Adaptive Kernel Kalman Filter

Mengwei Sun, Mike E. Davies, Ian Proudler, James R. Hopgood

2021 Sensor Signal Processing for Defence Conference (SSPD) | 978-1-6654-3314-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSPD51364.2021.9541455

***Abstract*—This paper presents a novel model-based Bayesian filter called the adaptive kernel Kalman filter (AKKF). The proposed filter approximates the arbitrary probability distribution functions (PDFs) of hidden states as empirical kernel mean embeddings (KMEs) in reproducing kernel Hilbert spaces (RKHSs). Specifically, particles are generated and updated in the data space to capture the properties of the dynamical system model, while the corresponding kernel weight vector and matrix associated with the particles' feature mappings are predicted and updated in the RKHS based on the kernel Kalman rule (KKR). We illustrate and confirm the advantages of our approach through simulation, offering detailed comparison with the unscented Kalman filter (UKF), particle filter (PF) and Gaussian particle filter (GPF) algorithms.**

***Index Terms*—Adaptive kernel Kalman filter, kernel Kalman rule, kernel mean embedding, non-linear Bayesian filter**

I. Introduction

Non-linear/non-Gaussian estimation problems in dynamic systems arise in many fields including statistical signal processing, target tracking, satellite navigation, and so on. In order to make inference about a dynamic system, dynamical state-space models (DSMs) are required, including a process model describing the evolution of the hidden states with time, and a measurement model relating the observations to the states. From a Bayesian perspective, the filter for dynamical system inference is designed to estimate the hidden states by recursively computing the posterior probability density function (PDF). Historically, the main focus of sequential Bayesian filters has been on model-based systems where there exists an explicit formulation of the DSM [1]. However, more recently data-driven Bayesian filters have been proposed where the DSM is unknown or partially known but data examples of state-observation pairs are provided [2]. In both scenarios the filters can be broken down into prediction and update stages.

The Kalman filter (KF) provides the optimal Bayesian solution for linear DSMs when both the prediction and posterior distributions are Gaussian. The extended Kalman filter (EKF) is a common form for the application of the KF to nonlinear systems [3], by using the first derivatives to approximate the observation function by a linear system of equations which can cause poor approximation performance when the model is highly non-linear or when the posterior distributions are multimodal. The unscented Kalman filter (UKF), an alternative to the EKF, was proposed in [4] and uses a weighted set of deterministic particles (so called sigma points) in the state space to approximate the state distribution. Compared with the EKF, the UKF can significantly improve the accuracy of the approximations, but divergence can still occur as in both filters the state distributions are essentially approximated as Gaussian. A more general solution to the non-linear Bayesian filter can be found in the bootstrap particle filter (PF) proposed in [5], in which the hidden state distributions are represented through a weighted set of random particles. Resampling is a necessary step in the bootstrap PF which induces an increase in complexity and is hard to parallelize [6]. To avoid the need for resampling, some specific implementations of the bootstrap PF have been proposed that further approximate the hidden state distribution at each time with a Gaussian, such as the Gaussian particle filter (GPF) [6], and the Gauss–Hermite filter [7].

Different from the approaches above, a number of works have used the recently formulated kernel Bayes rule (KBR) to develop data-driven Bayesian filters based on kernel mean embeddings (KMEs) [2], [8]. Here the unknown measurement model was inferred from prior training data. Owing to the virtue of KMEs, these methods can effectively deal with problems that involve unknown models or strong non-linear structures [9]. However, the feature space for the kernel embeddings remains restricted to the feature space defined by the training data set. Therefore, the performance of these filters relies heavily on there being sufficient similarity between the training data and the test data [10].

Inspired by the KBR [8] and kernel Kalman rule (KKR) [11], we explore the potential of KMEs within full model based filters and introduce a new hybrid filter called the adaptive kernel Kalman filter (AKKF). The main contributions of this paper can be summarized as:

- We derive a new model based Bayesian filter that is a hybrid of kernel based methods and PFs, in which both the prediction and posterior distributions are embedded into a kernel feature space but the known measurement and transition operators are used to calculate the update rules. This is in contrast to the PF where the prediction and posterior distributions are calculated through empirical PDF estimates in the data space.
- The proposed filter can avoid the problematic resampling in most PFs. In passing, we also highlight a missing link between the UKF sigma point method and the kernel conditional embedding operator.

The rest of the paper is set out as follows. The KME and KKR are reviewed in Section II. Section III presents the proposed AKKF. Simulation results for bearing–only tracking (BOT) problem are presented in Section IV and finally conclusions

M. W. Sun, M. E. Davies and J. R. Hopgood are with Institute of Digital Communications, University of Edinburgh, Edinburgh, EH9 3FG, U.K. E-mail: (msun; mike.davies; james.hopgood)@ed.ac.uk.

