A STOCHASTIC BEHAVIOR ANALYSIS OF STOCHASTIC RESTRICTED-GRADIENT DESCENT ALGORITHM IN REPRODUCING KERNEL HILBERT SPACES

Masa-aki Takizawa[†], *Masahiro Yukawa*[†*], *and Cédric Richard*[‡]

[†] Department of Electronics and Electrical Engineering, Keio University, Japan
[‡] Université de Nice Sophia-Antipolis, CNRS, France

ABSTRACT

This paper presents a stochastic behavior analysis of a kernel-based stochastic restricted-gradient descent method. The restricted gradient gives a steepest ascent direction within the so-called dictionary subspace. The analysis provides the transient and steady state performance in the mean squared error criterion. It also includes stability conditions in the mean and mean-square sense. The present study is based on the analysis of the kernel normalized least mean square (KNLMS) algorithm initially proposed by Chen *et al.* Simulation results validate the analysis.

Index Terms— kernel adaptive filter, reproducing kernel Hilbert space, the KLMS algorithm, performance analysis

1. INTRODUCTION

Kernel adaptive filtering [1] is an attractive approach for nonlinear estimation problems based on the theory of reproducing kernel Hilbert space (RKHS), and a number of kernel adaptive filtering algorithms have been proposed [2–8]. The existing kernel adaptive filtering algorithms are classified into two general categories according to the space in which optimization is performed [6]: (i) the functional-space approach (e.g., [2, 5, 7]) and (ii) the parameter-space approach (e.g., [4, 6, 9]). The kernel normalized least mean square (KNLMS) algorithm is a representative example of the parameter-space approach and its stochastic behavior analyses have been presented in [10–12]. The analyses have clarified the transient and steady-state performance in the mean squared error (MSE). A stochastic restricted-gradient descent algorithm studied in the present work is a functional-space counterpart of the KNLMS algorithm. We call it the constrained kernel least mean square (CKLMS) algorithm to distinguish it from the KLMS algorithm proposed in [13]. A primitive question is whether it is possible to give the same analyses as in [10–12] for the stochastic restricted-gradient descent algorithm. If this is possible, it will provide a theoretical basis to compare the performances of KNLMS and CKLMS. This will eventually give a new insight into the relationship between the two classes of kernel adaptive filtering algorithms.

To clarify the orientation of the CKLMS algorithm in the kernel adaptive filtering researches, let us give a short note on the functional-space approach. Dictionary sparsification is a common issue of kernel adaptive filtering [1, 3, 4, 14]. The KLMS algorithm [13] updates the filter only when the current input datum is added into the dictionary and this would cause severe performance degradations. A systematic scheme which eliminates such a limitation has been proposed in [15] under the name of *hyperplane projection along affine subspace (HYPASS)*. The HYPASS algorithm updates the filter using the projection onto the zero-instantaneous-error hyperplane along the so-called *dictionary subspace* $\mathcal{M}$, the subspace spanned by the dictionary elements. This is achieved by projecting the gradient direction onto $\mathcal{M}$. In a nutshell, HYPASS is the NLMS algorithm operated in the dictionary subspace $\mathcal{M}$. CKLMS is actually an LMS counterpart of HYPASS and we consider this LMS-based algorithm to make the analysis feasible. In [16] and [7], the mean square convergence analysis and the theoretical steady-state MSE have been presented for the KLMS and Quantized KLMS algorithms, respectively. However, transient performance analyses have not yet been reported due to the difficulty in treating the growing number of dictionary elements.

In this paper, we present a stochastic behavior analysis of the CKLMS algorithm with a Gaussian kernel under i.i.d. random inputs based on the framework presented in [12]. CKLMS is derived by using the *restricted gradient* which gives a steepest ascent direction within the dictionary subspace $\mathcal{M}$. The analysis provides theoretical MSEs during the transient phase as well as at the steady-state. We also derive stability conditions in the mean and mean-square sense. The key ingredients for the analysis are the restricted gradient and the isomorphism between the dictionary subspace $\mathcal{M}$ and a Euclidean space; these were also the key when the first and second authors developed a sparse version of HYPASS in [17]. The validity of the analysis is illustrated by simulations.

2. PRELIMINARIES

We address an adaptive estimation problem of a nonlinear system ψ with sequentially arriving input signals $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^L$, and its noisy output $d := \psi(\boldsymbol{u}) + \nu \in \mathbb{R}$, where $\boldsymbol{u}$ is assumed an i.i.d. random vector and ν is a zero-mean additive noise uncorrelated with any other signals. The function ψ is modeled as an element of the RKHS $\mathcal{H}$ associated with a Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right)$, $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$, where $\sigma > 0$ is the kernel parameter. We denote by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ the canonical inner product and the norm defined in $\mathbb{R}^L$, respectively, and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $\|\cdot\|_{\mathcal{H}}$ those in $\mathcal{H}$. A kernel adaptive filter is given as a finite order filter:

