Nonlinear Adaptive Filtering with Kernel Set-Membership Approach

Kewei Chen*, Stefan Werner†, Anthony Kuh‡ and Yih-Fang Huang*

***Abstract*—This paper develops nonlinear kernel adaptive filtering algorithms based on the set-membership filtering (SMF) framework. The set-membership-based filtering approach is distinct from the conventional adaptive filtering approaches in that it aims for the filtering error being bounded in magnitude, as opposed to seeking to minimize the time average or ensemble average of the squared errors. The proposed kernel SMF algorithms feature selective updates of their parameter estimates by making discerning use of the input data, and selective increase of the dimension in the kernel expansion. These result in less computational cost and faster tracking without compromising the mean-squared error performance. We show, through convergence analysis, that the sequences of parameter estimates of our proposed algorithms are convergent, and the filtering error is asymptotically upper bounded in magnitude. Simulations are performed which show clearly the advantages of the proposed algorithms in terms of lower computational complexity, reduced dictionary size, and steady-state mean-squared errors comparable to existing algorithms.**

***Index Terms*—Adaptive filters, set-membership filtering, kernel methods, nonlinear systems.**

I. Introduction

Adaptive filtering algorithms that adjust filter parameters according to the changes in signal characteristics are widely used in dynamical systems and signal processing applications. Traditional methodologies such as recursive least-squares (RLS) and least-mean squares (LMS), as well as their numerous variations, have been studied extensively over the past many decades; see, e.g., [1]–[3]. The RLS and LMS algorithms have been derived to minimize, respectively, the time average and the ensemble average of the squared filtering errors. To date, applications of those conventional adaptive filtering techniques still focus mostly on linear-in-parameter filters, even though nonlinear dynamical systems and signal processing problems arise in many practical situations, e.g., nonlinear communication systems [4] and nonlinear time series prediction [5].

Kernel methods based on reproducing kernel Hilbert space (RKHS) [6], [7] have gained much popularity in extending linear algorithms to nonlinear ones, due to the mathematical simplicity of RKHS. A Mercer kernel applied to a pair of input vectors can evaluate their inner product in the corresponding high-dimensional RKHS without the explicit knowledge of the mapping from the original space to the RKHS. Those kernel methods have been employed to derive kernel RLS (KRLS) algorithm, see, e.g., [8]; kernel LMS (KLMS) algorithm [9], [10] and kernel normalized LMS (KNLMS) algorithm [11]. In machine learning, kernel methods have played an important role in extending many algorithms, including the celebrated support-vector machines [12] and kernel principal component analysis [13]. While those algorithms are often developed in batch/offline mode, there has also been a considerable research on the design of kernel algorithms that work with streaming data [8]–[11], [14]–[18]. Particularly, in the framework of projections onto convex sets, a comprehensive review of various kernel adaptive learning algorithms for online system identification and modeling is provided in [19]. In another direction, techniques such as Nyström method [20] and random Fourier features approach [21] have been applied to approximate kernel functions with the purpose of speeding up kernel adaptive filtering algorithms; see, e.g., [22], [23].

This paper presents two nonlinear adaptive filtering algorithms, namely, K-BEACON and K-SM-NLMS, derived from the principles of set-membership filtering (SMF). The SMF is an adaptive filtering paradigm that features data-dependent selective update of the filter parameters [24], [25]. This feature, which is derived from the key objective that the filtering error is bounded in magnitude, is distinct from conventional algorithms such as RLS and LMS that update those parameters continuously, regardless of the benefits of such updates. In SMF, if the filtering error is less than the presumed magnitude bound, no update of the filter parameters is needed. Checking on whether or not the filtering error exceeds the presumed magnitude bound is sometimes termed *innovation check* in the SMF literature, see, e.g., [26]. The innovation check enables the SMF algorithms to update filter parameters only when there is sufficient *innovation*, which is measured by the filtering error. In this way, the updates of filter parameters in SMF are *event-triggered*, as opposed to *time-triggered* like those in RLS and LMS. It is important to note that while using only a fraction of the data to update the parameter estimates, the SMF algorithms perform comparably to their counterparts of traditional algorithms, namely, RLS and LMS, as measured by the steady-state mean-squared errors (MSE) as well as speed of convergence. The SMF's selective update feature offers opportunities for further exploration, see, e.g., [27]–[29].

To date, however, most of the SMF algorithms have been developed using linear models, see, e.g., [24], [25], [27]–[32]. Kernel set-membership NLMS algorithms have been proposed

K. Chen and Y.-F. Huang are with the Department of Electrical Engineering, University of Notre Dame, IN 46556, USA (e-mail: {kchen6, huang}@nd.edu). S. Werner is with the Department of Electronic Systems, NTNU-Norwegian University of Science and Technology, Trondheim 7491, Norway (e-mail: stefan.werner@ntnu.no). A. Kuh is with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96882, USA (e-mail: kuh@hawaii.edu). The work of S. Werner was supported, in part, by the Research Council of Norway and Academy of Finland under Grant 296849.

in [33], [34]. One of the challenges of those algorithms is to control kernel expansion, i.e., the increase of dictionary dimension. In [33], the approximate linear dependence (ALD) criterion was used for dictionary sparsification, which has a quadratic time complexity. In [34], no explicit sparsification rule was embedded except for the innovation check, which would likely yield excessively large dictionary. More detailed discussions on dictionary sparsification are given in Section II, as well as in Section V with simulation comparison. Another kernel SM algorithm was proposed in [35], which was derived as a nonlinear extension of [25]. Our work here focuses on employing kernel methods for two particular SMF algorithms, namely, BEACON [31] and SM-NLMS [30] to derive, respectively, the K-BEACON algorithm and the K-SM-NLMS algorithm. The main features of the proposed algorithms are:

- Data-dependent selective update of filter parameter estimates without compromising the performance in steady-state MSE, compared to other kernel algorithms.
- Less computation, better tracking performance, and sparser kernel expansion, thus smaller model order.
- Convergent parameter estimates sequence and asymptotically upper bounded error magnitude.
- Flexible trade-off between the computational complexity and the bound on the asymptotic filtering error.

The rest of this paper is organized as follows: Section II provides some fundamentals on kernel adaptive filtering. Section III presents the derivations of the K-BEACON algorithm and the K-SM-NLMS algorithm, which feature an innovation check as do all SMF algorithms. In the proposed algorithms, when the innovation check warrants an update of filter parameters, a coherence-based sparsification rule follows to control the model dimension (termed *coherence check*). The innovation check and coherence check together result in less computation complexity and smaller model dimensions, comparing to other existing algorithms. Section IV presents convergence analysis of our proposed algorithms that shows that the sequence of parameter estimates is convergent and the filtering error is asymptotically upper bounded in magnitude. The simulation results presented in Section V show that the proposed algorithms yield sparser kernel expansion (thus smaller, in some cases significantly smaller, dictionary dimension), lower computational cost, and comparable steady-state MSE. More interestingly, the proposed K-BEACON algorithm exhibits better tracking performance than the KRLS algorithm when the system model is time-varying. Conclusions are given in Section VI.

Notation: The set of real numbers is denoted by $\mathbb{R}$. Scalars, vectors and matrices are denoted by lowercase letters, lowercase boldface letters and uppercase boldface letters, respectively. The transpose operator is denoted by $(.)^{\mathrm{T}}$. The notation $\mathbf{A} \preceq \mathbf{B}$ means $\mathbf{B} - \mathbf{A}$ is a positive semidefinite matrix. The norm in an Euclidean space is denoted by $||.||_2$ while that of an RKHS is denoted by $||.||_{\mathcal{H}}$. The notation $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{A} \subset \mathcal{B}$ means $\mathcal{A}$ is a subset and strict subset of $\mathcal{B}$ respectively. The intersection of two sets $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} \cap \mathcal{B}$.

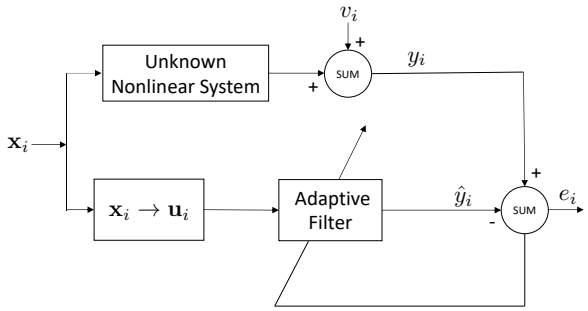

Fig. 1: System identification with kernel adaptive filtering. The step $\mathbf{x}_i \rightarrow \mathbf{u}_i$ represents kernel expansion of $\mathbf{x}_i$.

II. Kernel Adaptive Filtering

Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be two points in a space $\mathcal{X}$ that is a compact subspace of the L-dimensional Euclidean space $\mathbb{R}^L$. A reproducing kernel, $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, that maps from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$ is given by [6]

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \kappa(., \mathbf{x}_i), \kappa(., \mathbf{x}_j) \rangle_{\mathcal{H}},$$

where $\mathcal{H}$ is the induced reproducing kernel Hilbert space (RKHS) and $\langle ., . \rangle_{\mathcal{H}}$ is the corresponding inner product. Here, $\kappa(., \mathbf{x}_i)$ is a function in $\mathcal{H}$ called *representer of evaluation* at $\mathbf{x}_i$. Popular kernel choices include Gaussian kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\xi^2}\right),$$

and Laplacian kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2}{\xi}\right),$$

where ξ is the kernel bandwidth.

The problem considered here is the following: Given a set of data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, use the kernel method to find a function $h(\cdot)$ of $\mathcal{H}$ to minimize the sum of n squared errors. Specifically

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \left(y_i - h(\mathbf{x}_i)\right)^2, \tag{1}$$

where y_i is the system output and $h(\mathbf{x}_i)$ is the estimated output. The representer theorem [36], [37] shows that the function $h(\cdot)$ of (1) can be expressed as a kernel expansion in terms of the available data, i.e.,

$$h(\cdot) = \sum_{i=1}^{n} w_i \kappa(\cdot, \mathbf{x}_i). \tag{2}$$

Then (1) can be rewritten as $\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{K}\mathbf{w}||_2^2$, where $\mathbf{y} = [y_1 \cdots y_n]^{\mathrm{T}}$, $\mathbf{w} = [w_1 \cdots w_n]^{\mathrm{T}}$, and $\mathbf{K}$ is the Gram matrix with $[\mathbf{K}_{ij}] = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. The weights $\mathbf{w}$ can be found by solving $\mathbf{K}\mathbf{w} = \mathbf{y}$.

This formulation, however, is not applicable to problems of online adaptive filtering as depicted in Fig. 1, where data become available sequentially. For the online case, at

time instant i, the estimation of y_i (denoted by $\hat{y}_i$), given $\{(\mathbf{x}_j, y_j)\}_{j=1}^{i-1} \bigcup \{\mathbf{x}_i\}$ can be expressed as

$$\hat{y}_i = h_i(\mathbf{x}_i) = \sum_{j=1}^{i} w_i \kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{3}$$

According to (3), the model order of the problem grows every time when a new data point arrives. This presents a challenge for *real-time* implementation of online kernel algorithms. Therefore, there is a need to control the model order such that it would not keep increasing as the number of data points increases. Specifically, in the following form

$$h_i(\mathbf{x}_i) = \sum_{j=1}^{m(i)} w_j \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_j}), \tag{4}$$

where $\mathcal{J}_i = \{\alpha_j\}_{j=1}^{m(i)}$ is a subset of $\{j\}_{j=1}^{i}$, the dimension $m(i)$ of the dictionary $\mathcal{D}_i = \{\kappa(\cdot, \mathbf{x}_{\alpha_j})\}_{j=1}^{m(i)}$ should stop increasing at some point in time.