I. Proudler is with the Centre for Signal & Image Processing (CeSIP), Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, U.K. E-mail: ian.proudler@strath.ac.uk.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

are drawn in Section V.

II. Preliminaries

In this section, we briefly review the frameworks of the KME and data-driven KKR, see [8] and [11] for details.

A. Kernel Mean Embedding

A reproducing kernel Hilbert space (RKHS) denoted as $\mathcal{H}_x$ on the data space $\mathcal{X}$ with a Kernel function $k_x(x, x')$ is defined as a Hilbert space of functions with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_x}$ that has some additional properties [8]. The KME approach represents a conditional distribution $P(X|y)$ by an element in the RKHS as,

$$\mu_{X|y} := \mathbb{E}_{X|y}\left[\phi_x(X)\right] = \int_{\mathcal{X}} \phi_x(x) dP(x|y). \tag{1}$$

where $\phi_x(x) \in \mathcal{H}_x$ represents the feature mapping of x in RKHS $\mathcal{H}_x$ for all $x \in \mathcal{X}$. $\mu_{X|y}$ is a family of points, each indexed by fixing Y to a particular value y. By defining the conditional operator $C_{X|Y}$ as the linear operator which takes the feature mapping of a fixed value y as the input and outputs the corresponding conditional KME [10], the KME of a conditional distribution defined in (1), under certain conditions, is calculated as,

$$\mu_{X|y} = C_{X|Y}\phi_y(y) = C_{XY}\left(C_{YY} + \lambda I\right)^{-1}\phi_y(y). \tag{2}$$

Here, C_{XY} and C_{YY} represent the covariance operators in the tensor product feature spaces $\mathcal{H}_x \otimes \mathcal{H}_y$ and $\mathcal{H}_y \otimes \mathcal{H}_y$, respectively. The term λ is a regularization parameter to ensure that the inverse is well defined.

If instead of access to the true underlying distribution, as required by (1), an empirical estimate of the PDF is available through a particle representation, the KMEs can be estimated directly from these particles. Hence, given the sample set $\mathcal{D}_{XY} = \{(x^{\{1\}}, y^{\{1\}}), \ldots, (x^{\{M\}}, y^{\{M\}})\}$ which are drawn i.i.d. from $P(X, Y)$ with the feature mappings $\Phi := \left[\phi_x(x^{\{1\}}), \ldots, \phi_x(x^{\{M\}})\right]$ and $\Upsilon := \left[\phi_y(y^{\{1\}}), \ldots, \phi_y(y^{\{M\}})\right]$, the estimate of the conditional embedding operator $\hat{C}_{X|Y}$ is obtained as a linear regression in the RKHS [12], as shown in the illustration in Fig. 1. Then, the empirical KME of the conditional distribution is calculated by a linear algebra operation as,

$$\hat{\mu}_{X|y} = \hat{C}_{X|Y}\phi_y(y) = \Phi\left(G_{YY} + \lambda I\right)^{-1}\Upsilon^{\mathrm{T}}\phi_y(y) = \Phi\mathbf{w}, \tag{3}$$

$$\mathbf{w} = \left(G_{YY} + \lambda I\right)^{-1} G_{:,y}. \tag{4}$$

Here, $G_{YY} = \Upsilon^{\mathrm{T}}\Upsilon$ is the Gram matrix for the samples from the observation variable Y. The input test variable is $y \in \mathcal{Y}$. The kernel weight vector $\mathbf{w} = \left[w^{\{1\}}, \ldots, w^{\{M\}}\right]^{\mathrm{T}}$ includes M non-uniform weights and is calculated based on the vector of kernel functions $G_{:,y} = \left[k_y(y^{\{1\}}, y), \ldots, k_y(y^{\{M\}}, y)\right]^{\mathrm{T}}$. In summary, an empirical KME can represent a PDF over a basis at RKHS with the corresponding weight vector, which has the advantages of low computational cost and low sample complexity.

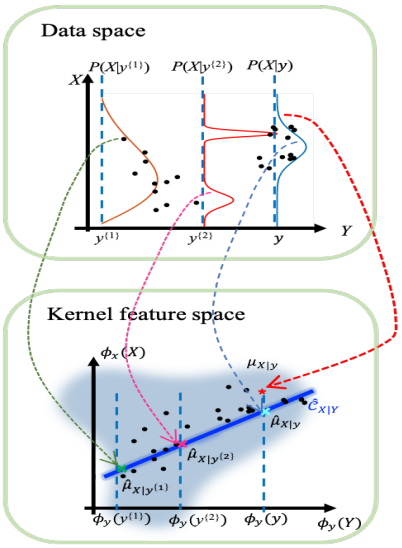

Fig. 1: KME of the conditional distribution $P(X|y)$ is embedded as a point in kernel feature space as $\mu_{X|y} = \int_X \phi_x(x) dP(x|y)$. Given the training data sampled from $P(X, Y)$, the empirical KME of $P(X|y)$ is approximated as a linear operation in RKHS, i.e., $\hat{\mu}_{X|y} = \hat{C}_{X|Y}\phi_y(y) = \Phi\mathbf{w}$. Legend: $\cdot$ samples, $\times$: empirical KME, $*$: KME.