$$\varphi_n := \sum_{j \in \mathcal{J}} \alpha_j^{(n)} \kappa(\cdot, \boldsymbol{u}_j),\ n \in \mathbb{N}, \tag{1}$$

where $\alpha_j^{(n)} \in \mathbb{R}$ are the filter coefficients and $\mathcal{J} := \{j_1, j_2, \cdots, j_r\}$ indicates the dictionary $\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j \in \mathcal{J}}$; n is the time index. Without loss of generality, we assume that the dictionary is a linearly independent set so that it spans an r dimensional subspace

$$\mathcal{M} := \operatorname{span}\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j \in \mathcal{J}} \subset \mathcal{H}, \tag{2}$$

which is called the *dictionary subspace*. Although the dictionary is updated typically during the learning process, we assume that the dictionary is fixed to make the analysis tractable.

*This work was partially supported by JSPS Grants-in-Aid (24760292).

The instantaneous error at time instant n is defined as $e_n := d_n - \langle \varphi, \kappa(\cdot, \boldsymbol{u}_n)\rangle_{\mathcal{H}} = d_n - \langle \boldsymbol{\alpha}, \boldsymbol{\kappa}_n\rangle$, where $\boldsymbol{\kappa}_n = [\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_1}), \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_2}), \cdots, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_r})]^{\mathsf{T}}$ is the vector of the kernelized input and $\boldsymbol{\alpha} = [\alpha_{j_1}, \alpha_{j_2}, \cdots, \alpha_{j_r}]^{\mathsf{T}}$ is the coefficient vector. The MSE cost function, with respect to the coefficient vector $\boldsymbol{\alpha}$, is given by

$$J(\boldsymbol{\alpha}) := E(e_n^2(\boldsymbol{\alpha})) = E(d_n^2) + \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{R}_\kappa \boldsymbol{\alpha} - 2\boldsymbol{p}^{\mathsf{T}}\boldsymbol{\alpha}, \tag{3}$$

where $\boldsymbol{R}_\kappa := E(\boldsymbol{\kappa}_n \boldsymbol{\kappa}_n^{\mathsf{T}})$ is the autocorrelation matrix of the kernelized input $\boldsymbol{\kappa}_n$ and $\boldsymbol{p} := E(d_n \boldsymbol{\kappa}_n)$ is the cross-correlation vector between $\boldsymbol{\kappa}_n$ and d_n. With the optimization in RKHS in mind, the MSE, with respect to φ, is given by:

$$\begin{aligned} J(\varphi) := E(e_n^2(\varphi)) =& E(d_n^2) + E(\langle \varphi, \kappa(\cdot, \boldsymbol{u}_n)\rangle_{\mathcal{H}}^2) \\ & - 2E(d_n \langle \varphi, \kappa(\cdot, \boldsymbol{u}_n)\rangle_{\mathcal{H}}). \end{aligned} \tag{4}$$

While the KNLMS algorithm optimizes $J(\boldsymbol{\alpha})$ in the Euclidean space $\mathbb{R}^L$, the constrained KLMS algorithm presented in the following section optimizes $J(\varphi)$ in the RKHS $\mathcal{H}$ under the restriction to the dictionary subspace $\mathcal{M}$, or in short, it optimizes $J(\varphi)$ in $\mathcal{M}$. Referring to [2], the stochastic gradient descent method for $J(\varphi)$ in $\mathcal{H}$ updates the filter φ_n along the 'line' (one dimensional subspace) spanned by the singleton $\{\kappa(\cdot, \boldsymbol{u}_n)\}$. This implies that the filter is updated only when $\kappa(\cdot, \boldsymbol{u}_n)$ is added into the dictionary, because otherwise $\varphi_n + \alpha\kappa(\cdot, \boldsymbol{u}_n) \notin \mathcal{M}$ for any $\alpha \neq 0$. We thus present the restricted gradient, which was initially introduced in [17], and derive the constrained KLMS algorithm in the following section.

3. THE CONSTRAINED KLMS ALGORITHM

The ordinary gradient of $J(\boldsymbol{\alpha})$ in $\mathbb{R}^r$ is given by $\nabla J(\boldsymbol{\alpha}) = 2(\boldsymbol{R}_\kappa \boldsymbol{\alpha} - \boldsymbol{p})$. Given any positive definite matrix $\boldsymbol{Q}$, $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\boldsymbol{Q}} := \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{y}$ and $\|\boldsymbol{x}\|_{\boldsymbol{Q}} := \sqrt{\boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{x}}$ define an inner product and its induced norm, respectively. The $\boldsymbol{G}$-gradient of (3) with the inner product $\langle \cdot, \cdot\rangle_{\boldsymbol{G}}$ is defined as [17]

$$\nabla_{\boldsymbol{G}} J(\boldsymbol{\alpha}) := \boldsymbol{G}^{-1} \nabla J(\boldsymbol{\alpha}), \tag{5}$$

where $[\boldsymbol{G}]_{\ell,m} = \kappa(\boldsymbol{u}_{j_\ell}, \boldsymbol{u}_{j_m})$ for $1 \le \ell, m \le r$ is the Gram matrix.[1]

