CONVERGENCE ANALYSIS OF KERNEL LMS ALGORITHM WITH PRE-TUNED DICTIONARY

Jie Chen[*]   *Wei Gao*[*†]   *Cédric Richard*[*]   *Jose-Carlos M. Bermudez*[‡]

[*] Université de Nice Sophia-Antipolis, CNRS, France
[†] College of Marine Engineering, Northwestern Polytechnical University, Xian, China
[‡] Federal University of Santa Catarina, Florianópolis, SC, Brazil
{jie.chen, cedric.richard}@unice.fr   gao_wei@mail.nwpu.edu.cn   j.bermudez@ieee.org

ABSTRACT

The kernel least-mean-square (KLMS) algorithm is an appealing tool for online identification of nonlinear systems due to its simplicity and robustness. In addition to choosing a reproducing kernel and setting filter parameters, designing a KLMS adaptive filter requires to select a so-called dictionary in order to get a finite-order model. This dictionary has a significant impact on performance, and requires careful consideration. Theoretical analysis of KLMS as a function of dictionary setting has rarely, if ever, been addressed in the literature. In an analysis previously published by the authors, the dictionary elements were assumed to be governed by the same probability density function of the input data. In this paper, we modify this study by considering the dictionary as part of the filter parameters to be set. This theoretical analysis paves the way for future investigations on KLMS dictionary design.

Index Terms— Kernel least-mean-square algorithm, convergence analysis, nonlinear adaptive filtering, dictionary learning

1. INTRODUCTION

Complex real-world applications often require nonlinear signal processing. During the last decade, adaptive filtering in reproducing kernel Hilbert spaces (RKHS) has become an appealing tool for online system identification and time series prediction [1]. By replacing inner products with a reproducing kernel, these algorithms provide an efficient and elegant way to map the input data into a high, even infinite, dimensional space with an implicit nonlinear application. The kernel recursive least-squares (KRLS) algorithm was introduced in [2]. The sliding-window KRLS and the extended KRLS algorithms were derived in [3,4], respectively. The kernel affine projection algorithm (KAPA) and, as a particular case, the kernel normalised LMS algorithm (KNLMS), were independently introduced in [5–8]. The kernel least-mean-square algorithm (KLMS), proposed in [9,10], has attracted much attention in recent years because of its simplicity and robustness. Along the same line, the quantized KLMS algorithm (QKLMS) was recently proposed in [11].

Kernel-based adaptive filters use more or less sophisticated criteria to construct a so-called dictionary, in order to operate with a finite-order model. This dictionary has a significant impact on performance, and requires careful consideration. One of the most informative criteria uses approximate linear dependency (ALD) condition to test the ability of the dictionary elements to linearly approximate the current kernelized input sample [2]. Other well-known criteria include the novelty criterion [12], the coherence criterion [6], the surprise criterion [13], and the closed-ball sparsification criterion [14]. Recently, KLMS algorithm with ℓ_1-norm regularization

has also been studied so that the algorithm can remove obsolete dictionary elements and then adapt to non-stationary environments [15–18].

Few theoretical studies have investigated the convergence behavior of kernel-based adaptive filters compared to the number of algorithm variants that have recently been derived. This situation is partly due to technical difficulties stemming from filter nonlinearity with respect to input samples. An extended analysis of the stochastic behavior of the KLMS algorithm with Gaussian kernel was proposed in [19], and a closed-form condition for convergence was introduced in [20]. Stability in the mean of this algorithm with ℓ_1-norm regularization was studied in [17, 18]. The aim of these works was to propose a procedure for designing KLMS filters, with appropriate parameter setting, given some desired performance. It was assumed that the dictionary elements are governed by the same probability density function as the input data. This situation typically arises with kernel-based adaptive filters that use a short-time sliding dictionary [9,21,22]. Nevertheless, this framework does not allow the user to pretune the dictionary to improve performance. In this paper, we propose a theoretical analysis of the KLMS algorithm with Gaussian kernel that considers the dictionary as part of the filter parameters to be set. We derive models for the mean and mean-square behavior of the adaptive weights, and for the mean-square estimation error. We also determine stability conditions. Finally, we illustrate the validity of these models with simulations. These derivations pave the way for future investigations on KLMS dictionary design.

