IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 12, 2021, accepted January 14, 2021, date of publication January 22, 2021, date of current version February 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053651

Joint Learning of Model Parameters and Coefficients for Online Nonlinear Estimation

MASA-AKI TAKIZAWA, (Student Member, IEEE), AND MASAHIRO YUKAWA, (Senior Member, IEEE)
Department of Electronics and Electrical Engineering, Keio University, Kanagawa 223-8522, Japan

Corresponding author: Masa-Aki Takizawa (takizawa@ykw.elec.keio.ac.jp)

This work was supported by the Grants-in-Aid for Scientific Research (KAKENHI) under Grant 18J21595 and Grant 18H01446.

ABSTRACT We propose a novel online algorithm for efficient nonlinear estimation. Target nonlinear functions are approximated with "unfixed" Gaussians of which the parameters are regarded as (a part of) variables. The Gaussian parameters (scales and centers), as well as the coefficients, are updated to suppress the instantaneous squared errors regularized by the ℓ_1 norm of the coefficients to enhance the model efficiency. Another point for enhancing the model efficiency is the multiscale screening method, which is a hierarchical dictionary growing scheme to initialize Gaussian scales with multiple choices. To reduce the computational complexity, a certain selection strategy is presented for growing the dictionary and updating the Gaussian parameters. Computer experiments show that the proposed algorithm enjoys high adaptation-capability and produces efficient estimates.

INDEX TERMS Nonlinear estimation, online learning, model parameter tuning, Gaussian function, sparse regularization.

I. INTRODUCTION

Many problems in signal processing and machine learning can be cast as online estimation of unknown "nonlinear" functions to which the simple linear model hardly fits. Finding an appropriate nonlinear model well-fitting the target unknown function has been a long-standing challenge of statistical inference. The primal goal of this article is to devise an online algorithm that finds an efficient model (and the coefficients simultaneously) being able to express the nonlinear function accurately with reasonably short expansion length. The efficient model would yield a number of practical benefits such as avoidance of over-fitting, reduction of computational complexity, saving of memory storage, improvements of convergence behavior, as well as disclosure of the latent dimension (interpretability of the resultant estimate).

As studied in the long history of online nonlinear estimation, there are many possible choices of nonlinear models and its learning algorithms. Extended and unscented Kalman filters [2], [3] are dominant choices when the target system has a state-space formulation, and parameter estimations for the state-space models have been studied in a variety of industrial applications, including the control of motor activity [4], [5], state estimation of power systems [6], [7], state-of-charge estimation for lithium-ion batteries [8], and control and estimation in vehicle systems [9], [10]. Volterra filter [11] is applicable to nonlinear functions, and it has widely been used in acoustics applications. The order of the Volterra series expansion is, however, limited typically to two (or three at most) due to the increase of computational loads, which means that the Volterra filter has limited capabilities to capture the nonlinearity of the target. The Gaussian model has widely been adopted such as in Gaussian process [12], radial basis function (RBF) network [13], spline interpolation [14], support vector regression [15], and kernel adaptive filtering [16]. Although the Gaussian model enjoys desirable properties for nonlinear estimation such as universal approximation property and smoothness [17], its performance heavily depends on the choice of the model parameters (the scales and centers of Gaussian functions). Specifically, undesirably small and large Gaussian scales cause the problems of overfitting and underfitting, respectively, which result in serious performance degradations for estimation process.

In the batch setting, a number of approaches to select the model parameters have been proposed in the contexts of kernel density estimation and kernel regression [18]–[23]. One may consider to apply those batch methods to online applications by selecting the model parameters with some training data. However, this approach is inefficient in real world applications as the following situations often happen:

The associate editor coordinating the review of this manuscript and approving it for publication was Di Zhang.

(i) the training data have different statistical properties from the test data (such as the case of covariate shift and/or colored signals), and (ii) the target function (and also its frequency components) changes over time. It is therefore of great importance to develop an online learning algorithm that adapts the model parameters as well as the coefficients so that the model becomes more efficient (i.e., the redundancy decreases) and, at the same time, the errors diminish as time goes by.

In the field of kernel adaptive filtering [24]–[35] and RBF network [36]–[38], a commonly used idea is selecting the centers of Gaussians from the input $\boldsymbol{u}_n$. During the last decade, online selecting and learning methods for Gaussian parameters have been studied. Center-selection schemes have been discussed in terms of novelty criteria to pick up only the necessary data from the input samples [25], [38]–[40]. An adaptive dictionary-refinement technique based on the proximity operator of a weighted (block) ℓ_1 norm has been proposed in [31], [41], [42]. The multikernel adaptive filtering [30]–[35], [43] has been proposed as a convex analytic approach with multiple different scales. The concept of online model selection and learning has been presented in [44], [45] based on the multikernel adaptive filtering framework, selecting appropriate scales from a hundred of possible scales by shrinking the coefficient vector for each scale while learning those parameters as well as for reducing the estimation errors simultaneously. Although those selection schemes for the Gaussian parameters yield reasonably good results, there is still sufficient room for improvements in the sense of ‘‘efficiency’’ of the estimate. In the kernel adaptive filtering context, moreover, some methods have been proposed to adapt the kernel scales [46]–[48] and centers [49], [50] in the dictionary. The method proposed in [47] uses a common scale parameter for all kernel functions. The method in [48], [51] updates both scales and centers individually, as in the way of the proposed approach.

In this paper, we propose an efficient adaptive method updating the Gaussian parameters (scales and centers) and the coefficients alternately to reduce the instantaneous squared errors. To enhance the model efficiency, we apply the ℓ_1 norm regularization to the cost function, which is applied in a variety of fields [52]–[55]. Specifically, the instantaneous cost is penalized by the weighted ℓ_1 norm of the coefficient vector, which leads to dictionary sparsification. The key difference between the proposed algorithm and the methods in [46]–[50] is a novel online dictionary growing technique, which builds a dictionary with multiple initial scales selected by a hierarchical selection strategy. The proposed dictionary growing technique is motivated by the fact that the performance of the aforementioned alternating update approach depends highly on the initial scales [56]. Specifically, the initial Gaussian scales affect the efficiency and the accuracy of the estimate significantly when the selected scale was far from the actual ones of the target function due to the ‘‘gradient vanishment’’ (see Section III-A).

The major properties of the proposed algorithm are summarized below.

- Multiple initial values for the Gaussian scales are employed to alleviate the sensitivity to the initial conditions. It is expected here that at least some of the initial scales are relevant to the estimation task. The use of multiple initial values, however, may cause undesirable growths of the dictionary size. To avoid it, we present an efficient dictionary growing strategy named *multiscale screening method*, which conducts an error test followed by multiple levels of novelty test for each input vector with coarse to fine ‘screens’ which correspond to large-to small- scale Gaussians.
- The computational complexity tends to be reasonably low thanks to a certain selection strategy for dictionary growing and scale/center updating.
- As revealed by computer experiments, the proposed algorithm enjoys high adaptation-capability while maintaining a small dictionary size compared with the single initialization case. The experiments are carried out in the context of online time-series data prediction with synthetic/real dataset. The proposed algorithm is compared with the state-of-the-art algorithms developed for (i) kernel adaptive filtering and (ii) online time-series prediction based on long short-term memory (LSTM) neural networks.

The rest of this paper is organized as follows. In Section II, the problem settings, model, and cost function are presented. In Section III, the proposed algorithm is presented, consisting of the dictionary growing step (Section III-A) and the parameter updating step (Section III-B). In Section IV, some discussions about the proposed algorithm are presented: the monotone decreasing property of the cost function, design schemes for parameters, a selection scheme for the initial Gaussian scales, and computational complexities. In Section V, computer experiments show the efficacy of the proposed algorithm, followed by conclusion in Section VI.

II. PROBLEM SETTING, NONLINEAR MODEL, AND COST FUNCTION

Let $\mathbb{R}$, $\mathbb{R}_{++}$, and $\mathbb{N}$ be the sets of real numbers, strictly positive real numbers, and nonnegative integers, respectively. We denote by $\mathcal{U} \subset \mathbb{R}^L$ the input space in which the input vectors $\boldsymbol{u}_n$ arise, where $n \in \mathbb{N}$ is the time index. The online nonlinear estimation problem considered in the present study is the following: estimate an unknown nonlinear function $\psi : \mathcal{U} \to \mathbb{R}$ by means of sequentially arriving input $\boldsymbol{u}_n \in \mathcal{U}$ and its output $d_n := \psi(\boldsymbol{u}_n)+\nu_n \in \mathbb{R}$ contaminated by additive noise $\nu_n \in \mathbb{R}$. No prior knowledge is assumed available about the structure of ψ and the input signals; i.e., none of the adequate number of Gaussians, the range of Gaussian centers/scales, and the input range is known prior to estimation.

Define the Gaussian function

$$g(\boldsymbol{u}; \xi, \boldsymbol{c}) := \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{c}\|^2}{2\xi}\right) \tag{1}$$

with the scale (variance) parameter $\xi > 0$ and the center (mean) vector $\boldsymbol{c} \in \mathbb{R}^L$, where, $\|\cdot\|$ denotes the Euclidean

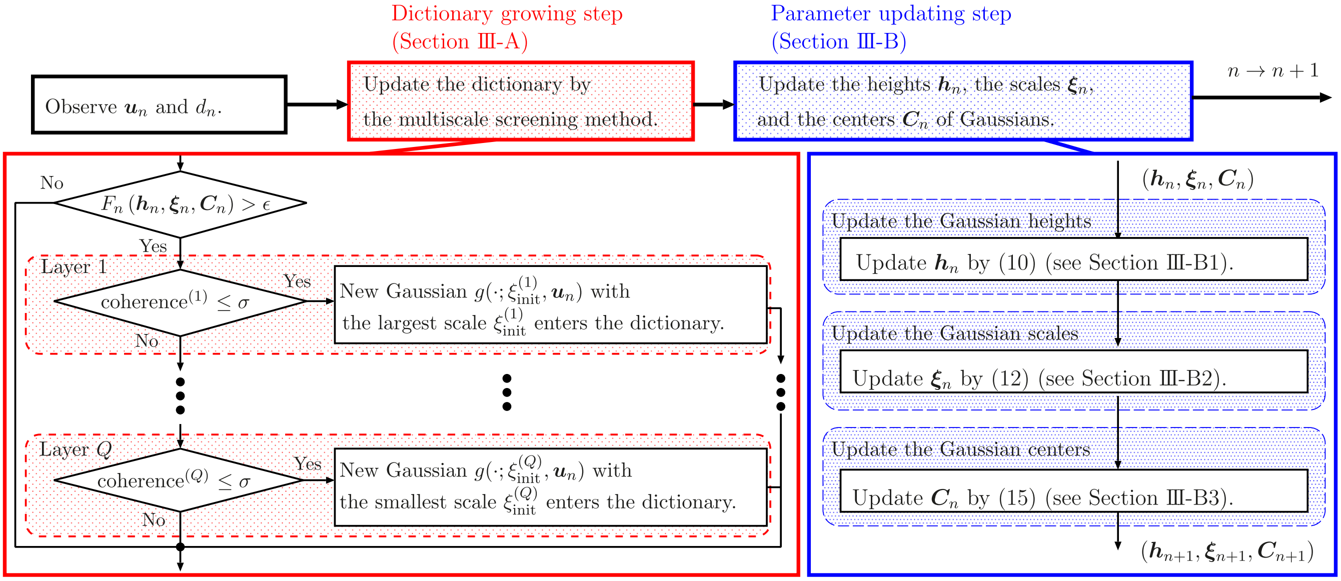

Dictionary growing step
(Section III-A)
Parameter updating step
(Section III-B)
Observe $\boldsymbol{u}_n$ and d_n.
Update the dictionary by the multiscale screening method.
Update the heights $\boldsymbol{h}_n$, the scales $\boldsymbol{\xi}_n$, and the centers $\boldsymbol{C}_n$ of Gaussians.
$n \to n+1$
No
$F_n(\boldsymbol{h}_n, \boldsymbol{\xi}_n, \boldsymbol{C}_n) > \epsilon$
Yes
Layer 1
coherence$^{(1)} \leq \sigma$
Yes
New Gaussian $g(\cdot; \xi_{\text{init}}^{(1)}, \boldsymbol{u}_n)$ with the largest scale $\xi_{\text{init}}^{(1)}$ enters the dictionary.
No
Layer Q
coherence$^{(Q)} \leq \sigma$
Yes
New Gaussian $g(\cdot; \xi_{\text{init}}^{(Q)}, \boldsymbol{u}_n)$ with the smallest scale $\xi_{\text{init}}^{(Q)}$ enters the dictionary.
No
$(\boldsymbol{h}_n, \boldsymbol{\xi}_n, \boldsymbol{C}_n)$
Update the Gaussian heights
Update $\boldsymbol{h}_n$ by (10) (see Section III-B1).
Update the Gaussian scales
Update $\boldsymbol{\xi}_n$ by (12) (see Section III-B2).
Update the Gaussian centers
Update $\boldsymbol{C}_n$ by (15) (see Section III-B3).
$(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}_{n+1}, \boldsymbol{C}_{n+1})$

FIGURE 1. A flow chart of the proposed algorithm. At each time instant *n*, the proposed algorithm updates the estimate in two steps.

norm. Our time-varying model is then given as

$$\varphi_n(\boldsymbol{u}) := \sum_{j=1}^{r_n} h_n^{(j)} g(\boldsymbol{u}; \xi_n^{(j)}, \boldsymbol{c}_n^{(j)}), \ \boldsymbol{u} \in \mathcal{U}, \tag{2}$$

with the height $h_n^{(j)} \in \mathbb{R}$, scale $\xi_n^{(j)} > 0$, and center $\boldsymbol{c}_n^{(j)} \in \mathbb{R}^L$ of the jth Gaussian. In this study, the scale $\xi_n^{(j)}$ and the center $\boldsymbol{c}_n^{(j)}$ of each Gaussian (atom) in the dictionary $\{g(\cdot; \xi_n^{(j)}, \boldsymbol{c}_n^{(j)})\}_{j=1}^{r_n}$, $n \in \mathbb{N}$, are regarded as *variables*. Namely, those parameters are updated iteratively so that our estimate φ_n becomes an efficient approximation of the target nonlinear function ψ.

