2020 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 21–24, 2020, ESPOO, FINLAND

DIFFUSION FIELD ESTIMATION USING DECENTRALIZED KERNEL KALMAN FILTER WITH PARAMETER LEARNING OVER HIERARCHICAL SENSOR NETWORKS

Shengdi Wang[⋆]*, Ban-Sok Shin*[‡]*, Dmitriy Shutin*[‡]*, Armin Dekorsy*[⋆]

[⋆]Department of Communications Engineering, University of Bremen, Germany
[‡]Institute of Communications and Navigation, German Aerospace Center, Germany
Email: {wang, dekorsy}@ant.uni-bremen.de, {ban-sok.shin, dmitriy.shutin}@dlr.de,

ABSTRACT

In this paper, a task is addressed to track a nonlinear time-varying diffusion field based on data collected by sensor networks. By exploiting kernel methods, the nonlinear field function is approximated by a linear combination of kernel functions in a reproducing kernel Hilbert space (RKHS). To capture the dynamical property of a diffusion field and the relation of system input and output data, a state-space model on weights of these kernel functions is constructed with unknown process noise. Thus, the nonlinear tracking problem is transformed into a linear state estimation solved by Kalman filter. Further, this kernel Kalman filter (KKF) is decomposed into a decentralized fashion in a way to collect sensor data efficiently over a hierarchical network structure with different clusters. To adapt the algorithm to unknown process noise, a decentralized variational Bayesian KKF is proposed to learn the distributions of system unknown variables.

Index Terms— Diffusion field estimation, nonlinear, kernel method, Kalman filter, variational Bayesian method

1. INTRODUCTION

Sensor networks can be applied to estimate spatially distributed fields. The main idea is to utilize collected sensor data to model the environment functions such as temperature, salinity or magnetic field in a specific region [1]. Usually, these functions are nonlinear. The kernel method has been successfully applied for nonlinear estimation in the past [2]. By transforming the input data to the high dimensional feature space, nonlinear problems can be modeled linearly. The kernel-based least squares (LS) estimation [3] and varieties of kernel adaptive filters [4], [5] have been studied for nonlinear estimation. However, when estimating dynamic fields, LS-based methods may have limited performance. The authors in [6] propose a kernel Kalman filter (KKF) for estimating time-varying nonlinear systems and it shows good performance compared to the kernel least mean squares algorithm. However, in [6] spatially distributed fields are not the main focus. In [7], a dynamic field is approximated by a linear combination of known basis functions which are not specified as kernels. By building up a proper linear dynamic model on coefficients of the linear combination, a decentralized Kalman filter is proposed to reconstruct the dynamic field. In practice, the absence of good model for system dynamics will restrict the performance of the Kalman filter. Hence, some model parameters should be estimated or learned, such as a variational Bayesian treatment of linear dynamic systems [8].

In this work, our main contribution is to exploit a kernel Kalman filter for time-varying diffusion field estimation in a decentralized fashion and adapt the algorithm to unknown system parameters with variational Bayesian method. Similar to [7], we specify the basis functions as kernels in the reproducing kernel Hilbert space (RKHS) and construct a simple process model to capture the dynamical properties of the field with unknown Gaussian noise. Then a kernel-based Kalman filer is analyzed in a Bayesian perspective for further variational Bayesian treatment of linear dynamic systems to learn distributions of unknown variables. In addition, we illustrate a decentralized estimation by efficiently collecting data over hierarchical sensor networks separated into clusters. The decentralized processing overcomes a single point of failure of a central processing unit and avoids complicated routing protocols. Finally, numerical simulation results are shown to compare the performance of different algorithms.

2. SYSTEM MODEL

Consider a diffusion field $f(\boldsymbol{x},t)$ with M instantaneous sources localized in a 2D plane in an isotropic medium. The function $f(\boldsymbol{x},t) : \mathcal{X} \to \mathbb{R}$ which maps 2D positions $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^2$ to a scalar real value at time t is modeled by [9]:

$$f(\boldsymbol{x},t)=\sum_{m=1}^{M} \frac{b_m}{4\pi\nu\cdot(t-t_m)} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{p}_m\|^2}{4\nu\cdot(t-t_m)}\right) h(t-t_m), \tag{1}$$

where the source m at Cartesian position $\boldsymbol{p}_m$ with intensity b_m is activated at time t_m. $h(t)$ is the Heaviside-function and parameter ν is the diffusion constant of the medium. This dif-

978-1-7281-6662-9/20/$31.00

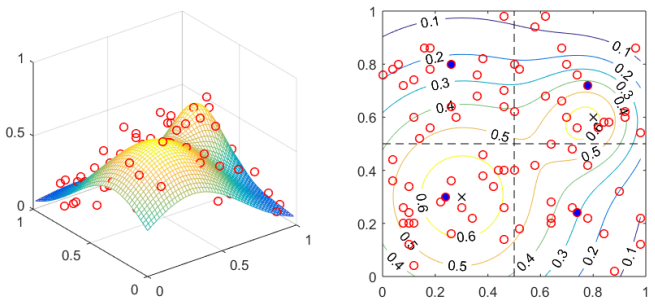

Fig. 1. Left: An example of the diffusion field (1) at a specific time instant with two sources and sensor measurements; Right: Contour of the field with source ($\times$), sensor ($\circ$) and cluster head ($\bullet$) positions.

