KERNEL LMS ALGORITHM WITH FORWARD-BACKWARD SPLITTING FOR DICTIONARY LEARNING

Wei Gao[†‡], *Jie Chen*[†], *Cédric Richard*[†], *Jianguo Huang*[‡], *Rémi Flamary*[†]

[†]Université de Nice Sophia-Antipolis, CNRS, Observatoire de la Côte d'Azur, Nice, France
{jie.chen, cedric.richard, remi.flamary}@unice.fr
[‡] College of Marine Engineering, Northwestern Polytechnical University, Xian, China
gao_wei@mail.nwpu.edu.cn; jghuang@nwpu.edu.cn

ABSTRACT

Nonlinear adaptive filtering with kernels has become a topic of high interest over the last decade. A characteristics of kernel-based techniques is that they deal with kernel expansions whose number of terms is equal to the number of input data, making them unsuitable for online applications. Kernel-based adaptive filtering algorithms generally rely on a two-stage process at each iteration: a model order control stage that limits the increase in the number of terms by including only valuable kernels into the so-called dictionary, and a filter parameter update stage. It is surprising to note that most existing strategies for dictionary update can only incorporate new elements into the dictionary. This unfortunately means that they cannot discard obsolete kernel functions, within the context of a time-varying environment in particular. Recently, to remedy this drawback, it has been proposed to associate an ℓ_1-norm regularization criterion with the mean-square error criterion. The aim of this paper is to provide theoretical results on the convergence of this approach.

Index Terms— Nonlinear adaptive filtering, reproducing kernel, sparsity, online forward-backward splitting, convergence

1. INTRODUCTION

During the last decade, adaptive filtering in reproducing kernel Hilbert spaces (RKHS) has become an attractive tool for nonlinear system identification. Indeed, this framework allows the use of linear structures to solve nonlinear estimation problems. For an overview of these approaches, we refer the reader to [1]. In the block diagram presented in Figure 1, the subspace $\mathcal{U}$ is a compact of $\mathbb{R}^q$, $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a reproducing kernel, and $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ is the induced RKHS with its inner product. The noise $z(n)$ is white, Gaussian with zero-mean. Considering the least-squares approach, given N input vectors $\boldsymbol{u}_n$ and desired outputs $d(n)$, the problem consists of identifying the function $\psi(\cdot)$ in $\mathcal{H}$ that solves the problem

$$\psi^* = \arg\min_{\psi\in\mathcal{H}} \left\{ J(\psi) = \frac{1}{2}\sum_{i=1}^{N} (d_i - \psi(\boldsymbol{u}_i))^2 \right\}. \qquad (1)$$

By virtue of the representer theorem [2], the function $\psi(\cdot)$ can be written as a kernel expansion in terms of available training data, namely, $\psi(\cdot) = \sum_{j=1}^{N} \alpha_j \, \kappa(\cdot, \boldsymbol{u}_j)$. The above optimization problem becomes

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ J(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N} (d_i - \boldsymbol{\alpha}^\top \boldsymbol{\kappa}_i)^2 \right\}. \qquad (2)$$

where $\boldsymbol{\kappa}_i$ is the $(N \times 1)$ vector with j-th entry $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j)$. Online prediction of time series data raises the question of how to process an increasing amount N of observations as new data is collected. To overcome this drawback, several authors have focused on fixed-size models of the form

$$\psi(\cdot) = \sum_{j=1}^{M} \alpha_j \, \kappa(\cdot, \boldsymbol{u}_{\omega_j}). \qquad (3)$$