Engel *et al*. proposed a kernel recursive least-squares (KRLS) algorithm [8] that adopted an approximate linear dependence (ALD) criterion as a sparsification rule to control the model dimension and used RLS algorithm to update the weights. At iteration i, the ALD-based sparsification rule suggests inserting $\kappa(\cdot, \mathbf{x}_i)$ into the dictionary only if it is not approximately linearly dependent on the existing dictionary, i.e.,

$$\min_{\mathbf{a}} \left\| \sum_{j=1}^{m(i-1)} a_j \kappa(., \mathbf{x}_{\alpha_j}) - \kappa(., \mathbf{x}_i) \right\|_{\mathcal{H}}^2 \geq \nu, \tag{5}$$

where $\mathbf{a} = [a_1\, a_2 \cdots a_{m(i-1)}]$ and ν is a pre-specified threshold. The major criticism on this approach is that it involves costly computations. To reduce the computational complexity at each iteration, Richard *et al*. [11] proposed the coherence-based sparsification rule to control the model dimension. Specifically, at time instant i, the coherence-based sparsification rule requires inserting $\kappa(\cdot, \mathbf{x}_i)$ into the dictionary only if the coherence is not greater than a pre-specified threshold μ, i.e.,

$$\max_{\alpha_j \in \mathcal{J}_{i-1}} |\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_j})| \leq \mu. \tag{6}$$

It was shown that the model dimension under the coherence-based sparsification rule remains finite as i goes to infinity. Along this direction, there has been considerable research on constructing and refining the dictionary in an adaptive fashion; see, e.g., [38]–[43]. In this paper, we adopt the coherence-based sparsification rule for dictionary construction, and focus on the derivations and analysis of kernel SMF algorithms.

III. Kernel SMF Algorithms

In this section, we employ the kernel method with the SMF principles to derive two kernel SMF algorithms. Consider the following nonlinear model that characterizes the input-output relationship

$$y_i = f(\mathbf{x}_i) + v_i, \tag{7}$$

where $y_i \in \mathbb{R}$ and $\mathbf{x}_i \in \mathbb{R}^{L\times 1}$ represent, respectively, the output signal and input signal vector at time instant i. Also, $v_i \in \mathbb{R}$ denotes the model uncertainty (noise). Similarly to the linear SMF algorithms, see, e.g., [25], [31], our goal here is to find a set of filter parameters such that, at any time instant i, the filter output error is upper bounded in magnitude, i.e.,

$$\mathcal{C}_i = \left\{ \mathbf{w} \in \mathbb{R}^{m(i)} : (y_i - \mathbf{w}^{\mathrm{T}} \mathbf{u}_i)^2 \leq \gamma^2 \right\}. \tag{8}$$

In (8), $\mathbf{u}_i$ is the vector consisting of kernel evaluations between $\mathbf{x}_i$ and each element in the current dictionary $\mathcal{D}_i = \{\kappa(\cdot, \mathbf{x}_{\alpha_j})\}_{j=1}^{m(i)}$. Specifically,

$$\mathbf{u}_i = \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i)}})\right]^{\mathrm{T}}. \tag{9}$$

Also, $\mathcal{C}_i$ is called the *constraint set* and it is a degenerate ellipsoid in the parameter space; while $\gamma > 0$ is a prescribed error bound.

Given a sequence of data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, if γ is properly chosen so that there exists a set of parameter estimates that satisfy (8) for all i, then that set is the intersection of all the constraint sets, namely,

$$\Omega_n = \cap_{i=1}^{n} \mathcal{C}_i = \Omega_{n-1} \cap \mathcal{C}_n. \tag{10}$$

The set Ω_n in the above equation is termed the *exact membership set*. Clearly, every point in the exact membership set is a legitimate parameter estimate, for it is consistent with the presumed model and the received data. We note that $\{\Omega_i\}_{i=1}^{n}$ is a sequence of monotone non-increasing sets, i.e., $\Omega_i \subseteq \Omega_{i-j}$ for any $1 \leq j \leq i$. Intuitively, if the data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ are rich enough, Ω_i will be small when i grows large, making it likely that $\Omega_{i-1} \subset \mathcal{C}_i$, thus $\Omega_i = \Omega_{i-1}$, as such, no update on the parameter estimates is needed at iteration i. Checking on whether or not an update is needed constitutes the basis of the so-called *innovation check*. This leads to the important feature of *data-dependent selective update* for all SMF algorithms.

A. K-BEACON

Conceptually, one would prefer to obtain a simple analytical expression of the exact membership set Ω_i for all i. In practice, however, it is usually more convenient to find some analytically tractable outer bounding set for Ω_i. Since the constraint set, $\mathcal{C}_i$, is a degenerate ellipsoid, a good candidate for the outer bounding set is an ellipsoid. In this section, we propose a kernel ellipsoidal-bounding SMF algorithm, namely, Kernel Bounding Ellipsoidal Adaptive CONstrained (K-BEACON) algorithm, which is an extension of the (linear) BEACON algorithm [31]. In the BEACON algorithm, at each time instant i, an ellipsoid $\mathcal{E}_i$ is derived to outer bound (in some optimum sense) Ω_i, and the center of the ellipsoid is used as the parameter estimate at that time instant. The recursive formulas of the bounding ellipsoid are derived as follows.

Let $\mathcal{E}_{i-1}$ be an optimum bounding ellipsoid, at time instant $i-1$, which outer bounds the exact membership set Ω_{i-1}, i.e., $\mathcal{E}_{i-1} \supset \Omega_{i-1}$. This bounding ellipsoid is formulated as

$$\mathcal{E}_{i-1} = \left\{ \mathbf{w} \in \mathbb{R}^{m(i-1)} : (\mathbf{w} - \mathbf{w}_{i-1})^{\mathrm{T}} \mathbf{P}_{i-1}^{-1} (\mathbf{w} - \mathbf{w}_{i-1}) \leq \sigma_{i-1}^2 \right\}, \tag{11}$$

where $\mathbf{w}_{i-1}$ is the center of the ellipsoid. The positive definite matrix $\mathbf{P}_{i-1}$ together with σ_{i-1} characterize the size (i.e., the lengths of the semi-axes) of the ellipsoid.

When the new data pair $(\mathbf{x}_i, y_i)$ becomes available at time instant i, the algorithm will decide:

1) whether or not the new data pair $(\mathbf{x}_i, y_i)$ contains sufficient innovation to warrant an update for the parameter estimates, i.e., innovation check; and
2) whether or not $\kappa(\cdot, \mathbf{x}_i)$ should be inserted into the existing dictionary. This is determined by the coherence-based sparsification rule as shown in (6), namely, coherence check.

In our proposed K-BEACON algorithm, the innovation check is conducted first, for there is no need to include it in the dictionary if the new data does not provide innovation. If the new data pair $(\mathbf{x}_i, y_i)$ is determined to be *innovative* enough, the parameter estimates will be updated. However, before calculating the updated parameter estimates, the proposed algorithm performs the coherence check to see if the dictionary's dimension needs to be increased, leading to two cases in the updating of the parameter estimates: (a) the dictionary remains the same; and (b) the dictionary dimension increases.

In both cases, updating the parameter estimates amounts to updating the optimal bounding ellipsoids. The process can be summarized as follows: Given $\mathcal{E}_{i-1}$, and the constraint set $\mathcal{C}_i$, the objective is to find an *optimal* ellipsoid $\mathcal{E}_i$ that tightly outer bounds the intersection of $\mathcal{E}_{i-1}$ and $\mathcal{C}_i$, i.e.,

$$\mathcal{E}_i \supset \mathcal{E}_{i-1} \cap \mathcal{C}_i. \tag{12}$$

The optimality of the bounding ellipsoid is defined here as a tight bounding ellipsoid with minimum σ_i^2 (see, (11) with subscript i). The resulting recursive formulas are summarized in the following theorem.

Theorem 1: Let $\mathcal{E}_{i-1}$ be the optimum bounding ellipsoid at iteration $i-1$, as formulated in (11). At time instant i, the following recursive expressions for $\mathbf{w}_i$, $\mathbf{P}_i$, and σ_i^2 defining an ellipsoid $\mathcal{E}_i$ that tightly outer bounds the intersection $\mathcal{E}_{i-1} \cap \mathcal{C}_i$ are obtained through a linear combination of (11) and (8), specifically, $\mathcal{E}_i = \mathcal{E}_{i-1} + \lambda_i \mathcal{C}_i$, with $0 \le \lambda_i < \infty$:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \lambda_i \mathbf{P}_i \mathbf{u}_i e_i \tag{13a}$$

$$\mathbf{P}_i^{-1} = \mathbf{P}_{i-1}^{-1} + \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}} \tag{13b}$$

$$e_i = y_i - \mathbf{w}_{i-1}^{\mathrm{T}} \mathbf{u}_i \tag{13c}$$

$$\sigma_i^2 = \sigma_{i-1}^2 + \lambda_i \gamma^2 - \frac{\lambda_i e_i^2}{1 + \lambda_i G_i} \tag{13d}$$

$$G_i = \mathbf{u}_i^{\mathrm{T}} \mathbf{P}_{i-1} \mathbf{u}_i. \tag{13e}$$

The optimum bounding ellipsoid is obtained by minimizing σ_i^2 with respect to λ_i, yielding the optimum λ_i^* as follows:

$$\lambda_i^* = \begin{cases} \frac{1}{G_i}\left(\frac{|e_i|}{\gamma} - 1\right), & \text{if } |e_i| > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Proof: See Appendix A. ∎

Notice that, in the innovation check at any iteration i, if $|e_i| \le \gamma$, then the optimal $\lambda_i^* = 0$, which results in $\mathbf{w}_i = \mathbf{w}_{i-1}, \mathbf{P}_i = \mathbf{P}_{i-1}$ and $\sigma_i^2 = \sigma_{i-1}^2$. Thus no update is needed and no coherence check ensues, either. In essence, the data-dependent selective update feature of SMF algorithms reduces computation cost by skipping the parameters update at such iterations. Graphically, the data-dependent selective update feature is demonstrated in Fig. 2 using a two-dimensional example. Note that if the previous parameter estimate belongs to the new constraint set $\mathcal{C}_i$, i.e., $\mathbf{w}_{i-1} \in \mathcal{C}_i$, then $\mathcal{C}_i$ is discarded and no update on the parameter estimates is needed.

Recall that the algorithm implements the coherence criterion to check if $\kappa(\cdot, \mathbf{x}_i)$ should be inserted into the dictionary only when the innovation check passes, which necessitates an update for the parameter estimates. Depending on whether or not the dimension of the dictionary increases, the algorithm should have two cases for the weights update, which are discussed separately below.