fusion field is sampled and measured at every Δt interval by N_s sensor nodes which are randomly deployed over a specific area. We use k to indicate the time index such that $t_k = k\Delta t$ where $k \in \mathbb{N}^+$. Each node $j \in \mathcal{N}_s$, where $\mathcal{N}_s$ is a set of nodes, has a time-invariant position $\boldsymbol{x}_j$. We assume that the positions of all sensor nodes are global knowledge which can be obtain at each sensor by broadcast. The local measurement model at each node $j \in \mathcal{N}_s$ at time instant k is

$$d_{j,k} = f(\boldsymbol{x}_j, t_k) + n_{j,k}, \tag{2}$$

where $n_{j,k} \sim \mathcal{N}(0, \sigma_n^2)$ is node and time uncorrelated, i.e., $\mathrm{E}\{n_{j,k} n_{p,q}\} = \sigma_n^2 \delta_{j,p} \delta_{k,q}$ with the Kronecker delta.

These sensors form a hierarchical network with N_c clusters without overlap. Each cluster involves: one cluster head (CH) and normal nodes (NNs). We assume that each NN only connects to the CH in its cluster and different CHs can exchange information. Thus, data over the network is aggregated more efficiently compared to homogeneous networks and only CHs process data with information exchange among neighboring CHs in order to diminish the processing efforts of NNs and save power. An example of the diffusion field measured by sensors in a hierarchical structure is shown in Fig. 1. Now our objective is to estimate/track the nonlinear time-varying diffusion field $f(\,\cdot\,, t_k)$ at each time instant k based on the input and sequential output data set $\{\boldsymbol{x}_j, d_{j,1:k}\}_{j=1}^{N_s}$.

3. PROBLEM FORMULATION

3.1. Reformulated Measurement Model with Kernel Method

If we only consider the instantaneous measurement model (2) at k to estimate the unknown $f(\,\cdot\,, t_k)$, a nonlinear estimation with *kernel trick* [2] can be performed. In this way, nonlinear algorithms are transformed into linear ones formulated by only inner products. Define a kernel function κ: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which represents an inner product of two input sample $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ in a high dimensional feature space with the relation: $\kappa(\boldsymbol{x}_j, \boldsymbol{x}_i) = \Phi^{\mathrm{T}}(\boldsymbol{x}_j)\Phi(\boldsymbol{x}_i)$. By choosing $\Phi(\boldsymbol{x}) = \kappa(\,\cdot\,, \boldsymbol{x})$, the input samples $\boldsymbol{x}$ in the input space $\mathcal{X}$ is mapped to a special case of feature space called reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ [2], which is a vector space of functions. Using such kernels, the nonlinear function $f(\,\cdot\,, t_k)$ at a time instant k has a linear representation in $\mathcal{H}$. Based on (2), we can apply the LS criterion by minimizing a sum of N_s squared residuals between measurements $d_{j,k}$ and estimated $\hat{f}(\,\cdot\,, t_k)$ to find the optimal solution $f^*(\,\cdot\,, t_k)$ in $\mathcal{H}$ [3] as

$$f^*(\,\cdot\,, t_k) = \arg\min_{\hat{f} \in \mathcal{H}} \sum_{j=1}^{N_s} |d_{j,k} - \hat{f}(\boldsymbol{x}_j, t_k)|^2. \tag{3}$$

According to the representer theorem [2], $\hat{f}(\cdot, t_k)$ as a solution of (3) can be represented by a linear combination of kernel functions centered at N_s input samples with weights $w_{j,k}$ as

$$\hat{f}(\,\cdot\,, t_k) = \sum_{j=1}^{N_s} w_{j,k} \kappa(\,\cdot\,, \boldsymbol{x}_j). \tag{4}$$

The terms of this linear combination can be reduced by applying dictionary learning [10]. Thus, some input data (here sensor positions) with high similarity will be removed from the calculation to enhance computation efficiency while keeping performance. We define a dictionary set $\mathcal{D} = \{\kappa(\,\cdot\,, \tilde{\boldsymbol{x}}_\ell)\}_{\ell=1}^{N_d}$ containing kernel functions centered at N_d input samples which can be selected locally from global known N_s sensor positions after dictionary learning. Then, (4) is changed into

$$\hat{f}(\,\cdot\,, t_k) = \sum_{\ell=1}^{N_d} w_{\ell,k} \kappa(\,\cdot\,, \tilde{\boldsymbol{x}}_\ell). \tag{5}$$