We call $\mathcal{D} = \{\kappa(\cdot, \boldsymbol{u}_{\omega_j})\}_{j=1}^{M}$ the dictionary, and M the order of the kernel expansion by analogy with linear transversal filters. Online identification of kernel-based models generally relies on a two-stage process at each iteration: a model order control step that updates the dictionary, and a parameter update step. The algorithms developed using this idea include the kernel recursive-least-square (KRLS) algorithm [3], the kernel least-mean-square (KLMS) algorithm [1, 4], the kernel normalized least-mean-square (KNLMS) algorithm and the kernel affine projection (KAPA) algorithm [5, 6, 7]. These methods use more or less sophisticated strategies to decide, at each time instant n, whether $\kappa(\cdot, \boldsymbol{u}_n)$ deserves to be included into the dictionary or not. One of the most informative criteria uses approximate linear dependence condition to test the ability of the dictionary elements to approximate the current input $\kappa(\cdot, \boldsymbol{u}_n)$ linearly [3]. Other well-known criteria include the novelty criterion [8], the coherence criterion [6], the surprise criterion [1], and the quantization criterion [4]. Without loss of generality, we focus on the KLMS algorithm with coherence-sparsification criterion (CS).

It is surprising to note that most, if not all, the existing strategies for dictionary update can only incorporate new elements into the dictionary. This unfortunately means that they cannot discard obsolete kernel functions, within the context of a time-varying environment in particular. Recently, sparsity-promoting regularization was considered within the context of linear adaptive filtering [9, 10, 11]. All these works propose to use, either the ℓ_1-norm of the vector of filter coefficients as a regularization term, or some other related regularizer to limit the induced bias. The optimization procedures consist of subgradient descent method [9], projection onto the ℓ_1 ball [10], or online forward-backward splitting [11]. Surprisingly, this idea was little used within the context of kernel-based adaptive filtering. To the best of our knowledge, only Yukawa suggested in [12] the use

This work was partly supported by the National Natural Science Foundation of China (61271415).

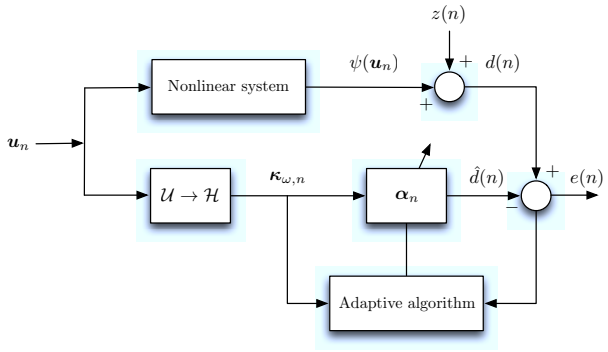

Fig. 1. Kernel-based adaptive system identification.

of ℓ_1-norm regularization with the mean-square error criterion. This study was conducted in the multi-kernel context.

In this paper, we focus the attention on Yukawa's approach [12] in a single-kernel setting. By using a sparsity-promoting regularization term, it allows minor contributors in the kernel dictionary to be automatically discarded. Our aim is to provide theoretical results on the convergence of this approach.

2. KERNEL LMS ALGORITHMS

Several versions of the KLMS algorithm were proposed in the literature, depending on if gradient descent is performed on $\psi(\cdot)$ in the functional space $\mathcal{H}$ by considering problem (1), or on vector $\boldsymbol{\alpha}$ by considering problem (2). The former strategy is considered in [4] for instance, and the latter in [6]. Before presenting them, note that the main drawback of all these approaches is that the growing number of training data $(\boldsymbol{u}_n, d_n)$ needs us to focus on fixed-size models of the form (3), and to define a strategy for updating the dictionary $\mathcal{D}$.

2.1. Dictionary update

Coherence is a fundamental parameter to characterize a dictionary in linear sparse approximation problems. In the kernel-based context, it is defined as [6]

$$\mu = \max_{i \neq j} |\kappa(\boldsymbol{u}_{\omega_i}, \boldsymbol{u}_{\omega_j})| \tag{4}$$

where κ is a unit-norm kernel. Coherence criterion suggests inserting the candidate input $\kappa(\cdot, \boldsymbol{u}_n)$ into the dictionary provided that its coherence remains below a given threshold μ_0

$$\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \mu_0, \tag{5}$$

where μ_0 is a parameter in $[0, 1[$ determining both the level of sparsity and the coherence of the dictionary. Before ending, note that the quantization criterion introduced in [4] consists of comparing the minimum distance of $\boldsymbol{u}_n$ from the elements of $\mathcal{D}$, namely, $\min_j \|\boldsymbol{u}_n - \boldsymbol{u}_{\omega_j}\|$, to a threshold δ_0. Quantization and coherence criteria are equivalent in the case of radial kernels, such as the Gaussian kernel for instance.