1) Case 1: Dimension Remains The Same: When the coherence-based sparsification rule does not suggest inserting $\kappa(\cdot, \mathbf{x}_i)$ into the existing dictionary, the dimension of the dictionary remains the same. The kernelized input vector in the constraint set $\mathcal{C}_i$ is expressed as

$$\mathbf{u}_i = \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i)}})\right]^{\mathrm{T}}, \tag{15}$$

where $m(i) = m(i-1)$. By employing the matrix inversion lemma, the computation of $\mathbf{P}_i$ in (13b) can be simplified as

$$\mathbf{P}_i = \mathbf{P}_{i-1} - \frac{\lambda_i^* \mathbf{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}} \mathbf{P}_{i-1}}{1 + \lambda_i^* G_i}. \tag{16}$$

Accordingly, $\mathbf{w}_i$ in (13a) is given by

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\lambda_i^* e_i \mathbf{P}_{i-1} \mathbf{u}_i}{1 + \lambda_i^* G_i}. \tag{17}$$

2) Case 2: Dimension Increases: When the coherence-based sparsification rule suggests inserting $\kappa(\cdot, \mathbf{x}_i)$ into the existing dictionary, then the kernelized input vector in the constraint set $\mathcal{C}_i$ is given by

$$\begin{aligned} \mathbf{u}_i &= \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i-1)}}) \ \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i)}})\right]^{\mathrm{T}} \\ &= \begin{bmatrix} \tilde{\mathbf{u}}_i^{\mathrm{T}} & u_{ai} \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{18}$$

where $\tilde{\mathbf{u}}_i = \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i-1)}})\right]^{\mathrm{T}}$, $u_{ai} = \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i)}})$, $m(i) = m(i-1) + 1$, and $\alpha_{m(i)} = i$. Before applying the recursive update formulas, the dimensions of $\mathbf{w}_{i-1}$ and $\mathbf{P}_{i-1}$ should be increased by one to match the dimension of $\mathbf{u}_i$. Specifically, we use

$$\tilde{\mathbf{w}}_{i-1} = \begin{bmatrix} \mathbf{w}_{i-1}^{\mathrm{T}} & 0 \end{bmatrix}^{\mathrm{T}}, \tag{19}$$

and

$$\tilde{\mathbf{P}}_{i-1} = \begin{bmatrix} \mathbf{P}_{i-1} & \mathbf{0}_{m(i-1)\times 1} \\ \mathbf{0}_{m(i-1)\times 1}^{\mathrm{T}} & 1 \end{bmatrix}, \tag{20}$$

where $\mathbf{0}_{m(i-1)\times 1}$ is the zero vector with dimension $m(i-1)\times 1$. In essence, the above equations are implemented to add a new semi-axis to the previous optimal bounding ellipsoid, resulting in an ellipsoid in the higher dimension. The recursive updating formulas can now be applied to find an ellipsoid $\mathcal{E}_i$ that tightly outer bounds the intersection between the enlarged $\mathcal{E}_{i-1}$ and $\mathcal{C}_i$ with minimum σ_i^2. The expressions for $\mathbf{w}_i$, $\mathbf{P}_i$ and σ_i^2 of $\mathcal{E}_i$ are given by (13) after replacing $\mathbf{w}_{i-1}$ and $\mathbf{P}_{i-1}$ with $\tilde{\mathbf{w}}_{i-1}$ and $\tilde{\mathbf{P}}_{i-1}$ respectively.

Algorithm 1 K-BEACON algorithm

Initialization
  Initialize w_0, P_0, γ, μ
  Insert $\kappa(\cdot, \mathbf{x}_1)$ into the dictionary, set $m(1) = 1$
  Denote the dictionary as $\mathcal{D}_1 = \{\kappa(\cdot, \mathbf{x}_{\alpha_1})\}$
For $i > 1$, repeat
  Given $(\mathbf{x}_i, y_i)$
  Compute $\tilde{\mathbf{u}}_i = \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i-1)}})\right]^{\mathrm{T}}$
  Compute $e_i = y_i - \mathbf{w}_{i-1}^{\mathrm{T}} \tilde{\mathbf{u}}_i$
  if $|e_i| > \gamma$ ▷ Innovation check
    if $\max_{j=1,\cdots,m(i-1)} |\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_j})| > \mu$ ▷ Case 1
      $m(i) = m(i-1)$
      $\mathbf{u}_i = \tilde{\mathbf{u}}_i$
      Update $\mathbf{P}_i$ using (16)
      Update $\mathbf{w}_i$ using (17)
    else ▷ Case 2
      $m(i) = m(i-1) + 1$
      Insert $\kappa(\cdot, \mathbf{x}_i)$ into the dictionary
      Denote $\kappa(\cdot, \mathbf{x}_i)$ as $\kappa(\cdot, \mathbf{x}_{\alpha_{m(i)}})$
      Update $\mathbf{P}_i$ using (22)
      Update $\mathbf{w}_i$ using (23)
    end
  end

The computation of the matrix $\mathbf{P}_i$ in (13b) is now obtained by substituting (18) and (20) to (13b):

$$\mathbf{P}_i^{-1} = \begin{bmatrix} \mathbf{P}_{i-1}^{-1} & \mathbf{0}_{m(i-1)\times 1} \\ \mathbf{0}_{m(i-1)\times 1}^{\mathrm{T}} & 1 \end{bmatrix} + \lambda_i^* \begin{bmatrix} \tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_i^{\mathrm{T}} & u_{ai}\tilde{\mathbf{u}}_i \\ u_{ai}\tilde{\mathbf{u}}_i^{\mathrm{T}} & u_{ai}^2 \end{bmatrix}. \tag{21}$$

Applying the matrix inversion lemma to (21) gives

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{p}_1 \\ \mathbf{p}_1^{\mathrm{T}} & r \end{bmatrix}, \tag{22}$$

where

$$\mathbf{P}_{11} = \mathbf{P}_{i-1} - \frac{\lambda_i^* \mathbf{P}_{i-1}\tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_i^{\mathrm{T}} \mathbf{P}_{i-1}}{1 + \lambda_i^*(u_{ai}^2 + \tilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i)}$$
$$\mathbf{p}_1 = -\frac{\lambda_i^* u_{ai}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i}{1 + \lambda_i^*(u_{ai}^2 + \tilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i)}$$
$$r = \frac{1 + \lambda_i^* \tilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i}{1 + \lambda_i^*(u_{ai}^2 + \tilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i)}.$$

Accordingly, the parameter estimate $\mathbf{w}_i$ in (13a) is given by

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{w}_{i-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{\lambda_i^* e_i \mathbf{P}_{i-1}\tilde{\mathbf{u}}_i}{1+\lambda_i^*(u_{ai}^2 + \tilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i)} \\ \frac{\lambda_i^* e_i u_{ai}}{1+\lambda_i^*(u_{ai}^2 + \tilde{\mathbf{u}}_i^{\mathrm{T}}\mathbf{P}_{i-1}\tilde{\mathbf{u}}_i)} \end{bmatrix}. \tag{23}$$

The pseudocode that summarizes the K-BEACON algorithm is shown in the table labeled as Algorithm 1.

B. K-SM-NLMS

An alternative approach to analytically track the exact membership set Ω_i is to use a bounding spheroid $\mathcal{S}_i$ that outer bounds Ω_i and use the center of the spheroid as the parameter estimate at time instant i. The recursive formulas of the bounding spheroid are derived as follows.

Let $\mathcal{S}_{i-1}$ be an optimum spheroid that, at time instant $i-1$, outer bounds the exact membership set Ω_{i-1}, i.e., $\mathcal{S}_{i-1} \supset \Omega_{i-1}$. This bounding spheroid can be formulated as

$$\mathcal{S}_{i-1} = \{\mathbf{w} \in \mathbb{R}^{m(i-1)} : ||\mathbf{w} - \mathbf{w}_{i-1}||_2^2 \leq \sigma_{i-1}^2\}, \tag{24}$$

where $\mathbf{w}_{i-1}$ is the center of the spheroid. Then, given $\mathcal{S}_{i-1}$ and the constraint set $\mathcal{C}_i$ obtained at time i, we shall find an optimal spheroid S_i that tightly outer bounds $\mathcal{S}_{i-1} \cap \mathcal{C}_i$, i.e.,

$$\mathcal{S}_i \supset \mathcal{S}_{i-1} \cap \mathcal{C}_i. \tag{25}$$

The resulting recursive formulas are summarized in the following theorem.

Theorem 2: Let spheroid $\mathcal{S}_{i-1}$ be an optimum bounding spheroid at iteration $i-1$, which is characterized by $\mathbf{w}_{i-1}$ and σ_{i-1}^2 as shown in (24). Given the constraint set $\mathcal{C}_i$ obtained at time instant i, the recursive expressions for $\mathbf{w}_i$ and σ_i^2 defining the optimum bounding spheroid $\mathcal{S}_i$ that tightly outer bounds $\mathcal{S}_{i-1} \cap \mathcal{C}_i$ are given by

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \lambda_i^* \frac{e_i \mathbf{u}_i}{\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i} \tag{26a}$$
$$e_i = y_i - \mathbf{w}_{i-1}^{\mathrm{T}}\mathbf{u}_i \tag{26b}$$
$$\sigma_i^2 = \sigma_{i-1}^2 - \lambda_i^{*2} \frac{e_i^2}{\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i}, \tag{26c}$$

where λ_i^* is given by

$$\lambda_i^* = \begin{cases} 1 - \frac{\gamma}{|e_i|}, & \text{if } |e_i| > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

Proof: Following similar derivations shown in [30], we see that, given a spheroid $\mathcal{S}_{i-1}$, the center $\mathbf{w}_i$ of the new optimum spheroid $\mathcal{S}_i$ that outer bounds the intersection of $\mathcal{S}_{i-1}$ and $\mathcal{C}_i$ with the smallest radius is found by projecting $\mathbf{w}_{i-1}$ perpendicularly to the nearest hyper-plane specified by $\mathcal{C}_i$. ■

Note that if $|e_i| \leq \gamma$ for any i, then $\lambda_i^* = 0$, which results in $\mathbf{w}_i = \mathbf{w}_{i-1}$ and $\sigma_i^2 = \sigma_{i-1}^2$ according to (26). In other words, if $|e_i| \leq \gamma$ holds, there is no update on the parameter estimate and the computation of (26) is not required. The data-dependent selective update feature is illustrated in Fig. 3 using a two-dimensional example.

Similarly to the K-BEACON algorithm, we implement the coherence criterion to check if $\kappa(\cdot, \mathbf{x}_i)$ should be inserted into the dictionary only when the new data pair warrants an update for the parameter estimates. Again, depending on whether or not the dimension of the dictionary increases, there are two cases for the parameter update.

1) Case 1: Dimension Remains The Same: When the coherence-based sparsification rule does not suggest inserting $\kappa(\cdot, \mathbf{x}_i)$ into the existing dictionary, the dimension remains the same, i.e.,

$$\mathbf{u}_i = \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i)}})\right]^{\mathrm{T}},$$

where $m(i) = m(i-1)$. The parameter estimates are updated according to (26a).