At time instant n, the variables (heights, scales, and centers of r_n Gaussians) can be expressed respectively as $\boldsymbol{h} := [h^{(1)}, h^{(2)}, \cdots, h^{(r_n)}]^\mathsf{T} \in \mathbb{R}^{r_n}$, $\boldsymbol{\xi} := [\xi^{(1)}, \xi^{(2)}, \cdots, \xi^{(r_n)}]^\mathsf{T} \in \mathbb{R}_{++}^{r_n}$, and $\boldsymbol{C} := [\boldsymbol{c}^{(1)} \ \boldsymbol{c}^{(2)} \ \cdots \ \boldsymbol{c}^{(r_n)}] \in \mathbb{R}^{L \times r_n}$. Here, the superscript $[]^\mathsf{T}$ stands for transpose of vector/matrix. The instantaneous cost function is then given by

$$J_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C}) := F_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C}) + \lambda \Omega_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C}), \tag{3}$$

where $\lambda > 0$ is the regularization parameter, and

$$F_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C}) := \frac{1}{2}(d_n - \varphi_n(\boldsymbol{u}_n))^2 \tag{4}$$

$$\Omega_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C}) := \sum_{j=1}^{r_n} \omega_n^{(j)} \left| h^{(j)} \right|, \tag{5}$$

for some positive weights $\omega_n^{(j)} > 0$. Here, $|\cdot|$ denotes the absolute value of a real number. A simple weight design is given by the weight $\omega_n^{(j)} := \frac{1}{\left|h_n^{(j)}\right| + \beta}$ for some small constant $\beta > 0$ which has been shown to promote sparsity of the coefficient vector without causing serious performance degradations [42], [57]; see also Section V for the efficiency of this weight design. The weighted ℓ_1 norm serves to discard those redundant/obsolete Gaussians that make no contribution to the estimation, yielding parsimonious estimates without causing serious performance degradations [27], [31], [41], [42], [58], [59]. Indeed, it has shown in [41] that obsolete Gaussians remaining in the dictionary may give negative impacts on the performance, and the weighted-ℓ_1 regularization mitigates such negative impacts. See [27], [31], [41], [42] for more details about the dictionary refinement techniques.

III. PROPOSED ALGORITHM

The proposed algorithm consists of two steps: (i) the dictionary growing step and (ii) the parameter updating step, where the latter includes the dictionary pruning process. The flowchart is given in Figure 1. In the first step, the dictionary is initialized to an empty set ($r_0 := 0$), and it grows under a hierarchical selection strategy using the sequentially coming data. In the second step, the variables $\boldsymbol{h}_n$, $\boldsymbol{\xi}_n$, and $\boldsymbol{C}_n$ are updated in sequence. Each step will be detailed below.

A. DICTIONARY GROWING STRATEGY UNDER MULTISCALE SCREENING

We pay our attention to the following fact: $F_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C})$ is nonconvex as a function of each scale parameter $\xi^{(j)}$, and it has shallow slopes at those points which are far from the optimal point (see [56]). This means that the gradient vanishes if the initial scale is undesirably large or smaller compared to the optimal one. In such a case, the learning speeds of the poorly-initialized $\xi^{(j)}$ become unacceptably slow, and this may cause a serious deterioration of the whole estimation process. In our preliminary experiments, the use of $\xi_{init}^{(1)}$ which is hundred times larger/smaller than an adequate scale caused slow convergence. We thus employ multiple initial values for the Gaussian scales so that at least some of the initial scales are suitable for the data. To avoid undesirable growths of the dictionary size due to the use of multiple initial scales, each input vector is tested from coarse to fine 'screens' corresponding to large- to small- scale Gaussians. This efficient dictionary growing strategy is named *multiscale screening method*. The multiscale screening method consists of the following two sub-steps: (i) the error test and (ii) the novelty test.

The error test is rather simple. When the estimation error is sufficiently small, the current estimate is good enough already for the current input $\boldsymbol{u}_n$ and therefore there is no need to add the new Gaussian (centered at the current input $\boldsymbol{u}_n$) into the dictionary, as there remains little space for improvements in estimation accuracy and such a redundant Gaussian function may even give negative impacts on the performance as mentioned in the previous section. The error condition is thus given as follows:

$$F_n\left(\boldsymbol{h}_n, \boldsymbol{\xi}_n, \boldsymbol{C}_n\right) > \epsilon, \tag{6}$$

where $\epsilon \geq 0$ is the threshold. Here, $\boldsymbol{h}_n := [h_n^{(1)}, h_n^{(2)}, \cdots, h_n^{(r_n)}]^\mathsf{T}$, $\boldsymbol{\xi}_n := [\xi_n^{(1)}, \xi_n^{(2)}, \cdots, \xi_n^{(r_n)}]^\mathsf{T}$, and $\boldsymbol{C}_n := [\boldsymbol{c}_n^{(1)}\ \boldsymbol{c}_n^{(2)}\ \cdots\ \boldsymbol{c}_n^{(r_n)}]$. If the error condition is satisfied, the novelty test is conducted to select a Gaussian function with an adequate scale parameter; otherwise, the dictionary does not grow at this time instant.

The novelty test is performed hierarchically based on the multiscale screening to select an adequate Gaussian scale. The multiscale screening aims to enhance the model efficiency. The global structures (the low frequency components) of the nonlinear function ψ can be captured efficiently by relatively large scale Gaussian functions, while the local structures (the fine parts) of ψ can be captured efficiently by Gaussian functions of appropriately small scales. The central philosophy of the multiscale screening is the following: (i) extract the global structures at the initial phase of estimation and (ii) extract the local structures (the estimation residual after removing the global structures) gradually once the dictionary for the global ones is well developed (for more discussions about the global-to-local order of the multiscale screening, see Section IV-B).

We now explain how to choose the initial scale at each iteration. A wide range of scales $\xi_{init}^{(1)} > \xi_{init}^{(2)} > \cdots > \xi_{init}^{(Q)} > 0$ are usually adopted. At time instant $n := 0$, the dictionary is empty, and the largest scale $\xi_{init}^{(1)}$ to extract the global structure is selected automatically without any novelty test, which means that the function $g(\cdot; \xi_{init}^{(1)}, \boldsymbol{u}_0)$ enters the dictionary. From the second iteration, the novelty test is conducted. At time instant $n \geq 2$, the similarity between $g(\cdot; \xi_{init}^{(1)}, \boldsymbol{u}_n)$ and (a selected subset of) the current dictionary is evaluated. (Indeed, the similarity is evaluated only with a subset of the dictionary selected under some criterion as explained later on for reducing the computational costs of the novelty test.) If the similarity is sufficiently low, $g(\cdot; \xi_{init}^{(1)}, \boldsymbol{u}_n)$ is regarded novel and it enters the dictionary. If, and only if, the similarity is high, it is regarded redundant and the second Gaussian $g(\cdot; \xi_{init}^{(2)}, \boldsymbol{u}_n)$ is tested in the same way. If the similarity is sufficiently low, $g(\cdot; \xi_{init}^{(2)}, \boldsymbol{u}_n)$ enters the dictionary, and, if (and only if) the similarity is high, it is regarded redundant and the third one is tested. This continues until some Gaussian is regarded novel; if all the Gaussian functions are regarded redundant, the dictionary does not grow at that time instant. Suppose that some $g(\cdot; \xi_{init}^{(q)}, \boldsymbol{u}_n)$ enters the dictionary. Then, the sizes of the variable vectors and matrix are augmented: the augmented vectors and matrices are given by $\hat{\boldsymbol{h}}_n := [\boldsymbol{h}_n^\mathsf{T}\ 0]^\mathsf{T} \in \mathbb{R}^{r_n+1}$, $\hat{\boldsymbol{\xi}}_n := [\boldsymbol{\xi}_n^\mathsf{T}\ \xi_{init}^{(q)}]^\mathsf{T} \in \mathbb{R}^{r_n+1}$, and $\hat{\boldsymbol{C}}_n := [\boldsymbol{C}_n\ \boldsymbol{u}_n] \in \mathbb{R}^{L\times(r_n+1)}$, respectively. Suppose in contrast that no Gaussian enters the dictionary. In this case, we let $\hat{\boldsymbol{h}}_n := \boldsymbol{h}_n$, $\hat{\boldsymbol{\xi}}_n := \boldsymbol{\xi}_n$, and $\hat{\boldsymbol{C}}_n := \boldsymbol{C}_n$.

Now, we present the selection strategy and the novelty criterion (the similarity measure). For computational efficiency, our strategy is the following: select a subset $\{g(\cdot; \xi_n^{(j)}, \boldsymbol{c}_n^{(j)})\}_{j\in\mathcal{J}_n} \subset \{g(\cdot; \xi_n^{(j)}, \boldsymbol{c}_n^{(j)})\}_{j=1}^{r_n}$ of the Gaussian functions in the dictionary that return the largest values at the current input $\boldsymbol{u}_n$ (see Figure 2). Here, $\mathcal{J}_n := \{j_1, \cdots, j_{s_n^{(NC)}}\}$ with its cardinality $|\mathcal{J}_n| = s_n^{(NC)}$ denotes the index set of the selected Gaussians. More specifically, we let $\{j_1, j_2, \cdots, j_{r_n}\} = \{1, 2, \cdots, r_n\}$ such that

$$g(\boldsymbol{u}_n; \xi_n^{(j_i)}, \boldsymbol{c}_n^{(j_i)}) \geq g(\boldsymbol{u}_n; \xi_n^{(j_k)}, \boldsymbol{c}_n^{(j_k)}), \quad 1 \leq i < k \leq r_n. \tag{7}$$

This selection strategy is computationally efficient and is expected to include such a dictionary atom that maximizes our novelty criterion of L^2 coherence (see [40] for the detail of the coherence criterion)

$$\begin{aligned} &c(\xi^{(u)}, \boldsymbol{u}, \xi^{(v)}, \boldsymbol{v}) \\ &\quad := \left| \frac{\left\langle g(\cdot; \xi^{(u)}, \boldsymbol{u}), g(\cdot; \xi^{(v)}, \boldsymbol{v}) \right\rangle_{L^2}}{\left\| g(\cdot; \xi^{(u)}, \boldsymbol{u}) \right\|_{L^2} \left\| g(\cdot; \xi^{(v)}, \boldsymbol{v}) \right\|_{L^2}} \right| \\ &\quad = \left(\frac{4\xi^{(u)}\xi^{(v)}}{(\xi^{(u)}+\xi^{(v)})^2} \right)^{L/4} \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2(\xi^{(u)}+\xi^{(v)})} \right) \in (0, 1], \end{aligned} \tag{8}$$

where $\langle \cdot, \cdot \rangle_{L^2}$ and $\|\cdot\|_{L^2}$ are the inner product and norm of the L^2 space (i.e., the space of square integrable functions). See Appendix A for the derivation of (8). Under the selection strategy (7) and the L^2-coherence criterion (8), the novelty test for the Gaussian $g(\cdot; \xi_{init}^{(q)}, \boldsymbol{u}_n)$ is given as follows:

$$\max_{j\in\mathcal{J}_n} \left| c(\xi_n^{(j)}, \boldsymbol{c}_n^{(j)}, \xi_{init}^{(q)}, \boldsymbol{u}_n) \right| \leq \sigma, \tag{9}$$

where $\sigma \in [0, 1]$ is the prespecified threshold. If the condition in (9) is satisfied, the similarity between $g(\cdot; \xi_{init}^{(q)}, \boldsymbol{u}_n)$ and the existing dictionary atoms is sufficiently low and therefore $g(\cdot; \xi_{init}^{(q)}, \boldsymbol{u}_n)$ is regarded to be novel. The complexity issue will be discussed in Section IV-D.

The error and novelty tests share its underlying philosophy with Platt's criteria which checks the estimation error and the Euclidean distance between the current input vector and its closest center of Gaussian in the dictionary.

B. UPDATES OF HEIGHTS, SCALES, AND CENTERS OF GAUSSIAN

The heights $\boldsymbol{h}_n$, scales $\boldsymbol{\xi}_n$, and centers $\boldsymbol{C}_n$ of the Gaussians are updated in a sequence. To reduce the computational costs, the selection strategy (7) presented in Section III-A is applied

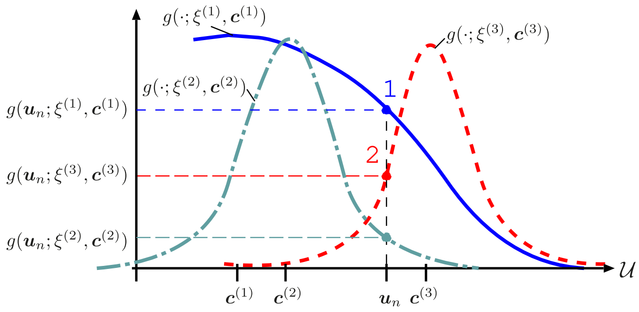

$g(\cdot;\xi^{(1)},\boldsymbol{c}^{(1)})$
$g(\cdot;\xi^{(3)},\boldsymbol{c}^{(3)})$
$g(\cdot;\xi^{(2)},\boldsymbol{c}^{(2)})$
$g(\boldsymbol{u}_n;\xi^{(1)},\boldsymbol{c}^{(1)})$
1
2
$g(\boldsymbol{u}_n;\xi^{(3)},\boldsymbol{c}^{(3)})$
$g(\boldsymbol{u}_n;\xi^{(2)},\boldsymbol{c}^{(2)})$
$\mathcal{U}$
$\boldsymbol{c}^{(1)}$
$\boldsymbol{c}^{(2)}$
$\boldsymbol{u}_n$
$\boldsymbol{c}^{(3)}$

FIGURE 2. The selection strategy for $r_n = 3$ (three Gaussians) and $s_n^{(NC)} = 2$. The numbers 1 and 2 denote the priority. In this illustration, $g(\cdot;\xi^{(1)},\boldsymbol{c}^{(1)})$ and $g(\cdot;\xi^{(3)},\boldsymbol{c}^{(3)})$ are selected. The unselected Gaussian $g(\cdot;\xi^{(2)},\boldsymbol{c}^{(2)})$ is not tested for the sake of computational efficiency.

to the updates of the scales and centers. We denote by $s_n^{(\xi)}$ and $s_n^{(c)}$ the sizes of the selected subsets for $\boldsymbol{\xi}_n$ and $\boldsymbol{C}_n$, respectively.