The functional Hilbert space $\left(\mathcal{M}, \langle \cdot, \cdot\rangle_{\mathcal{H}}\right)$ of dimension r is isomorphic to the Hilbert space $\left(\mathbb{R}^r, \langle \cdot, \cdot\rangle_{\boldsymbol{G}}\right)$ under the correspondence (see Fig. 1)

$$\mathcal{M} \ni \varphi := \sum_{j \in \mathcal{J}} \alpha_j \kappa(\cdot, \boldsymbol{u}_j) \longleftrightarrow \boldsymbol{\alpha} := [\alpha_{j_1}, \cdots, \alpha_{j_r}]^{\mathsf{T}} \in \mathbb{R}^r. \tag{6}$$

Note here that the isomorphism as Hilbert spaces includes, in addition to the one-to-one correspondence between the elements, the preservation of the inner product; i.e., $\langle \varphi_1, \varphi_2\rangle_{\mathcal{H}} = \langle \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2\rangle_{\boldsymbol{G}}$ for any $\varphi_1 \longleftrightarrow \boldsymbol{\alpha}_1$ and $\varphi_2 \longleftrightarrow \boldsymbol{\alpha}_2$. Under the correspondence in (6), the restricted gradient $\nabla_{|\mathcal{M}} J(\varphi)$ is defined, through the $\boldsymbol{G}$-gradient in $\mathbb{R}^L$, as follows [17]:

$$\nabla_{|\mathcal{M}} J(\varphi) \longleftrightarrow \nabla_{\boldsymbol{G}} J(\boldsymbol{\alpha}) = \boldsymbol{G}^{-1} \nabla J(\boldsymbol{\alpha}). \tag{7}$$

The *restricted gradient* $\nabla_{|\mathcal{M}} J(\varphi)$ gives the steepest ascent direction, within the dictionary subspace $\mathcal{M}$, of the tangent plane of the functional (4) at the point φ. See the derivation of the restricted gradient in [17]. An instantaneous approximation of the restricted gradient $\nabla_{|\mathcal{M}} J(\varphi_n) \longleftrightarrow \nabla_{\boldsymbol{G}} J(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n := [\alpha_{j_1}^{(n)}, \alpha_{j_2}^{(n)}, \cdots, \alpha_{j_r}^{(n)}]^{\mathsf{T}} \in \mathbb{R}^r$ is given by $\tilde{\nabla}_{|\mathcal{M}} J(\varphi_n) \longleftrightarrow \tilde{\nabla}_{\boldsymbol{G}} J(\boldsymbol{\alpha}_n) := \boldsymbol{G}^{-1}\tilde{\nabla} J(\boldsymbol{\alpha}_n) := 2\boldsymbol{G}^{-1}(\boldsymbol{\kappa}_n \boldsymbol{\kappa}_n^{\mathsf{T}} \boldsymbol{\alpha}_n - d_n \boldsymbol{\kappa}_n) = -2e_n \boldsymbol{G}^{-1}\boldsymbol{\kappa}_n$. Hence, for the initial vector $\boldsymbol{\alpha}_0 := \boldsymbol{0}$, the stochastic restricted-gradient descent method, which we call the *constrained KLMS (CKLMS) algorithm*, is given by

$$\boldsymbol{\alpha}_{n+1} := \boldsymbol{\alpha}_n - \frac{\eta}{2}\tilde{\nabla}_{\boldsymbol{G}} J(\boldsymbol{\alpha}_n) = \boldsymbol{\alpha}_n + \eta e_n \boldsymbol{G}^{-1}\boldsymbol{\kappa}_n, \quad n \in \mathbb{N}, \tag{8}$$

where $\eta > 0$ is the step size. The CKLMS algorithm (8) requires r^2 complexity for each time update, and this would make a significant impact on the overall complexity of the algorithm. In [15, 18], a simple selective-updating idea for complexity reduction without serious performance degradations has been presented; it will be shown in Section 5 that the selective-updating works well. We finally remark that the metric $\boldsymbol{G}$ is naturally derived from the inner product $\langle \cdot, \cdot\rangle_{\mathcal{H}}$ in the functional space $\mathcal{H}$.

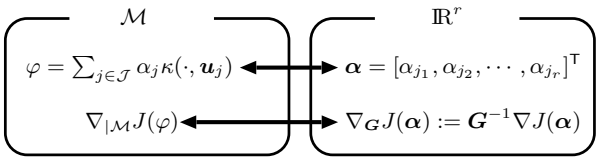

Fig. 1. The isomorphism between $\mathbb{R}^r$ and $\mathcal{M}$ and the restricted gradient.