2. KLMS ALGORITHM

2.1. Problem formulation

Let $\mathcal{X}$ be a compact domain in Euclidean space $\mathbb{R}^L$ and let $\mathcal{Y} = \mathbb{R}$. Let $\rho_{\mathcal{Z}}$ be a Borel probability measure on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ whose regularity properties will be assumed as needed. Considering a sample

$$\boldsymbol{z} \in \mathcal{Z}^N, \qquad \boldsymbol{z} = \{(\boldsymbol{x}(i), y(i))\}_{i=1}^N, \qquad (1)$$

we aim to find a regression function ψ using $\boldsymbol{z}$. Let $\mathcal{H}$ be a reproducing kernel Hilbert space with kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Restricting the solution to this space, the function ψ^* that minimizes the regularized mean-square error

$$\min_{\psi \in \mathcal{H}} \sum_{i=1}^N |y(i) - \psi(\boldsymbol{x}(i))|^2 + \lambda \|\psi\|_{\mathcal{H}}^2, \quad \lambda \geq 0 \qquad (2)$$

is of the form

$$\psi^* = \sum_{i=1}^N \alpha_i \, \kappa(\cdot, \boldsymbol{x}(i)) \qquad (3)$$

with $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^\top$ the unique solution of the well-posed linear system in $\mathbb{R}^N$:

$$(\boldsymbol{K} + \lambda\,\boldsymbol{I})\,\boldsymbol{\alpha} = \boldsymbol{y} \tag{4}$$

where $\boldsymbol{K}$ is the $N \times N$ Gram matrix with (i, j)-th entry $\kappa(\boldsymbol{x}(i), \boldsymbol{x}(j))$, $\boldsymbol{y}$ is the $N \times 1$ vector with i-th entry $y(i)$, and $\boldsymbol{I}$ is the $N \times N$ identity matrix. This framework is inappropriate to solve problem (2) in an online manner, since the algorithm would suffer from the on-going increase of the size N of the model as new data arrives. A well-known strategy is to use the fixed-size model:

$$\psi(\cdot) = \sum_{i=1}^{M} \alpha_i\, \kappa(\cdot, \boldsymbol{x}_{\omega_i}). \tag{5}$$

The set $\boldsymbol{\omega} = \{\kappa(\cdot,\,\boldsymbol{x}_{\omega_i})\}$ with $i = \{1, \ldots, M\}$ is the so-called dictionary. Several rules have been proposed in the literature to construct the dictionary in an online manner. Consider a dictionary $\boldsymbol{\omega}$, and a time sequence $\{\boldsymbol{x}(n), y(n)\}_n$, the KLMS algorithm is an efficient strategy to estimate (5). The update rule is given by

$$\boldsymbol{\alpha}(n + 1) = \boldsymbol{\alpha}(n) + \eta\, e(n)\, \boldsymbol{\kappa}_\omega(n) \tag{6}$$

where $\boldsymbol{\kappa}_\omega(n) = [\kappa(\boldsymbol{x}(n), \boldsymbol{x}_{\omega_1}), \ldots, \kappa(\boldsymbol{x}(n), \boldsymbol{x}_{\omega_M})]^\top$, and $e(n)$ is the estimation error at instant n

$$\begin{aligned} e(n) &= y(n) - \psi(\boldsymbol{x}(n)) \\ &= y(n) - \sum_{j=1}^{M} \alpha_j(n)\, \kappa(\boldsymbol{x}(n), \boldsymbol{x}_{\omega_j}) \end{aligned} \tag{7}$$

Consider the mean-square error criterion

$$E\{e^2(n)\} = \int_\Omega \int_{\mathcal{X} \times \mathcal{Y}} e^2(n)\, d\rho_{\mathcal{Z}}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\omega})\, d\rho_\Omega \tag{8}$$

with ρ_Ω a Borel probability measure on the dictionary space Ω. Except with simplified assumptions such as in [19], where the authors consider that the dictionary elements are governed by the same probability density function as input data, distributions of $\boldsymbol{\omega}$ generated by dictionary learning methods cannot be expressed in closed forms. In this paper, we consider the dictionary as part of the filter parameters to be set. Our objective is to characterize both transient and steady-state of the mean-square criterion conditionally to $\boldsymbol{\omega}$, that is,

$$E_Z\{e^2(n)|\boldsymbol{\omega}\} = \int_{\mathcal{X} \times \mathcal{Y}} e^2(n)\, d\rho(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\omega}). \tag{9}$$

We shall use the subscript ω for quantities conditioned on the dictionary, and X to expectation with respect to input data distribution.