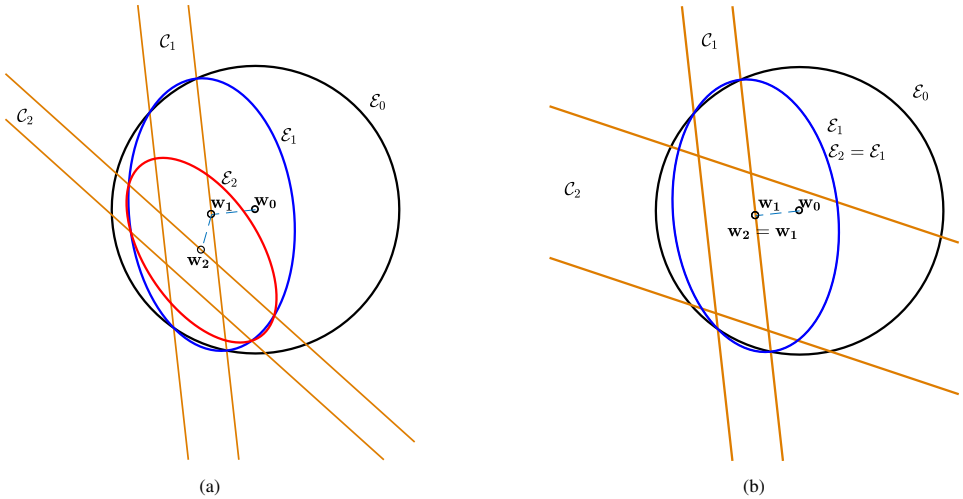

Fig. 2: Data-dependent selective update of K-BEACON. (a) The constraint set $\mathcal{C}_2$ is innovative and yields an update; (b) The constraint set $\mathcal{C}_2$ is not innovative and does not yield an update.

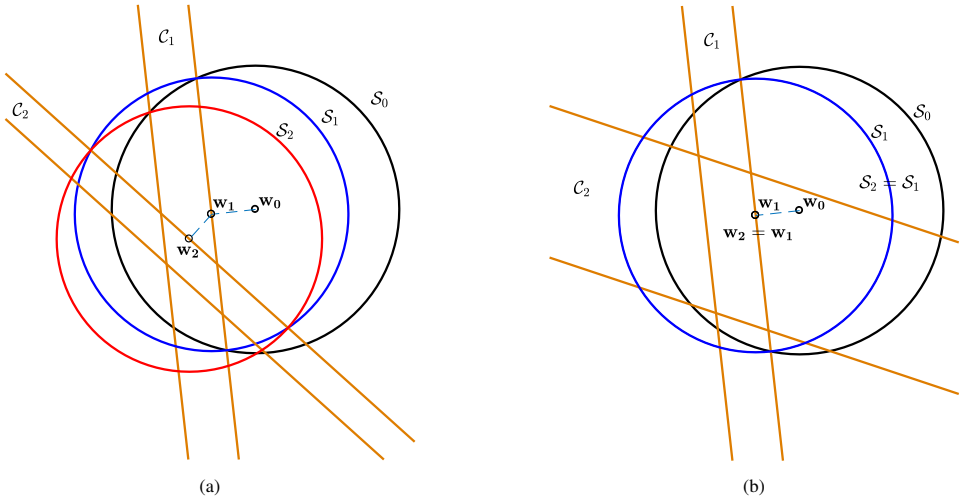

Fig. 3: Data-dependent selective update of K-SM-NLMS. (a) The constraint set $\mathcal{C}_2$ is innovative and yields an update; (b) The constraint set $\mathcal{C}_2$ is not innovative and does not yield an update.

2) Case 2: Dimension Increases: When the coherence-based sparsification rule suggests inserting $\kappa(\cdot, \mathbf{x}_i)$ into the existing dictionary, then the kernelized input vector is expressed as

$$\begin{aligned}\mathbf{u}_i &= \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i-1)}}) \ \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i)}})\right]^{\mathrm{T}} \\ &= \left[\begin{array}{cc} \tilde{\mathbf{u}}_i^{\mathrm{T}} & u_{ai} \end{array}\right]^{\mathrm{T}},\end{aligned} \tag{28}$$

where $m(i) = m(i-1) + 1$ and $\alpha_{m(i)} = i$. The dimension of the weight $\mathbf{w}_{i-1}$ should also increase by one to match the dimension of $\mathbf{u}_i$, i.e.,

$$\tilde{\mathbf{w}}_{i-1} = \left[\begin{array}{cc} \mathbf{w}_{i-1}^{\mathrm{T}} & 0 \end{array}\right]^{\mathrm{T}}. \tag{29}$$

In this case, the expression of $\mathbf{w}_i$ of $\mathcal{S}_i$ is given by (26) after replacing $\mathbf{w}_{i-1}$ with $\tilde{\mathbf{w}}_{i-1}$, i.e.,

$$\mathbf{w}_i = \left[\begin{array}{c} \mathbf{w}_{i-1} \\ 0 \end{array}\right] + \frac{\lambda_i^* e_i}{\tilde{\mathbf{u}}_i^{\mathrm{T}} \tilde{\mathbf{u}}_i + u_{ai}^2} \left[\begin{array}{c} \tilde{\mathbf{u}}_i \\ u_{ai} \end{array}\right]. \tag{30}$$

The pseudocode that summarizes the K-SM-NLMS algorithm is shown in the table labeled as Algorithm 2.

C. Further Interpretations of K-BEACON and K-SM-NLMS

Examining the recursive formulas derived in the previous sections, one can see that the K-BEACON algorithm can be viewed as a KRLS [8] algorithm with a data-dependent forgetting factor, while the K-SM-NLMS algorithm can be viewed as a KNLMS [11] algorithm with a data-dependent step size. This can also be shown by solving bounded error constrained optimization problems as follows.

Similarly to the linear BEACON algorithm [31], when the dictionary dimension does not increase at iteration i, the recursive formula of $\mathbf{w}_i$ of K-BEACON can be derived from

Algorithm 2 K-SM-NLMS algorithm

Initialization
  Initialize w_0, γ, μ
  Insert $\kappa(\cdot, \mathbf{x}_1)$ into the dictionary, set $m(1) = 1$
  Denote the dictionary as $\mathcal{D}_1 = \{\kappa(\cdot, \mathbf{x}_{\alpha_1})\}$
For $i > 1$, repeat
  Given $(\mathbf{x}_i, y_i)$
  Compute $\tilde{\mathbf{u}}_i = \left[\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_1}) \cdots \kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_{m(i-1)}})\right]^{\mathrm{T}}$
  Compute $e_i = y_i - \mathbf{w}_{i-1}^{\mathrm{T}} \tilde{\mathbf{u}}_i$
  if $|e_i| > \gamma$ ▷ Innovation check
    if $\max_{j=1,\cdots,m(i-1)} |\kappa(\mathbf{x}_i, \mathbf{x}_{\alpha_j})| > \mu$ ▷ Case 1
      $m(i) = m(i-1)$
      $\mathbf{u}_i = \tilde{\mathbf{u}}_i$
      Update $\mathbf{w}_i$ using (26a)
    else ▷ Case 2
      $m(i) = m(i-1) + 1$
      Insert $\kappa(\cdot, \mathbf{x}_i)$ into the dictionary
      Denote $\kappa(\cdot, \mathbf{x}_i)$ as $\kappa(\cdot, \mathbf{x}_{\alpha_{m(i)}})$
      Update $\mathbf{w}_i$ using (30)
    end
  end

solving the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{w}_i} \quad & (\mathbf{w}_i - \mathbf{w}_{i-1})^{\mathrm{T}} \mathbf{P}_{i-1}^{-1} (\mathbf{w}_i - \mathbf{w}_{i-1}) - \sigma_{i-1}^2 \\ s.t. \quad & (y_i - \mathbf{w}_i^{\mathrm{T}} \mathbf{u}_i)^2 \leq \gamma^2. \end{aligned} \tag{31}$$

When the dictionary dimension increases by one at iteration i, the recursive formula of $\mathbf{w}_i$ of K-BEACON can be derived from solving (31) after replacing $\mathbf{w}_{i-1}$ and $\mathbf{P}_{i-1}$ with $\tilde{\mathbf{w}}_{i-1}$ and $\tilde{\mathbf{P}}_{i-1}$ as shown in (19) and (20) respectively. The objective function in (31) can be re-written as

$$\begin{aligned} & (\mathbf{w}_i - \mathbf{w}_{i-1})^{\mathrm{T}} \mathbf{P}_{i-1}^{-1} (\mathbf{w}_i - \mathbf{w}_{i-1}) - \sigma_{i-1}^2 \\ = & \left(\mathbf{w}_i - \begin{bmatrix} w_0 \\ \mathbf{0}_{(m(i)-1)\times 1} \end{bmatrix} \right)^{\mathrm{T}} \\ & \begin{bmatrix} P_0 & \mathbf{0}_{1\times(m(i)-1)} \\ \mathbf{0}_{(m(i)-1)\times 1} & \mathbf{I}_{(m(i)-1)\times(m(i)-1)} \end{bmatrix}^{-1} \\ & \left(\mathbf{w}_i - \begin{bmatrix} w_0 \\ \mathbf{0}_{(m(i)-1)\times 1} \end{bmatrix} \right) - \sigma_0^2 \\ & + \sum_{j=1}^{i-1} \lambda_j^* \left(\left(y_j - \mathbf{w}_i^{\mathrm{T}} \begin{bmatrix} \mathbf{u}_j \\ \mathbf{0}_{(m(i)-m(j))\times 1} \end{bmatrix} \right)^2 - \gamma^2 \right). \end{aligned} \tag{32}$$

The first term in (32) represents the confidence in the initial guess and the last term is the weighted sum of the filtering errors with all the previous data points. It is clear from (32) that the objective of K-BEACON is to find the estimate $\mathbf{w}_i$ to minimize the weighted sum of all previous estimation errors, with λ_j^* as the weighting factor for each previous time instant $1 \leq j \leq i-1$. Thus, if many of the λ_j^*'s are zero (due to selective update), the filtering errors of the corresponding time instants are not included in the weighted sum. This can be interpreted as K-BEACON algorithm offering an optimal way (λ_j^* is optimized at each iteration j) to *forget* previous data. This is in contrast to RLS for which the forgetting factor is set *a priori* as a constant, independent of the received data. As will be seen later in Section V, K-BEACON exhibits better tracking properties than KRLS.

As for the recursive formulas of $\mathbf{w}_i$ in K-SM-NLMS, it can be derived from the following constrained optimization problem [30], [34] when dictionary dimension at iteration i does not increase:

$$\begin{aligned} \min_{\mathbf{w}_i} \quad & ||\mathbf{w}_i - \mathbf{w}_{i-1}||_2 \\ s.t. \quad & (y_i - \mathbf{w}_i^{\mathrm{T}} \mathbf{u}_i)^2 \leq \gamma^2, \end{aligned} \tag{33}$$

where the objective function is based on the minimal disturbance principle [44] with the constraint that the filtering error is bounded. When the dictionary dimension at iteration i increases by one, $\mathbf{w}_{i-1}$ in (33) should be replaced with $\tilde{\mathbf{w}}_{i-1}$.