With (5), the original LS problem (3) can be reformulated into a kernel least squares (KLS) estimation with a stacked weight vector $\boldsymbol{w}_k = [w_{1,k}, \cdots, w_{N_d,k}]^{\mathrm{T}} \in \mathbb{R}^{N_d}$ and a vector of kernel evaluations $\boldsymbol{\kappa}(\boldsymbol{x}_j) = [\kappa(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}_1), \cdots, \kappa(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}_{N_d})]^{\mathrm{T}} \in \mathbb{R}^{N_d}$:

$$\boldsymbol{w}_k^* = \arg\min_{\boldsymbol{w}_k \in \mathbb{R}^{N_d}} \sum_{j=1}^{N_s} |d_{j,k} - \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{\kappa}(\boldsymbol{x}_j)|^2. \tag{6}$$

Thus, the objective is reduced to finding the optimal weight vector $\boldsymbol{w}$ instead of searching for a function f in $\mathcal{H}$.

Based on (6), we can approximate the local measurement model (2) at each node $j \in \mathcal{N}_s$ by

$$d_{j,k} = \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{\kappa}(\boldsymbol{x}_j) + n_{j,k}. \tag{7}$$

All local measurement models can be stacked into a centralized form at each time instant k as

$$\boldsymbol{d}_k = \boldsymbol{K} \boldsymbol{w}_k + \boldsymbol{n}_k, \tag{8}$$

where $\boldsymbol{d}_k = [d_{1,k}, \cdots, d_{N_s,k}]^{\mathrm{T}} \in \mathbb{R}^{N_s}$ is the stacked measurement vector, $\boldsymbol{K} = [\boldsymbol{\kappa}(\boldsymbol{x}_1), \cdots, \boldsymbol{\kappa}(\boldsymbol{x}_{N_s})]^{\mathrm{T}} \in \mathbb{R}^{N_s \times N_d}$ is the feature matrix and $\boldsymbol{n}_k = [n_{1,k}, \cdots, n_{N_s,k}]^{\mathrm{T}} \in \mathbb{R}^{N_s}$ is the stacked measurement noise vector with covariance $\boldsymbol{R}_k = \sigma_n^2 \boldsymbol{I}_{N_s}$. Note that the feature matrix $\boldsymbol{K}$ is time-invariant because we assume that the sensor positions are fixed during the whole measurement procedure.

3.2. Process Model

So far, we generate a kernel-based measurement model (8) to deal with the nonlinear estimation more easily. Since the diffusion field is time-varying, using only the instantaneous data set at a specific time instant k for estimation may degrade the performance. Definitely, the weight vector $\boldsymbol{w}_k$ should also be time-varying to capture the change of diffusion field $f(\cdot, t_k)$ along with the increasing time instant k. As [6], $\boldsymbol{w}_k$ can be modeled by a first-order Markov process corrupted by Gaussian random noise:

$$\boldsymbol{w}_{k+1} = \boldsymbol{F}_k \boldsymbol{w}_k + \boldsymbol{q}_k, \tag{9}$$

where $\boldsymbol{F}_k \in \mathbb{R}^{N_d \times N_d}$ is a process matrix and $\boldsymbol{q}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ is zero mean Gaussian noise with covariance matrix $\boldsymbol{Q}_k$. However, both parameters $\boldsymbol{F}_k$ and $\boldsymbol{Q}_k$ highly depend on the properties of the true diffusion field and are unknown in our case. In the absence of good knowledge on system dynamics, we can set $\boldsymbol{F}_k = \boldsymbol{I}_{N_d}$ [7]. The process noise covariance matrix $\boldsymbol{Q}_k$ should be estimated or learned. Thus, a Gaussian random walk process model is formulated. Equations (9) and (8) can be regarded as a kernel-based state-space model of a linear dynamic system with system state variable $\boldsymbol{w}_k$.

3.3. Problem Formulation

The main task is to estimate the system state $\boldsymbol{w}_k$ based on available sequential data set from time instant 1 to k: $\{\boldsymbol{x}_j, d_{j,1:k}\}_{j=1}^{N_s}$. We can apply the Kalman filter to solve this state estimation problem. The Kalman filter [11] is an optimal filter in the minimum mean square error (MMSE) sense to estimate system state based on the state-space model (9) and (8) with accurate noise parameters. It can be derived in a Bayesian viewpoint [12] by inferring the posterior distribution based on the likelihood and prior distribution as

$$p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k}) \propto p(\boldsymbol{d}_k|\boldsymbol{w}_k) p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k-1}). \tag{10}$$