4. PERFORMANCE ANALYSIS

4.1. Key idea and assumption

We derive a theoretical MSE and stability conditions for CKLMS given by (8) with Gaussian kernel, given the dictionary $\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j\in\mathcal{J}}$. Left-multiplying both-sides of (8) by the square root $\boldsymbol{G}^{\frac{1}{2}}$ of $\boldsymbol{G}$ yields[2]

$$\tilde{\boldsymbol{\alpha}}_{n+1} = \tilde{\boldsymbol{\alpha}}_n + \eta e_n \tilde{\boldsymbol{\kappa}}_n, \tag{9}$$

where $\tilde{\boldsymbol{\kappa}}_n = \boldsymbol{G}^{-\frac{1}{2}}\boldsymbol{\kappa}_n$, $\tilde{\boldsymbol{\alpha}}_n = \boldsymbol{G}^{\frac{1}{2}}\boldsymbol{\alpha}_n$. The cost function $J(\boldsymbol{\alpha})$ in (3) can be rewritten by

$$(J(\boldsymbol{\alpha}) =)\ \tilde{J}(\tilde{\boldsymbol{\alpha}}) = E(d_n^2) + \tilde{\boldsymbol{\alpha}}^{\mathsf{T}}\tilde{\boldsymbol{R}}_\kappa \tilde{\boldsymbol{\alpha}} - 2\tilde{\boldsymbol{p}}^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}, \tag{10}$$

as a function of $\tilde{\boldsymbol{\alpha}} := \boldsymbol{G}^{\frac{1}{2}}\boldsymbol{\alpha}$, and (9) can be regarded as a stochastic gradient descent method for this cost function $\tilde{J}(\tilde{\boldsymbol{\alpha}})$. Here

$$\tilde{\boldsymbol{R}}_\kappa := E(\tilde{\boldsymbol{\kappa}}_n \tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}) = \boldsymbol{G}^{-\frac{1}{2}}\boldsymbol{R}_\kappa \boldsymbol{G}^{-\frac{1}{2}}, \tag{11}$$

and

$$\tilde{\boldsymbol{p}} := E(d_n \tilde{\boldsymbol{\kappa}}_n) = \boldsymbol{G}^{-\frac{1}{2}}\boldsymbol{p}, \tag{12}$$

are the autocorrelation matrix and the cross-correlation vector for the modified vector $\tilde{\boldsymbol{\kappa}}_n$, respectively.

As $\tilde{\boldsymbol{R}}_\kappa$ is positive definite provided that $\boldsymbol{R}_\kappa$ is non-singular, the optimum weight vector is given by

$$\tilde{\boldsymbol{\alpha}}^* := \tilde{\boldsymbol{R}}_\kappa^{-1}\tilde{\boldsymbol{p}}, \tag{13}$$

and with $\tilde{\boldsymbol{\alpha}}^*$, we define the weight error vector

$$\tilde{\boldsymbol{v}}_n := \tilde{\boldsymbol{\alpha}}_n - \tilde{\boldsymbol{\alpha}}^*. \tag{14}$$

[1] The Gram matrix $\boldsymbol{G}$ is ensured to be positive definite due to the assumption that the elements of the dictionary are linearly independent. The definition of the $\boldsymbol{G}$-gradient is validated by observing that $\langle \boldsymbol{\beta} - \boldsymbol{\alpha}, \nabla_{\boldsymbol{G}} J(\boldsymbol{\alpha})\rangle_{\boldsymbol{G}} + J(\boldsymbol{\alpha}) = \langle \boldsymbol{\beta} - \boldsymbol{\alpha}, \nabla J(\boldsymbol{\alpha})\rangle + J(\boldsymbol{\alpha}) \le J(\boldsymbol{\beta})$ for any $\boldsymbol{\beta} \in \mathbb{R}^L$.

[2] For any positive semi-definite matrix $\boldsymbol{Q}$, there exists a unique square root $\boldsymbol{Q}^{\frac{1}{2}}$ satisfying $\boldsymbol{Q} = \boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{Q}^{\frac{1}{2}}$.

In the present analysis, $\tilde{\boldsymbol{\kappa}}_n\tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}$ needs to be independent of $\tilde{\boldsymbol{v}}_n$, which is guaranteed by making the following conditioned modified independence assumption (CMIA) [12].

