa_{\text{nonl}} \|e_k\|_2^2 + \kappa_{\text{sec}} \|e_k\|_2^2 + c_{\text{sec}} + \kappa_{\text{noise}} \delta,
\]
where the bound of Lemma 4 is applied. But, for $\|e_k\|_2 \leq \epsilon'$,
\[
\kappa_{\text{nonl}} \|e_k\|_2^2 + \kappa_{\text{sec}} \|e_k\|_2^2 \leq (\kappa_{\text{nonl}} \epsilon' + \kappa_{\text{sec}}) \|e_k\|_2 \|e_k\|_2.
\]
Choosing $\epsilon = \min \left( \epsilon', \frac{\alpha}{2(\kappa_{\text{nonl}} \epsilon' + \kappa_{\text{sec}})} \right)$, we have for $\|e_k\|_2 \leq \epsilon$,
\[
\kappa_{\text{nonl}} \|e_k\|_2^2 + \kappa_{\text{sec}} \|e_k\|_2^2 \leq \frac{\alpha}{2} V_k(e_k),
\]
which implies
\[
E[V_{k+1}(e_{k+1}) | e_k] - V_k(e_k) \leq \frac{\alpha}{2} V_k(e_k) + c_{\text{sec}} + \kappa_{\text{noise}} \delta.
\]
Hence, Lemma 1 of the companion paper (Part I) [1] applies here. However, to have negative mean drift, we require $\delta$ to be small enough such that there exists some $\tilde{\epsilon} \leq \epsilon$ to satisfy (22). This condition ensures that for $\tilde{\epsilon} \leq \|e_k\|_2 \leq \epsilon$, $E[V_{k+1}(e_{k+1}) | e_k] - V_k(e_k) \leq 0$ is fulfilled and the estimation error $e_k$ remains exponentially bounded in mean-square bound if the error is within $\epsilon$ bound.

APPENDIX B

PROOF OF THEOREM 2

From the state transition function of I-SOEFK (17), we have
\[
\nabla^2 \tilde{f}_k(\hat{x}_k) = \nabla^2 \left[ f(\hat{x}_k) \right] - \sum_{j=1}^{p} [K_k]_{i,j} \nabla^2 \left[ h(\hat{x}_k) \right]_{j}.
\]
Using the upper bounds on Hessian matrices from the assumptions of Theorem 1, we have
\[
\nabla^2 \tilde{f}_k(\hat{x}_k) \preceq \left( \bar{a} + \bar{b} \|\sqrt{\beta} \right) \begin{bmatrix} K_k \end{bmatrix}_{i,j} \preceq I.
\]
But, the row sum $\sum_{j=1}^{p} [K_k]_{i,j} \leq \|K_k\|_{\infty}$. Further, using equivalence of norms i.e. $\|K_k\|_{\infty} \leq \sqrt{\rho} \|K_k\|$, we have
\[
\nabla^2 \tilde{f}_k(\hat{x}_k) \preceq \left( \bar{a} + \bar{b} \|\sqrt{\rho} \right) \begin{bmatrix} K_k \end{bmatrix}_{i,j} \preceq I.
\]
Similarly,
\[
\nabla^2 \tilde{f}_k(\hat{x}_k) \preceq \left( \bar{a} + \bar{b} \|\sqrt{\beta} \right) \begin{bmatrix} K_k \end{bmatrix}_{i,j} \preceq I.
\]
Hence, with $\bar{a} = \bar{a} - (\bar{b} \|\sqrt{\rho} (\bar{T} \bar{H} + \beta) / \bar{\ell})$ and $\bar{b} = \bar{a} + (\bar{b} \|\sqrt{\rho} (\bar{T} \bar{H} + \beta) / \bar{\ell})$, we have $\bar{a} \leq \nabla^2 \tilde{f}_k(\hat{x}_k) \preceq \bar{b}$ for all $i \in \{1, 2, \ldots, n\}$.

The proof of the remaining conditions of Theorem 1 for I-SOEFK dynamics follows from the proof of Theorem 5 in the companion paper (Part I) [1].