Because of assumed Gaussian noise in (9) and (8), all above distributions are Gaussian distributed. Here, the prior distribution is defined by $p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k-1}) = \mathcal{N}(\boldsymbol{w}_k; \hat{\boldsymbol{w}}_{k|k-1}, \boldsymbol{P}_{k|k-1})$ with prediction mean $\hat{\boldsymbol{w}}_{k|k-1}$ and covariance $\boldsymbol{P}_{k|k-1}$, where the subscript notation $k|k-1$ is used to denote the estimate at time instant k given measurements up to $k-1$. From (8), the likelihood distribution is determined by $p(\boldsymbol{d}_k|\boldsymbol{w}_k) = \mathcal{N}(\boldsymbol{d}_k; \boldsymbol{K}\boldsymbol{w}_k, \boldsymbol{R}_k)$. The posterior distribution is defined by $p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k}) = \mathcal{N}(\boldsymbol{w}_k; \hat{\boldsymbol{w}}_{k|k}, \boldsymbol{P}_{k|k})$ with posterior mean $\hat{\boldsymbol{w}}_{k|k}$ and covariance $\boldsymbol{P}_{k|k}$. To estimate the system state $\hat{\boldsymbol{w}}_{k|k}$, the maximum-a-posteriori (MAP) estimation problem can be solved:

$$\begin{aligned} \hat{\boldsymbol{w}}_{k|k} &= \underset{\boldsymbol{w}_k \in \mathbb{R}^{N_d}}{\arg\max}\ p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k}) \\ &= \underset{\boldsymbol{w}_k \in \mathbb{R}^{N_d}}{\arg\max}\ p(\boldsymbol{d}_k|\boldsymbol{w}_k) p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k-1}), \end{aligned} \tag{11}$$

which is equivalent to the MMSE estimation under Gaussian assumptions [13], i.e., the a-posteriori state estimate $\hat{\boldsymbol{w}}_{k|k}$ is given by the mean of $p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k})$.

4. DECENTRALIZED KERNEL KALMAN FILTER

In the Kalman filter, the mean and covariance of prior distribution can be predicted based on the process model (9). Then, by plugging the Gaussian likelihood and prior distribution into (11), the MAP problem can be solved to update the mean and covariance of posterior distribution. Thus, centralized KKF (CKKF) update equations can be obtained [13]:

Predict step:

$$\hat{\boldsymbol{w}}_{k|k-1} = \boldsymbol{F}_{k-1} \hat{\boldsymbol{w}}_{k-1|k-1}, \tag{12}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1} \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}, \tag{13}$$

Update step:

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{K}^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{K} \right)^{-1}, \tag{14}$$

$$\hat{\boldsymbol{w}}_{k|k} = \boldsymbol{P}_{k|k} (\boldsymbol{K}^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{d}_k + \boldsymbol{P}_{k|k-1}^{-1} \hat{\boldsymbol{w}}_{k|k-1}). \tag{15}$$

Note that equations (14) and (15) are equivalent to the traditional form of Kalman filter. This form is more suitable for decentralized implementation of the filter. With a hierarchical sensor network structure, each NN first transmit local measurement $d_{j,k}$ to its CH forming a stacked vector $\boldsymbol{d}_{c,k}$, $\forall c \in \mathcal{N}_c$, where $\mathcal{N}_c$ is a set of clusters. The centralized stacked feature matrix and measurement noise covariance matrix can also be separated into blocks corresponding to clusters as $\boldsymbol{K} = [\boldsymbol{K}_1^{\mathrm{T}}, \cdots, \boldsymbol{K}_{N_c}^{\mathrm{T}}]^{\mathrm{T}}$ and block diagonal $\boldsymbol{R}_k = \mathrm{blkdiag}[\boldsymbol{R}_{1,k}, \cdots, \boldsymbol{R}_{N_c,k}]$. Thus, the following two terms in (14) and (15) collect the global data from all N_c CHs as:

$$\boldsymbol{K}^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{K} = \sum_{c=1}^{N_c} \boldsymbol{K}_c^{\mathrm{T}} \boldsymbol{R}_{c,k}^{-1} \boldsymbol{K}_c, \tag{16}$$

$$\boldsymbol{K}^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{d}_k = \sum_{c=1}^{N_c} \boldsymbol{K}_c^{\mathrm{T}} \boldsymbol{R}_{c,k}^{-1} \boldsymbol{d}_{c,k}. \tag{17}$$

If the communication network of CHs is fully connected, summations in (16) and (17) are immediately obtained by information exchange on terms $\boldsymbol{K}_c^{\mathrm{T}} \boldsymbol{R}_{c,k}^{-1} \boldsymbol{K}_c$ and $\boldsymbol{K}_c^{\mathrm{T}} \boldsymbol{R}_{c,k}^{-1} \boldsymbol{d}_{c,k}$ among CHs. If not, an average consensus scheme [14] can be applied to iteratively calculate (16) and (17). An alternative for consensus is to directly decompose (11) subject to a consensus constraint [13]. In [15], a proposed diffusion Kalman filter is another choice for an arbitrary network. There, (16) and (17) are approximated at each CH c by a collection of information in a neighboring cluster set $\mathcal{A}_c$ including c itself as

$$\boldsymbol{U}_{c,k} = \sum_{i \in \mathcal{A}_c} \boldsymbol{K}_i^{\mathrm{T}} \boldsymbol{R}_{i,k}^{-1} \boldsymbol{K}_i, \tag{18}$$

$$\boldsymbol{u}_{c,k} = \sum_{i \in \mathcal{A}_c} \boldsymbol{K}_i^{\mathrm{T}} \boldsymbol{R}_{i,k}^{-1} \boldsymbol{d}_{i,k}. \tag{19}$$