Assumption 1 *$\boldsymbol{\kappa}_n\boldsymbol{\kappa}_n^{\mathsf{T}}$ is independent of $\boldsymbol{v}_n(=\boldsymbol{G}^{-\frac{1}{2}}\tilde{\boldsymbol{v}}_n)$.*

4.2. Mean weight error analysis

The estimation error can be expressed by

$$e_n = d_n - \tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}\tilde{\boldsymbol{v}}_n - \tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}^*. \tag{15}$$

Substituting (15) to (9), we obtain the recursive expression for $\tilde{\boldsymbol{v}}_n$:

$$\tilde{\boldsymbol{v}}_{n+1} = \tilde{\boldsymbol{v}}_n + \eta d_n\tilde{\boldsymbol{\kappa}}_n - \eta\tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}\tilde{\boldsymbol{v}}_n\tilde{\boldsymbol{\kappa}}_n - \eta\tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}^*\tilde{\boldsymbol{\kappa}}_n. \tag{16}$$

Using CMIA, we obtain the mean weight error model

$$E(\tilde{\boldsymbol{v}}_{n+1}) = (\boldsymbol{I}_r - \eta\tilde{\boldsymbol{R}}_\kappa)E(\tilde{\boldsymbol{v}}_n), \tag{17}$$

where $\boldsymbol{I}_r$ denotes the $r\times r$ identity matrix for any positive integer r. Let the input $\boldsymbol{u}_n$ be a random vector following a Gaussian distribution with zero mean and the covariance matrix $\boldsymbol{R}_u := E(\boldsymbol{u}_n\boldsymbol{u}_n^{\mathsf{T}})$. Then, the (ℓ, m) component ($1\le\ell, m\le r$) of the autocorrelation matrix $\boldsymbol{R}_\kappa$ of $\boldsymbol{\kappa}_n$ is given by [12]:

$$[\boldsymbol{R}_\kappa]_{\ell,m} = |\boldsymbol{I}_L + \frac{2}{\sigma^2}\boldsymbol{R}_u|^{-\frac{1}{2}} \exp\left[-\frac{1}{4\sigma^2}\left(2\|\bar{\boldsymbol{u}}_{\ell m}\|^{(2)} - \|\bar{\boldsymbol{u}}_{\ell m}\|^2_{\left(\boldsymbol{I}_L+\frac{\sigma^2}{2}\boldsymbol{R}_u^{-1}\right)^{-1}}\right)\right],$$

where $\bar{\boldsymbol{u}}_{\ell m} = \boldsymbol{u}_{j_\ell} + \boldsymbol{u}_{j_m}$, $\|\bar{\boldsymbol{u}}_{\ell m}\|^{(2)} = \|\boldsymbol{u}_{j_\ell}\|^2 + \|\boldsymbol{u}_{j_m}\|^2$, and $|\cdot|$ stands for determinant.

From the recursion in (17), we obtain the mean stability condition of CKLMS as follows.

Theorem 1 (Stability in the mean) *Assume CMIA holds. Then, for any initial condition, given dictionary $\{\kappa(\cdot,\boldsymbol{u}_j)\}_{j\in\mathcal{J}}$, the CKLMS algorithm asymptotically converges in the mean if the step size is chosen to satisfy*

$$0 < \eta < \frac{2}{\lambda_{\max}(\tilde{\boldsymbol{R}}_\kappa)}, \tag{18}$$

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

Proof: It is clear from the well-known mean stability results (see, e.g., [19]).

4.3. Mean-square error analysis

Squaring (15) and taking its expectation under CMIA, the MSE (10) of CKLMS can be rewritten as

$$\tilde{J}(\tilde{\boldsymbol{\alpha}}_n) = J_{\min} + \mathrm{tr}(\tilde{\boldsymbol{R}}_\kappa\tilde{\boldsymbol{C}}_n), \tag{19}$$

where $\tilde{\boldsymbol{C}}_n := E(\tilde{\boldsymbol{v}}_n\tilde{\boldsymbol{v}}_n^{\mathsf{T}})$ is the correlation matrix of $\tilde{\boldsymbol{v}}_n$ and $J_{\min} := E(d_n^2) - \tilde{\boldsymbol{p}}^{\mathsf{T}}\tilde{\boldsymbol{R}}_\kappa^{-1}\tilde{\boldsymbol{p}}$ is the minimum MSE. We assume that $e_n^* := d_n - \tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}\tilde{\boldsymbol{h}}^*$ and $\tilde{\boldsymbol{\kappa}}_n\tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}$ are uncorrelated and that $E(e_n^*)\approx 0$. Following the arguments in [10, Section Ⅲ. D] with $\boldsymbol{\kappa}_\omega$ and $\boldsymbol{v}$ replaced respectively by $\tilde{\boldsymbol{\kappa}}$ and $\tilde{\boldsymbol{v}}$, we arrive, with simple manipulations, at the following recursion:

$$\tilde{\boldsymbol{C}}_{n+1} \approx \tilde{\boldsymbol{C}}_n + \eta^2(\tilde{\boldsymbol{T}}_n + J_{\min}\tilde{\boldsymbol{R}}_\kappa) - \eta(\tilde{\boldsymbol{R}}_\kappa\tilde{\boldsymbol{C}}_n + \tilde{\boldsymbol{C}}_n\tilde{\boldsymbol{R}}_\kappa), \tag{20}$$

where $\tilde{\boldsymbol{T}}_n := E(\tilde{\boldsymbol{\kappa}}_n\tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}}\tilde{\boldsymbol{v}}_n\tilde{\boldsymbol{v}}_n^{\mathsf{T}}\tilde{\boldsymbol{\kappa}}_n\tilde{\boldsymbol{\kappa}}_n^{\mathsf{T}})$ and its (ℓ, m) component can be approximated as

$$[\tilde{\boldsymbol{T}}_n]_{\ell,m} \approx \mathrm{tr}(\tilde{\boldsymbol{S}}_{\ell,m}\tilde{\boldsymbol{C}}_n), \quad 1\le\ell, m\le r. \tag{21}$$