One can see from (31) and (33) that if the previous parameter estimate lies in the new constraint set $\mathcal{C}_i$, i.e., $(y_i - \mathbf{w}_{i-1}^{\mathrm{T}} \mathbf{u}_i)^2 \leq \gamma^2$, then the minimum of the objective function for both algorithms is obtained by not updating the parameter estimates, i.e., $\mathbf{w}_i = \mathbf{w}_{i-1}$. Otherwise, the parameter estimates will be updated to satisfy the constraint, i.e., the *a posteriori error* is upper bounded by γ in magnitude in both K-BEACON algorithm and K-SM-NLMS algorithm.

Observation: The *a posteriori error* defined as $\delta_i = y_i - \mathbf{w}_i^{\mathrm{T}} \mathbf{u}_i$ in both K-BEACON algorithm and K-SM-NLMS algorithm is always less than or equal to the error bound γ in magnitude. Specifically,

$$|\delta_i| = \begin{cases} \gamma, & \text{if } |e_i| > \gamma \\ |e_i|, & \text{if } |e_i| \leq \gamma. \end{cases} \tag{34}$$

Proof: We prove the results of K-BEACON algorithm. The proof of K-SM-NLMS follows similarly.

In the case when $|e_i| > \gamma$ and dimension remains the same at iteration i, from the recursive update equation (17), we have

$$\begin{aligned} \mathbf{w}_i^{\mathrm{T}} \mathbf{u}_i &= \mathbf{w}_{i-1}^{\mathrm{T}} \mathbf{u}_i + \frac{\lambda_i^* \mathbf{u}_i^{\mathrm{T}} \mathbf{P}_{i-1} \mathbf{u}_i e_i}{1 + \lambda_i^* G_i} \\ &= \mathbf{w}_{i-1}^{\mathrm{T}} \mathbf{u}_i + \frac{\lambda_i^* G_i}{1 + \lambda_i^* G_i} e_i. \end{aligned}$$

In the case when $|e_i| > \gamma$ and dimension increases by one at iteration i, from the recursive update equation (23), we have

$$\begin{aligned} \mathbf{w}_i^{\mathrm{T}} \mathbf{u}_i &= \mathbf{w}_{i-1}^{\mathrm{T}} \tilde{\mathbf{u}}_i + \frac{\lambda_i^* \tilde{\mathbf{u}}_i^{\mathrm{T}} \mathbf{P}_{i-1} \tilde{\mathbf{u}}_i e_i + \lambda_i^* u_{ai}^2 e_i}{1 + \lambda_i^* (u_{ai}^2 + \tilde{\mathbf{u}}_i^{\mathrm{T}} \mathbf{P}_{i-1} \tilde{\mathbf{u}}_i)} \\ &= \mathbf{w}_{i-1}^{\mathrm{T}} \tilde{\mathbf{u}}_i + \frac{\lambda_i^* G_i}{1 + \lambda_i^* G_i} e_i. \end{aligned}$$

Hence,

$$\delta_i = y_i - \mathbf{w}_i^{\mathrm{T}} \mathbf{u}_i = e_i - \frac{\lambda_i^* G_i}{1 + \lambda_i^* G_i} e_i = \frac{e_i}{|e_i|} \gamma,$$

where the last equality comes from the optimal assignment of λ_i^* given by (14). In the case when $|e_i| \leq \gamma$, there is no update, i.e., $\mathbf{w}_i = \mathbf{w}_{i-1}$, and $\delta_i = e_i$. ■

IV. Convergence Analysis

In this section, convergence properties of the proposed algorithms are examined. We show that for both K-BEACON algorithm and K-SM-NLMS algorithm, if there exists at least one $\hat{\mathbf{w}}$ that satisfies the constraint set for all i with a given γ, the sequence of parameter estimates is convergent, and the magnitude of filtering error is asymptotically upper bounded by γ.

Theorem 3: Consider a sequence of data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\infty}$ generated by the nonlinear model (7). Denote by m, $\{\mathbf{u}_i\}_{i=1}^{\infty}$ $\{\mathbf{w}_i\}_{i=1}^{\infty}$ and $\{e_i\}_{i=1}^{\infty}$, respectively, the final dictionary dimension, the sequence of kernelized inputs, the sequence of parameter estimates and the sequence of filtering errors generated by either the K-BEACON algorithm or the K-SM-NLMS algorithm. Define

$$\bar{\mathbf{u}}_i = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{0}_{(m-m(i))\times 1} \end{bmatrix},$$

$$\bar{\mathbf{w}}_i = \begin{bmatrix} \mathbf{w}_i \\ \mathbf{0}_{(m-m(i))\times 1} \end{bmatrix},$$

and

$$\Omega_\infty = \cap_{i=1}^{\infty}\{\mathbf{w} \in \mathbb{R}^m : \left(y_i - \mathbf{w}^{\mathrm{T}}\bar{\mathbf{u}}_i\right)^2 \leq \gamma^2\}.$$

If Ω_∞ is non-empty with a prescribed error bound $\gamma > 0$, then $\lim_{i\to\infty} \bar{\mathbf{w}}_i = \bar{\mathbf{w}}_\infty$. Further, if $||\bar{\mathbf{u}}_i||_2^2 \leq u_{\max}$ holds for all i, then $\limsup_{i\to\infty} |e_i| \leq \gamma$.

Proof: See Appendix B. ∎

Note that, for both algorithms, the innovation check is performed by comparing the filtering error to γ to determine if the algorithm should update the parameter estimate. The choice of γ thus offers a flexible trade-off between the asymptotic upper bound of the filtering error magnitude and update frequency. In general, a larger γ requires less frequent update and smaller dimension of the dictionary, hence less computation cost.

The size of the exact membership set is also studied here under different choices of γ. This analysis extends that of set-membership-based system identification of linear models presented in [45]. Formally, we need the following definitions to establish the convergence of the exact membership set.

Definition 1: (Tightness of the error bound γ). Under the setting stated in Theorem 3, if there exists a target parameter $\hat{\mathbf{w}} \in \Omega_\infty$, the error bound γ is said to be tight for $\hat{\mathbf{w}}$ if the filtering errors with filter parameter $\hat{\mathbf{w}}$ are upper bounded by γ in magnitude, and the filtering errors achieve $-\gamma$ and γ with non-zero probability, i.e.,

$$-\gamma \leq y_i - \hat{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i \leq \gamma,$$

$$\text{Prob}\left\{-\gamma \leq y_i - \hat{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i \leq -\gamma + \epsilon\right\} \geq p,$$

and

$$\text{Prob}\left\{\gamma - \epsilon \leq y_i - \hat{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i \leq \gamma\right\} \geq p$$

hold for any $\epsilon > 0$ and some $p \in (0, 1]$ and for all i.

Definition 2: (Persistent excitation). The kernelized input vector $\bar{\mathbf{u}}_i$ is said to be persistently exciting if there exist $0 < q_\alpha < q_\beta < \infty$, and $l > 0$ such that

$$q_\alpha^2\mathbf{I} \preceq \frac{1}{l}\sum_{i=i_0+1}^{i_0+l} \bar{\mathbf{u}}_i\bar{\mathbf{u}}_i^{\mathrm{T}} \preceq q_\beta^2\mathbf{I}$$

for all i_0.

Definition 3: (Diameter of membership set). The diameter of the membership set is defined to be the longest distance between any two points in the exact membership set, i.e.,

$$\mathcal{L}(\Omega_i) = \sup_{\bar{\mathbf{w}}_i, \bar{\mathbf{w}}_j \in \Omega_i} ||\bar{\mathbf{w}}_i - \bar{\mathbf{w}}_j||_2.$$

Theorem 4: With the same formulation as in Theorem 3, if γ is a tight bound for a target parameter $\hat{\mathbf{w}}$, and $\bar{\mathbf{u}}_i$ is persistently exciting (as defined in Definitions 1 and 2, respectively), then the diameter sequence of the exact membership set converges to zero with probability one (w.p.1), i.e.,

$$\text{Prob}\left\{\lim_{i\to\infty} \mathcal{L}(\Omega_i) = 0\right\} = 1. \tag{35}$$

Otherwise, if the error bound is set to be $\gamma = \sqrt{\gamma'^2 + b^2}$, where γ' is the tight bound and the over-estimated part b is an arbitrary positive real number, then

$$\text{Prob}\left\{\lim_{i\to\infty} \mathcal{L}(\Omega_i) \leq \frac{2b}{q_\alpha}\right\} = 1. \tag{36}$$

Proof: See Appendix C. ∎

The above theorem states that the sequence of the exact membership sets converges to a point with probability one under persistent excitation and tight error bound assumptions. When the error bound is over-estimated, then the sequence of the diameters of the exact membership sets converges to a value whose upper bound increases linearly with the over-estimated part with probability one.

V. Simulations

This section presents three simulation examples to demonstrate the benefits of the two proposed kernel SMF algorithms, and to compare their performance with that of KRLS algorithm [8], that of KNLMS algorithm [11] and that of NLR-SM-NLMS algorithm [34]. The first example shows our proposed K-BEACON algorithm and K-SM-NLMS algorithm compare favorably to KRLS algorithm and KNLMS algorithm, respectively, in terms of computational costs while achieving comparable steady state prediction error. It also shows that the error bounds of our proposed algorithms offer flexible trade-off between steady state prediction error and computational costs. The second example demonstrates the K-BEACON algorithm's advantage in terms of tracking time varying systems, compared with the KRLS algorithm. The third example shows the advantage of embedding the coherence criterion in the K-SM-NLMS algorithm. Comparing to the NLR-SM-NLMS algorithm, which did not include coherence check in constructing its dictionary, our proposed K-SM-NLMS algorithm achieves more efficient dictionary sparsification.

A. Example 1

Consider the discrete-time nonlinear dynamical system studied in [11]:

$$\begin{cases} s_i = 1.1 \cdot \exp(-|s_{i-1}|) + x_i, \\ d_i = s_i^2, \\ y_i = d_i + v_i, \end{cases}$$

where x_i is the system input, d_i is the desired system output, and y_i is the observed output. The additive noise $v_i, i = 1, 2, \cdots$, is a sequence of independent and identically distributed (*i.i.d.*) random variables distributed according to $\mathcal{N}(0,\, 1)$. For each i, the input x_i is sampled from an *i.i.d.* Gaussian random sequence with marginal distribution $\mathcal{N}(0,\, 0.25^2)$. The initial condition is set as $s_0 = 0.5$. In this example, SNR $= -4.0$ dB, which is defined as the ratio of the power of desired output d_i to that of additive noise v_i. The objective here is to construct an estimated model of the form $\hat{d}_i = h_i(x_i)$. A Laplacian kernel $\kappa(x_i, x_j) = \exp(-\frac{||x_i - x_j||_2}{0.35})$ is used in this example. The threshold μ in the coherence criterion is set to be 0.3. The experimental setting and parameters of the KNLMS algorithm and the KRLS algorithm are set in this paper to be the same as those in [11]. The details on how the parameters are set, such as kernel bandwidth, the step size η and the regularization parameter ϵ of the KNLMS algorithm, can be also found in [11]. The same step size η in the KNLMS algorithm is also used to upper bound the step size λ^* of the K-SM-NLMS algorithm and the same regularization parameter ϵ in the KNLMS algorithm is also used in the K-SM-NLMS and K-BEACON algorithms. In order to find the error bound γ (ranging from 0 to 3 with increment of 0.01) for K-BEACON algorithm and K-SM-NLMS algorithm, preliminary experiments were conducted following the same approaches used in [11]: Based on 10 independent runs of 3000-sample sequences, and the error bounds were selected to minimize the mean-square prediction error over the last 500 samples.