Then a update step is performed at each $c \in \mathcal{N}_c$:

$$\boldsymbol{P}_{c,k|k} = \left(\boldsymbol{P}_{c,k|k-1}^{-1} + \boldsymbol{U}_{c,k}\right)^{-1}, \tag{20}$$

$$\hat{\boldsymbol{\varphi}}_{c,k|k} = \boldsymbol{P}_{c,k|k}\left(\boldsymbol{u}_{c,k} + \boldsymbol{P}_{c,k|k-1}^{-1}\hat{\boldsymbol{w}}_{c,k|k-1}\right), \tag{21}$$

$$\hat{\boldsymbol{w}}_{c,k|k} = \sum_{i\in\mathcal{A}_c} r_{c,k}\hat{\boldsymbol{\varphi}}_{c,k|k}, \tag{22}$$

where (22) is a weighted average step to combine the intermediate state estimate $\hat{\boldsymbol{\varphi}}_{c,k|k}$ in $\mathcal{A}_c$ and the weight $r_{c,k}$ is decided by network topologies [15]. However, the result can be suboptimal compared to the central solution. In our work, to fast reach the optimal central solution, we suppose a fully connected communication network among CHs. Thus, each CH acts as a central node to gather the whole network information using (18), (19) and perform a local update step as (20), (21) which are identical to the central form (14), (15). Then, the subsequent predict step is calculated locally at each CH.

5. NOISE ADAPTIVE KERNEL KALMAN FILTER

To adapt the KKF to the unknown process noise, we first present a parameter estimation applied in [16] and [6] as a comparison. Then, we exploit the variational Bayesian method to learn distributions of unknown variables in the linear dynamic system in a decentralized way.

5.1. Parameter Estimation

As proposed in [16], the process noise covariance matrix is assumed to be $\boldsymbol{Q}_k = \sigma_q^2\boldsymbol{I}_{N_d}$. Then the parameter σ_q^2 at each time instant k is estimated by

$$\hat{\sigma}_q^2(k) = \frac{\|\hat{\boldsymbol{w}}_{k|k} - \hat{\boldsymbol{w}}_{k|k-1}\|_2^2}{N_d}. \tag{23}$$

An estimated $\hat{\sigma}_q^2(k)$ achieves a compromise between good tracking when the KF at the beginning of the convergent stage or when a sudden change comes to the system and low misalignment when the KF converges to the steady-state. However, a simple diagonal matrix $\boldsymbol{Q}_k$ may not sufficient to capture the statistics of the time-varying state $\boldsymbol{w}_k$.

5.2. Parameter Learning

Another noise adaptive Kalman filter is a variational Bayesian method to online learn distributions of unknown variables based on sequential measurement data. There are different considerations to adapt the KF algorithm to an unknown $\boldsymbol{Q}_k$ such as [17] and [18]. Recall the update equations of KF in (12)-(14). With (13), it is obvious that $\boldsymbol{P}_{k|k-1}$ depends on $\boldsymbol{Q}_{k-1}$ and will further influence the update of posterior distribution on $\boldsymbol{w}_k$. Instead of learning the unknown $\boldsymbol{Q}_{k-1}$, here we refer to the idea of [17] to learn the distributions of covariance matrix $\boldsymbol{P}_{k|k-1}$ and system state $\boldsymbol{w}_k$ together based on sequential data set $\{\boldsymbol{x}_j, d_{j,1:k}\}_{j=1}^{N_s}$ up to time k. Note that here $\boldsymbol{P}_{k|k-1}$ with subscript notation $k|k-1$ is regarded as a random variable. The problem now is to calculate the joint posterior distribution which is proportionate to the likelihood times joint prior distribution:

$$p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}|\boldsymbol{d}_{1:k}) \propto p(\boldsymbol{d}_k|\boldsymbol{w}_k)p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}|\boldsymbol{d}_{1:k-1}). \tag{24}$$

To update the posterior distribution in (24) along with time instants, conjugate priors are chosen which ensure the prior and posterior distribution are in the same model given a likelihood in order to make the calculation mathematically tractable. Following the Bayesian statistics, the inverse Wishart (IW) distribution is used to model the covariance matrix of a Gaussian form given its mean [19]. Thus, the prior term $p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}|\boldsymbol{d}_{1:k-1})$ in (24) can be written as a product of a Gaussian distribution and an IW distribution:

$$\begin{aligned} p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}|\boldsymbol{d}_{1:k-1}) =& \mathcal{N}(\boldsymbol{w}_k; \hat{\boldsymbol{w}}_{k|k-1}, \boldsymbol{P}_{k|k-1})\times \\ & \mathcal{IW}(\boldsymbol{P}_{k|k-1}; v_{k|k-1}, \boldsymbol{W}_{k|k-1}), \end{aligned} \tag{25}$$