2.2. Filter parameter update

At iteration n, upon the arrival of new data, one of the following alternatives holds. If $\kappa(\cdot, \boldsymbol{u}_n)$ does not satisfy the coherence rule (5), the dictionary remains unaltered. On the other hand, if condition (5) is met, $\kappa(\cdot, \boldsymbol{u}_n)$ is inserted into the dictionary where it is now denoted by $\kappa(\cdot, \boldsymbol{u}_{\omega_{M+1}})$. The LMS algorithm applied to the parametric form (2) leads to the following algorithm [6]

- Case 1: $\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| > \mu_0$

$$\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_{n-1} + \eta\, e_n\, \boldsymbol{\kappa}_{\omega,n} \tag{6}$$

- Case 2: $\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \mu_0$

$$\boldsymbol{\alpha}_n = \begin{pmatrix} \boldsymbol{\alpha}_{n-1} \\ 0 \end{pmatrix} + \eta\, e_n\, \boldsymbol{\kappa}_{\omega,n} \tag{7}$$

where $e_n = d_n - \boldsymbol{\alpha}_{n-1}^\top \boldsymbol{\kappa}_{\omega,n}$ represents the estimation error, and $\boldsymbol{\kappa}_{\omega,n}$ the $(M \times 1)$ vector with i-th entry defined by $\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_i})$. The coherence criterion guarantees that the dictionary dimension is finite for any input sequence $\{\boldsymbol{u}_n\}_{n=1}^{\infty}$ due to the compactness of the input space $\mathcal{U}$ [6]. The steady-state behavior and the transient behavior of the KLMS algorithm with Gaussian kernel were carefully studied in [13], for Gaussian input signals. Recursive expressions for the mean-weight-error vector and the mean-square-error were derived. A condition for convergence was proposed in [13, 14].

The KLMS algorithm derived in [4] adopts the Fréchet's notion of differentiability to derive a gradient descent direction with respect to $\psi(\cdot)$ in problem (1), that is,

$$\nabla E\{(d_n - \psi(\boldsymbol{u}_n))^2\} = -2E\{e_n \kappa(\cdot, \boldsymbol{u}_n)\} \tag{8}$$

This leads to the following update equations where, without loss of generality but to be consistent with (6)-(7), we apply the coherence rule rather than the quantization rule [4]

- Case 1: $\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| > \mu_0$

$$\alpha_n(j_0) = \alpha_{n-1}(j_0) + \eta\, e_n \tag{9}$$

  where $j_0 = \arg\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})|$

- Case 2: $\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \mu_0$

$$\boldsymbol{\alpha}_n = \begin{pmatrix} \boldsymbol{\alpha}_{n-1} \\ \eta\, e_n \end{pmatrix} \tag{10}$$

  or, equivalently, $\alpha_n(M+1) = \eta\, e_n$.

The update equation (10) is directly obtained from a stochastic approximation of the gradient (8) in the case where $\kappa(\cdot, \boldsymbol{u}_n)$ is inserted into the dictionary. If not, the heuristic rule (9) is used. It consists of using the correction term ηe_n to update the j_0-th entry of $\boldsymbol{\alpha}_{n-1}$, where $\kappa(\cdot, \boldsymbol{u}_{j_0})$ is the closest dictionary element to $\kappa(\cdot, \boldsymbol{u}_n)$ in the sense of the coherence parameter (4).