REFERENCES

[1] H. Singh, A. Chattopadhyay, and K. V. Mishra, “Inverse extended Kalman filter—Part I: Fundamentals,” IEEE Trans. Signal Process., vol. 71, 2023-2025, 2023.
[2] V. Krishnamurthy and M. Rangaswamy, “How to calibrate your adversary’s capabilities? Inverse filtering for counter-autonomous systems,” IEEE Trans. Signal Process., vol. 67, no. 24, pp. 6511–6525, Dec. 2019.
[3] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, “Cognitive radar framework for target detection and tracking,” IEEE J. Sel. Topics Signal Process., vol. 9, no. 8, pp. 1427–1439, Dec. 2015.
[4] N. Sharara, J. Fabrikian, and H. Messer, “Optimal cognitive beamforming for target tracking in MIMO radar/sonar,” IEEE J. Sel. Topics Signal Process., vol. 9, no. 8, pp. 1440–1450, Dec. 2015.
[5] R. Mattila, C. Rojas, V. Krishnamurthy, and B. Wahlberg, “Inverse filtering for hidden Markov models,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–10.
[6] A. Y. Ng et al., “Algorithms for inverse reinforcement learning,” in Proc. Int. Conf. Mach. Learn., vol. 1, 2000, p. 2.
[7] R. Mattila, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, “Inverse filtering for hidden Markov models with applications to counter-adversarial autonomous systems,” IEEE Trans. Signal Process., vol. 68, pp. 4987–5002, 2020.
[8] V. Krishnamurthy, K. Pattanayak, S. Gogineni, B. Kang, and M. Rangaswamy, “Adversarial radar inference: Inverse tracking, identifying cognition, and designing smart interference,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 4, pp. 2067–2081, Aug. 2021.
[9] B. Kang, V. Krishnamurthy, K. Pattanayak, S. Gogineni, and M. Rangaswamy, “Smart interference signal design to a cognitive radar,” in Proc. IEEE Radar Conf., San Antonio, TX, USA, May 2023, pp. 1–6.
[10] S. Haykin, Kalman Filtering and Neural Networks, vol. 47. New York, NY, USA: Wiley, 2004.
[11] D. Simon, Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches. New Jersey, NJ, USA: Wiley, 2006.
[12] R. Tenney, R. Hebbert, and N. Sandell, “A tracking filter for maneuvering sources,” IEEE Trans. Autom. Control, vol. AC-22, no. 2, pp. 246–251, Apr. 1977.
[13] S. Zhao and B. Huang, “Trial-and-error or avoiding a guess? Initialization of the Kalman filter,” Automatica, vol. 121, 2020, Art. no. 109184.
[14] M. Salvaldi and D. Choukroun, “Process noise covariance design in Kalman filtering via bounds optimization,” IEEE Trans. Autom. Control, vol. 64, no. 2, pp. 834–840, Feb. 2019.
[15] A. H. Jazwinski, Stochastic Processes and Filtering Theory. Mineola, NY, USA: Courier Corporation, 2007.
[16] X. Wang and E. E. Yaz, “Second-order fault tolerant extended Kalman filter for discrete time nonlinear systems,” IEEE Trans. Autom. Control, vol. 64, no. 12, pp. 5086–5093, Dec. 2019.
[17] P. Tam and J. Moore, “A Gaussian sum approach to phase and frequency estimation,” IEEE Trans. Commun., vol. 25, no. 9, pp. 935–942, Sep. 1977.
[18] P. R. Wishner, J. A. Tabaczynski, and M. Athans, “A comparison of three non-linear filters,” Automatica, vol. 5, no. 4, pp. 487–496, 1969.
[19] H. Weiss and J. Moore, “Improved extended Kalman filter design for passive tracking,” IEEE Trans. Autom. Control, vol. AC-25, no. 4, pp. 807–811, Aug. 1980.
[20] R. S. Bucy and K. D. Senne, “Digital synthesis of non-linear filters,” Automatica, vol. 7, no. 3, pp. 287–298, 1971.
[21] L. Schwartz and E. Stear, “A computational comparison of several nonlinear filters,” IEEE Trans. Autom. Control, vol. AC-13, no. 1, pp. 83–86, Feb. 1968.
[22] M. Netto, L. Gimeno, and M. Mendes, “On the optimal and suboptimal nonlinear filtering problem for discrete-time systems,” IEEE Trans. Autom. Control, vol. AC-23, no. 6, pp. 1062–1067, Dec. 1978.
[23] J. Vila-Valls, E. Chaumette, F. Vincent, and P. Closas, “Robust linearly constrained Kalman filter for general mismatched linear state-space models,” IEEE Trans. Autom. Control, vol. 67, no. 12, pp. 6794–6801, Dec. 2021.
[24] Q. Ge, T. Shao, Z. Duan, and C. Wen, “Performance analysis of the Kalman filter with mismatched noise covariances,” IEEE Trans. Autom. Control, vol. 61, no. 12, pp. 4014–4019, Dec. 2016.
[25] 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 Trans. Autom. Control, vol. 63, no. 2, pp. 594–601, Feb. 2018.