where $\hat{\boldsymbol{w}}_{k|k-1}$, $\boldsymbol{P}_{k|k-1}$, $v_{k|k-1}$ and $\boldsymbol{W}_{k|k-1}$ are the learning parameters. To capture the prior information of $\boldsymbol{P}_{k|k-1}$, we select $v_{k|k-1}$ and $\boldsymbol{W}_{k|k-1}$ such that the mean of IW distribution $\boldsymbol{W}_{k|k-1}/(v_{k|k-1} - N_d - 1)$ is set to be the prediction of $\boldsymbol{P}_{k|k-1}$ in (13) with a pre-selected nominal $\tilde{\boldsymbol{Q}}_{k-1}$ as

$$\frac{\boldsymbol{W}_{k|k-1}}{v_{k|k-1} - N_d - 1} = \tilde{\boldsymbol{P}}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_{k-1}^{\mathrm{T}} + \tilde{\boldsymbol{Q}}_{k-1}. \tag{26}$$

The selected prior parameters of IW distribution are

$$v_{k|k-1} = \theta + N_d + 1, \tag{27}$$

$$\boldsymbol{W}_{k|k-1} = \theta\tilde{\boldsymbol{P}}_{k|k-1} \tag{28}$$

with a defined tuning parameter $\theta > 0$ which reflects how the algorithm balances process model information in the update step [17]. After choosing proper conjugate priors as shown in (25), the VB-approach is seeking a function $q(\boldsymbol{\Theta}_k)$ with $\boldsymbol{\Theta}_k = \{\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}\}$ to approximate the posterior distribution $p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}|\boldsymbol{d}_{1:k})$ by minimizing the Kullback-Leibler (KL) divergence between the function q and the true posterior [19]:

$$\mathrm{KL}[q(\boldsymbol{\Theta}_k)\|p(\boldsymbol{\Theta}_k|\boldsymbol{d}_{1:k})] = \int q(\boldsymbol{\Theta}_k)\log\frac{q(\boldsymbol{\Theta}_k)}{p(\boldsymbol{\Theta}_k|\boldsymbol{d}_{1:k})}d\boldsymbol{\Theta}_k \geqslant 0. \tag{29}$$

To obtain an analytical solution, the standard VB-approach applies the mean-field approximation [19] to factorize q in disjoint groups:

$$q(\boldsymbol{\Theta}_k) = q(\boldsymbol{w}_k)q(\boldsymbol{P}_{k|k-1}), \tag{30}$$

where the two separated functions have the following distributions:

$$q(\boldsymbol{w}_k) = \mathcal{N}(\boldsymbol{w}_k; \hat{\boldsymbol{w}}_{k|k}, \boldsymbol{P}_{k|k}), \tag{31}$$

$$q(\boldsymbol{P}_{k|k-1}) = \mathcal{IW}(\boldsymbol{P}_{k|k-1}; v_{k|k}, \boldsymbol{W}_{k|k}). \tag{32}$$

Then, substitute (30) into (29) to minimize KL divergence, we get the optimal solution of each partition:

$$\ln q^*(\boldsymbol{w}_k) = \mathrm{E}_{\boldsymbol{P}_{k|k-1}}\left\{\ln p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}, \boldsymbol{d}_{1:k})\right\} + \text{const.}, \tag{33}$$

$$\ln q^*(\boldsymbol{P}_{k|k-1}) = \mathrm{E}_{\boldsymbol{w}_k}\left\{\ln p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}, \boldsymbol{d}_{1:k})\right\} + \text{const.}, \tag{34}$$

where the joint distribution $p(\boldsymbol{w}_k, \boldsymbol{P}_{k|k-1}, \boldsymbol{d}_{1:k}) = p(\boldsymbol{d}_k|\boldsymbol{w}_k)$ $p(\boldsymbol{w}_k|\boldsymbol{d}_{1:k-1}, \boldsymbol{P}_{k|k-1})p(\boldsymbol{P}_{k|k-1}|\boldsymbol{d}_{1:k-1})p(\boldsymbol{d}_{1:k-1})$. By calculating (33) and (34), the parameters in (31) and (32) are updated iteratively in an inner VB-iteration l as

$$v_{k|k}^l = v_{k|k-1} + 1, \tag{35}$$

$$\boldsymbol{W}_{k|k}^l = \boldsymbol{W}_{k|k-1} + \boldsymbol{P}_{k|k}^{l-1} + (\hat{\boldsymbol{w}}_{k|k}^{l-1} - \hat{\boldsymbol{w}}_{k|k-1})\times (\hat{\boldsymbol{w}}_{k|k}^{l-1} - \hat{\boldsymbol{w}}_{k|k-1})^{\mathrm{T}}, \tag{36}$$

$$\boldsymbol{P}_{k|k-1}^l = \boldsymbol{W}_{k|k}^l/(v_{k|k}^l - N_d - 1), \tag{37}$$

$$\boldsymbol{P}_{k|k}^l = \left((\boldsymbol{P}_{k|k-1}^l)^{-1} + \boldsymbol{K}^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{K}\right)^{-1}, \tag{38}$$

$$\hat{\boldsymbol{w}}_{k|k}^l = \boldsymbol{P}_{k|k}^l\left(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{d}_k + (\boldsymbol{P}_{k|k-1}^l)^{-1}\hat{\boldsymbol{w}}_{k|k-1}\right) \tag{39}$$

with initialization $\hat{\boldsymbol{w}}_{k|k}^0 = \hat{\boldsymbol{w}}_{k|k-1}$, $\boldsymbol{P}_{k|k}^0 = \boldsymbol{P}_{k|k-1}$ and (27) (28). After VB-iteration, the subsequent predict step follows (12) and (13). Due to limited space of this paper, readers can refer to [17] for details on derivation and parameter tuning.