B. Kernel Kalman Rule

The KKR was proposed in [11] as a recursive least squares estimator for KMEs of posterior distributions. In the proposed empirical KKR [11], the mean embedding and covariance operator are predicted and updated similar to the way a conventional KF does but relying on the training data set $\mathcal{D}_{\tilde{X}XY} = \{(\tilde{x}^{\{1\}}, x^{\{1\}}, y^{\{1\}}), \ldots, (\tilde{x}^{\{M\}}, x^{\{M\}}, y^{\{M\}})$. Here, $\tilde{x}^{\{i\}}$ denotes the preceding state of $x^{\{i\}}, i = 1, \ldots, M$, and $y^{\{i\}}$ is the corresponding observation. The feature mappings of training data are represented as $\tilde{\Phi} := \left[\phi_x(\tilde{x}^{\{1\}}) \ldots, \phi_x(\tilde{x}^{\{M\}})\right]$, $\Phi := \left[\phi_x(x^{\{1\}}) \ldots, \phi_x(x^{\{M\}})\right]$ and $\Upsilon := \left[\phi_y(y^{\{1\}}) \ldots, \phi_y(y^{\{M\}})\right]$, respectively. The estimate of the preceding state is given by the KME $\hat{\mu}^{+}_{x_{n-1}}$ and the covariance operator $\hat{C}^{+}_{x_{n-1},x_{n-1}}$. Based on the derivations in [11], the kernel Kalman filter prediction and update steps consist of the following:

$$\hat{\mu}^{-}_{x_n} = \hat{C}_{X|\tilde{X}}\hat{\mu}^{+}_{x_{n-1}}, \tag{5}$$

$$\hat{C}^{-}_{x_n,x_n} = \hat{C}_{X|\tilde{X}}\hat{C}^{+}_{x_{n-1},x_{n-1}}\hat{C}^{\mathrm{T}}_{X|\tilde{X}} + \mathcal{V}. \tag{6}$$

$$\hat{\mu}^{+}_{x_n} = \hat{\mu}^{-}_{x_n} + \mathcal{Q}_n\left(\phi_y(y_n) - \hat{C}_{Y|X}\hat{\mu}^{-}_{x_n}\right), \tag{7}$$

$$\hat{C}^{+}_{x_n,x_n} = \hat{C}^{-}_{x_n,x_n} - \mathcal{Q}_n\hat{C}_{Y|X}\hat{C}^{-}_{x_n,x_n}. \tag{8}$$

Here, the conditional embedding operators for the distributions $P(X|\tilde{X})$ and $P(Y|X)$, represented by $\hat{C}_{X|\tilde{X}}$ and $\hat{C}_{Y|X}$, are calculated based on training data as $\hat{C}_{X|\tilde{X}} = \Phi\left(K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}}\right)^{-1}\tilde{\Phi}$ and $\hat{C}_{Y|X} = \Upsilon\left(K_{xx} + \lambda_K\right)^{-1}\Phi$, respectively. The Gram matrices are $K_{\tilde{x}\tilde{x}} = \tilde{\Phi}^{\mathrm{T}}\tilde{\Phi}$ and $K_{xx} = \Phi^{\mathrm{T}}\Phi$. The covariance of the transition residual matrix is represented as $\mathcal{V}$, and the kernel Kalman gain operator $\mathcal{Q}_n$ is given by [11],

$$\mathcal{Q}_n = \hat{C}^{-}_{x_n,x_n}\hat{C}^{\mathrm{T}}_{Y|X}\left(\hat{C}_{Y|X}\hat{C}^{-}_{x_n,x_n}\hat{C}^{\mathrm{T}}_{Y|X} + \mathcal{R}\right)^{-1}. \tag{9}$$

where $\mathcal{R}$ is the covariance of the residual of the observation operator.

It should be noted that the existing filters based on the KKR are fully data driven and therefore of use when the DSM is not available and the test data has high similarities to the training data. Data-driven based KKR filters have been used for tracking problems that include the position estimate of a target which follows rotation in a circle or oscillatory rotation [11]. However, the data-driven based filters are only effective when training data provides a good description for the current state, and will fail when the target moves out of the training space. To mitigate this shortcoming, we present a new type of kernel Kalman filter defined for model based scenarios in Section III.