1) UPDATE OF THE HEIGHTS WITH DICTIONARY PRUNING

We employ the adaptive proximal forward backward splitting (APFBS) algorithm [60] for the cost function in (3) which is a superposition of the smooth fidelity function F_n and the nonsmooth regularizer $\lambda\Omega_n$. Although F_n and Ω_n are defined for the variable vectors and matrix compatible with the dictionary of size r_n, we keep the same notation to denote the same functions for the resized variables in accordance with the dictionary growing and pruning.

Step 1 (Height update including dictionary pruning): Let $\mu_h > 0$ be the stepsize parameter.

0. Switch $\boldsymbol{h}_n$, $\boldsymbol{\xi}_n$, and $\boldsymbol{C}_n$ to the possibly augmented counterparts $\hat{\boldsymbol{h}}_n \in \mathbb{R}^{\hat{r}_n}$, $\hat{\boldsymbol{\xi}}_n \in \mathbb{R}^{\hat{r}_n}_{++}$, and $\hat{\boldsymbol{C}}_n \in \mathbb{R}^{L\times\hat{r}_n}$, where $\hat{r}_n$ is either r_n+1 or r_n depending on whether the dictionary grows or not (see Section III-A).
1. Update the Gaussian coefficients by

$$\boldsymbol{h}_{n+1} := T_d\left[prox_{\mu_h\lambda\Omega_n}\left(\hat{\boldsymbol{h}}_n - \mu_h \frac{\partial F_n\left(\hat{\boldsymbol{h}}_n, \hat{\boldsymbol{\xi}}_n, \hat{\boldsymbol{C}}_n\right)}{\partial \hat{\boldsymbol{h}}} \right)\right] \in \mathbb{R}^{r_{n+1}}, \tag{10}$$

where

- $prox_{\mu_h\lambda\Omega_n}(\boldsymbol{h}) := \operatorname{argmin}_{\boldsymbol{x}}\left(\lambda\Omega_n(\boldsymbol{x}) + \frac{1}{2\mu_h}\|\boldsymbol{h}-\boldsymbol{x}\|^2\right)$ is the proximity operator of which the jth output can be computed as $[prox_{\mu_h\lambda\Omega_n}(\boldsymbol{h})]_j = \max\left\{\left|h^{(j)}\right| - \mu_h\lambda\omega^{(j)}, 0\right\} sign(h^{(j)})$, and
- $T_d : \mathbb{R}^{\hat{r}_n} \to \mathbb{R}^{r_{n+1}}$ resizes the argument vector, say $\hat{\boldsymbol{h}} := [\hat{h}^{(1)}, \hat{h}^{(2)}, \cdots, \hat{h}^{(\hat{r}_n)}]^{\mathsf{T}} \in \mathbb{R}^{\hat{r}_n}$, to its support size by discarding the zero components, i.e., $T_d(\hat{\boldsymbol{h}}) := (\hat{h}^{(j)})_{j\in supp(\hat{\boldsymbol{h}})}$, where $supp(\hat{\boldsymbol{h}}) := \{j \in \{1, 2, \cdots, \hat{r}_n\} \mid \hat{h}^{(j)} \neq 0\}$.

The dictionary is resized accordingly by discarding those Gaussian functions associated with the zero components.

The partial differential in (10) is given by

$$\frac{\partial F_n\left(\hat{\boldsymbol{h}}_n, \hat{\boldsymbol{\xi}}_n, \hat{\boldsymbol{C}}_n\right)}{\partial \hat{\boldsymbol{h}}} = -e_n\left(\hat{\boldsymbol{h}}_n, \hat{\boldsymbol{\xi}}_n, \hat{\boldsymbol{C}}_n\right)\boldsymbol{g}_n, \tag{11}$$

where $\boldsymbol{g}_n := [g(\boldsymbol{u}_n; \hat{\xi}_n^{(1)}, \hat{\boldsymbol{c}}_n^{(1)}), \cdots, g(\boldsymbol{u}_n; \hat{\xi}_n^{(\hat{r}_n)}, \hat{\boldsymbol{c}}_n^{(\hat{r}_n)})]^{\mathsf{T}} \in \mathbb{R}^{\hat{r}_n}$ and $e_n\left(\hat{\boldsymbol{h}}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{C}}\right) := d_n - \sum_{j=1}^{\hat{r}_n}\hat{h}^{(j)} g(\boldsymbol{u}_n; \hat{\xi}^{(j)}, \hat{\boldsymbol{c}}^{(j)})$, where $\hat{h}^{(j)}$, $\hat{\xi}^{(j)}$, and $\hat{\boldsymbol{c}}^{(j)}$ are the jth components and column of $\hat{\boldsymbol{h}}$, $\hat{\boldsymbol{\xi}}$, and $\hat{\boldsymbol{C}}$, respectively. The same notation e_n will be used to denote the same instantaneous error function for the resized variables compatible with the size-r_{n+1} dictionary.

2) UPDATE OF THE SCALES

To update the scale parameters over the set $\mathbb{R}_{++}$, we derive the multiplicative gradient update for the cost function in (3).

Step 2 (Scale update): Let $\mu_\xi^{(j)} > 0$ be the stepsize parameter.

0. Switch $\hat{\boldsymbol{\xi}}_n$ and $\hat{\boldsymbol{C}}_n$ again to its resized counterparts $\check{\boldsymbol{\xi}}_n \in \mathbb{R}^{r_{n+1}}_{++}$ and $\check{\boldsymbol{C}}_n \in \mathbb{R}^{L\times r_{n+1}}$, respectively, compatible with the downsized dictionary after pruning.
1. Select the index set $\{j_1, j_2, \cdots, j_{s_n^{(\xi)}}\}$ by (7) from the renewed dictionary $\{g(\cdot; \check{\xi}_n^{(j)}, \check{\boldsymbol{c}}_n^{(j)})\}_{j=1}^{r_{n+1}}$.
2. Update the scales of the selected Gaussians in a pseudo-code style as follows:[1]

```
for $i = 1 : s_n^{(\xi)}$
    $$\check{\xi}_n^{(j_i)} \leftarrow \check{\xi}_n^{(j_i)} \exp\left(-\mu_\xi^{(j_i)}\check{\xi}_n^{(j_i)} \frac{\partial F_n\left(\boldsymbol{h}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n\right)}{\partial \check{\xi}^{(j_i)}}\right) \tag{12}$$
end
```

Note that the updated scale $\check{\xi}_n^{(j_i)}$ will be used to evaluate the partial differential in (12) for updating its subsequent scales. The same applies to Step 3 (center update) in Section III-B3.

3. $\boldsymbol{\xi}_{n+1} \leftarrow \check{\boldsymbol{\xi}}_n$.

The partial differential in (12) is given by

$$\frac{\partial F_n\left(\boldsymbol{h}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n\right)}{\partial \check{\xi}^{(j)}} = -\frac{e_n\left(\boldsymbol{h}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n\right) h_{n+1}^{(j)} \left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2 g(\boldsymbol{u}_n; \check{\xi}_n^{(j)}, \check{\boldsymbol{c}}_n^{(j)})}{2(\check{\xi}_n^{(j)})^2}. \tag{13}$$

The multiplicative update (12) together with (13) is derived as follows. To ensure the strict positivity of $\check{\xi}^{(j)}$, we change the variable $\check{\xi}^{(j)}$ into $\check{\eta}^{(j)} := \log\check{\xi}^{(j)} \in \mathbb{R}$ which can take any real number. Then, the gradient update for the corresponding parameter $\check{\eta}_n^{(j)} := \log\check{\xi}_n^{(j)}$ is given, in a pseudo-code style, as

[1] The notation $a \leftarrow b$ in the pseudo code means "substitute b to a".

$$\begin{aligned}\check{\eta}_n^{(j)} &\leftarrow \check{\eta}_n^{(j)} - \mu_\xi^{(j)} \frac{\partial F_n\left(\boldsymbol{h}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n\right)}{\partial \check{\eta}^{(j)}} \\ &= \check{\eta}_n^{(j)} - \mu_\xi^{(j)} \check{\xi}_n^{(j)} \frac{\partial F_n\left(\boldsymbol{h}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n\right)}{\partial \check{\xi}^{(j)}}, \end{aligned} \tag{14}$$

where the equality is due to $\partial F_n/\partial\check{\eta}^{(j)} = (\partial F_n/\partial\check{\xi}^{(j)}) \times (\partial\check{\xi}^{(j)}/\partial\check{\eta}(j)) = \check{\xi}^{(j)}\partial F_n/\partial\check{\xi}^{(j)}$. Operating the inverse map $\exp(\cdot)$ of the logarithmic function to the both sides of (14) yields (12).

3) UPDATE OF $\check{\boldsymbol{C}}_n$

For the Gaussian centers $\check{\boldsymbol{c}}_n^{(j)}$, we employ the standard gradient descent update.

Step 3 (Center update): Let $\mu_c^{(j)} > 0$ be the stepsize parameter.

1. Select the index $\{j_1, j_2, \cdots, j_{s_n^{(c)}}\}$ by (7) from the renewed dictionary $\{g(\cdot; \xi_{n+1}^{(j)}, \check{\boldsymbol{c}}_n^{(j)})\}_{j=1}^{r_{n+1}}$.
2. Update the centers of the selected Gaussians as follows: for $i = 1 : s_n^{(c)}$

$$\check{\boldsymbol{c}}_n^{(j_i)} \leftarrow \check{\boldsymbol{c}}_n^{(j_i)} - \mu_c^{(j_i)} \frac{\partial F_n\left(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}_{n+1}, \check{\boldsymbol{C}}_n\right)}{\partial \check{\boldsymbol{c}}^{(j_i)}} \tag{15}$$

   end
3. $\boldsymbol{C}_{n+1} \leftarrow \check{\boldsymbol{C}}_n,\ j = 1, \cdots, r_{n+1}$.

The partial differential in (15) is given by

$$\frac{\partial F_n\left(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}_{n+1}, \check{\boldsymbol{C}}_n\right)}{\partial \check{\boldsymbol{c}}^{(j)}} = -\frac{e_n\left(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}_{n+1}, \check{\boldsymbol{C}}_n\right) h_{n+1}^{(j)} g(\boldsymbol{u}_n; \xi_{n+1}^{(j)}, \check{\boldsymbol{c}}_n^{(j)})(\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)})}{\xi_{n+1}^{(j)}}. \tag{16}$$

Remark 1: The selective update strategy of the proposed algorithm is related to the set-membership approach [61], [62], which updates the estimate only when the current error is sufficiently large for reducing the computational complexity of the parameter update. Some kernel adaptive filtering algorithms based on the set-membership approach have been proposed [63]–[65]. Although the proposed selection strategy shares the same motivation, the criterion for selecting the Gaussians to be updated differs significantly from that of the set-membership approach: the proposed selection strategy checks the values of Gaussians at the current input $\boldsymbol{u}_n$ (see Figure 2). In the proposed selection strategy, moreover, the number of the Gaussians to be updated can be designed by the user.

IV. DISCUSSIONS

A. MONOTONE DECREASING PROPERTY OF COST FUNCTION

The proposed algorithm alternates the (proximal) gradient updates for the Gaussian coefficients $\boldsymbol{h}_n$, scales $\boldsymbol{\xi}_n$, and centers $\boldsymbol{C}_n$. The standard analysis of the proximal gradient algorithm can be applied with (local) Lipschitz continuity of the function. Here, given a metric space $(X, d(\cdot,\cdot))$, a mapping $T : X \to X$ is said to be locally Lipschitz continuous on a subset $C \subset X$ if, for any pair $(x, y) \in C \times C$, $d(Tx, Ty) \leq \gamma d(x, y)$ for some constant $\gamma \geq 0$ [66]. If in particular $C = X$, T is Lipschitz continuous.

For simplicity, we introduce the following shorthand notation to express F_n as a function of a specific entry $\xi^{(j)}$ of $\boldsymbol{\xi}$:

$$F_n^{(\xi^{(j)})}(\xi^{(j)}) := F_n(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}, \check{\boldsymbol{C}}_n)\Big|_{\xi^{(i)} = \check{\xi}_n^{(i)},\ i \neq j}. \tag{17}$$

Note here that all the variables excluding $\xi^{(j)}$ are fixed to the up-to-date values. Likewise, define

$$F_n^{(\boldsymbol{c}^{(j)})}(\boldsymbol{c}^{(j)}) := F_n(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}_{n+1}, \boldsymbol{C})\Big|_{\boldsymbol{c}^{(i)} = \check{\boldsymbol{c}}_n^{(i)},\ i \neq j}. \tag{18}$$

As F_n is quadratic in $\boldsymbol{h}$, $\frac{\partial F_n}{\partial \boldsymbol{h}}$ is clearly Lipschitz continuous with constant $\gamma_n^{(h)} = \left\|\boldsymbol{g}_n\right\|^2$. As the multiplicative update of $\check{\xi}_n^{(j_i)}$ in (12) is derived from the (additive) gradient update of $\check{\eta}_n^{(j_i)}(:= \log\check{\xi}_n^{(j_i)})$, we consider the local Lipschitz continuity of $\frac{\partial F_n^{(\xi^{(j)})}}{\partial \eta^{(j)}}$, $\eta^{(j)} := \log \xi^{(j)}$.[2] The (local) Lipschitz continuity of $\frac{\partial F_n^{(\xi^{(j)})}}{\partial \eta^{(j)}}$ and $\frac{\partial F_n^{(\boldsymbol{c}^{(j)})}}{\partial \boldsymbol{c}^{(j)}}$ is given below.