We have observed that the functional approach (9)-(10) converges slightly slower than the parametric approach (6)-(7). The use of the heuristic coordinate descent (9) rather than the steepest gradient descent (6), in the most common case where the dictionary has reached a steady-state form, justifies this difference. Because of this, and our understanding of its convergence behavior [13, 14], we shall focus on algorithm (6)-(7) in the following.

3. KERNEL LMS ALGORITHM WITH SPARSITY-PROMOTING REGULARIZATION

In order to possibly remove kernels from the dictionary $\mathcal{D}$, Yukawa proposed to use a sparsity-promoting regularization term with criterion (2), which considers the contribution of each element to the performance of the filter [12]. With this algorithm, minor contributors are automatically discarded. In order to solve the resulting stochastic optimization problem, forward-backward splitting can be performed as follows.

3.1. Forward-backward splitting method

Let us consider the following optimization problem

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \{Q(\boldsymbol{\alpha}) = J(\boldsymbol{\alpha}) + \lambda\Omega(\boldsymbol{\alpha})\} \tag{11}$$

where $J(\cdot)$ is a convex empirical loss function with Lipschitz continuous gradient and Lipschitz constant $1/\eta_0$, $\Omega(\cdot)$ is a convex, continuous, but not necessarily differentiable regularization function, and λ is a regularization parameter. This composite problem has been extensively studied in the literature, and can be solved efficiently using forward-backward splitting. See [15, 16] for an overview. In a nutshell, this approach consists of iteratively minimizing the following quadratic approximation of $Q(\boldsymbol{\alpha})$ at a given point $\boldsymbol{\alpha}_n$

$$\begin{aligned} Q_\eta(\boldsymbol{\alpha}, \boldsymbol{\alpha}_n) = J(\boldsymbol{\alpha}_n) + \nabla J(\boldsymbol{\alpha}_n)^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}_n) \\ + \frac{1}{2\eta}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_n\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}) \end{aligned} \tag{12}$$

since $Q(\boldsymbol{\alpha}) \leq Q_\eta(\boldsymbol{\alpha}, \boldsymbol{\alpha}_n)$ for any $\eta \leq \eta_0$. Function $Q_\eta(\boldsymbol{\alpha}, \boldsymbol{\alpha}_n)$ admits a unique minimizer, denoted by $\boldsymbol{\alpha}_{n+1}$. Simple algebra shows

$$\boldsymbol{\alpha}_{n+1} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ \lambda\Omega(\boldsymbol{\alpha}) + \frac{1}{2\eta}\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}_n\|_2^2 \right\} \tag{13}$$

where $\hat{\boldsymbol{\alpha}}_n = \boldsymbol{\alpha}_n - \eta\nabla J(\boldsymbol{\alpha}_n)$ can be interpreted as an intermediate gradient descent step on the cost function $J(\cdot)$. Problem (13) is called the proximity operator for the regularization function $\Omega(\cdot)$, denoted by $\text{Prox}_{\lambda\eta\Omega(\cdot)}(\cdot)$. Let us recall that convergence of the process (13) to a global minimum is ensured if $1/\eta$ is a Lipschitz constant of $\nabla J(\boldsymbol{\alpha})$. With $J(\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{d} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2$ as in problem (2), a typical condition ensuring convergence of $\boldsymbol{\alpha}_{n+1}$ to a minimizer of problem (11) is to require that [16]

$$0 < \eta < 2/\lambda_{\max}(\boldsymbol{K}^\top\boldsymbol{K}) \tag{14}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue.

Forward-backward splitting is an efficient method for minimizing empirical risk with sparse regularization, which was originally derived for offline learning. A generalization of this algorithm for stochastic optimization, called Fobos, was recently proposed in [17]. Practically, it consists of using a stochastic approximation for the gradient ∇J at each iteration. This online approach can be coupled with the KLMS algorithm. For convenience, in the next subsection, we shall begin by describing the offline setup based on problem (2).