III. Adaptive Kernel Kalman Kilter

Inspired by the data-driven based KKR [11] and PF, the proposed adaptive kernel Kalman filter aims to take all the benefits of the KKR and PF. The proposed AKKF is executed in both data space and kernel feature space. In the kernel feature space, the kernel weight vector and positive definite weight matrix are estimated using the KKR, which requires an embedding of the state update function to update the estimate from time $n-1$ to time n. Then an embedding of the measurement function is used to update the prior estimate at time n to the posterior estimate at time n. In data space, the embeddings for the state process and measurement functions are obtained as follows: a proposal distribution is generated using information from the kernel space at time $n-1$, which is then propagated through the non-linear DSM. The following subsections will derive the proposed AKKF, with the implementation is summarized in Algorithm I.

A. Embedding the Posterior Distribution at Time $n-1$

Let the particles and corresponding kernel feature mappings at time $n-1$ be represented by $x_{n-1}^{\{i=1:M\}}$ and $\phi_x(x_{n-1}^{\{i=1:M\}})$, respectively. Given also the previous weight vector $\mathbf{w}_{n-1}^{+}$ and positive definite weight matrix S_{n-1}^{+}, the empirical KME and covariance operator for the posterior $p(x_{n-1}|y_{1:n-1})$ are:

$$\hat{\mu}_{x_{n-1}}^{+} = \Phi_{n-1}\mathbf{w}_{n-1}^{+}, \tag{10}$$

$$\hat{C}_{x_{n-1}x_{n-1}}^{+} = \Phi_{n-1}S_{n-1}^{+}\Phi_{n-1}^{\mathrm{T}}. \tag{11}$$

where the feature mappings is calculated as $\Phi_{n-1} = \left[\phi_x(x_{n-1}^{\{1\}}), \ldots, \phi_x(x_{n-1}^{\{M\}})\right]$ [11]. Specifically, suppose $x_{n-1} = [x_{n-1,1}, \ldots, x_{n-1,d}]^{\mathrm{T}}$ is a d-dimension vector and the quadratic kernelis utilized. Then, its feature mapping is [13],

$$\begin{aligned}\phi_x(x_{n-1}) = \Big[&x_{n-1,d}^2, \ldots, x_{n-1,1}^2, \sqrt{2}x_{n-1,d}x_{n-1,d-1}, \ldots, \\ &\sqrt{2}x_{n-1,2}x_{n-1,1}, \sqrt{2}x_{n-1,d}, \ldots, \sqrt{2}x_{n-1,1}, 1\Big]^{\mathrm{T}}.\end{aligned} \tag{12}$$

Therefore, from (10), the empirical KME $\hat{\mu}_{x_{n-1}}^{+}$ is represented in terms of the expectations of X_{n-1}^2 and X_{n-1}, as $\hat{\mu}_{x_{n-1}}^{+} = \left[\mathbb{E}(X_{n-1}^2), \sqrt{2}\mathbb{E}(X_{n-1}X'_{n-1}), \sqrt{2}\mathbb{E}(X_{n-1}), 1\right]^{\mathrm{T}}$. Then, the $\mathbb{E}(X_{n-1})$ and $\mathbb{E}(X_{n-1}^2)$ are extracted from $\hat{\mu}_{x_{n-1}}^{+}$ and passed to the data space. As pointed out in [14], the approximation of a Gaussian distribution is easier to realize than the approximation of an arbitrary non-linear function. Hence, the proposed AKKF uses a new weighted sample representation called proposal particles to approximate the KME that can be exactly propagated through the non-linearity. The proposal particles are generated according to the importance distribution as,

$$\tilde{x}_{n-1}^{\{i=1:M\}} \sim \mathcal{N}\left(\mathbb{E}\left(X_{n-1}\right), \mathrm{Cov}\left(X_{n-1}\right)\right), \tag{13}$$

$$\mathrm{Cov}\left(X_{n-1}\right) = \mathbb{E}(X_{n-1}^2) - \mathbb{E}\left(X_{n-1}\right)\mathbb{E}\left(X_{n-1}\right)^{\mathrm{T}}. \tag{14}$$

The feature mappings of the proposal particles are represented as $\Psi_{n-1} = \left[\phi_x(\tilde{x}_{n-1}^{\{1\}}), \ldots, \phi_x(\tilde{x}_{n-1}^{\{M\}})\right]$. Then, the posterior distribution $p(x_{n-1}|y_{1:n-1})$ can also be embedded using the new basis Ψ_{n-1} and therefore the weight vector and covariance operator are transformed into Ψ_{n-1} as,

$$\hat{\mu}_{x_{n-1}}^{+} = \Psi_{n-1}\tilde{\mathbf{w}}_{n-1}^{+}, \tag{15}$$

$$\hat{C}_{x_{n-1}x_{n-1}}^{+} = \Psi_{n-1}\tilde{S}_{n-1}^{+}\Psi_{n-1}^{\mathrm{T}}. \tag{16}$$