Fig. 4 shows the learning curves averaged over 200 independent runs with $\gamma = 0.91$ in the K-BEACON algorithm and $\gamma = 0.88$ in the K-SM-NLMS algorithm. To reduce the fluctuation of the learning curves, the MSE is computed with a moving average window that averages over 20 previous consecutive iterations in each run. Table I summarizes the simulation setup and performance of each algorithm obtained from averaging 200 independent 10,000-iteration runs. The normalized mean-square prediction error (NMSE) over the last 5,000 iterations was computed as

$$\text{NMSE} = \mathbb{E}\left[\frac{\sum_{i=5001}^{10000}(d_i - h_{i-1}(x_i))^2}{\sum_{i=5001}^{10000} d_i^2}\right].$$

Update frequency of K-SM-NLMS algorithm and K-BEACON algorithm is presented as a percentage of iterations (averaged over 200 independent runs). The dictionary dimension of each algorithm is also averaged over 200 runs. In this particular example, K-BEACON with $\gamma = 0.91$ yields a dimension smaller than that of KRLS and a significantly less frequent parameter updates while achieving a comparable steady-state MSE. Similar observation can be made for the comparison of K-SM-NLMS and KNLMS. The performance of K-BEACON and K-SM-NLMS algorithm is shown with different choices of the error bound γ. A larger γ yields larger MSE but results in less frequent update and smaller dimension of the dictionary, indicating a trade-off between the steady-state error and update frequency.

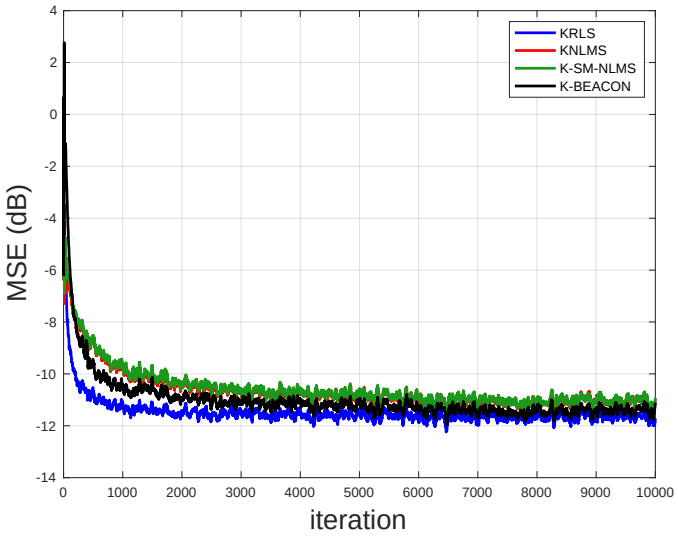

Fig. 4: Learning curves averaged over 200 independent runs of Example 1.

Table II shows the computation cost of each algorithm, measured by the number of real multiplications and the number of real additions per iteration for each algorithm, where m denotes the dictionary dimension at that iteration. The computation cost of KRLS algorithm is $\mathcal{O}(m^2)$, while that of K-BEACON algorithm is only a fraction of $\mathcal{O}(m^2)$, due to its selective update feature which makes the algorithm skip updates whenever the data pairs are judged to be not *innovative*. Similarly, the computational cost of KNLMS is $\mathcal{O}(m)$, while that of K-SM-NLMS is only a fraction of that, and the fraction is directly related to the percentage of updates.

B. Example 2

Consider the following time-varying nonlinear system

$$\begin{cases} d_i = 0.1 \cdot \sin(d_{i-1}\pi) + 0.4 \cdot \cos(d_{i-2}\pi), & i < 5,000, \\ d_i = 0.9 \cdot \sin(d_{i-1}\pi) + 0.15 \cdot \cos(d_{i-2}\pi), & i \geq 5,000, \\ y_i = d_i + v_i, \end{cases}$$

where d_i is the desired system output and the initial condition is set to be $d_0 = d_1 = 0.5$. At each iteration, the system output is corrupted by an additive noise v_i. $\{v_i\}$ is a sequence of *i.i.d.* random variables uniformly distributed in $\mathcal{U}(-1, 1)$. The SNR in this case is -0.8 dB. The algorithms are implemented to construct an approximate model of the form $\hat{d}_i = h_i(\mathbf{x}_i)$ with $\mathbf{x}_i = [y_{i-1}\ y_{i-2}]^{\mathrm{T}}$.

Table III summarizes the simulation setup and performance of each algorithm. The NMSE is computed by averaging over the last 2,000 iterations of each period, before and after system change, and averaging over 200 independent runs. Update frequency of K-BEACON algorithm in Table III is calculated as a percentage of iterations (averaged over 200 independent runs) in each period.

Fig. 5 shows the learning curves averaged over 200 independent runs. It is well known that, for RLS algorithms, a larger forgetting factor yields a smaller steady-state error but worse

TABLE I: Simulation setup and performance of Example 1.

Algorithm	Parameter	Dimension	NMSE	Update frequency
KNLMS [11]	$\mu = 0.3, \epsilon = 0.0009, \eta = 0.01$	4.16	0.200	100%
KRLS [8]	$\nu = 0.7$	6.85	0.172	100%
K-SM-NLMS	$\mu = 0.3, \gamma = 0.88$	4.02	0.201	39.8%
	$\mu = 0.3, \gamma = 1.88$	3.61	0.214	7.4%
K-BEACON	$\mu = 0.3, \gamma = 0.91$	4.02	0.183	38.1%
	$\mu = 0.3, \gamma = 1.91$	3.60	0.207	7.0%

TABLE II: Computation cost per iteration.

Algorithm	$\times$	$+$
KNLMS [11]	$3m + \mathcal{O}(1)$	$3m + \mathcal{O}(1)$
KRLS [8]	$4m^2 + 4m + \mathcal{O}(1)$	$4m^2 + 4m + \mathcal{O}(1)$
K-SM-NLMS No Update	$m + \mathcal{O}(1)$	$m + \mathcal{O}(1)$
K-SM-NLMS Update	$3m + \mathcal{O}(1)$	$3m + \mathcal{O}(1)$
K-BEACON No Update	$m + \mathcal{O}(1)$	$m + \mathcal{O}(1)$
K-BEACON Update	$3m^2 + 3m + \mathcal{O}(1)$	$2m^2 + 3m + \mathcal{O}(1)$

tracking performance. Comparing to the KRLS algorithm with different forgetting factors, β, K-BEACON algorithm shows comparable MSE and good tracking performance. In fact, K-BEACON has better tracking performance than KRLS for the forgetting factors chosen, while achieving comparable NMSE performance. Although KRLS with forgetting factor $\beta = 1$ performs slightly better in terms of NMSE before system change, its tracking performance after system change is much worse than K-BEACON algorithm's. In addition, compared with KRLS with forgetting factor $\beta = 0.999$, K-BEACON algorithm yields smaller MSE both before and after system change as well as faster tracking performance.

The better tracking performance of the K-BEACON algorithm can be explained by understanding that the optimal combining parameter, λ_i^*, serves as a *data-dependent* forgetting factor, see, e.g., (14) and (32). From (32), we can see easily that the influence of past data is much reduced because the optimal $\lambda_i^* = 0$ for a large number of previous iterations. This enables the K-BEACON algorithm to track (*adapt to*) system changes quickly without compromising the MSE performance. In comparison, the forgetting factor in KRLS needs to be set *a priori*, independent of the data received.

C. Example 3

In this example, we compare the performance of our proposed K-SM-NLMS algorithm with that of the NLR-SM-KNLMS algorithm proposed in [34] and that of the KNLMS algorithm proposed in [11]. Note that in the NLR-SM-KNLMS algorithm, only innovation check is used to construct its dictionary and the maximum dictionary length is fixed by a pre-specified number. At each iteration after the dictionary length achieves the maximum number, if the data point is innovative, this data point is added to replace the oldest element in the dictionary, regardless of its relation with those elements that already exist in the dictionary. In our proposed K-SM-NLMS algorithm, adding the coherence criterion results in more efficient dictionary with smaller dimensions.

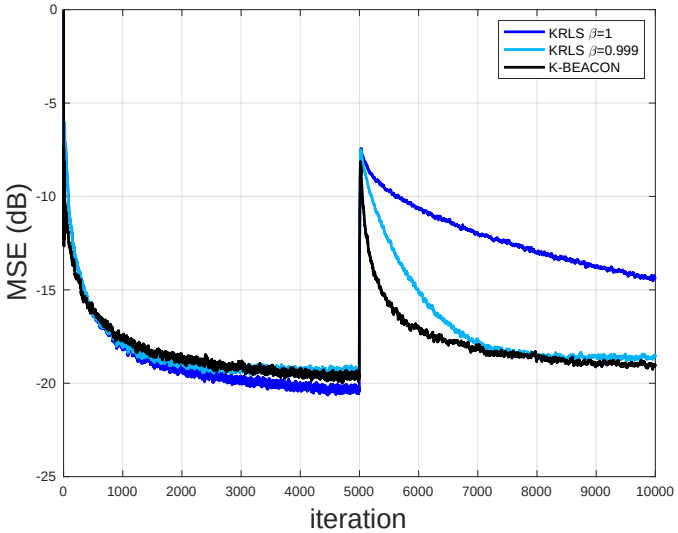

Fig. 5: Learning curves averaged over 200 independent runs of Example 2.

Consider the nonlinear problem studied in [34]

$$\begin{cases} d_i = \frac{d_{i-1}}{1+d_{i-1}^2} + x_{i-1}^3, \\ y_i = d_i + v_i, \end{cases}$$

where the input x_i is sampled from an *i.i.d.* Gaussian random sequence with a marginal distribution $\mathcal{N}(0, 0.15^2)$. The output signal y_i is corrupted by a zero-mean white Gaussian noise with variance $\sigma_v^2 = 10^{-4}$. Following the same settings presented in [34], a Gaussian kernel with bandwidth 0.025 is employed for both algorithms and the maximum dictionary length for the NLR-SM-KNLMS algorithm is set at 16. Setting $\gamma = \sqrt{10}\sigma_v$ as used in [34] yields the learning curves (averaged over 200 independent runs) shown in Fig. 6. The step size η and the regularization parameter ϵ of the KNLMS algorithm were determined by grid search. Based on 10 independent runs of 1000-sample sequences, they were selected to minimize the mean-square prediction error over the last 200 samples. Table IV summarizes the setup and performance of each algorithm. The NMSE is computed by averaging over the last 1,000 iterations. This example clearly demonstrates that the K-SM-NLMS algorithm compares favorably to both KNLMS and NLR-SM-KNLMS algorithms, for it yields smaller dictionary dimension and less update frequency while achieving smaller NMSE.

TABLE III: Simulation setup and performance of Example 2.

Algorithm	Parameter	Dimension	NMSE_1	Update frequency_1	NMSE_2	Update frequency_2
KRLS [8]	$\nu = 0.5, \beta = 1$	30.1	0.099	100%	0.093	100%
	$\nu = 0.5, \beta = 0.999$	30.1	0.120	100%	0.030	100%
K-BEACON	$\mu = 0.5, \gamma = 0.96$	24.7	0.116	7.3%	0.028	7.7%

MSE_1 and Update frequency_1 (MSE_2 and Update frequency_2) are evaluated in the period before (after) system change.