Here, the (p, q) component ($1\le p, q\le r$) of $\tilde{\boldsymbol{S}}_{\ell,m}$ is defined as

$$[\tilde{\boldsymbol{S}}_{\ell,m}]_{p,q} := E(\tilde{\kappa}_{n,\ell}\tilde{\kappa}_{n,m}\tilde{\kappa}_{n,p}\tilde{\kappa}_{n,q}) = \boldsymbol{g}_\ell^{\mathsf{T}}\boldsymbol{H}_{m,p}\,\boldsymbol{g}_q, \tag{22}$$

where $\tilde{\kappa}_{n,\ell} := [\tilde{\boldsymbol{\kappa}}_n]_\ell$, $\boldsymbol{g}_\ell$ ($1\le\ell\le r$) is the ℓ-th column vector of $\boldsymbol{G}^{-\frac{1}{2}}$, and $\boldsymbol{H}_{m,p} := E(\boldsymbol{\kappa}_n\boldsymbol{\kappa}_n^{\mathsf{T}}\boldsymbol{g}_m\boldsymbol{g}_p^{\mathsf{T}}\boldsymbol{\kappa}_n\boldsymbol{\kappa}_n^{\mathsf{T}})$. The approximation in (21) can be developed by following the arguments in [12, Section 3.3] with $\boldsymbol{\kappa}_\omega$ and $\boldsymbol{v}_\omega$ replaced by $\tilde{\boldsymbol{\kappa}}$ and $\tilde{\boldsymbol{v}}$, respectively. Finally, the (i, j) component of $\boldsymbol{H}_{m,p}$ can be written as

$$[\boldsymbol{H}_{m,p}]_{i,j} = \boldsymbol{g}_m^{\mathsf{T}}\boldsymbol{S}_{i,j}\boldsymbol{g}_p, \quad 1\le i, j\le r, \tag{23}$$

where $[\boldsymbol{S}_{i,j}]_{s,t} := E(\kappa_{n,i}\kappa_{n,j}\kappa_{n,s}\kappa_{n,t})$, $1\le s, t\le r$, with $\kappa_{n,i} := \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_i})$ can be computed by [12, Eq. (35)].

Let us now establish the mean-square stability condition and derive the steady-state MSE. Due to the presence of $\tilde{\boldsymbol{R}}_\kappa\tilde{\boldsymbol{C}}_n + \tilde{\boldsymbol{C}}_n\tilde{\boldsymbol{R}}_\kappa$ in (20), we exploit the lexicographic representation of $\tilde{\boldsymbol{C}}_n$, i.e, the columns of each matrix are stacked on top of each other into a vector. The recursion (20) can be rewritten as

$$\tilde{\boldsymbol{c}}_{n+1} = \boldsymbol{K}\tilde{\boldsymbol{c}}_n + \eta^2 J_{\min}\tilde{\boldsymbol{r}}_\kappa, \tag{24}$$

where $\tilde{\boldsymbol{c}}_n$ and $\tilde{\boldsymbol{r}}_\kappa$ are the lexicographic forms of $\tilde{\boldsymbol{C}}_n$ and $\tilde{\boldsymbol{R}}_\kappa$, respectively, and

$$\boldsymbol{K} := \boldsymbol{I}_{r^2} - \eta(\boldsymbol{K}_1 + \boldsymbol{K}_2) + \eta^2\boldsymbol{K}_3, \tag{25}$$

where $\boldsymbol{K}_1 := \boldsymbol{I}_r\otimes\tilde{\boldsymbol{R}}_\kappa$, $\boldsymbol{K}_2 := \tilde{\boldsymbol{R}}_\kappa\otimes\boldsymbol{I}_r$, and $\boldsymbol{K}_3$ is an $r^2\times r^2$ matrix entries are: $[\boldsymbol{K}_3]_{\ell+(m-1)r,p+(q-1)r} := [\tilde{\boldsymbol{S}}_{\ell,m}]_{p,q}$with $1\le\ell, m, p, q\le r$. Here, $\otimes$ denotes the Kronecker product. By (24) and (25), we obtain the following results.