Substituting (15) into (10), and (16) into (11), respectively, the proposal kernel weight vector $\tilde{\mathbf{w}}_{n-1}^{+}$ and matrix $\tilde{S}_{n-1}^{+}$ are calculated as,

$$\tilde{\mathbf{w}}_{n-1}^{+} = \left(K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}}I\right)^{-1}K_{\tilde{x}x}\mathbf{w}_{n-1}^{+} = \Gamma_{n-1}\mathbf{w}_{n-1}^{+}, \tag{17}$$

$$\tilde{S}_{n-1}^{+} = \Gamma_{n-1}S_{n-1}^{+}\Gamma_{n-1}^{\mathrm{T}}, \tag{18}$$

where Γ_{n-1} defined in (17) represents the change of basis from Φ_{n-1} to Ψ_{n-1}, $K_{\tilde{x}\tilde{x}} = \Psi_{n-1}^{\mathrm{T}}\Psi_{n-1}$ represents the Gram matrix of the proposal particles at time $n-1$, $K_{\tilde{x}x} = \Psi_{n-1}^{\mathrm{T}}\Phi_{n-1}$ is the matrix between the particles and proposal particles at time $n-1$, and $\lambda_{\tilde{K}}$ is the regularization parameter to modify $K_{\tilde{x}\tilde{x}}$.

B. Prediction from Time $n-1$ to Time n

The proposal particles at time $n-1$ are propagated through the process function to achieve the prediction particles, i.e.,

$$x_n^{\{i\}} = f(\tilde{x}_{n-1}^{\{i\}}, u_n^{\{i\}}), \quad i = 1 \ldots M. \tag{19}$$

Algorithm 1 Adaptive kernel Kalman filter

Require: DSM: process model and measurement model.

1: **Initialization**: Set the initial particles in the data space $\tilde{x}_0^{\{i=1:M\}} \sim P_{\mathrm{init}}$, $\mathbf{w}_0 = 1/M\,[1, \ldots, 1]^{\mathrm{T}}$.
2: **for** $n = 1 : N$ **do**
3: Prediction:
- First, in the data space: $x_n^{\{i\}} = f(\tilde{x}_{n-1}^{\{i\}}, u_n^{\{i\}})$,

$\Rightarrow$ Second, in the kernel feature space with basis Φ_n: $\mathbf{w}_n^{-} = \Gamma_{n-1}\mathbf{w}_{n-1}^{+}$, $S_n^{-} = \tilde{S}_{n-1}^{+} + V_n$.

4: Update:
- First, in the data space: $y_n^{\{i\}} = h(x_n^{\{i\}}, v_n^{\{i\}})$,

$\Rightarrow$ Second, in the kernel feature space with basis Φ_n: $\mathbf{w}_n^{+} = \mathbf{w}_n^{-} + Q_n\left(G_{:,y_n} - G_{yy}\mathbf{w}_n^{-}\right)$, $S_n^{+} = S_n^{-} - Q_nG_{yy}S_n^{-}$. $\hat{\mu}_{x_n} = \Phi_n\mathbf{w}_n^{+}$.

5: Proposal particles draw:
- First, in the data space: $\tilde{x}_n^{\{i=1:M\}} \sim \mathcal{N}\left(\mathbb{E}\left(X_n\right), \mathbb{E}\left(X_n^2\right) - \mathbb{E}\left(X_n\right)\mathbb{E}\left(X_n\right)^{\mathrm{T}}\right)$.

$\Rightarrow$ Second, in the kernel feature space with basis Ψ_n: $\tilde{\mathbf{w}}_n^{+} = \Gamma_n\mathbf{w}_n^{+}$, $\tilde{S}_n^{+} = \Gamma_nS_n^{+}\Gamma_n^{\mathrm{T}}$.

6: **end for**

where $u_n^{\{i\}}$ represents a process noise sample drawn from the process noise distribution. Then, the transitional probability $p(x_n|x_{n-1})$ is embedded using the new basis defined by the feature mappings of the prediction particles $\Phi_n = \left[\phi_x(x_n^{\{1\}}), \ldots, \phi_x(x_n^{\{M\}})\right]$, and is approximated as:

$$\begin{aligned} p(x_n|x_{n-1}) \mapsto \hat{\mu}_{x_n}^- &= \Phi_n \mathbf{w}_n^- = \hat{C}_{x_n|x_{n-1}} \hat{\mu}_{x_{n-1}}^+, \\ &= \Phi_n \left(K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}} I\right)^{-1} K_{\tilde{x}x} \mathbf{w}_{n-1}^+, \end{aligned} \tag{20}$$