TABLE IV: Simulation setup and performance of Example 3.

Algorithm	Parameter	Dimension	NMSE	Update frequency
KNLMS [11]	$\mu = 0.1, \epsilon = 0.2, \eta = 0.35$	15.2	0.369	100%
NLR-SM-KNLMS [34]	dimension limit $= 16, \gamma = 0.0316$	16.0	0.462	51.1%
K-SM-NLMS	$\mu = 0.1, \gamma = 0.0316$	14.7	0.351	45.0%

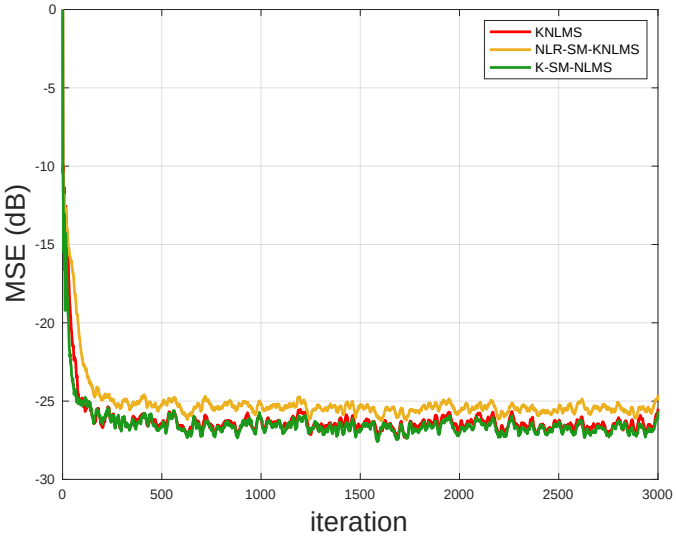

Fig. 6: Learning curves averaged over 200 independent runs of Example 3.

VI. Conclusion

This paper has presented two nonlinear adaptive filtering algorithms derived from the SMF principles and kernel method. In addition to the coherence-based sparsification rule that controls the size of kernel expansion, the data-dependent selective update property embedded in the SMF framework results in a sparser model and, more importantly, much less frequent updates of the parameter estimates. The selective update property makes the proposed kernel SMF algorithms more effective in terms of less computation cost and faster tracking performance. In the convergence analysis, it is shown that the sequences of parameter estimates of our proposed algorithms are convergent, and the filtering errors are asymptotically upper bounded in magnitude. The error bound of our proposed algorithm offers a flexible trade-off between the steady-state performance and computational costs, which is directly related to the parameter update frequency. Simulation results showed that our proposed kernel algorithms achieve comparable steady-state MSE to that of KRLS and KNLMS algorithms with only a small fraction of parameter updates. Further, the proposed algorithms exhibit good tracking performance due to its selective update feature.

Appendix A
Proof of Theorem 1

Linear combination of (11) and (8) yields the bounding ellipsoid at time instant i, i.e.,

$$\begin{aligned}\mathcal{E}_i &= \{\mathbf{w} : (\mathbf{w}-\mathbf{w}_i)^{\mathrm{T}}\mathbf{P}_i^{-1}(\mathbf{w}-\mathbf{w}_i) \le \sigma_i^2\} \\ &= \{\mathbf{w} : (\mathbf{w}-\mathbf{w}_{i-1})^{\mathrm{T}}\mathbf{P}_{i-1}^{-1}(\mathbf{w}-\mathbf{w}_{i-1}) \\ &\quad + \lambda_i(y_i - \mathbf{w}^{\mathrm{T}}\mathbf{u}_i)^2 \le \sigma_{i-1}^2 + \lambda_i\gamma^2\}. \end{aligned} \tag{37}$$

The following relations are directly obtained via identification-by-terms of the expanded expressions in (37)

$$\mathbf{P}_i^{-1} = \mathbf{P}_{i-1}^{-1} + \lambda_i\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}} \tag{38}$$

$$\mathbf{w}_i = \mathbf{P}_i(\mathbf{P}_{i-1}^{-1}\mathbf{w}_{i-1} + \lambda_i y_i\mathbf{u}_i) \tag{39}$$

$$\begin{aligned}\sigma_i^2 &= \sigma_{i-1}^2 + \lambda_i\gamma_i^2 - \lambda_i y_i^2 \\ &\quad -\mathbf{w}_{i-1}^{\mathrm{T}}\mathbf{P}_{i-1}^{-1}\mathbf{w}_{i-1} + \mathbf{w}_i^{\mathrm{T}}\mathbf{P}_i^{-1}\mathbf{w}_i. \end{aligned} \tag{40}$$

Pre-multiplying (38) with $\mathbf{P}_i$ and rearranging the terms yield the relation $\mathbf{P}_i\mathbf{P}_{i-1}^{-1} = \mathbf{I} - \lambda_i\mathbf{P}_i\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}$, which, after substitution in (39) together with (13c), gives (13a).

By employing the matrix inversion lemma, the computation of $\mathbf{P}_i$ can be expressed as

$$\mathbf{P}_i = \mathbf{P}_{i-1} - \frac{\lambda_i\mathbf{P}_{i-1}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}\mathbf{P}_{i-1}}{1+\lambda_i G_i}, \tag{41}$$

which combined with (13a) yields:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\lambda_i\mathbf{P}_{i-1}\mathbf{u}_i e_i}{1+\lambda_i G_i}. \tag{42}$$

Equation (13d) can be established by substituting (38) and (42) into (40).

We now prove (14). Since σ_{i-1}^2 and γ are fixed, minimizing σ_i^2 is equivalent to minimizing the following

$$J_i = \lambda_i - \frac{c_i\lambda_i}{1+\lambda_i G_i},$$

where $c_i = \frac{e_i^2}{\gamma^2}$. The first derivative of J_i over λ_i is

$$\frac{\mathrm{d}J_i}{\mathrm{d}\lambda_i} = \frac{(1+\lambda_i G_i)^2 - c_i}{(1+\lambda_i G_i)^2}.$$

The second derivative is given by

$$\frac{\mathrm{d}^2 J_i}{\mathrm{d}\lambda_i^2} = \frac{2c_i G_i(1+\lambda_i G_i)}{(1+\lambda_i G_i)^4}.$$

Since $\mathbf{P}_{i-1}$ is positive definite, $G_i = \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_{i-1}\mathbf{u}_i > 0$ and the second derivative is positive. Recall that λ_i is non-negative, if $c_i > 1$, then the first derivative achieves zero at $\lambda_i^* = \frac{1}{G_i}(\frac{|e_i|}{\gamma} - 1)$; if $c_i \le 1$, J_i is minimized at $\lambda_i^* = 0$.

APPENDIX B
PROOF OF THEOREM 3

We first show that the sequence $\{\sigma_i^2\}_{i=1}^{\infty}$ is convergent if Ω_∞ is non-empty. Note that this sequence is non-increasing. It suffices to prove that it is lower bounded. Since Ω_∞ is non-empty, there exists a large σ_0^2 such that the initial $\mathcal{E}_0$ and $\mathcal{S}_0$ include Ω_∞, i.e., $\Omega_\infty \subseteq \mathcal{E}_0$ and $\Omega_\infty \subseteq \mathcal{S}_0$. Since $\Omega_\infty \subseteq \mathcal{C}_i$ for all i, $\mathcal{E}_i = \mathcal{E}_{i-1} \cap \mathcal{C}_i$ and $\mathcal{S}_i = \mathcal{S}_{i-1} \cap \mathcal{C}_i$, then $\Omega_\infty \subseteq \mathcal{E}_\infty$ and $\Omega_\infty \subseteq \mathcal{S}_\infty$, which implies that $\sigma_i^2 \geq 0$ for all i. Note that the difference $\sigma_i^2 - \sigma_{i-1}^2$ does not depend on the initial value σ_0^2 for all $i > 1$, which implies that $\{\sigma_i^2\}_{i=1}^{\infty}$ is convergent with any arbitrary σ_0^2.

Now we use the result that $\{\sigma_i^2\}_{i=1}^{\infty}$ is a Cauchy sequence to prove that $\{\bar{\mathbf{w}}_i\}_{i=1}^{\infty}$ is also a Cauchy sequence. We prove the results of K-BEACON algorithm. The proof of K-SM-NLMS follows similarly.

Consider the subsequence where every iteration in the subsequence is an updating iteration. For every positive real number ϵ, there is a positive integer N such that for all $m, n > N$, $|\sigma_m^2 - \sigma_n^2| < \epsilon$. Without loss of generality, let $m > n$ in the sequel, we have

$$\begin{aligned} |\sigma_m^2 - \sigma_n^2| &= \sum_{i=n+1}^{m} \left(\sigma_{i-1}^2 - \sigma_i^2\right) \\ &= \sum_{i=n+1}^{m} \left(\frac{\lambda_i^* e_i^2}{1+\lambda_i^* G_i} - \lambda_i^* \gamma^2\right) \\ &= \sum_{i=n+1}^{m} \frac{(|e_i|-\gamma)^2}{G_i}, \end{aligned} \tag{43}$$

where the last equality comes from assigning the optimal λ_i^* given by (14). Define

$$\bar{\mathbf{P}}_i = \begin{bmatrix} \mathbf{P}_i & \mathbf{0}_{m(i)\times(m-m(i))} \\ \mathbf{0}_{(m-m(i))\times m(i)} & \mathbf{0}_{(m-m(i))\times(m-m(i))} \end{bmatrix}.$$

According to (17), it can be verified that

$$\bar{\mathbf{w}}_i = \bar{\mathbf{w}}_{i-1} + \frac{\lambda_i^* e_i \bar{\mathbf{P}}_{i-1}\bar{\mathbf{u}}_i}{1+\lambda_i^* G_i}.$$

Therefore,

$$\begin{aligned} ||\bar{\mathbf{w}}_m - \bar{\mathbf{w}}_n||_2^2 &\leq \sum_{i=n+1}^{m} ||\bar{\mathbf{w}}_i - \bar{\mathbf{w}}_{i-1}||_2^2 \\ &= \sum_{i=n+1}^{m} \frac{\lambda_i^{*2} e_i^2 \bar{\mathbf{u}}_i^{\mathrm{T}} \bar{\mathbf{P}}_{i-1}\bar{\mathbf{P}}_{i-1}\bar{\mathbf{u}}_i}{\left(1+\lambda_i^* G_i\right)^2} \\ &\leq \sum_{i=n+1}^{m} \frac{\lambda_i^{*2} e_i^2 \bar{\mathbf{u}}_i^{\mathrm{T}} \bar{\mathbf{P}}_{i-1}\bar{\mathbf{u}}_i}{\left(1+\lambda_i^* G_i\right)^2} \\ &= \sum_{i=n+1}^{m} \frac{(|e_i|-\gamma)^2}{G_i} < \epsilon, \end{aligned} \tag{44}$$

where the last equality comes from assigning the optimal λ_i^* given by (14) and the fact that $G_i = \bar{\mathbf{u}}_i^{\mathrm{T}} \bar{\mathbf{P}}_{i-1} \bar{\mathbf{u}}_i$.