To distinguish these two noise adaptive approaches, here we name the first one as noise adaptive KKF with $\boldsymbol{Q}$ estimation (NAKKF_Q) and the second one as variational Bayesian KKF (VBKKF). To process the VBKKF in a decentralized mode, only the updates (38) and (39) on parameters of state $\boldsymbol{w}_k$ require information from neighboring clusters following the procedure in section 4. The parameters $v_{k|k}$ and $\boldsymbol{W}_{k|k}$ of $\boldsymbol{P}_{k|k-1}$ can be updated locally.

6. SIMULATION RESULTS

To evaluate the performance of KKF for diffusion field estimation, we generate a $M = 2$ sources field with diffusion constant $\nu = 0.01$ based on (1). These two sources have intensities $b_1 = 1$, $b_2 = 0.7$ at positions $\boldsymbol{p}_1 = [0.3, 0.3]^{\mathrm{T}}$, $\boldsymbol{p}_2 = [0.8, 0.6]^{\mathrm{T}}$ activating at time $t_1 = 0$, $t_2 = 10$. This field is sampled at every $\Delta t = 0.2$ up to the total simulation time $T = 20$ and measured by a randomly deployed sensor network with $N_s = 80$ sensors in a unit square area separated into $N_c = 4$ clusters. The communication links among CHs are assumed to be ideal and fully connected. We set parameters $\boldsymbol{F}_k = \boldsymbol{I}_{N_d}$, $\sigma_n^2 = 0.01$ and a nominal $\tilde{\boldsymbol{Q}}_k = \sigma_q^2\boldsymbol{I}_{N_d}$ with $\sigma_q^2 = 10^{-4}$ in the state-space model. Here, a Gaussian kernel [2] $\kappa(\boldsymbol{x}_j, \boldsymbol{x}_i) = \exp\{-\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2/(2\zeta^2)\}$ is applied to match the form of system model (1) with adaptive kernel bandwidth $\zeta = \sqrt{2\nu k}$ [3]. The decentralized KKF and two noise adaptive algorithms are applied to track

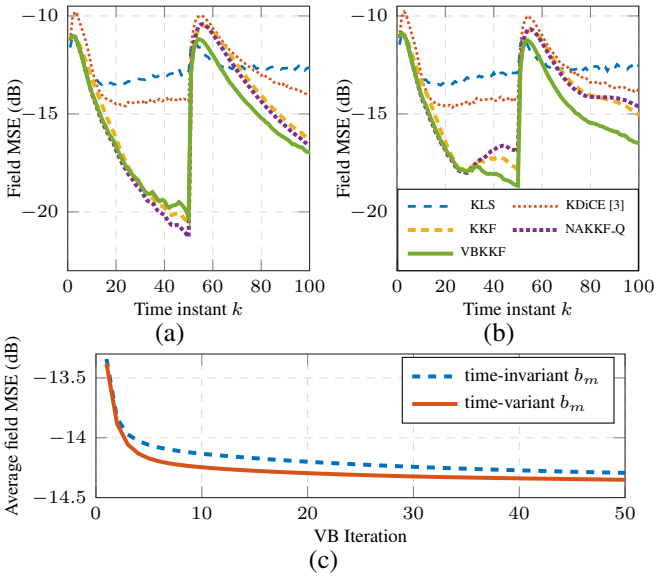

Fig. 2. (a) Field MSE vs time instant k with time-invariant intensities; (b) Field MSE vs time instant k with time-variant intensities; (c) Averaged filed MSE over time vs inner VB iteration for VBKKF.

this time-varying diffusion field over a hierarchical sensor network compared to the central KLS solution and another KDiCE algorithm [3] which is designed for distributed KLS regression. To reduce computation effort, a dictionary set $\mathcal{D}$ is selected by the coherence criterion [3, eq. (18)] with threshold $\tau = 0.8$. The initial mean and covariance of system state are $\hat{\boldsymbol{w}}_{0|0} = \boldsymbol{0}_{N_d}$, $\boldsymbol{P}_{0|0} = \boldsymbol{I}_{N_d}$. For VBKKF, we choose $\theta = 3$ and run 5 inner VB-iteration under each time instant k. For KDiCE, the step-size is set to be 1 and only one inner iteration is simulated per k. We calculate the mean square error (MSE) per k between the true and estimated field averaged over N_c clusters and N_g grid sample points in a unit square area:

$$\mathrm{MSE}_k = \frac{1}{N_c}\frac{1}{N_g}\sum_{c=1}^{N_c}\sum_{g=1}^{N_g}\|f(\boldsymbol{x}_g, t_k) - \hat{\boldsymbol{w}}_{c,k|k}^{\mathrm{T}}\boldsymbol{\kappa}(\boldsymbol{x}_g)\|^2, \tag{40}$$

which is further averaged over 200 Monte Carlo simulations with random realizations of $n_{j,k}$ and sensor positions.