where $\mathbf{w}_n^-$ is the prior kernel weight vector and $\hat{C}_{x_n|x_{n-1}}$ represents the empirical transition operator. Next, the empirical predictive covariance operator $\hat{C}_{x_n x_n}^-$ with the corresponding prior kernel weight matrix S_n^- is computed as,

$$\begin{aligned} &\hat{C}_{x_n x_n}^- = \Phi_n S_n^- \Phi_n^{\mathrm{T}} = \hat{C}_{x_n|\tilde{x}_{n-1}} \hat{C}_{x_{n-1}x_{n-1}}^+ \hat{C}_{x_n|\tilde{x}_{n-1}}^{\mathrm{T}} + \mathcal{V}_n, \\ =&\hat{C}_{x_n|\tilde{x}_{n-1}} \Psi_{n-1} \tilde{S}_{n-1}^+ \Psi_{n-1}^{\mathrm{T}} \hat{C}_{x_n|\tilde{x}_{n-1}}^{\mathrm{T}} + \mathcal{V}_n = \Phi_n \tilde{S}_{n-1}^+ \Phi_n^{\mathrm{T}} + \mathcal{V}_n. \end{aligned} \tag{21}$$

Here, $\mathcal{V}_n$ represents the transition residual matrix,

$$\begin{aligned} \mathcal{V}_n &= \frac{1}{M} \left(\hat{C}_{x_n|\tilde{x}_{n-1}} \Psi_{n-1} - \Phi_n\right) \left(\hat{C}_{x_n|\tilde{x}_{n-1}} \Psi_{n-1} - \Phi_n\right)^{\mathrm{T}}, \\ =&\Phi_n \underbrace{\left[\frac{1}{M} \left(\left(K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}} I\right)^{-1} K_{\tilde{x}\tilde{x}} - I\right) \left(\left(K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}} I\right)^{-1} K_{\tilde{x}\tilde{x}} - I\right)^{\mathrm{T}}\right]}_{V_n} \Phi_n^{\mathrm{T}}, \end{aligned} \tag{22}$$

where V_n is the finite matrix representation of $\mathcal{V}_n$. Based on (20)-(22), the prior $\mathbf{w}_n^-$ and S_n^- are calculated as,

$$\mathbf{w}_n^- = \left(K_{\tilde{x}\tilde{x}} + \lambda_{\tilde{K}} I\right)^{-1} K_{\tilde{x}x} \mathbf{w}_{n-1}^+ = \Gamma_{n-1} \mathbf{w}_{n-1}^+, \tag{23}$$

$$S_n^- = \tilde{S}_{n-1}^+ + V_n. \tag{24}$$

C. Update at Time n

The observation particles are updated based on the observation model as,

$$y_n^{\{i\}} = g(\tilde{x}_n^{\{i\}}, v_n^{\{i\}}), \quad i = 1 \ldots M. \tag{25}$$

where $v_n^{\{i\}}$ represents a measurement noise sample drawn from the measurement noise distribution. The kernel mappings of observation particles in the kernel feature space are $\Upsilon_n = \left[\phi_y(y_n^{\{1\}}), \ldots, \phi_y(y_n^{\{M\}})\right]$. The posterior KME and the corresponding covariance operator are calculated as [11],

$$\hat{\mu}_{x_n}^+ = \Phi_n \mathbf{w}_n^+ = \hat{\mu}_{x_n}^- + \mathcal{Q}_n \left[\phi_y(y_n) - \hat{C}_{y_n|x_n} \hat{\mu}_{x_n}^-\right], \tag{26}$$

$$\hat{C}_{x_n x_n}^+ = \Phi_n S_n^+ \Phi_n^{\mathrm{T}} = \mathrm{cov}(\phi_x(x_n) - \hat{\mu}_{x_n}^+). \tag{27}$$

where $\mathbf{w}_n^+$ and S_n^+ represent the posterior kernel weight and matrix, respectively. The kernel Kalman gain operator denoted as $\mathcal{Q}_n$ is derived by minimizing the residual error $\hat{C}_{x_n x_n}^+$. According to derivations in [11], $\mathcal{Q}_n$ is calculated as,

$$\mathcal{Q}_n = \hat{C}_{x_n x_n}^- C_{y_n|x_n}^{\mathrm{T}} \left(\hat{C}_{y_n|x_n} \hat{C}_{x_n x_n}^- \hat{C}_{y_n|x_n}^{\mathrm{T}} + \mathcal{R}\right)^{-1}. \tag{28}$$

where $\mathcal{R}$ is the covariance matrix of the observation operator residual and is approximated as $\mathcal{R} = \kappa I$. The empirical likelihood operator is calculated as,

$$\begin{aligned} \hat{C}_{y_n|x_n} &= \hat{C}_{y_n x_n} \hat{C}_{x_n x_n}^{-1} = \Upsilon_n \left(\Phi_n^{\mathrm{T}} \Phi_n + \lambda_K I\right)^{-1} \Phi_n^{\mathrm{T}}, \\ &= \Upsilon_n \left(K_{xx} + \lambda_K I\right)^{-1} \Phi_n^{\mathrm{T}} = \Upsilon_n K_{xx}^{-1} \Phi_n^{\mathrm{T}}. \end{aligned} \tag{29}$$