3.2. Application to kernel LMS algorithm

In order to automatically discard irrelevant elements from the dictionary $\mathcal{D}$, we shall now consider the optimization problem (2) with sparsity-promoting convex regularization function $\Omega(\cdot)$

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ Q(\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{d} - \boldsymbol{K}\boldsymbol{\alpha}\|^2 + \lambda\Omega(\boldsymbol{\alpha}) \right\} \tag{15}$$

where $\boldsymbol{K}$ is the $(N\times N)$ Gram matrix with (i,j)-th entry $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j)$. Obviously, problem (15) is of the general form (11) and can be solved with forward-backward splitting method described previously. Two regularization processes are successively considered.

Firstly, we consider the celebrated ℓ_1-norm regularization function defined as $\Omega_1(\boldsymbol{\alpha}) = \sum_j |\alpha(j)|$. It is often used for sparse regression and its proximity operator is separable. The j-th entry of the latter can be expressed as

$$\left(\text{Prox}_{\lambda\|\cdot\|_1}(\boldsymbol{\alpha})\right)(j) = \text{sign}\{\alpha(j)\}\max\{|\alpha(j)| - \lambda, 0\} \tag{16}$$

known as the soft thresholding operator. Secondly, we suggest an adaptive ℓ_1-norm function of the form $\Omega_a(\boldsymbol{\alpha}) = \sum_j w_j|\alpha(j)|$ where the w_j's are weights to be dynamically adjusted. The proximity operator for this regularization function is defined by

$$\begin{aligned} &\left(\text{Prox}_{\lambda\Omega_a(\cdot)}(\boldsymbol{\alpha})\right)(j) \\ &\qquad = \text{sign}\{\alpha(j)\}\max\{|\alpha(j)| - \lambda\, w(j), 0\} \end{aligned} \tag{17}$$

This regularization function has been proven to be more consistent than the usual ℓ_1-norm [18], and tends to reduce the induced bias. The weights are usually chosen as $w(j) = 1/|\alpha^o(j)|$, where $\boldsymbol{\alpha}^o$ is the least-square solution of the problem (2). Since $\boldsymbol{\alpha}^o$ is not available in our case, we chose $w(j) = 1/(|\alpha_{n-1}(j)| + \epsilon_\alpha)$ at each iteration n, where ϵ_α is a small constant to prevent the denominator from vanishing [19]. This technique, also referred to as reweighted least square, is performed at each iteration of the stochastic optimization process. Note that a similar regularization term was used in [9] in order to approximate the ℓ_0-norm.

The pseudocode for KLMS algorithm with sparsity-promoting regularization is provided in Algorithm 1. It can be noticed that the proximity operator is applied after the gradient descent step. The dictionary elements associated with null coefficients in vector $\boldsymbol{\alpha}_n$ are removed. This approach reduces to the KLMS algorithm if $\lambda = 0$.

3.3. Stability in the mean

From now on, the environment is assumed stationary. The system inputs $\boldsymbol{u}(n)$ are zero-mean, independent, and identically distributed Gaussian $(q\times 1)$ vectors. The components of $\boldsymbol{u}(n)$ can, however, be correlated. Let $\boldsymbol{R}_{\boldsymbol{u}\boldsymbol{u}}$ be their autocorrelation matrix. A direct consequence of the above independence assumption is that the kernelized inputs $\boldsymbol{\kappa}_{\omega,n}$ are also statistically independent $(M\times 1)$ vectors.

To conduct the stability analysis in the mean of the KLMS algorithm with the sparsity-inducing regularization (16), we first observe that the update equation can be rewritten as

$$\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_{n-1} + \eta\, e_n\, \boldsymbol{\kappa}_{\omega,n} - \boldsymbol{f}_{n-1} \tag{18}$$

with

$$f_{n-1}(j) = \begin{cases} \lambda\,\text{sgn}(\hat{\alpha}_{n-1}(j)) & \text{if} \quad |\hat{\alpha}_{n-1}(j)| \geq \lambda \\ \hat{\alpha}_{n-1}(j) & \text{otherwise} \end{cases} \tag{19}$$

where $\hat{\boldsymbol{\alpha}}_{n-1} = \boldsymbol{\alpha}_{n-1} + \eta\, e_n\, \boldsymbol{\kappa}_{\omega,n}$. Up to a change of variable in λ, the general form (18)–(19) remains the same for the regularization function (17). It is important to note that the sequence $|f_n(j)|$ is bounded in both cases, by λ and λ/ϵ_α, respectively.