Fig. 2(a) shows online tracking curves of different algorithms along with the increasing time instant k. Note that in simulations, KKF, NAKKF_Q and VBKKF are all decentralized algorithms. Due to the activation of the second source, there is a sudden change at $k = 50$. It is obvious that various KKF algorithms which estimate the field based on all data set up to the current time instant outperforms the KLS and KLS-based KDiCE algorithm. The NAKKF_Q has the best steady-state performance at the first stage but has a slower adaptation to the sudden change compared to VBKKF. Then, we change the field with time-varying intensities $b_m \times (1 - 0.2\sin(2\pi 10^{-1.2}k))$ to observe the tracking capabilities of different algorithms for this continuous changing. The results

are shown in Fig. 2(b) indicating that VBKKF adapts to the unknown process noise well in this case and achieves the best tracking performance. Fig. 2(c) illustrates that the averaged field MSE over time instant k is decreasing along with the increasing number of VB-iterations for VBKKF. The curves asymptotically converge to certain solutions. However, each solution is different from the true joint posterior distribution in (24), i.e., KL divergence will not reach zero. The reason is variable $\boldsymbol{w}_k$ and $\boldsymbol{P}_{k|k-1}$ are dependent, but in VB-method they are assumed to be independent with the mean-field approximation (30). In the previous simulations, we choose VB-iteration to 5, which is a good choice confirmed in Fig. 2(c). This choice makes VBKKF achieve a good estimation result meanwhile keeping low computation effort. In other words, the tracking performance of VBKKF in Fig. 2(a)(b) can be further improved by increasing VB-iterations.

7. CONCLUSION

In this paper, a kernel-based Kalman filter is investigated to track time-varying diffusion fields over hierarchical sensor networks in a decentralized fashion. We further exploit a decentralized VBKKF to learn the distribution of system unknown variables. Simulations show that KKF outperforms the KLS-based approaches. Compared to NAKKF_Q, the learning-based VBKKF has the potential to track complicated diffusion fields, e.g., with continuously changing intensities.

8. REFERENCES

[1] S. N. Simić and S. Sastry, "Distributed environmental monitoring using random sensor networks," in *Information Processing in Sensor Networks*. Springer, 2003, pp. 582–592.

[2] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[3] B. S. Shin, H. Paul, and A. Dekorsy, "Distributed kernel least squares for nonlinear regression applied to sensor networks," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, Sep. 2016, pp. 1588–1592.

[4] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel adaptive filtering*, John Wiley & Sons, 2010.

[5] B. S. Shin, M. Yukawa, R. L. G. Cavalcante, and A. Dekorsy, "Distributed adaptive learning with multiple kernels in diffusion networks," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5505–5519, Nov. 2018.

[6] J. B. Rosinha, S. J. M. de Almeida, and J. C. M. Bermudez, "A new kernel Kalman filter algorithm for estimating time-varying nonlinear systems," in *2017 IEEE Int. Symposium on Circuits and Systems (ISCAS)*. IEEE, May. 2017, pp. 1–4.

[7] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. Robotics*, vol. 24, no. 3, pp. 710–724, Jun. 2008.

[8] M. J. Beal, *Variational algorithms for approximate Bayesian inference*, University of London, 2003.

[9] Y. M. Lu, P. L. Dragotti, and M. Vetterli, "Localizing point sources in diffusion fields from spatiotemporal samples," in *Proc. Int. Conf. on Sampling Theory and Applications (SampTA)*, Jun. 2011.

[10] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[12] Z. Chen, "Bayesian filtering: from Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[13] S. Wang, H. Paul, and A. Dekorsy, "Distributed optimal consensus-based Kalman filtering and its relation to MAP estimation," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2018, pp. 3664–3668.

[14] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[15] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Automat. Contr.*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[16] C. Paleologu, J. Benesty, and S. Ciochin, "Study of the general Kalman filter for echo cancellation," *IEEE Trans. Audio Speech Language Process.*, vol. 21, no. 8, pp. 1539–1549, Aug. 2013.

[17] 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. Automat. Contr.*, vol. 63, no. 2, pp. 594–601, Feb. 2018.

[18] J. Ma, H. Lan, Z. Wang, X. Wang, Q. Pan, and B. Moran, "Improved adaptive Kalman filter with unknown process noise covariance," in *2018 21st Int. Conf. on Information Fusion (FUSION)*, Jul. 2018, pp. 1–5.

[19] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.