Lemma 1:

1) The partial derivative $\frac{\partial F_n^{(\xi^{(j)})}}{\partial \eta^{(j)}}$ is locally Lipschitz on $[t, +\infty)$, $t \in \mathbb{R}$, with constant

$$\gamma_{j,n}^{(\eta)}(t) := \frac{\left|h_{n+1}^{(j)}\right|\left(\left|\hat{d}_n^{(j)}\right| + \left|h_{n+1}^{(j)}\right|\right)}{2}\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2 e^{-t}, \tag{19}$$

where $\hat{d}_n^{(j)} := d_n - \sum_{i\neq j} h_{n+1}^{(i)} \exp\left(-\frac{\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(i)}\right\|^2}{2\check{\xi}_n^{(i)}}\right)$.

2) The partial derivative $\frac{\partial F_n^{(\boldsymbol{c}^{(j)})}}{\partial \boldsymbol{c}^{(j)}}$ is Lipschitz continuous with constant

$$\begin{aligned}\gamma_{j,n}^{(c)} &= \delta_{j,n}^* \frac{\left|h_{n+1}^{(j)}\right|}{\xi_{n+1}^{(j)}}\left(\left|\check{d}_n^{(j)}\right| + \delta_{j,n}^*\left|h_{n+1}^{(j)}\right|\right) \\ &\leq \frac{\left|h_{n+1}^{(j)}\right|}{\xi_{n+1}^{(j)}}\left(\left|\check{d}_n^{(j)}\right| + \left|h_{n+1}^{(j)}\right|\right),\end{aligned} \tag{20}$$

where $\delta_{j,n}^* := \max_{i=1,2,\cdots,L} \delta_{j,n}^{(i)} \in (0, 1]$ with $\delta_{j,n}^{(i)} := \exp\left(-\frac{\sum_{k\neq i}(u_n^{(k)} - \check{c}_n^{(k)})^2}{2\xi_{n+1}^{(j)}}\right) \in (0, 1]$ and $\check{d}_n^{(j)} := d_n - \sum_{i\neq j} h_{n+1}^{(i)} \exp\left(-\frac{\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(i)}\right\|^2}{2\xi_{n+1}^{(i)}}\right)$.

Proof: See Appendices B and C. □

Inspecting (19), we can see that the local Lipschitz constant $\gamma_{j,n}^{(\eta)}(t)$ decreases monotonically in t, meaning that the stepsize range allowed for updating the scale parameters becomes narrower as t decreases. This implies that the stepsize bound must be smaller than $2/\gamma_{j,n}^{(\eta)}(t)$ when $\check{\eta}_n^{(j)}$ is updated to a

[2] The multiplicative update can also be updated with the mirror descent update with negative entropy [67], as mentioned in [1].

smaller value (i.e., when its corresponding gradient is positive valued). A question of theoretical interest here is the following: under the standard stepsize range $(0, 2/\gamma_{j,n}^{(\eta)}(t))$, what is the maximal possible $t > 0$ for which the cost function is locally Lipschitz with constant $\gamma_{j,n}^{(\eta)}(t)$ at the current estimate $\check{\eta}_n^{(j)}$ both before and after the update, so that the monotone decreasing property is ensured. Such a t is clearly the updated estimate (which is smaller than before the update) when the gradient is positive. To ensure the local Lipschitz continuity for all possible stepsizes within the range $(0, 2/\gamma_{j,n}^{(\eta)}(t))$, we consider the following equation:

$$t = \check{\eta}_n^{(j)} - \frac{2}{\gamma_{j,n}^{(\eta)}(t)} \frac{\partial F_n^{(\xi^{(j)})}(\check{\xi}_n^{(j)})}{\partial \eta^{(j)}}. \tag{21}$$

We can now present the monotone decreasing property.

Theorem 1: Let $\mu_h \in \left(0, \frac{2}{\gamma_n^{(h)}}\right)$, $\mu_\xi^{(j)} \in \left(0, \frac{2}{\gamma_{j,n}^{(\eta)}(t_n^{(j)})}\right)$, and $\mu_c^{(j)} \in \left(0, \frac{2}{\gamma_{j,n}^{(c)}}\right)$ for[3]

$$t_n^{(j)} = \begin{cases} \check{\eta}_n^{(j)}, & \dfrac{\partial F_n(\hat{\boldsymbol{h}}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n)}{\partial \eta^{(j)}} \leq 0 \\ \tilde{t}_n^{(j)}, & \dfrac{\partial F_n(\hat{\boldsymbol{h}}_{n+1}, \check{\boldsymbol{\xi}}_n, \check{\boldsymbol{C}}_n)}{\partial \eta^{(j)}} > 0, \end{cases} \tag{22}$$

where $\tilde{t}_n^{(j)}$ is a unique solution of (21). Then, after each iteration, it holds that

$$J_n\left(\boldsymbol{h}_n, \boldsymbol{\xi}_n, \boldsymbol{C}_n\right) - J_n\left(\boldsymbol{h}_{n+1}, \boldsymbol{\xi}_{n+1}, \boldsymbol{C}_{n+1}\right) \geq 0. \tag{23}$$

Proof: The claims for μ_h and $\mu_c^{(j)}$ are verified directly by applying the monotone decreasing properties of the (proximal) gradient descent [68], [69] in light of Lemma 1. In the rest, we verify the stepsize range of $\mu_\xi^{(j)}$. If $\frac{\partial F_n^{(\xi^{(j)})}(\check{\xi}_n^{(j)})}{\partial \eta^{(j)}} \leq 0$, the estimate $\check{\eta}_n^{(j)}$ increases after the gradient update, and therefore the (local) Lipschitz constant $\gamma_{j,n}^{(\eta)}(\check{\eta}_n^{(j)})$ is valid over $[\check{\eta}_n^{(j)}, \infty)$ in which the updated $\check{\eta}_n^{(j)}$ lies. If, on the other hand, $\frac{\partial F_n^{(\xi^{(j)})}(\check{\xi}_n^{(j)})}{\partial \eta^{(j)}} > 0$, $\check{\eta}_n^{(j)}$ decreases after the update in (12), and therefore the maximal t ensuring the local Lipschitz continuity is characterized by (21). In this case, (21) has a unique solution since $f(t) := t - \left(\check{\eta}_n^{(j)} - \frac{2}{\gamma_{j,n}^{(\eta)}(t)} \frac{\partial F_n^{(\xi^{(j)})}(\check{\xi}_n^{(j)})}{\partial \eta^{(j)}}\right)$ is continuous and monotonically increasing with $\lim_{t\to+\infty} f(t) = +\infty$ and $\lim_{t\to-\infty} f(t) = -\infty$. □

Equation (21) has no closed form solution, and an iterative method needs to be used to find the $\tilde{t}_n^{(j)}$. This is unfavorable in online estimation. We therefore present efficient designs of the stepsize parameters based on Theorem 1 without solving (21) explicitly in Section IV-C.

B. ON GLOBAL-TO-LOCAL ORDER OF MULTISCALE SCREENING

We discuss the global-to-local (large-to-small scale) order of the proposed multiscale screening method. To find an economic way of expressing the unknown function ψ with our Gaussian model given in (2), the appropriate centers and scales of Gaussian need to be known. This is certainly unrealistic in online scenarios in which the amount of available data is rather limited especially at the early phase of estimation. The local structures need to be expressed with delicate adjustments of center points (as well as scale), and thus small-scale Gaussians are more sensitive to the mismatch of the center position than large-scale ones. This is one of the reasons for the global-to-local order.

Another reason comes from the characteristics of the data-fidelity function $F_n(\boldsymbol{h}, \boldsymbol{\xi}, \boldsymbol{C})$ in (4). As pointed out in Section III-A, the learning speeds of the proposed algorithm become slow when the initial scales are far from the ones of the target due to the gradient vanishment. The sole use of an undesirably-large initial scale tends to yield an underfitting estimate, since the corresponding Gaussian does not fit the nonlinear function ψ. In contrast, the sole use of an undesirably-small initial scale tends to yield an overfitting estimate and it also causes an explosion of the dictionary size, since the learning algorithm seeks to express every detail of ψ with a peaky Gaussian individually. From the current perspective of the authors, this is caused mainly by the gradient vanishment issue mentioned above, but nevertheless we cannot deny the possibility of falling into some local minima. The goal of the present study is to build an adaptive algorithm which generates an efficient approximation of ψ, and the use of small initial scale, especially at the early learning-phrase, is therefore not recommended from this efficiency aspect. The proposed global-to-local strategy works quite well in practice.

C. PARAMETER DESIGN

The parameters σ, ϵ, and Q control the tradeoff between the computational complexity and the performance of the algorithm, and users can design those parameters for each application. As a rule of thumb, the larger the parameters σ, ϵ, and Q, the larger the maximal dictionary size. Although the use of the large dictionary tends to yield fast convergence and low MSEs, this may cause also an explosion of the computational complexity. See Section IV-D for more details about the computational complexity. Empirically, setting the selection parameters $s_n^{(NC)}$, $s_n^{(\xi)}$, $s_n^{(c)}$ from 3 to 7 gives a reasonably low computational complexity (see Section V). The parameter β in the weight of the ℓ_1 norm (see (4)) is the regularization parameter to avoid division by zero, and it is thus set to some small value such as 10^{-4}.

The stepsize parameters μ_h, $\mu_\xi^{(j)}$, and $\mu_c^{(j)}$ and the regularization parameter λ affect the accuracy of the final estimate as well as the convergence speed, and thus it needs to be carefully designed. In particular, the stepsizes $\mu_\xi^{(j)}$ and $\mu_c^{(j)}$ as well as λ govern the dictionary size and thus the efficiency

[3] In (22), $\check{\eta}_n^{(j)}$ is the parameter before update, as it cannot be used to update the $\check{\eta}_n^{(j)}$ itself otherwise.

of the final estimate. We present efficient designs of $\mu_\xi^{(j)}$ and $\mu_c^{(j)}$ in the following subsections.

1) DESIGN OF $\mu_c^{(j)}$

To ensure the monotone decreasing property in Theorem 1, an appropriate stepsize depends on the Lipschitz constant $\gamma_{j,n}^{(c)}$. Unfortunately, $\gamma_{j,n}^{(c)}$ in (20) is defined with $\xi_{n+1}^{(j)}$, $\check{d}_n$, and $h_{n+1}^{(j)}$, which are unavailable for designing $\mu_c^{(j)}$ prior to adaptation as no prior knowledge is assumed available about the structure of the target system (see Section II). Fortunately, the initial scale $\xi_{init}^{(q)}$ could be used as an alternative of $\xi_{n+1}^{(j)}$, since the current $\xi_{n+1}^{(j)}$ is expected to be closer to $\xi_{init}^{(q)}$ than (at least most of) the others $\xi_{init}^{(\tilde{q})}$, $\tilde{q} = 1, \cdots, q-1, q+1, \cdots, Q$. Replacing $\xi_{n+1}^{(j)}$ by $\xi_{init}^{(q)}$, $\gamma_{j,n}^{(c)}$ is inversely proportional to $\xi_{init}^{(q)}$, and an appropriate stepsize is thus proportional to $\xi_{init}^{(q)}$. Based on the above discussion, below is a design scheme for the stepsize $\mu_c^{(j)}$.

Example 1 (Design scheme for $\mu_c^{(j)}$): Set $\mu_c > 0$. For the Gaussians initialized by $\xi_{init}^{(1)}$, set the stepsize to $\mu_c^{(j)} = \mu_c$. For the Gaussians initialized by each $\xi_{init}^{(q)}$, $q = 2, \cdots, Q$, set the stepsize to $\mu_c^{(j)} = \frac{\xi_{init}^{(q)}}{\xi_{init}^{(1)}}\mu_c$.

2) DESIGN OF $\mu_\xi^{(j)}$

In contrast to $\mu_c^{(j)}$, one can employ the same value for all $\mu_\xi^{(j)}$ due to the following reason. Since $\gamma_{j,n}^{(\xi)}(t)$ is monotonically increasing in $\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2$ with $\lim_{\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2 \to \infty} \gamma_{j,n}^{(\xi)}(t) = \infty$, we need to set an upper bound for $\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2$ to design $\mu_\xi^{(j)}$ based on $\gamma_{j,n}^{(\xi)}(t)$. Meanwhile, a large $\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2$ implies a small output of the Gaussian $\exp\left(-\frac{\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2}{2\check{\xi}_n^{(j)}}\right)$. This means that the Gaussian gives a negligible impact on the estimation in the vicinity of $\boldsymbol{u}_n$ when $\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2$ is sufficiently large. Due to the above discussions, we now make an upper bound as $\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2 \leq -2\check{\xi}_n^{(j)} \log a$ for some small constant $0 < a \ll 1$ so that $\exp\left(-\frac{\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2}{2\check{\xi}_n^{(j)}}\right) \geq a$. In the same way as the design scheme for $\mu_c^{(j)}$, we replace $\check{\xi}_n^{(j)}$ in (19) by the initial scale $\xi_{init}^{(q)}$ as in the design of $\mu_c^{(j)}$ with $t := \log \xi_{init}^{(q)}$ (see also (22)). Substituting $\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2 = -2\xi_{init}^{(q)} \log a$ to (19), we obtain $\gamma_{j,n}^{(\xi)}(\eta_{init}^{(q)})|_{\left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(i)}\right\|^2 = -2\xi_{init}^{(q)} \log a} = -\log a \left|h_{n+1}^{(j)}\right| \left(\left|\hat{d}_n^{(j)}\right| + \left|h_{n+1}^{(j)}\right|\right)$ which no longer depends on $\xi_{init}^{(q)}$. It is thus reasonable to use the same stepsizes for all js.

3) DESIGN OF INITIAL GAUSSIAN SCALES

If the components of the target function were known, one could select the initial Gaussian scales $\xi_{init}^{(q)}$ that are appropriate for those components. In many applications, however, the target components are unknown prior to estimation. In such a case, one may want to use a set of Gaussians with a variety of scales which are regular in a certain sense. As the set of inflection points of each Gaussian forms a hypersphere of radius $\sqrt{\xi_{init}^{(j)}}$, the idea is to place the hyperspheres in a regular fashion. We present a selection example of the initial Gaussian scales.