Let $\boldsymbol{v}_n = \boldsymbol{\alpha}^o - \boldsymbol{\alpha}_n$ be the weight-error vector, where $\boldsymbol{\alpha}^o$ is the solution of the non-regularized problem ($\lambda = 0$). In order to make the study of the stochastic behavior of $\boldsymbol{v}_n$ mathematically feasible, the following simplifying assumption is required. It has been successfully employed in several adaptive filter analyses [13].

Assumption 1. *The modified independence assumption (MIA) considers that $\boldsymbol{\kappa}_{\omega,n}\boldsymbol{\kappa}_{\omega,n}^\top$ is statistically independent of $\boldsymbol{v}(n)$.*

The following theorem guarantees the asymptotic mean stability of the KLMS algorithm with sparsity-promoting regularization (16) and (17).

Theorem 1. *Assume MIA holds. For any initial condition $\boldsymbol{\alpha}_0$, the regularized KLMS algorithm asymptotically converges in the mean if the step-size is chosen to satisfy*

$$0 < \eta < 2/\lambda_{\max}(\boldsymbol{R}_{\kappa\kappa}) \tag{20}$$

To prove this theorem, let us rewrite the equation (18) as follows

$$\boldsymbol{v}_n = \boldsymbol{v}_{n-1} - \eta\,\boldsymbol{\kappa}_{\omega,n}(\boldsymbol{\kappa}_{\omega,n}^{\top}\boldsymbol{v}_{n-1} + z(n)) + \boldsymbol{f}_{n-1} \tag{21}$$

Taking the expected value of both sides and using the MIA, the recursion (21) leads to

$$\begin{aligned} E\{\boldsymbol{v}_n\} = (\boldsymbol{I} - \eta\,\boldsymbol{R}_{\kappa\kappa})^n\, E\{\boldsymbol{v}_0\} \\ + \sum_{i=0}^{n-1} (\boldsymbol{I} - \eta\,\boldsymbol{R}_{\kappa\kappa})^i\, E\{\boldsymbol{f}_{n-i-1}\} \end{aligned} \tag{22}$$

where $\boldsymbol{R}_{\kappa\kappa}$ is the autocorrelation matrix of $\boldsymbol{\kappa}_{\omega,n}$, and $E\{\boldsymbol{v}_0\}$ is the initial condition. To prove the convergence of $E\{\boldsymbol{v}_n\}$, we have to demonstrate that both terms on the r.h.s. of this expression converge as n goes to infinity. The first term converges to zero if we can ensure that $\delta \triangleq \|\boldsymbol{I} - \eta\,\boldsymbol{R}_{\kappa\kappa}\| < 1$. We can easily check that this condition is met for any step-size η satisfying (20) since

$$\delta = |1 - \eta\,\lambda_{\max}(\boldsymbol{R}_{\kappa\kappa})| \tag{23}$$

Let us show now that condition (20) also implies that the second term on the r.h.s. of equation (22) asymptotically converges to a finite value, thus leading to the overall convergence of this recursion. Each term of this series is bounded because

$$\begin{aligned} \|(\boldsymbol{I} - \eta\,\boldsymbol{R}_{\kappa\kappa})^i\, E\{\boldsymbol{f}_{n-i-1}\}\| \\ \leq \|(\boldsymbol{I} - \eta\,\boldsymbol{R}_{\kappa\kappa})\|^i\, E\{\|\boldsymbol{f}_{n-i-1}\|\} \\ \leq \sqrt{M}\,\delta^i\, f_{\max} \end{aligned} \tag{24}$$