[26] H. Zhu, G. Zhang, Y. Li, and H. Leung, “Adaptive Kalman filter with inaccurate noise covariances in the presence of outliers,” IEEE Trans. Autom. Control, vol. 67, no. 1, pp. 374–381, Jan. 2022.

[27] S. Yi and M. Zorzi, “Robust Kalman Filtering under model uncertainty: The case of degenerate densities,” IEEE Trans. Autom. Control, vol. 67, no. 7, pp. 3458–3471, Jul. 2022.

[28] S. M. Mohamed and S. Nahavandi, “Robust finite-horizon Kalman filtering for uncertain discrete-time systems,” IEEE Trans. Autom. Control, vol. 57, no. 6, pp. 1548–1552, Jun. 2012.

[29] W. Liu, I. Park, Y. Wang, and J. C. Príncipe, “Extended kernel recursive least squares algorithm,” IEEE Trans. Signal Process., vol. 57, no. 10, pp. 3801–3814, Oct. 2009.

[30] Y. Engel, S. Mannor, and R. Meir, “The kernel recursive least-squares algorithm,” IEEE Trans. Signal Process., vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[31] P. Zha, B. Shen, and J. C. Príncipe, “Extended Kalman filter using a kernel recursive least squares observer,” in Proc. Int. Joint Conf. Neural Netw., Piscataway, NJ, USA: IEEE, 2011, pp. 1402–1408.

[32] U. K. Singh, M. Alaei-Kerahroodi, and M. B. Shankar, “RKHS based state estimator for radar sensor in indoor application,” in Proc. IEEE Radar Conf. (RadarConf), 2022, pp. 1–6.

[33] P. Zha, B. Chen, and J. C. Príncipe, “Learning nonlinear generative models of time series with a Kalman filter in RKHS,” IEEE Trans. Signal Process., vol. 62, no. 1, pp. 141–155, Jan. 2014.

[34] F. Tobar, P. M. Djurić, and D. P. Mandic, “Unsupervised state-space modeling using reproducing kernels,” IEEE Trans. Signal Process., vol. 63, no. 19, pp. 5210–5221, Oct. 2015.

[35] K. Berntorp, “Online Bayesian inference and learning of Gaussian-process state-space models,” Automatica, vol. 129, 2021, Art. no. 109613.

[36] Q. Wen, Z. Ge, and Z. Song, “Data-based linear Gaussian state-space model for dynamic process monitoring,” AIChE J., vol. 58, no. 12, pp. 3763–3776, 2012.

[37] D. Alspach and H. Sorenson, “Nonlinear Bayesian estimation using Gaussian sum approximations,” IEEE Trans. Autom. Control, vol. AC-17, no. 4, pp. 439–448, Aug. 1972.

[38] P. Tichavsky, C. H. Muravchik, and A. Nehorai, “Posterior Cramér-Rao bounds for discrete-time nonlinear filtering,” IEEE Trans. Signal Process., vol. 46, no. 5, pp. 1386–1396, May 1998.