Example 2 (Initial Gaussian scales): The user sets the largest and smallest scales $\xi_{init}^{(1)}$ and $\xi_{init}^{(Q)}$ to some appropriate values. The other scales are then set to $\xi_{init}^{(q)} := \sqrt{\xi_{init}^{(1)}} - (q - 1)\frac{\sqrt{\xi_{init}^{(1)}} - \sqrt{\xi_{init}^{(Q)}}}{Q-1}$, $q = 2, \cdots, Q - 1$.
The one-dimensional case is illustrated in Figure 3. Here, the scales $\xi_{init}^{(2)}$ and $\xi_{init}^{(3)}$ are determined so that the inflection points $\sqrt{\xi_{init}^{(q)}}$ of all Gaussians $g(\cdot; \xi_{init}^{(q)}, 0)$, $q = 1, 2, 3, 4$, are equally spaced in the interval $\left[\sqrt{\xi_{init}^{(4)}}, \sqrt{\xi_{init}^{(1)}}\right]$. The parameter design scheme presented in Example 2 works well in practice as shown in Section V.

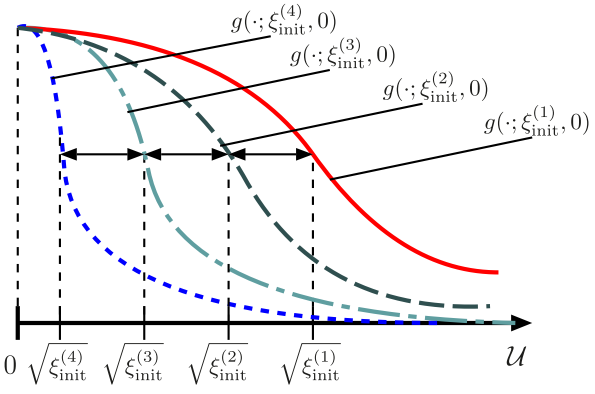


FIGURE 3. An example of initial Gaussian scales for $Q = 4$.

D. COMPUTATIONAL COMPLEXITY

The computational complexity of the proposed algorithm at each time instant n is generally given in terms of the dictionary size r_n as well as the input dimension L. The computational complexity of the proposed algorithm depends also on $s^{(NC)}$, $s^{(\xi)}$, $s^{(c)}$, and Q which are supposed to be constant during the adaptation.

Table 1 summarizes the overall complexities (the number of real multiplications) per iteration of the proposed algorithm and the related algorithms. Here, the kernel normalized least mean square (KNLMS) algorithm [40], which is a kernelized version of the normalized least mean square (NLMS) [70] algorithm, is a benchmark algorithm in the kernel adaptive filtering. The quantized kernel least mean square algorithm with adaptive kernel size (QKLMS-AKS) [46] and the kernel algorithm with adaptive width (KAW) [71] are kernel adaptive filtering algorithms which adapt the Gaussian scales. The resource allocating network (RAN) is a benchmark algorithm in the RBF network. The complexity

TABLE 1. Computational complexities of the proposed and related algorithms.

Proposed	$(L+5)r_n + 8s^{(\xi)} + (2L+4)s^{(c)} + Q(L/4+4)s^{(NC)}$
QKLMS-AKS	$r_n(L+2)$
KAW	$r_n(L+4) + (r_n-1)^2$
RAN	$r_n(2L+6)$
KNLMS	$r_n(L+3)$
NLMS	$2L$

contains all the multiplications required at each time n including those for dictionary growing and parameter updating. For the proposed algorithm, the case of the non-selective update (i.e., $s^{(NC)} = s^{(\xi)} = s^{(c)} = r_n$) is also considered to show the effectiveness of the selective update.

Figure 4 illustrates the complexities as a function of the dictionary size r_n for $L = 6$ and $Q = 3$. The figure shows that the complexity of the proposed algorithm ($s^{(NC)} = 3$, $s^{(\xi)} = s^{(c)} = 5$) is lower than those of the proposed algorithm (non-selective update) and RAN. This is due to the selection strategy for dictionary growing and parameter updating. Although the proposed algorithm ($s^{(NC)} = 3$, $s^{(\xi)} = s^{(c)} = 5$) requires the slightly higher complexity than QKLMS-AKS, it enjoys significant gains in MSE, as shown in the next section.

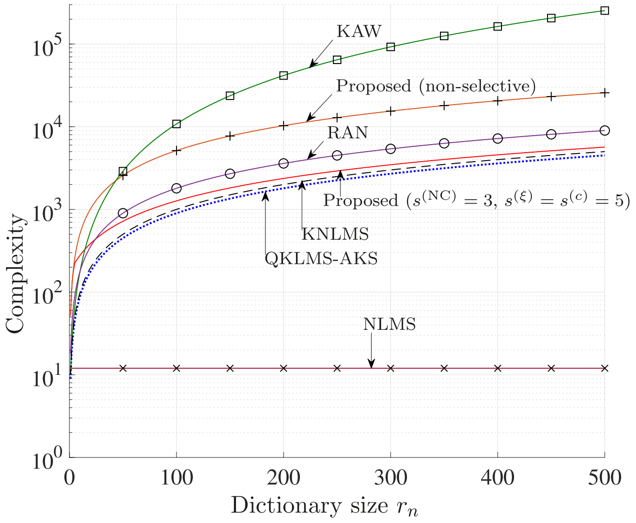


FIGURE 4. Computational complexities of the proposed and related algorithms.

V. SIMULATION RESULTS

We show the efficacy of the proposed algorithm for system identification problems with two sets of synthetic data and time-series prediction problems with two benchmark data. For the proposed algorithm, the dictionaries are constructed by the multiscale screening method presented in Section III-A. For the weighted ℓ_1 norm, $\omega_n^{(j)} = \frac{1}{\left|h_n^{(j)}\right|+\beta}$ [57] with $\beta = 10^{-4}$ is employed. The numbers of checked/updated Gaussians are set to $s_n^{(NC)} = s_n^{(\xi)} = s_n^{(c)} = 5$ in Experiment 1 and 2, and $s_n^{(NC)} = s_n^{(\xi)} = s_n^{(c)} = 4$ in Experiment 3, for all $n \in \mathbb{N}$.

EXPERIMENT 1: EFFECTIVENESS OF THE ADAPTATION OF GAUSSIAN SCALES AND CENTERS

We consider the following nonlinear function

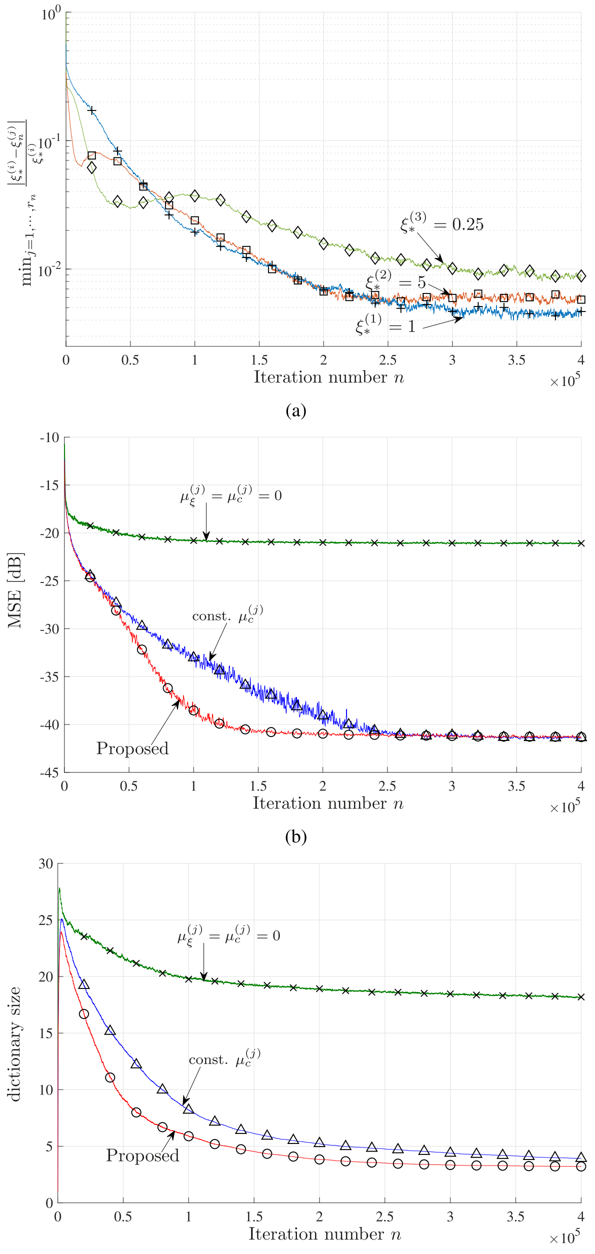
$$\psi(\boldsymbol{u}) = \exp\left(-\frac{\|\boldsymbol{u}-0.75\mathbf{1}\|^2}{2\xi_*^{(1)}}\right) - 3\exp\left(-\frac{\|\boldsymbol{u}-1.5\mathbf{1}\|^2}{2\xi_*^{(2)}}\right) + 2\exp\left(-\frac{\|\boldsymbol{u}-2.25\mathbf{1}\|^2}{2\xi_*^{(3)}}\right),\ \boldsymbol{u} \in \mathbb{R}^5, \quad (24)$$

which is the sum of three Gaussian functions with $\xi_*^{(1)} = 1$, $\xi_*^{(2)} = 5$, and $\xi_*^{(3)} = 0.25$, where $\mathbf{1} = [1, \cdots, 1]^{\mathsf{T}} \in \mathbb{R}^5$. The observed signal is generated as $d_n := \psi(\boldsymbol{u}_n) + v_n$, $n \in \mathbb{N}$, where $\boldsymbol{u}_n$ is the input data of which each element is randomly generated from a uniform distribution over $[0, 3]^5$ and $v_n \sim \mathcal{N}(0, 1.0 \times 10^{-2})$ is the additive white Gaussian noise.

To show that the proposed algorithm adapts the Gaussian scale and center efficiently, the performance of the proposed algorithm is compared with the performance of the proposed algorithm without the adaptation of the Gaussian scales ξ and centers $\boldsymbol{c}$ ($\mu_\xi^{(j)} = \mu_c^{(j)} = 0$). For the proposed algorithm, the initial Gaussian scales are selected according to Example 2 with $\xi_{init}^{(1)} = 10^{0.5}$ and $\xi_{init}^{(3)} = 10^{-0.5}$; the stepsizes $\mu_c^{(j)}$ for the Gaussian scales are according to Example 1 with $\mu_c = 0.1$; and the other stepsizes are set to $\mu_h = 0.1$ and $\mu_\xi^{(j)} = 0.1$. The regularization parameter $\lambda = 10^{-3}$ and the parameters of the multiscale screening method are chosen so that the dictionary size is close to the number of Gaussians contained in the target and the MSE becomes as low as possible at the steady state. To show the effectiveness of the design scheme for $\mu_c^{(j)}$, we test the performance of the proposed algorithm with the constant stepsizes $\mu_c^{(j)} = 0.05$, $j = 1, \cdots, r_{n+1}$. For the proposed algorithm ($\mu_\xi^{(j)} = \mu_c^{(j)} = 0$), $\xi_{init}^{(1)} = 2.5$, $\xi_{init}^{(2)} = 0.75$, and $\xi_{init}^{(3)} = 0.1$ are chosen so that the algorithm achieves the best performance. We empirically found that the use of a large number of Gaussians with small scales yields small errors when adequate Gaussian parameters are unknown. Even if the scales of the target function is known, the estimation errors may become large when the centers are located at inadequate positions. For the algorithm ($\mu_\xi^{(j)} = \mu_c^{(j)} = 0$), the best parameters are chosen such that the maximal dictionary size is as close as possible to that of the proposed algorithm.

Figure 5 depicts (a) the normalized minimum differences of the Gaussian scales between the target function and the estimate, i.e., $\min_{j=1,\cdots,r_n} \frac{\left|\xi_*^{(i)} - \xi_n^{(j)}\right|}{\xi_*^{(i)}}$, $i = 1, 2, 3$, (b) the MSE, and (c) the dictionary size. All results are averaged over 200 runs.

From Figure 5(a), one can see that the errors of the Gaussian scales are reasonably small. Furthermore, Figures 5(b) and 5(c) show that the dictionary size of the proposed algorithm at the steady state is nearly identical to the number of Gaussians of the target function, and also the proposed algorithm is superior to the algorithm ($\mu_\xi^{(j)} = \mu_c^{(j)} = 0$) in the sense of MSE. The proposed algorithm achieves the



(a)

(b)

(c)

FIGURE 5. Results of Experiment 1: (a) the minimum difference of the Gaussian scales between the target function and the atoms, (b) MSE, and (c) dictionary size.

considerably small steady-state MSE (-41 [dB]) which is 20 [dB] lower than the algorithm ($\mu_\xi^{(j)} = \mu_c^{(j)} = 0$), thanks to the adaptations of the Gaussian scales and centers. The algorithm (const. $\mu_c^{(j)}$) requires many iterations to reach the steady-state MSE. The proposed design scheme for $\mu_c^{(j)}$ enables to use adequate stepsizes for each of Gaussians and consequently the proposed algorithm achieves the fast convergence. These results show that the proposed algorithm quickly yields the 'efficient' estimates by adapting the Gaussian scales and centers with adequate stepsizes $\mu_c^{(j)}$, although the proposed algorithm has no guarantee to yield perfectly efficient estimates.