where $f_{\max} = \lambda$ or λ/ϵ_α, depending if one uses the regularization function (16) or (17). Condition (20) implies that $\delta < 1$ and, as a consequence,

$$\sum_{i=0}^{n-1} \|(\boldsymbol{I} - \eta\,\boldsymbol{R}_{\kappa\kappa})^i\, E\{\boldsymbol{f}_{n-i-1}\}\| \leq \frac{\sqrt{M}\,f_{\max}}{1-\delta} \tag{25}$$

To summarize, because the two terms of the r.h.s. of equation (22) asymptotically converge to finite values, we conclude that $E\{\boldsymbol{v}_n\}$ will converge to a steady-state value. Finally, we shall now evaluate the constant δ. Following [13], it can be shown that

$$\boldsymbol{R}_{\kappa\kappa} = (r_{\text{md}} - r_{\text{od}})\,\boldsymbol{I} + r_{\text{od}}\,\boldsymbol{1}\boldsymbol{1}^{\top} \tag{26}$$

for any reproducing kernel κ, with $\boldsymbol{1}$ the all-one vector, and

$$\begin{aligned} r_{\text{md}} &= E\{\kappa^2(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_i})\} \\ r_{\text{od}} &= E\{\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_i})\,\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})\}, \quad i \neq j \end{aligned} \tag{27}$$

the main-diagonal and off-diagonal entries of $\boldsymbol{R}_{\kappa\kappa}$. Simple algebra shows that $\lambda_{\max}(\boldsymbol{R}_{\kappa\kappa}) = r_{\text{md}} + (M-1)r_{\text{od}}$.

4. EXPERIMENTAL RESULTS

This section presents simulation examples to illustrate the algorithm performance. We used the coherence sparsification criterion (CS) with the same threshold μ_0 for all the methods, in order to compare the dictionary sizes. KRLS served as a reference for comparing KLMS with CS (KLMS-CS), KLMS-CS with ℓ_1-norm regularization (KLMS-CSL1), and adaptive ℓ_1-norm (KLMS-CSAL1). The Gaussian kernel defined as $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2/2\beta_0^2)$ with the same the kernel bandwidth β_0 was used with all the methods mentioned above.

Algorithm 1 KLMS with sparsity-inducing regularization.

Initialization
  Select the step size η, and the parameters of the kernel;
  Insert $\kappa(\cdot, \boldsymbol{u}_0)$ into the dictionary, $\boldsymbol{\alpha}_0 = 0$.
for $n = 1, 2, \cdots$, do
  if $\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| > \mu_0$
    Compute $\boldsymbol{\kappa}_{\omega,n}$ and $\boldsymbol{\alpha}_n$ using equation (6);
  elseif $\max_{j=1,\ldots,M} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \mu_0$
    Incorporate $\kappa(\cdot, \boldsymbol{u}_n)$ into the dictionary;
    Compute $\boldsymbol{\kappa}_{\omega,n}$ and $\boldsymbol{\alpha}_n$ using equation (7);
  end if
  $\boldsymbol{\alpha}_n = \text{Prox}_{\lambda\eta\Omega(\cdot)}(\boldsymbol{\alpha}_n)$
  Remove $\kappa(\cdot, \boldsymbol{u}_{\omega_j})$ from the dictionary if $\alpha_n(j) = 0$.
end for

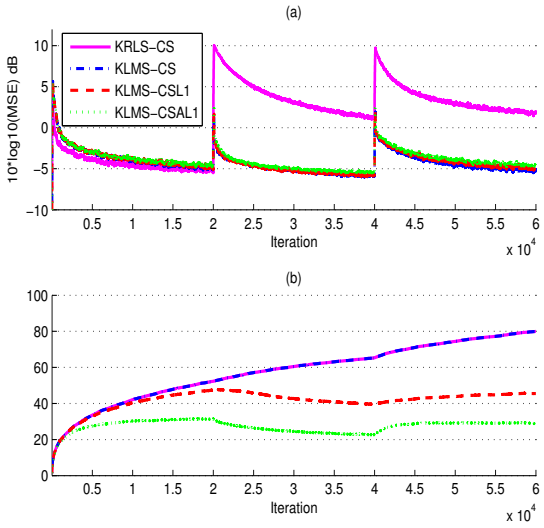

Fig. 2. (a) Ensemble-average learning curves; (b) Evolution of size of the dictionaries.