Theorem 2 (Mean-square stability) *Assume CMIA holds. For any initial conditions and the step size η satisfying* (18)*, given a dictionary $\{\kappa(\cdot,\boldsymbol{u}_j)\}_{j\in\mathcal{J}}$, the CKLMS algorithm with Gaussian kernel is mean-square stable, if the matrix $\boldsymbol{K}$ is stable (i.e., the spectral radius of $\boldsymbol{K}$ is less than one).*

Proof: The algorithm is said to be mean-square stable if, and only if, the state vector remains bounded and tends to a steady-state value, regardless of the initial condition [19]. To complete the proof, it is sufficient to show that $\|\tilde{\boldsymbol{v}}_n\|^2$ remains bounded and tends to a steady-state value. This is verified by the fact that $\tilde{\boldsymbol{c}}_n$ is bounded and tends to a steady-state value if the matrix $\boldsymbol{K}$ is stable. □

Theorem 3 (MSE in the steady state) *Assume that the step size η satisfies* (18) *and the matrix $\boldsymbol{K}$ is stable. The steady-state MSE is given by* (19) *with the lexicographic representation of $\tilde{\boldsymbol{C}}_\infty$ given by*

$$\tilde{\boldsymbol{c}}_\infty = \eta^2 J_{\min}(\boldsymbol{I}_{r^2} - \boldsymbol{K})^{-1}\tilde{\boldsymbol{r}}_\kappa. \tag{26}$$

Proof: Letting $\tilde{\boldsymbol{c}}_{n+1} = \tilde{\boldsymbol{c}}_n$ in (24) and rearranging the equation, we obtain (26). □

5. SIMULATION RESULTS

We shall compare simulated learning curves and analytic models to validate the analysis. We conduct two experiments under the same settings as in [12]. In the first experiment, the input sequence is generated by

$$u_n := \rho u_{n-1} + \sigma_u\sqrt{1-\rho^2}\omega_n, \tag{27}$$

where ω_n is the noise following the i.i.d standard normal distribution. The nonlinear system is defined as follows:

$$\begin{cases} x_n &:= 0.5u_n - 0.3u_{n-1} \\ d_n &:= x_n - 0.5x_n^2 + 0.1x_n^3 + \nu_n, \end{cases} \tag{28}$$

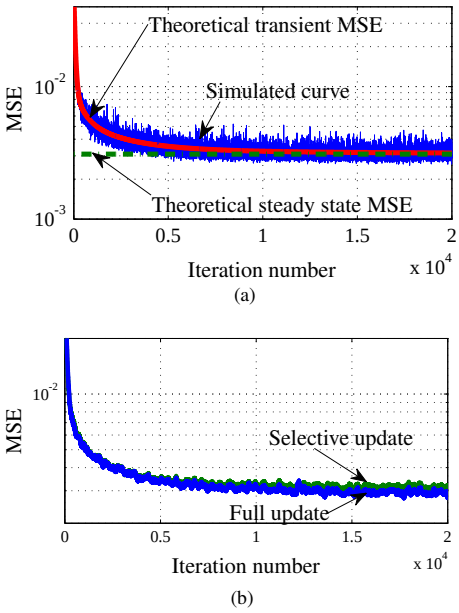

Fig. 2. Simulation results of the first experiment.

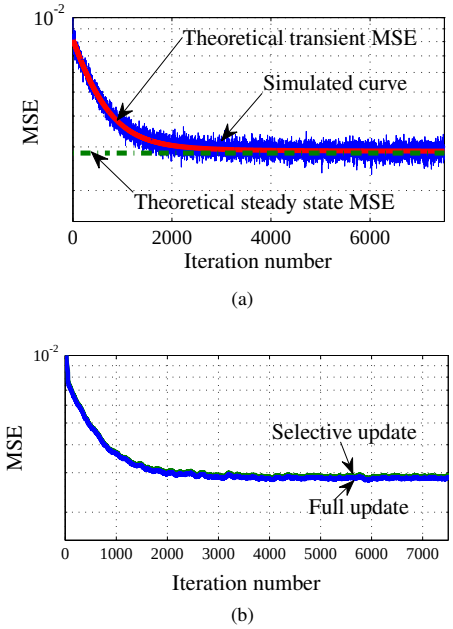

Fig. 3. Simulation results of the second experiment.

Table 1. Computational complexity of the CKLMS algorithm.

Selective update	$(L+s_n+1)r+O(s_n^3)$
Full update	$(L+r+2)r$

where ν_n is an additive zero-mean Gaussian noise with the standard deviation $\sigma_\nu = 0.05$. The input vector is $\boldsymbol{u}_n = [u_n\ u_{n-1}]^\mathsf{T}$. The step size, the standard deviation of the input, the input correlation parameter, the kernel parameter and the dictionary size are set to $\eta = 0.075$, $\sigma_u = 0.5$, $\rho = 0.5$, $\sigma = 0.7$ and $r = 25$, respectively. The dictionary is r samples on a uniform grid defined on $[-1, 1] \times [-1, 1]$.

Fig. 2(a) depicts the results: the learning curves, the theoretical transient MSE curve, and the theoretical steady state MSE line are presented in blue, red, and green (dotted line), respectively. The simulated curve is obtained by averaging over 1000 Monte-Carlo runs. The theoretical MSE is estimated by (19) with $\tilde{\boldsymbol{C}}_n$ recursively evaluated by (20). The steady state MSE is computed by Theorem 3. Although the input is correlated, the theoretical MSE presented in this paper well represents the behavior of CKLMS.