EXPERIMENT 2: EFFECTIVENESS OF THE MULTISCALE SCREENING

To show the effectiveness of the multiscale screening method presented in Section III-A, estimation performances of the proposed algorithm are studied for such functions that consist of extremely large and small Gaussians. Specifically, we consider the nonlinear function $\psi(\boldsymbol{u}) = \sum_i^5 h_*^{(i)} \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{c}^{(i)}\|^2}{2\xi_*^{(i)}}\right)$, $\boldsymbol{u} \in \mathbb{R}^3$. Here $\boldsymbol{c}^{(i)} \in \mathbb{R}^3$ is generated from a uniform distribution over $[0, 1]^3$, $h^{(1)} = 1$, $h^{(i)} = 5, \ i = 2, \cdots, 5$, and $\xi_*^{(i)}$ is generated as $\xi_*^{(i)} = \left|\xi_*^{\tilde{(i)}}\right|$ with

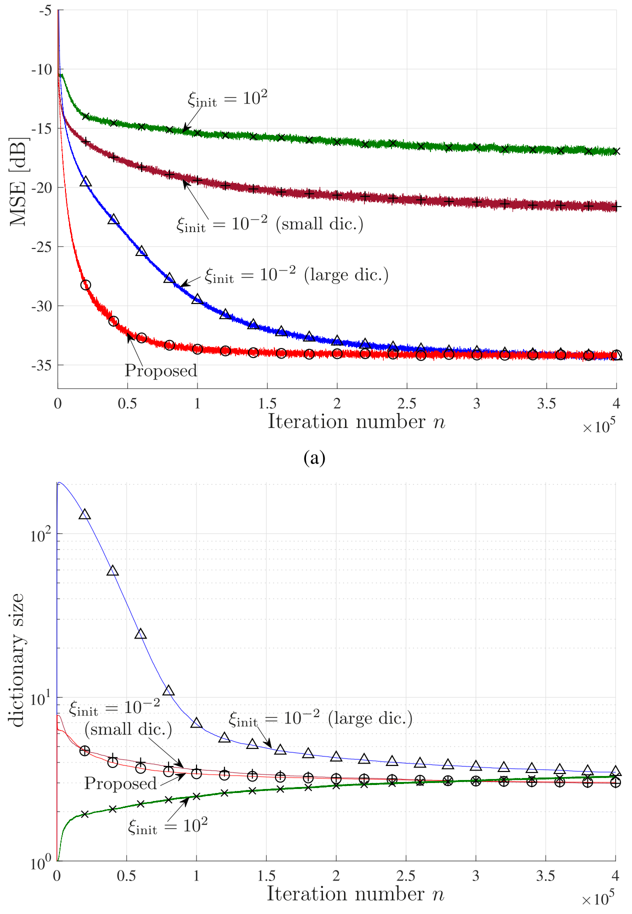
$$\xi_*^{\tilde{(i)}} \sim \begin{cases} \mathcal{N}(100, 10.0), & i = 1 \\ \mathcal{N}(1 \times 10^{-2}, 5.0 \times 10^{-3}), & i = 2, \cdots, 5. \end{cases} \qquad (25)$$

The observed signal is generated as $d_n := \psi(\boldsymbol{u}_n) + v_n, n \in \mathbb{N}$, where $\boldsymbol{u}_n$ is the input data of which each element is randomly generated from a uniform distribution over $[0, 1]^3$ and $v_n \sim \mathcal{N}(0, 1.0 \times 10^{-2})$ is the additive white Gaussian noise. The proposed algorithm is tested with (i) the multiple initial scales ($Q = 3$) and (ii) the single initial scale ($Q = 1$). The results are averaged over 500 independent trials.

For the proposed algorithm, the initial Gaussian scales are selected according to Example 2 with $\xi_{init}^{(1)} = 10^2$ and $\xi_{init}^{(3)} = 10^{-2}$; the stepsizes $\mu_c^{(j)}$ for the Gaussian scales are according to Example 1 with $\mu_c = 0.1$; and the other stepsizes are set to $\mu_h = 0.1$ and $\mu_\xi^{(j)} = 0.1$. The other stepsizes are set to $\mu_h = 0.1$ and $\mu_\xi^{(j)} = 0.1$, $j = 1, \cdots, r_{n+1}$. The regularization parameter and the parameters of the multiscale screening method are chosen in the same way as in Experiment 1.

For the proposed algorithm ($Q = 1$), the two settings for the initial scales are tested: the large initial-scale $\xi_{init} = \xi_{init}^{(1)} = 10^2$ and the small initial-scale $\xi_{init} = \xi_{init}^{(3)} = 10^{-2}$. For the small initial-scale, we consider two cases: (i) small dictionary for the same maximal dictionary size, and (ii) large dictionary for the same steady-state MSE, as the proposed algorithm. For the large initial-scale $\xi_{init} = 10^2$, the parameters are selected so that the lowest MSE is attained.

Figure 6 illustrates the results in terms of (a) the MSEs and (b) the dictionary sizes. In Figure 6(b), one can see that the algorithms with $Q = 1$ yield notably high MSE for $\xi_{init} = 10^2$ and $\xi_{init} = 10^{-2}$ (small dic.). This is because, with the extremely large initial Gaussian scale $\xi_{init} = 10^2$, the adaptation of the Gaussian scales tends to stop at large scales, failing to capture fine fluctuations caused by small scale Gaussians. In contrast, in the extremely small initial Gaussian scale $\xi_{init} = 10^{-2}$, the adaptation tends to stop at small scales, failing to capture the global structure of the target. Although the MSEs of $\xi_{init} = 10^{-2}$ (large dic.) is reasonably small due to the use of the large number of small Gaussians in the initial phase of estimation, the maximal dictionary size is unacceptably large and the redundant Gaussians remain for a while as seen in Figure 6 (b). These results



(a)

(b)

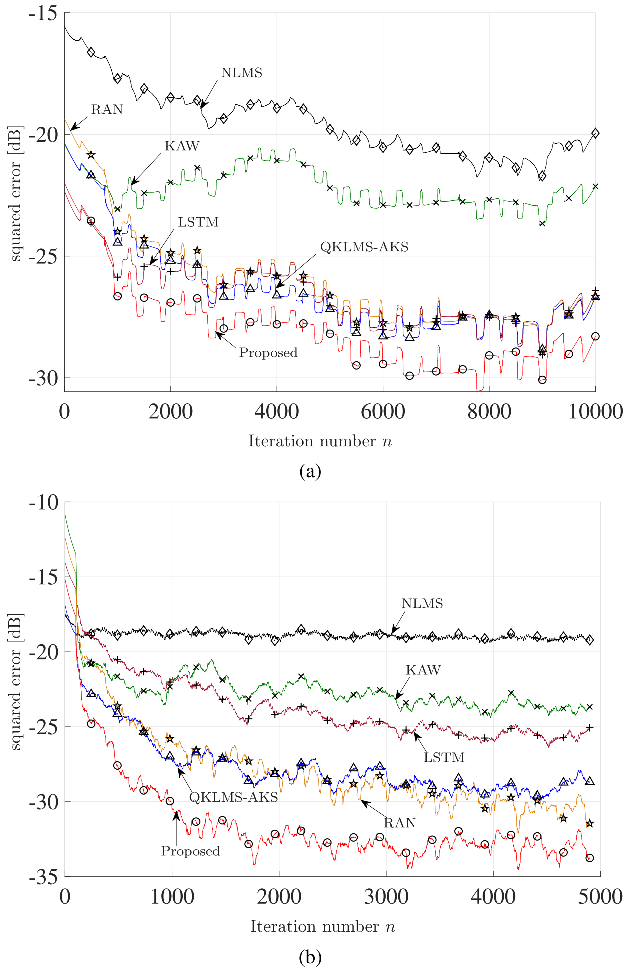
FIGURE 6. Results of Experiment 2: (a) the MSEs and (b) the dictionary sizes.

are due to the gradient vanishment caused by the nonconvexity of the cost function as pointed out in Sections. III-A and IV-B. The proposed algorithm attains the efficient estimates, preventing from the sharp rise of the dictionary size. This shows the effectiveness the global-to-local order of the multiscale screening method.

EXPERIMENT 3: APPLICATION TO PREDICTION OF REAL AND SYNTHETIC TIME-SERIES DATA

We demonstrate the performance of the proposed algorithm in application to online predictions of (a) the laser data [72] from SantaFe dataset and (b) the sequence generated by Mackey-Glass equation [73].[4] Each datum d_n is predicted with $\boldsymbol{u}_n := [d_{n-1}, d_{n-2}, \cdots, d_{n-L}]^{\mathsf{T}} \in \mathcal{U} \subset \mathbb{R}^L$ for $L = 6$.

The proposed algorithm is compared with the following algorithms: (i) NLMS, (ii) RAN (a benchmark algorithm in RBF network field), (iii) the state-of-the-art algorithm [74] for online time-series estimation which is based on LSTM neural network architecture, and (iv) the state-of-the-art nonlinear estimation algorithms that adapt the Gaussian scales



(a)

(b)

FIGURE 7. Learning curves of Experiment 3: (a) laser data from SantaFe data set and (b) Mackey-Glass equation.

with single initial values: QKLMS-AKS [46] and KAW [71]. In contrast to the proposed algorithm which discards redundant atoms from the dictionary by the ℓ_1 norm regularization, QKLMS-AKS has no structure to discard the atoms, i.e., the dictionary size of QKLMS-AKS increases monotonically. Unlike the proposed algorithm and QKLMS-AKS, KAW fixes the dictionary size at some predefined values. To be more precise, the dictionary grows at every iteration until the dictionary size reaches the predefined value. If the dictionary size exceeds the predefined value, one atom is discarded from the dictionary. For the proposed algorithm, the initial Gaussian scales are determined by Example 2 with $\xi_{init}^{(1)} = 10$ and $Q = 3$. Figures 7(a) and 7(b) depict the learning curves for (a) Santa-Fe dataset and (b) Mackey-Glass data. Here, (a) 500- and (b) 200- point moving averages of the results are taken, respectively. Table 2 summarizes the squared errors, computational complexities, and maximal dictionary sizes averaged over all iterations. The complexities of the proposed algorithm, RAN, QKLMS-AKS, and KAW are shown in Table 1 with the dictionary sizes averaged over iterations. The complexities of NLMS and LSTM are counted as $2L$ and $N(m^2 + mp)$, respectively, where N, m, p are the

[4] The sequence is generated by $\frac{dx_n}{dn} = -bx_n + \frac{ax_{n-t}}{1+x_{n-t}^{10}}$. with $b = 0.1$, $a = 0.2$ and time delay $t = 30$

TABLE 2. Results of Experiment 3.

		squared error	complexity	dic. size (max)
Proposed	(a) SantaFe	**−28.4** [dB]	339	18
	(b) Mackey-Glass	**−27.37** [dB]	355	38
QKLMS -AKS [46]	(a) SantaFe	−25.8 [dB]	382	52
	(b) Mackey-Glass	−26.5 [dB]	365	49
KAW [71]	(a) SantaFes	−22.1 [dB]	2901	50
	(b) Mackey-Glass	−21.5 [dB]	2485	46
RAN [38]	(a) SantaFe	−25.1 [dB]	579	54
	(b) Mackey-Glass	−25.0 [dB]	697	51
LSTM [74]	(a) SantaFe	−26.2 [dB]	3040	-
	(b) Mackey-Glass	−22.7 [dB]	2560	-
NLMS [70]	(a) SantaFe	−19.2 [dB]	12	-
	(b) Mackey-Glass	−18.9 [dB]	12	-

numbers of particles, output nodes, and input nodes of the network, respectively. The dictionary size of QKLMS-AKS is selected so that the complexity of QKLMS-AKS is approximately the same as that of the proposed algorithm. The dictionary sizes of KAW and RAN is selected so that the maximal dictionary sizes of KAW and RAN are approximately the same as the proposed algorithm. The particle number N of LSTM is selected so as to attain the same complexity as KAW, which requires the largest complexity in these algorithms. Note that the dictionary sizes of all algorithms change dynamically.

Figures 7(a) and 7(b), and Table 2 show that the MSEs of the proposed algorithm are smaller than those of the others for both data. Furthermore, it can be seen from Table 2 that the proposed algorithm requires a lower complexity than KAW, RAN, and LSTM. The performance of NLMS is limited since its model is linear and is thus inadequate for the nonlinear time-series prediction. Again, the performance of KAW is inferior to the other nonlinear algorithms due to the use of the same Gaussian scales for all atoms. Note that KAW and LSTM may achieve lower MSEs, but with high complexity, if a larger-sized dictionary and many particles are used, respectively.

VI. CONCLUSION

We proposed a learning algorithm which adapts the model parameters, as well as the coefficients, of a weighted sum of the Gaussians. The proposed algorithm consisted of two steps: the dictionary growing step and the parameters updating step. In the dictionary growing step, a novel multiple initialization scheme was presented as a remedy for the gradient vanishing problem without serious increases of the dictionary size. In the parameter updating step, the Gaussian parameters were updated as well as the coefficients by the proximal gradient based algorithm. Due to the use of the ℓ_1 norm regularization, the model efficiency was enhanced. Thanks to the selection strategy for dictionary growing and scale/center updating, the complexity of the proposed algorithm was reasonably low. Computer simulations for the toy examples showed that the proposed algorithm successfully attains efficient estimates. In application to the time-series data predictions, the proposed algorithm achieved approximately 4.7 [dB] lower MSE than the state-of-the-art online prediction algorithm.

ACKNOWLEDGMENT

The authors are thankful to Prof. Isao Yamada of the Tokyo Institute of Technology for his suggestions. This article was presented in part at the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2019).

APPENDIX A
SKETCH OF THE DERIVATION OF (8)

The inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ between the two normalized Gaussians under the uniform distribution with infinite interval is given as [75]

$$\left\langle \tilde{g}(\cdot;\xi^{(u)},\boldsymbol{u}), \tilde{g}(\cdot;\xi^{(v)},\boldsymbol{v})\right\rangle_{\mathcal{H}} = \frac{1}{\left(2\pi(\xi^{(u)}+\xi^{(v)})\right)^{L/2}}\exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2(\xi^{(u)}+\xi^{(v)})}\right), \quad \text{(A.1)}$$

where $\tilde{g}(\cdot;\xi,\boldsymbol{c}) := \frac{1}{(2\pi\xi)^{L/2}}\exp\left(-\frac{\|\cdot-\boldsymbol{c}\|^2}{2\xi}\right)$, $\boldsymbol{c}\in\mathbb{R}^L$, *and* $\xi>0$. We can also easily verify the following:

$$\left\langle g(\cdot;\xi^{(u)},\boldsymbol{u}), g(\cdot;\xi^{(v)},\boldsymbol{v})\right\rangle_{\mathcal{H}} = \left(2\pi\frac{\xi^{(u)}\xi^{(v)}}{\xi^{(u)}+\xi^{(v)}}\right)^{L/2}\exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2(\xi^{(u)}+\xi^{(v)})}\right) \quad \text{(A.2)}$$

and

$$\left\|g(\cdot;\xi^{(u)},\boldsymbol{u})\right\|_{\mathcal{H}} = \sqrt{\langle g(\cdot;\xi^{(u)},\boldsymbol{u}), g(\cdot;\xi^{(u)},\boldsymbol{u})\rangle_{\mathcal{H}}} = \left(\pi\xi^{(u)}\right)^{L/4}. \quad \text{(A.3)}$$

By using the above inner product and the norm, the coherence (8) is obtained.