In this experiment described in [20], an input Gaussian signal d_n with 3 distinct mean values $(-4, 0, 4)$ and unit variance was considered to generate 3 segments of 2×10^4 data. It was transmitted in a channel consisting of the linear part $t_n = -0.8d_n + 0.7d_{n-1}$ and the memoryless nonlinear part $q_n = t_n + 0.25t_n^2 + 0.11t_n^3$. Then, this signal was corrupted with an additive white Gaussian noise and observed as $\boldsymbol{u}_n$. The equalizer length and the equalization time delay were set to 5 and 2, respectively. The SNR was set to 15 dB. The kernel bandwidth β_0 and step size η of the KLMS were respectively set to 3.536 and 0.1. Parameter λ was set to 0.0005 for KLMS-CSL1 and KLMS-CSAL1. The coherence sparsification threshold μ_0 was set to 0.3 for all the methods. The simulation experiments were conducted on 6×10^4 samples, and averaged over 200 Monte Carlo runs. The ensemble-average learning curves and the evolution of size of dictionary are shown in Figure 2. Observe that each change in the statistics of the input signal causes the KLMS-CS and the RLS-CS algorithms to insert new elements into the dictionaries. As expected, KLMS-CSL1 and KLMS-CSAL1 were able to discard the obsolete dictionary elements, with an advantage to KLMS-CSAL1.

5. REFERENCES

[1] W. Liu, J. Principe, and S. Haykin, *Kernel Adaptive Filtering*, Wiley, New Jersey, 2010.

[2] B. Schölkopf, R. Herbrich, and R. Williamson, "A generalized representer theorem," Tech. Rep. NC2-TR-2000-81, NeuroCOLT, Royal Holloway College, University of London, UK, 2000.

[3] Y. Engel, S. Mannor, and R. Meir, "Kernel recursive least squares," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[4] B. Chen, S. Zhao, P. Zhu, and J. Principe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, Jan. 2012.

[5] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE ISIT*, 2007, pp. 956–960.

[6] C. Richard, J.-C M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, March 2009.

[7] K. Slavakis and S. Theodoridis, "Sliding window generalized kernel affine projection algorithm using projection mappings," in *EURASIP Journal on Advances in Signal Processing*, 2008.

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

[9] Y. Chen, Y. Gu, and A. O. Hero, "Sparse lms for system identification," in *Proc. IEEE ICASSP*, 2009, pp. 3125–3128.

[10] K. Slavakis, Y. Kopsinis, and S. Theodoridis, "Adaptive algorithm for sparse system identification using projections onto weighted l1 balls," in *Proc. IEEE ICASSP*, 2010, pp. 3742–3745.

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

[12] M. Yukawa, "Multikernel adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4672–4682, Sept. 2012.

[13] 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 Transactions on Signal Processing*, vol. 60, no. 5, pp. 2208–2222, 2012.

[14] C. Richard and J.-C M. Bermudez, "Closed-form conditions for convergence of the gaussian kernel-least-mean-square algorithm," in *Proc. Asilomar*, 2012.

[15] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, *Optimization for Machine Learning*, chapter Convex optimization with sparsity-inducing norms, MIT Press, 2011.

[16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[17] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, December 2009.

[18] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[19] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[20] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Online learning in reproducing kernel hilbert spaces," *Signal Processing, E-Reference*, 2012 (to appear).