Here, the Gram matrix of particles at time n is calculated as $K_{xx} = \Phi_n^{\mathrm{T}} \Phi_n$, and λ_K is the regularization parameter to modify the covariance operator K_{xx}. In this paper, λ_K is set to be 0. Substituting (21) and (29) into (28), $\mathcal{Q}_n$ can be calculated as,

$$\mathcal{Q}_n = \Phi_n S_n^- \Upsilon_n^{\mathrm{T}} \left(\Upsilon_n S_n^- \Upsilon_n^{\mathrm{T}} + \kappa I\right)^{-1} = \Phi_n \underbrace{S_n^- \left(G_{yy} S_n^- + \kappa I\right)^{-1}}_{Q_n} \Upsilon_n^{\mathrm{T}}. \tag{30}$$

where Q_n is the finite matrix representation of $\mathcal{Q}_n$ in terms of the current basis Φ_n. The Gram matrix of the observation at time n is $G_{yy} = \Upsilon_n^{\mathrm{T}} \Upsilon_n$. Then, the updated KME vector and matrix are given by,

$$\hat{\mu}_{x_n}^+ = \Phi_n \mathbf{w}_n^+ = \Phi_n \left[\mathbf{w}_n^- + Q_n \left(G_{:,y_n} - G_{yy} \mathbf{w}_n^-\right)\right], \tag{31}$$

$$\hat{C}_{x_n x_n}^+ = \Phi_n S_n^+ \Phi_n^{\mathrm{T}} = \hat{C}_{x_n x_n}^- - \Phi_n Q_n G_{yy} S_n^- \Phi_n^{\mathrm{T}}. \tag{32}$$

where the kernel vector of the measurement at time n is $G_{:,y_n} = \Upsilon_n^{\mathrm{T}} \phi_y(y_n)$. Based on the derivations above, the weight mean vector and covariance matrix are finally updated as,

$$\mathbf{w}_n^+ = \mathbf{w}_n^- + Q_n \left(G_{:,y_n} - G_{yy} \mathbf{w}_n^-\right), \tag{33}$$

$$S_n^+ = S_n^- - Q_n G_{yy} S_n^-. \tag{34}$$

IV. Simulation Results

Bearing–only tracking (BOT) is one of the fundamental problems in target tracking systems. In this section, we report the tracking performance of different filters applied to BOT problems of a single target using a single sensor in a 2-D space. The corresponding dynamical state-space model (DSM) is described by the equations:

$$\mathbf{x}_n = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{n-1} + \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix} \mathbf{u}_n, \tag{35}$$

$$y_n = \tan^{-1}\left(\frac{\eta_n}{\xi_n}\right) + v_n. \tag{36}$$

Here, n represents time index and $n = 1, \ldots, N$. The hidden states are $\mathbf{x}_n = [\xi_n, \dot{\xi}_n, \eta_n, \dot{\eta}_n]^T$, where (ξ_n, η_n) and $(\dot{\xi}_n, \dot{\eta}_n)$ represent the target position and the corresponding velocity in X-axis and Y-axis, y_n is the corresponding observation. The process noise $\mathbf{u}_n$ follows Gaussian distribution $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 \mathbf{I}_2)$ and $\sigma_u = 0.001$. Following [6], the prior distribution for the initial state is specified as $\mathbf{x}_0 \sim \mathcal{N}(\mu_0, \mathbf{P}_0)$ with $\mu_0 = [-0.05, 0.001, 0.7, -0.05]^{\mathrm{T}}$ and,

$$\mathbf{P}_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.1 & 1 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

Fig. 2 and Fig. 3 display two representative trajectories and the tracking performance obtained by four filters: UKF, GPF, PF, and the proposed AKKF using a quadratic kernel. The observer is located at $[0, 0]$. The numbers of particles used for

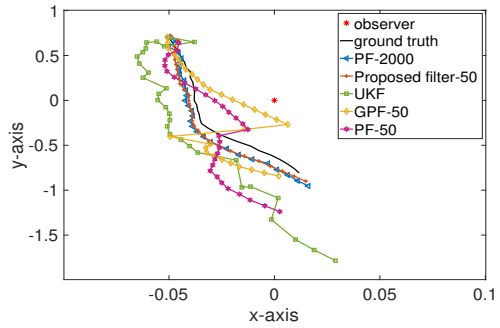

Fig. 2: Trajectory-1: Example of tracking a moving target in two dimensions with the PF, UKF, GPF, and AKKF filters.