APPENDIX B
LIPSCHITZ CONTINUITY OF $\frac{\partial F_n^{(\xi^{(j)})}}{\partial\eta^{(j)}}$

For brevity, we let $\rho_n^{(j)} := \left\|\boldsymbol{u}_n - \check{\boldsymbol{c}}_n^{(j)}\right\|^2 \geq 0$. The function $F_n^{(\xi^{(j)})}$ of $\xi^{(j)}$ can then be written as

$$F_n^{(\xi^{(j)})}\left(\xi^{(j)}\right) = \frac{1}{2}\left(\hat{d}_n^{(j)} - h_{n+1}^{(j)}\exp\left(-\frac{\rho_n^{(j)}}{2\xi^{(j)}}\right)\right)^2. \quad \text{(B.1)}$$

Using the chain rule, the partial derivative of $F_n^{(\xi^{(j)})}$ with respect to $\eta^{(j)} = \log\xi^{(j)}$ is then given by

$$\frac{\partial F_n^{(\xi^{(j)})}\left(\xi^{(j)}\right)}{\partial\eta^{(j)}} = -\frac{h_{n+1}^{(j)}\rho_n^{(j)}}{2\xi^{(j)}}\exp\left(-\frac{\rho_n^{(j)}}{2\xi^{(j)}}\right)\left(\hat{d}_n^{(j)} - h_{n+1}^{(j)}\exp\left(-\frac{\rho_n^{(j)}}{2\xi^{(j)}}\right)\right). \quad \text{(B.2)}$$

For $\xi^{(j)},\tilde{\xi}^{(j)} > 0$, one can verify by the triangle inequality that

$$\left|\frac{\partial F_n^{(\xi^{(j)})}(\xi^{(j)})}{\partial\eta^{(j)}} - \frac{\partial F_n^{(\xi^{(j)})}(\tilde{\xi}^{(j)})}{\partial\eta^{(j)}}\right|$$

$$\begin{aligned}&\leq \left|h_{n+1}^{(j)}\right| \left[\left|\hat{d}_n^{(j)}\right| \left|\frac{\rho_n^{(j)}}{2\xi^{(j)}} \exp\left(-\frac{\rho_n^{(j)}}{2\xi^{(j)}}\right) - \frac{\rho_n^{(j)}}{2\tilde{\xi}^{(j)}} \exp\left(-\frac{\rho_n^{(j)}}{2\tilde{\xi}^{(j)}}\right)\right|\right.\\ &\quad \left. + \left|h_{n+1}^{(j)}\right| \left|\frac{\rho_n^{(j)}}{2\xi^{(j)}} \exp\left(-\frac{\rho_n^{(j)}}{\xi^{(j)}}\right) - \frac{\rho_n^{(j)}}{2\tilde{\xi}^{(j)}} \exp\left(-\frac{\rho_n^{(j)}}{\tilde{\xi}^{(j)}}\right)\right|\right]. \end{aligned} \tag{B.3}$$

Letting $x := \frac{\rho_n^{(j)}}{2\xi^{(j)}}$ and $\tilde{x} := \frac{\rho_n^{(j)}}{2\tilde{\xi}^{(j)}} > 0$ in (B.3) yields

$$\begin{aligned}&\left|\frac{\partial F_n^{(\xi^{(j)})}(\xi^{(j)})}{\partial \eta^{(j)}} - \frac{\partial F_n^{(\xi^{(j)})}(\tilde{\xi}^{(j)})}{\partial \eta^{(j)}}\right|\\ &\leq \left|h_{n+1}^{(j)}\right| \left(\left|\hat{d}_n^{(j)}\right| \left|xe^{-x} - \tilde{x}e^{-\tilde{x}}\right| + \left|h_{n+1}^{(j)}\right| \left|xe^{-2x} - \tilde{x}e^{-2\tilde{x}}\right|\right). \end{aligned} \tag{B.4}$$

Considering the maximal magnitude of the gradient, the following inequalities are readily verified:

$$\left|xe^{-ax} - \tilde{x}e^{-a\tilde{x}}\right| \leq |x - \tilde{x}|, \ a > 0. \tag{B.5}$$

Combining (B.4) and (B.5), we obtain

$$\begin{aligned}&\left|\frac{\partial F_n^{(\xi^{(j)})}(\xi^{(j)})}{\partial \eta^{(j)}} - \frac{\partial F_n^{(\xi^{(j)})}(\tilde{\xi}^{(j)})}{\partial \eta^{(j)}}\right|\\ &\qquad \leq \left|h_{n+1}^{(j)}\right| \left(\left|\hat{d}_n^{(j)}\right| + \left|h_{n+1}^{(j)}\right|\right) |x - \tilde{x}|. \end{aligned} \tag{B.6}$$

On the other hand, by using $\xi^{(j)} = e^{\eta^{(j)}}$, $|x - \tilde{x}|$ is rewritten as

$$|x - \tilde{x}| = \left|\frac{\rho_n^{(j)}}{2\xi^{(j)}} - \frac{\rho_n^{(j)}}{2\tilde{\xi}^{(j)}}\right| = \left|\frac{\rho_n^{(j)}}{2}\right| \left|e^{-\eta^{(j)}} - e^{-\tilde{\eta}^{(j)}}\right|. \tag{B.7}$$

Here, due to the convexity of $e^{-\eta}$, $\eta \in \mathbb{R}$, one can verify that $\left|e^{-\eta^{(j)}} - e^{-\tilde{\eta}^{(j)}}\right| \leq e^{-t} \left|\eta^{(j)} - \tilde{\eta}^{(j)}\right|$, $\forall \eta^{(j)}, \tilde{\eta}^{(j)} \geq t \in \mathbb{R}$, from which (B.7) leads to

$$|x - \tilde{x}| \leq \left|\frac{\rho_n^{(j)}}{2}\right| e^{-t} \left|\eta^{(j)} - \tilde{\eta}^{(j)}\right|, \ \forall \eta^{(j)}, \tilde{\eta}^{(j)} \geq t. \tag{B.8}$$

Combining (B.6) and (B.8) yields

$$\begin{aligned}&\left|\frac{\partial F_n^{(\xi^{(j)})}(\xi^{(j)})}{\partial \eta^{(j)}} - \frac{\partial F_n^{(\xi^{(j)})}(\tilde{\xi}^{(j)})}{\partial \eta^{(j)}}\right|\\ &\leq \frac{\left|h_{n+1}^{(j)}\rho_n^{(j)}\right|}{2} \left(\left|\hat{d}_n^{(j)}\right| + \left|h_{n+1}^{(j)}\right|\right) e^{-t} \left|\eta^{(j)} - \tilde{\eta}^{(j)}\right|, \ \forall \eta^{(j)}, \tilde{\eta}^{(j)} \geq t. \end{aligned}$$

APPENDIX C

LIPSCHITZ CONTINUITY OF $\frac{\partial F_n^{(c^{(j)})}}{\partial c^{(j)}}$

We first prove the following lemma.

Lemma 2: For $f(c) := c \exp\left(\frac{-c^2}{\xi}\right)$, $c \in \mathbb{R}$, $\xi > 0$, the following inequality holds:

$$|f(c) - f(\tilde{c})| \leq |c - \tilde{c}|. \tag{C.1}$$

Proof: The first and second derivatives of f are given by $f'(c) = \exp\left(\frac{-c^2}{\xi}\right) - \frac{2c^2}{\xi} \exp\left(\frac{-c^2}{\xi}\right)$ and $f''(c) = \frac{4c}{\xi^2}\left(-\frac{3}{2}\xi + c^2\right) \exp\left(\frac{-c^2}{\xi}\right)$, respectively. By solving $f''(c) = 0$, we obtain the inflection points $c = 0$ and $c = \pm\sqrt{\frac{3}{2}\xi}$ of f, and at those points f' has the following values: $f'(0) = -1$ and $f'\left(\pm\sqrt{\frac{3}{2}\xi}\right) = -2e^{-\frac{3}{2}}$, respectively. Since $\left|f'(0)\right| > \left|f'\left(\pm\sqrt{\frac{3}{2}\xi}\right)\right|$ and $\lim_{c \to \pm\infty} f'(c) = 0$, we obtain (C.1). □

Proof of (20): For brevity, we drop the time index n. We shall then prove the following inequality:

$$\begin{aligned}&\left\|\frac{\partial F^{(\boldsymbol{c}^{(j)})}(\boldsymbol{c})}{\partial \boldsymbol{c}^{(j)}} - \frac{\partial F^{(\boldsymbol{c}^{(j)})}(\tilde{\boldsymbol{c}})}{\partial \boldsymbol{c}^{(j)}}\right\|\\ &\quad \leq \delta^* \frac{\left|h^{(j)}\right|}{\xi^{(j)}} \left(\left|\check{d}^{(j)}\right| + \delta^* \left|h^{(j)}\right|\right) \|\boldsymbol{c} - \tilde{\boldsymbol{c}}\|,\\ &\quad \boldsymbol{c}, \tilde{\boldsymbol{c}} \in \mathbb{R}^L. \end{aligned} \tag{C.2}$$

(Note that the inequality in (20) can readily be verified by $\delta^* \leq 1$.) The function $F^{(\boldsymbol{c}^{(j)})}$ of $\boldsymbol{c}^{(j)}$ can be written as

$$F^{(\boldsymbol{c}^{(j)})}\left(\boldsymbol{c}^{(j)}\right) = \frac{1}{2}\left(\check{d}^{(j)} - h^{(j)} \exp\left(-\frac{\|\boldsymbol{u} - \boldsymbol{c}^{(j)}\|^2}{2\xi^{(j)}}\right)\right)^2, \tag{C.3}$$

and the ith component its partial derivative is given as

$$\begin{aligned}&\left[\frac{\partial F^{(\boldsymbol{c}^{(j)})}(\boldsymbol{c})}{\partial \boldsymbol{c}^{(j)}}\right]_i\\ &\quad = -\frac{h^{(j)}}{\xi^{(j)}}(u^{(i)} - c^{(i)}) \left(\check{d}^{(j)} - h^{(j)} \exp\left(-\frac{\|\boldsymbol{u} - \boldsymbol{c}\|^2}{2\xi^{(j)}}\right)\right)\\ &\quad \times \exp\left(-\frac{\|\boldsymbol{u} - \boldsymbol{c}\|^2}{2\xi^{(j)}}\right), \end{aligned} \tag{C.4}$$

where $u^{(i)}$ and $c^{(i)}$ denote the ith components of $\boldsymbol{u}$ and $\boldsymbol{c}$, respectively. By (C.4) and $\exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{c}\|^2}{2\xi^{(j)}}\right) = \delta^{(i)} \exp\left(-\frac{(u^{(i)}-c^{(i)})^2}{2\xi^{(j)}}\right)$, we can verify, for $\boldsymbol{c}, \tilde{\boldsymbol{c}} \in \mathbb{R}^L$, that

$$\begin{aligned}&\left|\left[\frac{\partial F^{(\boldsymbol{c}^{(j)})}(\boldsymbol{c})}{\partial \boldsymbol{c}^{(j)}}\right]_i - \left[\frac{\partial F^{(\boldsymbol{c}^{(j)})}(\tilde{\boldsymbol{c}})}{\partial \boldsymbol{c}^{(j)}}\right]_i\right|\\ &\leq \frac{\left|h^{(j)}\check{d}^{(j)}\delta^{(i)}\right|}{\left|\xi^{(j)}\right|} \left|(u^{(i)} - c^{(i)}) \exp\left(-\frac{(u^{(i)} - c^{(i)})^2}{2\xi^{(j)}}\right)\right.\\ &\quad \left. - (u^{(i)} - \tilde{c}^{(i)}) \exp\left(-\frac{(u^{(i)} - \tilde{c}^{(i)})^2}{2\xi^{(j)}}\right)\right|\\ &\quad + \frac{\left|h^{(j)}\delta^{(i)}\right|^2}{\left|\xi^{(j)}\right|} \left|(u^{(i)} - c^{(i)}) \exp\left(-\frac{(u^{(i)} - c^{(i)})^2}{\xi^{(j)}}\right)\right.\\ &\quad \left. - (u^{(i)} - \tilde{c}^{(i)}) \exp\left(-\frac{(u^{(i)} - \tilde{c}^{(i)})^2}{\xi^{(j)}}\right)\right|. \end{aligned} \tag{C.5}$$

Direct applications of Lemma 2 to the two terms in the right side of (C.5) for $c = (u^{(i)} - c^{(i)})$ and $c = \frac{(u^{(i)}-c^{(i)})}{\sqrt{2}}$ yields

$$\left|\left[\frac{\partial F^{(\boldsymbol{c}^{(j)})}(\boldsymbol{c})}{\partial \boldsymbol{c}^{(j)}}\right]_i - \left[\frac{\partial F^{(\boldsymbol{c}^{(j)})}(\tilde{\boldsymbol{c}})}{\partial \boldsymbol{c}^{(j)}}\right]_i\right|$$

$$
\begin{aligned}
&\leq \frac{\left|h^{(j)}\check{d}^{(j)}\delta^{(i)}\right|}{\left|\xi^{(j)}\right|}\left|c^{(i)}-\tilde{c}^{(i)}\right| + \frac{\left|h^{(j)}\delta^{(i)}\right|^2}{\left|\xi^{(j)}\right|}\left|c^{(i)}-\tilde{c}^{(i)}\right| \\
&= \frac{\delta^{(i)}\left|h^{(j)}\right|}{\xi^{(j)}}\left(\left|\check{d}^{(j)}\right| + \delta^{(i)}\left|h^{(j)}\right|\right)\left|c^{(i)}-\tilde{c}^{(i)}\right|. \qquad \text{(C.6)}
\end{aligned}
$$

By (C.6), we finally obtain the following bound:

$$
\begin{aligned}
&\left\| \frac{\partial F^{(\boldsymbol{c}^{(j)})}(\boldsymbol{c})}{\partial \boldsymbol{c}^{(j)}} - \frac{\partial F^{(\boldsymbol{c}^{(j)})}(\tilde{\boldsymbol{c}})}{\partial \boldsymbol{c}^{(j)}} \right\|^2 \\
&= \sum_{i=1}^{L}\left(\frac{\delta^{(i)}\left|h^{(j)}\right|}{\xi^{(j)}}\left(\left|\check{d}^{(j)}\right| + \delta^{(i)}\left|h^{(j)}\right|\right)\left|c^{(i)}-\tilde{c}^{(i)}\right|\right)^2 \\
&\leq \left(\delta^{*}\frac{\left|h^{(j)}\right|}{\xi^{(j)}}\left(\left|\check{d}^{(j)}\right| + \delta^{*}\left|h^{(j)}\right|\right)\right)^2 \sum_{i=1}^{L}\left|c^{(i)}-\tilde{c}^{(i)}\right|^2 \\
&= \left(\delta^{*}\frac{\left|h^{(j)}\right|}{\xi^{(j)}}\left(\left|\check{d}^{(j)}\right| + \delta^{*}\left|h^{(j)}\right|\right)\right)^2 \left\|\boldsymbol{c}-\tilde{\boldsymbol{c}}\right\|^2, \qquad \text{(C.7)}
\end{aligned}
$$

where $\delta^{*} := \max_{i=1,\cdots,L}\delta^{(i)}$.