In the second experiment, the fluid-flow control problem is considered [20]:

$$\begin{cases} x_n := & 0.1044u_n + 0.0883u_{n-1} \\ & +1.4138x_{n-1} - 0.6065x_{n-2} \\ d_n := & 0.3163x_n/\sqrt{0.1+0.9x_n^2} + \nu_n, \end{cases} \tag{29}$$

where the input u_n is generated again by (27) with $\sigma_u = 0.5$ and $\rho = 0.5$, and the standard deviation of the additive Gaussian noise ν_n is set to $\sigma_\nu = 0.05$. The kernel parameter is set to $\sigma = 0.75$. The input vector is $\boldsymbol{u}_n = [u_n\ u_{n-1}]^\mathsf{T}$. 31 dictionary elements are selected from the inputs $\boldsymbol{u}_n$ based on the coherence criterion [4] in advance. The step size is set to $\eta = 0.01$. The simulated curves are obtained by averaging over 1000 Monte-Carlo runs, and the same theoretical model as the first experiment is used. Fig 3(a) depicts the results. Again, the simulation results show the validity of the analysis. Table 1 summarizes the overall per-iteration complexity (the number of real multiplications) of CKLMS with full update and selective update (see [15, 18]), and Fig. 4 illustrates the complexity as a function of the dictionary size r for $L = 2$ and $s_n = 1$; $O(s_n^3)$ is counted simply as s_n^3. Here, $s_n = 1$ means that only one coefficient is updated at each iteration and hence the complexity is reduced drastically. Fig 2(b) and 3(b) depict the MSE learning curves of CKLMS with full update and selective update for $s_n = 1$. It can be seen that CKLMS with the selective update exhibits a steady-state MSE comparable to the full-update case with drastically lower complexity.

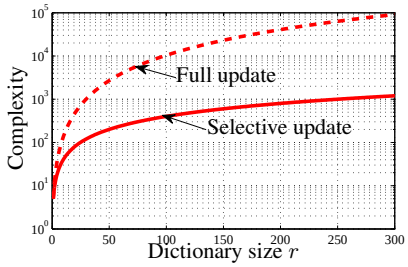

Fig. 4. Computational complexity.

6. CONCLUSION

This paper presented a stochastic behavior analysis of the CKLMS algorithm which is a stochastic restricted-gradient descent method. The analysis provided a transient and steady-state MSEs of the algorithm. We also derived stability conditions in the mean and mean-square sense. Simulation results showed that the theoretical MSE curves given by the analysis well meet the simulated MSE curves. The outcomes of this study will serve as a theoretical basis to compare the performances of KNLMS and CKLMS.

7. REFERENCES

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

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

[3] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

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

[5] K. Slavakis, S. Theodoridis, and I. Yamada, "Adaptive constrained learning in reproducing kernel Hilbert spaces: the robust beamforming case," *IEEE Trans. Signal Processing*, vol. 57, no. 12, pp. 4744–4764, Dec. 2009.

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

[7] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, Dec. 2012.

[8] S. V. Vaerenbergh, M. Lazaro-Gradilla, and I. Santamaria, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neural Network and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, Aug 2012.

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

[10] 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 Processing*, vol. 60, no. 5, pp. 2208–2222, May 2012.

[11] C. Richard and J.-C. M. Bermudez, "Closed-form conditions for convergence of the gaussian kernel-least-mean-square algorithm," in *Proc. Asilomar*, Pacific Grove, CA, USA, Nov. 2012, pp. 1797–1801.

[12] J. Chen, W. Gao, C. Richard, and J.-C. M. Bermudez, "Convergence analysis of kernel LMS algorithm with pre-tuned dictionary," in *Proc. IEEE ICASSP*, 2014, pp. 7243–7247.

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

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

[15] M. Yukawa and R. Ishii, "An efficient kernel adaptive filtering algorithm using hyperplane projection along affine subspace," in *Proc. EUSIPCO*, 2012, pp. 2183–2187.

[16] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Mean square convergence analysis for kernel least mean square algorithm," *Signal Processing*, vol. 92, pp. 2624–2632, 2012.

[17] M. Takizawa and M. Yukawa, "An efficient sparse kernel adaptive filtering algorithm based on isomorphism between functional subspace and euclidean space," in *Proc. IEEE ICASSP*, 2014, pp. 4508–4512.

[18] ——, "Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space," *IEEE Trans. Signal Processing*, submitted for publication.

[19] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, 2008.

[20] H. Al-Duwaish, M. N. Karim, and V. Chandrasekar, "Use of multilayer feedforward neural networks in identification and control of wiener model," in *Proc. IEEE Control Theory Appl.*, vol. 143, 1996, pp. 255–258.