[39] I. Lourenço, R. Mattila, C. R. Rojas, and B. Wahlberg, “How to protect your privacy? A framework for counter-adversarial decision making,” in Proc. 59th IEEE Conf. Decis. Control (CDC), 2020, pp. 1785–1791.

[40] I. Lourenço, R. Mattila, C. R. Rojas, X. Hu, and B. Wahlberg, “Hidden Markov models: Inverse filtering, belief estimation and privacy protection,” J. Syst. Sci. Complexity, vol. 34, pp. 1801–1820, 2021.

[41] K. Pattanayak, V. Krishnamurthy, and C. Berry, “Inverse-reverse reinforcement learning. How to hide strategy from an adversarial inverse reinforcement learner,” in Proc. IEEE 61st Conf. Decis. Control (CDC), 2022, pp. 3631–3636.

[42] K. Pattanayak, V. Krishnamurthy, and C. Berry, “Meta-cognition. An inverse-inverse reinforcement learning approach for cognitive radars,” in Proc. 25th Int. Conf. Inf. Fusion (FUSION), Piscataway, NJ, USA: IEEE, 2022, pp. 1–8.

[43] K. Reif, S. Gunther, E. Yuz, and R. Unbehauen, “Stochastic stability of the discrete-time extended Kalman filter,” IEEE Trans. Autom. Control, vol. 44, no. 4, pp. 714–728, Apr. 1999.

[44] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software. New York, NY, USA: Wiley, 2004.

[45] M. Roth and F. Gustafsson, “An efficient implementation of the second order extended Kalman filter,” in Proc. IEEE 14th Int. Conf. Inf. Fusion, 2011, pp. 1–6.

[46] B. D. Anderson and J. B. Moore, Optimal Filtering. Mineola, NY, USA: Courier Corporation, 2012.

[47] I. Bilik and J. Tabrikian, “MMSE-based filtering in presence of non-Gaussian system and measurement noise,” IEEE Trans. Aerosp. Electron. Syst., vol. 46, no. 3, pp. 1153–1170, Jul. 2010.

[48] M. Raitoharju and R. Piché, “On computational complexity reduction methods for Kalman filter extensions,” IEEE Aeros. Electron. Syst. Mag., vol. 34, no. 10, pp. 2–19, Oct. 2019.

[49] N. Aronszajn, “Theory of reproducing kernels,” Trans. Am. Math. Soc., vol. 68, no. 3, pp. 337–404, 1950.

[50] L. Ralaivola and F. d’Alché Buc, “Dynamical modeling with kernels for nonlinear time series prediction,” in Proc. Adv. Neural Inf. Process. Syst., vol. 16, pp. 1–8, 2003.

[51] W. Liu, I. Park, Y. Wang, and J. C. Príncipe, “Extended kernel recursive least squares algorithm,” IEEE Trans. Signal Process., vol. 57, no. 10, pp. 3801–3814, Oct. 2009.

[52] S. Van Vaerenbergh, J. Via, and I. Santamaría, “A sliding-window kernel RLS algorithm and its application to nonlinear channel identification,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. Proc., vol. 5, 2006, pp. V–V.

[53] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in Proc. Int. Conf. Comput. Learn. Theory. Berlin, Germany: Springer, 2001, pp. 416–426.

[54] I. Steinwart, “On the influence of the kernel on the consistency of support vector machines,” J. Mach. Learn. Res., vol. 2, pp. 67–93, 2001.

[55] B. Hajek, Random Processes for Engineers. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[56] K. Xiong, H. Zhang, and C. Chan, “Performance evaluation of UKF-based nonlinear filtering,” Automatica, vol. 42, no. 2, pp. 261–270, 2006.

[57] C. D. Richmond and L. L. Horowitz, “Parameter bounds on estimation accuracy under model misspecification,” IEEE Trans. Signal Process., vol. 63, no. 9, pp. 2263–2278, May 2015.

[58] S. Fortunati, F. Gini, M. S. Greco, and C. D. Richmond, “Performance bounds for parameter estimation under misspecified models: Fundamental findings and applications,” IEEE Signal Process. Mag., vol. 34, no. 6, pp. 142–157, Nov. 2017.