Now we prove that $\limsup_{i\to\infty} |e_i| \leq \gamma$ if $||\bar{\mathbf{u}}_i||_2^2 \leq u_{\max}$ holds for all i. Suppose $\limsup_{i\to\infty} |e_i| \leq \gamma$ is false, then there exists a positive real number ϵ such that

$$\limsup_{i\to\infty} |e_i| = \gamma + \epsilon,$$

which implies there are infinitely many i such that $(|e_i| - \gamma) > \epsilon/c$, for some $c > 1$. Consider the $\{\sigma_i^2\}_{i=1}^{\infty}$ sequence,

$$\sigma_i^2 = \sigma_{i-1}^2 - \frac{(|e_i|-\gamma)^2}{G_i} < \sigma_{i-1}^2 - \frac{\epsilon^2}{c^2 u_{\max}},$$

which implies that $\sigma_N^2 < 0$ for some N large enough, regardless of what initial value σ_0^2 is chosen. This contradicts the fact that there exists a large initial value σ_0^2 such that $\sigma_i^2 \geq 0$ for all i.

APPENDIX C
PROOF OF THEOREM 4

According to Definition 1, $\hat{\mathbf{w}}$ is a point in the exact membership set. Denote by $\check{\mathbf{w}}$ another arbitrary fixed point such that $\check{\mathbf{w}} \neq \hat{\mathbf{w}}$. Define

$$\begin{aligned} z_k = \max\Bigg(&\frac{1}{l}\sum_{i=1}^{l}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2, \frac{1}{l}\sum_{i=l+1}^{2l}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2, \cdots, \\ &\frac{1}{l}\sum_{i=(k-1)l+1}^{kl}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2\Bigg). \end{aligned} \tag{45}$$

We first show that the membership set excludes $\check{\mathbf{w}}$ in probability, i.e., we show that $\text{Prob}\{z_k > \gamma^2\} \to 1$ as $k \to \infty$. Suppose

$$\text{Prob}\left\{\frac{1}{l}\sum_{i=j+1}^{j+l}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \leq \gamma^2\right\} \leq 1 - p \tag{46}$$

holds for some $p \in (0, 1]$, then we have

$$\begin{aligned} &\text{Prob}\left\{z_k > \gamma^2\right\} \\ &= 1 - \text{Prob}\left\{z_k \leq \gamma^2\right\} \\ &= 1 - \prod_{t=0}^{k-1} \text{Prob}\left\{\frac{1}{l}\sum_{i=j+1}^{j+l}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \leq \gamma^2\right\} \\ &= 1 - (1-p)^k \to 1 \quad \text{as} \quad k \to \infty \end{aligned}$$

In the sequel, it suffices to prove (46). Denote by Δ_i the filtering error at iteration i with parameter $\hat{\mathbf{w}}$, i.e., $\Delta_i = y_i - \hat{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i$, then

$$\begin{aligned} &\frac{1}{l}\sum_{i=j+1}^{j+l}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \\ &= \frac{1}{l}\sum_{i=j+1}^{j+l}(y_i - \hat{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i + \hat{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \\ &= \frac{1}{l}\sum_{i=j+1}^{j+l}(\Delta_i + \tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \\ &= \sum_{i=j+1}^{j+l}\left(\frac{1}{l}(\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 + \frac{1}{l}\Delta_i^2 + \frac{2}{l}\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i\Delta_i\right), \end{aligned} \tag{47}$$

where $\tilde{\mathbf{w}} = \hat{\mathbf{w}} - \check{\mathbf{w}} \neq 0$. Since $\bar{\mathbf{u}}_i$ is persistently exciting, see Definition 2, then

$$\frac{1}{l}\sum_{i=j+1}^{j+l}(\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \geq q_\alpha^2 ||\tilde{\mathbf{w}}||_2^2 > 0. \tag{48}$$

Recall that in Definition 1, Δ_i achieves the error bound γ and $-\gamma$ with non-zero probability, and there is non-zero probability that $\text{sgn}(\Delta_i) = \text{sgn}(\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)$, hence

$$\text{Prob}\left\{\frac{1}{l}\sum_{i=j+1}^{j+l}(\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 \geq \gamma^2 - \frac{1}{l}\sum_{i=j+1}^{j+l}\Delta_i^2 \text{ and } \sum_{i=j+1}^{j+l}\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i\Delta_i > 0\right\} \geq p > 0, \tag{49}$$

from some $p \in (0, 1]$. Then we have

$$\text{Prob}\left\{\frac{1}{l}\sum_{i=j+1}^{j+l}(\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 > \gamma^2 - \sum_{i=j+1}^{j+l}\left(\frac{1}{l}\Delta_i^2 + \frac{2}{l}\tilde{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i\Delta_i\right)\right\} \geq p > 0. \tag{50}$$

Rearranging the terms in (50) and combining (47) yields

$$\text{Prob}\left\{\frac{1}{l}\sum_{i=j+1}^{j+l}(y_i - \check{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{u}}_i)^2 > \gamma^2\right\} \geq p,$$

which completes the proof of (46).

Now, the membership set excludes any $\check{\mathbf{w}} \neq \hat{\mathbf{w}}$ in probability implies that $\mathcal{L}(\Omega_i)$ converges to zero in probability. Therefore, there is at least one subsequence of $\mathcal{L}(\Omega_i)$ converges to zero w.p.1. Further, since the sequence $\mathcal{L}(\Omega_i)$ itself is non-increasing, it implies that $\mathcal{L}(\Omega_i)$ converges to zero w.p.1.

In the case where $\gamma = \sqrt{\gamma'^2 + b^2}$ is the pre-specified error bound, where γ' is a tight bound for $\hat{\mathbf{w}}$. Denote by $\check{\mathbf{w}}$ another fixed point such that $||\check{\mathbf{w}} - \hat{\mathbf{w}}||_2 > \frac{b}{q_\alpha}$. Define z_k as shown in (45). To show that the exact membership set excludes $\check{\mathbf{w}}$ in probability, it can be shown that $\text{Prob}\{z_k > \gamma'^2 + b^2\} \to 1$ as $k \to \infty$. Therefore, the diameter sequence of the membership set with $\gamma = \sqrt{\gamma'^2 + b^2}$ converges to a value not greater than $\frac{2b}{q_\alpha}$ w.p.1.

REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[2] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. Hoboken, NJ, USA: Wiley, 2009.

[3] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ, USA: Wiley, 2003.

[4] C.-H. Cheng and E. J. Powers, "Optimal volterra kernel estimation algorithms for a nonlinear communication system for PSK and QAM inputs," *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 147–163, Jan. 2001.

[5] P. Zhu, B. Chen, and J. C. Principe, "Learning nonlinear generative models of time series with a Kalman filter in RKHS," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 141–155, Jan. 2014.

[6] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[7] D. Duttweiler and T. Kailath, "RKHS approach to detection and estimation problems–Part V: Parameter estimation," *IEEE Trans. Inf. Theory*, vol. 19, no. 1, pp. 29–37, Jan. 1973.

[8] 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.

[9] W. Liu, P. P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, Feb. 2008.

[10] P. P. Pokharel, W. Liu, and J. C. Principe, "Kernel least mean square algorithm with constrained growth," *Signal Process.*, vol. 89, no. 3, pp. 257–265, Mar. 2009.

[11] 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.

[12] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[13] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[14] 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.

[15] K. Slavakis and S. Theodoridis, "Sliding window generalized kernel affine projection algorithm using projection mappings," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, p. 735351, 2008.

[16] M. 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.

[17] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*. New York, NY, USA: Wiley, 2010.

[18] S. Y. Kung, *Kernel Methods and Machine Learning*. Cambridge, U.K.: Cambridge University Press, 2014.

[19] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, 2010.

[20] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *NIPS*, 2001, pp. 682–688.

[21] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2008, pp. 1177–1184.

[22] A. Singh, N. Ahuja, and P. Moulin, "Online learning with kernels: Overcoming the growing sum problem," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2012, pp. 1–6.

[23] P. Bouboulis, S. Chouvardas, and S. Theodoridis, "Online distributed learning over networks in rkh spaces using random fourier features," *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1920–1932, 2017.

[24] E. Fogel and Y.-F. Huang, "On the value of information in system identification-—bounded noise case," *Automatica*, vol. 18, no. 2, pp. 229–238, Mar. 1982.

[25] S. Dasgupta and Y.-F. Huang, "Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise," *IEEE Trans. Inf. Theory*, vol. 33, no. 3, pp. 383–392, May 1987.

[26] S. Werner, Y.-F. Huang, M. L. De Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 2849–2852.

[27] S. Gollamudi, S. Kapoor, S. Nagaraj, and Y.-F. Huang, "Set-membership adaptive equalization and an updator-shared implementation for multiple channel communications systems," *IEEE Trans. Signal Process.*, vol. 46, no. 9, pp. 2372–2385, Sep. 1998.

[28] P. Ghofrani, T. Wang, and A. Schmeink, "A fast converging channel estimation algorithm for wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3169–3184, Jun. 2018.

[29] P. S. Diniz, "On data-selective adaptive filtering," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4239–4252, Aug. 2018.

[30] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Process. Lett.*, vol. 5, no. 5, pp. 111–114, May 1998.

[31] S. Nagaraj, S. Gollamudi, S. Kapoor, and Y.-F. Huang, "BEACON: An adaptive set-membership filtering technique with sparse updates," *IEEE Trans. Signal Process.*, vol. 47, no. 11, pp. 2928–2941, Nov. 1999.

[32] R. Arablouei and K. Doğançay, "Steady-state mean squared error and tracking performance analysis of the quasi-OBE algorithm," *Signal Process.*, vol. 93, no. 1, pp. 100–108, Jan. 2013.

[33] A. V. Malipatil, Y.-F. Huang, S. Andra, and K. Bennett, "Kernelized set-membership approach to nonlinear adaptive filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 149–152.

[34] A. Flores and R. C. de Lamare, "Set-membership adaptive kernel NLMS algorithms: Design and analysis," *Signal Process.*, vol. 154, pp. 1–14, Jan. 2019.

[35] K. Chen, S. Werner, A. Kuh, and Y.-F. Huang, "Nonlinear online learning—a kernel SMF approach," in *Proc. Asia-Pacific Signal Inf. Process. Assoc.*, 2018, pp. 218–223.
[36] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Appl.*, vol. 33, no. 1, pp. 82–95, 1971.
[37] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. Ann. Conf. Comput. Learn. Theory*, 2001, pp. 416–426.
[38] M. Yukawa, "Multikernel adaptive filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4672–4682, Sep. 2012.
[39] W. D. Parreira, J. C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the gaussian kernel least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2208–2222, May 2012.
[40] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1313–1326, Aug. 2012.
[41] S. Zhao, B. Chen, P. Zhu, and J. C. Principe, "Fixed budget quantized kernel least-mean-square algorithm," *Signal Process.*, vol. 93, no. 9, pp. 2759–2770, Sep. 2013.
[42] 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.
[43] M. 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.
[44] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.
[45] E.-W. Bai, H. Cho, and R. Tempo, "Convergence properties of the membership set," *Automatica*, vol. 34, no. 10, pp. 1245–1249, Oct. 1998.