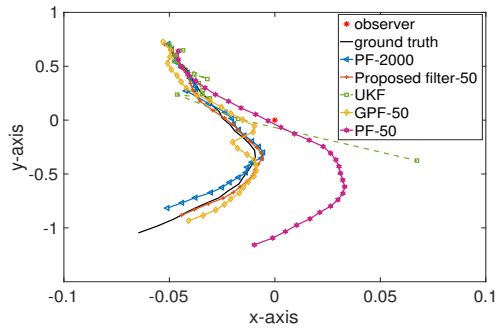

Fig. 3: Trajectory-2: Example of tracking a moving target in two dimensions with the PF, UKF, GPF, and AKKF filters, where UKF diverges.

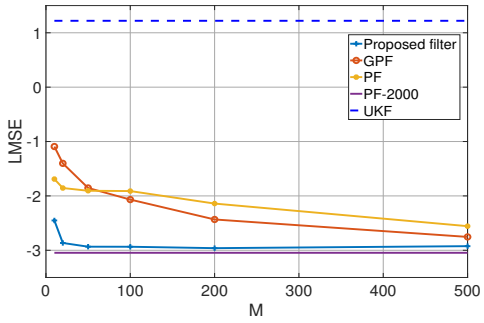

Fig. 4: LMSE performance comparison of the PF, UKF, GPF, and AKKF filters for Trajectory-1.

PF, GPF and AKKF are 50, while the benchmark performance is given by a PF with 2000 particles. Fig. 4 and Fig. 5 shows the average logarithmic mean square error (LMSE) obtained for 100 random realizations of trajectory-1 and trajectory-2 as a function of particle number denoted as M. From the simulation results, we can arrive at the following conclusions. First, for BOT problems, the tracking performance of PF, GPF and AKKF is obviously better than UKF which shows divergence for trajectory-2 as shown in Fig. 3. Second, the proposed AKKF shows significant improvement compared to the PF and GPF with small particle numbers, as shown in Fig. 4 and Fig. 5.

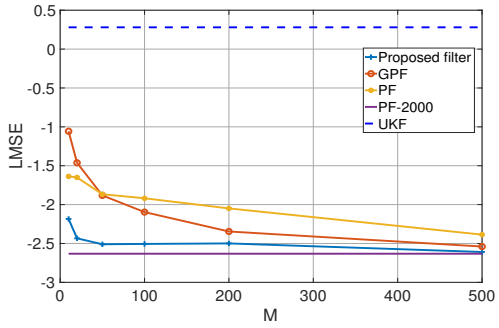

Fig. 5: LMSE performance comparison of the PF, UKF, GPF, and AKKF filters for Trajectory-2.

V. Conclusions

This paper provided a novel model based kernel Kalman filter. By embedding the probabilities into kernel spaces, more feature information of the hidden states and observations can be captured and recorded. Therefore, the proposed AKKF out performs existing algorithms when applied to a BOT problem.

References

[1] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Kalman Filtering*, 2020, pp. 355–417.

[2] M. Kanagawa, Y. Nishiyama, A. Gretton, and K. Fukumizu, "Filtering with state-observation examples via kernel monte carlo filter," *Neural Computation*, vol. 28, no. 2, pp. 382–444, 2016.

[3] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[4] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., vol. 3068, International Society for Optics and Photonics. SPIE, 1997, pp. 182 – 193. [Online]. Available: https://doi.org/10.1117/12.280797

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[6] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, 2003.

[7] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using gauss–hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[8] L. Song, K. Fukumizu, and A. Gretton, "Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models," *IEEE Signal Processing Mag.*, vol. 30, no. 4, pp. 98–111, Jun. 2013.

[9] M. Kanagawa, Y. Nishiyama, A. Gretton, and K. Fukumizu, "Monte carlo filtering using kernel embedding of distributions," in *Proc. Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[10] K. Fukumizu, L. Song, and A. Gretton, "Kernel bayes' rule: Bayesian inference with positive definite kernels," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3753– 3783, Dec. 2013.

[11] G. Gebhardt, A. Kupcsik, and G. Neumann , "The kernel kalman rule," *Machine Learning*, pp. 2113–2157, Jun. 2019.

[12] S. Grünewälder, G. Lever, L. Baldassarre, S. Patterson, A. Gretton, and Massimiliano, "Conditional mean embeddings as regressors," in *Proc. the 29th International Coference on International Conference on Machine*, vol. 1, 2012, pp. 1803–1810.

[13] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, "Kernel mean embedding of distributions: A review and beyond," *Foundations and Trends in Machine Learning*, vol. 10, no. 1-2, pp. 1–141, 2017.

[14] S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Tech. Rep., 1996.