REFERENCES

[1] M. Takizawa and M. Yukawa, "Online learning with self-tuned Gaussian kernels: Good kernel-initialization by multiscale screening," in *Proc. IEEE ICASSP*, May 2019, pp. 4863–4867.

[2] M. Grewal and A. Andrews, "Kalman filtering: Theory and applications," Tech. Rep., Jan. 1985.

[3] 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*, vol. 3068, I. Kadar, Ed. Bellingham, WA, USA: SPIE, pp. 182–193, 1997, doi: 10.1117/12.280797.

[4] Y. Shi, K. Sun, L. Huang, and Y. Li, "Online identification of permanent magnet flux based on extended Kalman filter for IPMSM drive with position sensorless control," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4169–4178, Nov. 2012.

[5] E. Laroche, E. Sedda, and C. Durieu, "Methodological insights for online estimation of induction motor parameters," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 5, pp. 1021–1028, Sep. 2008.

[6] E. Ghahremani and I. Kamwa, "Dynamic state estimation in power system by applying the extended Kalman filter with unknown inputs to phasor measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2556–2566, Nov. 2011.

[7] E. Ghahremani and I. Kamwa, "Online state estimation of a synchronous generator using unscented Kalman filter from phasor measurements units," *IEEE Trans. Energy Convers.*, vol. 26, no. 4, pp. 1099–1108, Dec. 2011.

[8] M. Partovibakhsh and G. Liu, "An adaptive unscented Kalman filtering approach for online estimation of model parameters and state-of-charge of lithium-ion batteries for autonomous mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 357–363, Jan. 2015.

[9] C. Wang, Z. Wang, L. Zhang, D. Cao, and D. G. Dorrell, "A vehicle rollover evaluation system based on enabling state and parameter estimation," *IEEE Trans. Ind. Informat.*, early access, Jul. 27, 2020, doi: 10.1109/TII.2020.3012003.

[10] X. Ding, Z. Wang, L. Zhang, and C. Wang, "Longitudinal vehicle speed estimation for four-wheel-independently-actuated electric vehicles based on multi-sensor fusion," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12797–12806, Nov. 2020.

[11] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Process. Mag.*, vol. 8, no. 3, pp. 10–26, Jul. 1991.

[12] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2, no. 3. Cambridge, MA, USA: MIT Press, 2006.

[13] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, Jun. 1991.

[14] G. Wahba, *Spline Models for Observational Data*, vol. 59. Philadelphia, PA, USA: SIAM, 1990.

[15] V. Vapnik, S. E. Golowich, and A. J. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 281–287.

[16] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering*. Hoboken, NJ, USA: Wiley, 2010.

[17] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 67–93, Mar. 2002.

[18] J. Racine, "An efficient cross-validation algorithm for window width selection for nonparametric kernel regression," *Commun. Statist.-Simul. Comput.*, vol. 22, no. 4, pp. 1107–1114, Jan. 1993.

[19] G. C. Cawley and N. L. C. Talbot, "Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers," *Pattern Recognit.*, vol. 36, no. 11, pp. 2585–2592, Nov. 2003.

[20] S. An, W. Liu, and S. Venkatesh, "Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression," *Pattern Recognit.*, vol. 40, no. 8, pp. 2154–2162, Aug. 2007.

[21] E. Herrmann, "Local bandwidth choice in kernel regression estimation," *J. Comput. Graph. Statist.*, vol. 6, no. 1, pp. 35–54, Mar. 1997.

[22] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. New York, NY, USA: Routledge, 1998.

[23] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *J. Amer. Stat. Assoc.*, vol. 91, no. 433, pp. 401–407, Mar. 1996.

[24] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[25] 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.

[26] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, Jul. 2008.

[27] B. Chen, S. Zhao, P. Zhu, S. Seth, and J. C. Príncipe, "Online efficient learning with quantized KLMS and L_1 regularization," in *Proc. Int. Joint Conf. Neural Netw.*, 2012, pp. 1–6.

[28] S. Van Vaerenbergh, M. Lazaro-Gredilla, and I. Santamaria, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1313–1326, Aug. 2012.

[29] M.-A. Takizawa and M. Yukawa, "Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4257–4269, Aug. 2015.

[30] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 6.

[31] M. Yukawa, "Multikernel adaptive filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4672–4682, Sep. 2012.

[32] M. Yukawa, "Adaptive learning in Cartesian product of reproducing kernel Hilbert spaces," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6037–6048, Nov. 2015.

[33] M. Kasparick, R. L. G. Cavalcante, S. Valentin, S. Stanczak, and M. Yukawa, "Kernel-based adaptive online reconstruction of coverage maps with side information," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5461–5473, Jul. 2016.

[34] D. A. Awan, R. L. G. Cavalcante, M. Yukawa, and S. Stanczak, "Detection for 5G-NOMA: An online adaptive machine learning approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[35] 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.

[36] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, no. 6, pp. 954–975, Nov. 1993.

[37] N. Sundararajan, P. Saratchandran, and Y. W. Lu, *Radial Basis Function Neural Networks With Sequential Learning: MRAN and Its Applications*, vol. 11. Singapore: World Scientific, 1999.

[38] J. Platt, "A resourse-allocating network for function interpolation," *Neural Comput.*, vol. 3, no. 2, pp. 213–225, 1991.

[39] N. B. Karayiannis and G. W. Mi, "Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques," *IEEE Trans. Neural Netw.*, vol. 8, no. 6, pp. 1492–1506, Nov. 1997.

[40] 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.

[41] W. Gao, J. Chen, C. Richard, and J. Huang, "Online dictionary learning for kernel LMS," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2765–2777, Jun. 2014.

[42] M.-A. Takizawa and M. Yukawa, "Efficient dictionary-refining kernel adaptive filter with fundamental insights," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4337–4350, Aug. 2016.

[43] D. A. Awan, R. L. G. Cavalcante, M. Yukawa, and S. Stanczak, *Adaptive Learning for Symbol Detection: A Reproducing Kernel Hilbert Space Approach* (Machine Learning for Future Wireless Communications). New York, NY, USA: Wiley, 2020, ch. 11, pp. 197–211.

[44] M. Yukawa and R. ishii, "Online model selection and learning by multikernel adaptive filtering," in *Proc. EUSIPCO*, 2013, pp. 1–5.

[45] O. Toda and M. Yukawa, "Online model-selection and learning for nonlinear estimation based on multikernel adaptive filtering," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E100.A, no. 1, pp. 236–250, 2017.

[46] B. Chen, J. Liang, N. Zheng, and J. C. Principe, "Kernel least mean square with adaptive kernel size," *Neurocomputing*, vol. 191, pp. 95–105, May 2013.

[47] T. Wada and T. Tanaka, "Doubly adaptive kernel adaptive filtering," in *Proc. APSIPA*, 2017, pp. 904–909, Paper tA-P3.6.

[48] T. Wada, K. Fukumori, and T. Tanaka, "Dictionary learning for Gaussian kernel adaptive filtering with variable kernel center and width," in *Proc. IEEE ICASSP*, Apr. 2018, pp. 2766–2770.

[49] C. Saide, R. Lengelle, P. Honeine, C. Richard, and R. Achkar, "Dictionary adaptation for online prediction of time series data with kernels," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Aug. 2012, pp. 604–607.

[50] C. Saide, R. Lengelle, P. Honeine, and R. Achkar, "Online kernel adaptive algorithms with dictionary adaptation for MIMO models," *IEEE Signal Process. Lett.*, vol. 20, no. 5, pp. 535–538, May 2013.

[51] H. Chen, Y. Gong, X. Hong, and S. Chen, "A fast adaptive tunable RBF network for nonstationary systems," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2683–2692, Dec. 2016.

[52] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York, NY, USA: Springer-Verlag, 2010.

[53] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[54] J. Lesouple, T. Robert, M. Sahmoudi, J.-Y. Tourneret, and W. Vigneau, "Multipath mitigation for GNSS positioning in an urban environment using sparse estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1316–1328, Apr. 2019.

[55] D. Meng, X. Wang, M. Huang, L. Wan, and B. Zhang, "Robust weighted subspace fitting for DOA estimation via block sparse recovery," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 563–567, Mar. 2020.

[56] M.-A. Takizawa and M. Yukawa, "Steepening squared error function facilitates online adaptation of Gaussian scales," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5450–5454.

[57] M. Yukawa, Y. Tawara, M. Yamagishi, and I. Yamada, "Sparsity-aware adaptive filters based on Lp-norm inspired soft-thresholding technique," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 2749–2752.

[58] M. Yukawa, "On use of multiple kernels in adaptive learning—Extended reproducing kernel Hilbert space with Cartesian product," in *Proc. IEICE Signal Process. Symp.*, Nov. 2010, pp. 59–64.

[59] M. Yukawa, "Nonlinear adaptive filtering techniques with multiple kernels," in *Proc. EUSIPCO*, 2011, pp. 136–140.

[60] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. IEEE ICASSP*, Mar. 2010, pp. 3734–3737.

[61] P. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Cham, Switzerland: Springer, Jan. 2008.

[62] W. A. Martins, M. V. S. Lima, P. S. R. Diniz, and T. N. Ferreira, "Optimal constraint vectors for set-membership affine projection algorithms," *Signal Process.*, vol. 134, pp. 285–294, May 2017.

[63] A. Flores and R. C. de Lamare, "Set-membership kernel adaptive algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2676–2680.

[64] A. V. Malipatil, Y.-F. Huang, S. Andra, and K. Bennett, "Kernelized set-membership approach to nonlinear adaptive filtering," in *Proc. IEEE ICASSP*, Mar. 2005, pp. 149–152.

[65] K. Chen, S. Werner, A. Kuh, and Y.-F. Huang, "Nonlinear adaptive filtering with kernel set-membership approach," *IEEE Trans. Signal Process.*, vol. 68, pp. 1515–1528, 2020.

[66] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer, 2011.

[67] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*. Hoboken, NJ, USA: Wiley, 1983.

[68] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. New York, NY, USA: Springer, 2013.

[69] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.

[70] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.

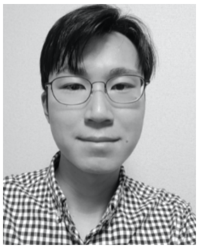
[71] H. Fan, Q. Song, and S. B. Shrestha, "Kernel online learning with adaptive kernel width," *Neurocomputing*, vol. 175, pp. 233–242, Jan. 2016.

[72] A. S. Weigend and E. N. A. Gershenfeld, Eds., *Time Series Prediction: Forecasting the Future and Understanding the Past Reading*. Boston, MA, USA: Addsion-Wesly, 1994.

[73] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, Jul. 1977.

[74] T. Ergen and S. Serdar Kozat, "Efficient online learning algorithms based on LSTM neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3772–3783, Aug. 2018.

[75] M. Ohnishi and M. Yukawa, "Online nonlinear estimation via iterative L^2-space projections: Reproducing kernel of subspace," *IEEE Trans. Signal Process.*, vol. 66, no. 15, pp. 4050–4064, Aug. 2018.



MASA-AKI TAKIZAWA (Student Member, IEEE) received the B.E. degree from Niigata University, in 2013, and the M.E. degree from Keio University, Japan, in 2015, where he is currently pursuing the Ph.D. degree in electronics and electrical engineering. His research interest includes adaptive signal processing. He was a recipient of the Research Fellowship of the Japan Society for the Promotion of Science in April 2018.

MASAHIRO YUKAWA (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees from Tokyo Institute of Technology, in 2002, 2004, and 2006, respectively. He studied at the University of York, U.K., as a Visiting Researcher for six months, and at the Technical University of Munich, Germany, as a Guest Researcher for four months, and worked at RIKEN, Japan, as a Special Postdoctoral Researcher for three years, and at the Niigata University, Japan, as an Associate Professor for another three years. In 2016, he studied with the Machine Learning Group, Technical University of Berlin, Germany. He is currently an Associate Professor with the Department of Electronics and Electrical Engineering, Keio University, Japan. His research interests include mathematical adaptive signal processing, convex/sparse optimization, and machine learning. He is a member of the IEICE. He was a recipient of the Research Fellowship of the Japan Society for the Promotion of Science (JSPS), from April 2005 to March 2007. He received the Excellent Paper Award and the Young Researcher Award from the IEICE, in 2006 and 2010, respectively, the Yasujiro Niwa Outstanding Paper Award in 2007, the Ericsson Young Scientist Award, in 2009, TELECOM System Technology Award, in 2014, the Young Scientists' Prize, the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, in 2014, the KDDI Foundation Research Award, in 2015, and the Funai Foundation Academic Award, in 2016. He served as an Associate Editor for the *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* (2009–2013), the *Multidimensional Systems and Signal Processing* (Springer) (2012–2016), and the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2015–2019).