2.2. Optimal solution

Given $\boldsymbol{\omega}$, the estimation error at instant n writes

$$e_\omega(n) = y(n) - \psi_\omega(\boldsymbol{x}(n)) \tag{10}$$

with $\psi_\omega(\boldsymbol{x}(n)) = \psi(\boldsymbol{x}(n))|\boldsymbol{\omega}$. Squaring both sides of equation (10), and taking the expected value, leads to the mean-square error (MSE) criterion

$$\begin{aligned} J_{\text{MSE},\omega} &= E\{e_\omega^2(n)\} \\ &= E\{y^2(n)\} - 2\,\boldsymbol{p}_{\kappa y,\omega}^\top \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \boldsymbol{R}_{\kappa\kappa,\omega} \boldsymbol{\alpha} \end{aligned} \tag{11}$$

where

$$\boldsymbol{R}_{\kappa\kappa,\omega} = E\{\boldsymbol{\kappa}_\omega(n)\boldsymbol{\kappa}_\omega^\top(n)|\boldsymbol{\omega}\}$$

is the correlation matrix of the kernelized input, and

$$\boldsymbol{p}_{\kappa y,\omega} = E\{y(n)\boldsymbol{\kappa}_\omega(n)|\boldsymbol{\omega}\}$$

is the cross-correlation vector between $\boldsymbol{\kappa}_\omega(n)$ and $y(n)$. As $\boldsymbol{R}_{\kappa\kappa,\omega}$ is positive definite, the optimum weight vector is given by

$$\boldsymbol{\alpha}_\omega^* = \arg\min_{\boldsymbol{\alpha}} J_{\text{MSE},\omega}(\boldsymbol{\alpha}) = \boldsymbol{R}_{\kappa\kappa,\omega}^{-1}\, \boldsymbol{p}_{\kappa y,\omega} \tag{12}$$

and the minimum MSE is

$$J_{\min,\omega} = E\{y^2(n)\} - \boldsymbol{p}_{\kappa y,\omega}^\top \boldsymbol{R}_{\kappa\kappa,\omega}^{-1}\, \boldsymbol{p}_{\kappa y,\omega}. \tag{13}$$

3. PERFORMANCE ANALYSIS

We shall now derive the convergence model and stability conditions for the KLMS algorithm with Gaussian kernel, given $\boldsymbol{\omega}$. The Gaussian kernel is defined as

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\tfrac{1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2) \tag{14}$$

where σ denotes the kernel bandwidth. Inputs $\boldsymbol{x}(n)$ are assumed independent zero-mean Gaussian random vectors with autocorrelation matrix $\boldsymbol{R}_{xx} = E\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\}$. Let $\boldsymbol{v}_\omega(n)$ be the weight error vector defined as

$$\boldsymbol{v}_\omega(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}_\omega^*. \tag{15}$$

3.1. Preliminaries and assumptions

Before starting to derive the model, let us recall the following result on the moment generating function of any quadratic form of a Gaussian vector. Let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_L)^\top$ be a random vector following Gaussian distribution with mean and covariance matrix

$$E\{\boldsymbol{\xi}\} = 0 \quad \text{and} \quad \boldsymbol{R}_{\xi\xi} = E\{\boldsymbol{\xi}\,\boldsymbol{\xi}^\top\} \tag{16}$$

Let the random variable ζ be the quadratic form of $\boldsymbol{\xi}$ defined as

$$\zeta = \boldsymbol{\xi}^\top \boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{b}^\top \boldsymbol{\xi} \tag{17}$$

The moment generating function of ζ is given by [23]

$$\begin{aligned} \Psi_\zeta(s) =&|\boldsymbol{I} - 2s\boldsymbol{H}\boldsymbol{R}_{\xi\xi}|^{-\frac{1}{2}} \\ &\cdot \exp\left(\tfrac{s^2}{2}\boldsymbol{b}^\top \boldsymbol{R}_{\xi\xi}(\boldsymbol{I} - 2s\boldsymbol{H}\boldsymbol{R}_{\xi\xi})^{-1}\boldsymbol{b}\right) \end{aligned} \tag{18}$$

This result will be useful to determine expected values. Simplifying assumptions are required in order to make the study of the stochastic behavior of $\boldsymbol{v}_\omega(n)$ mathematically feasible. The following statistical assumption is required in the analysis:

Assumption 1 $\boldsymbol{\kappa}_\omega(n)\boldsymbol{\kappa}_\omega^\top(n)$ *is independent of* $\boldsymbol{v}_\omega(n)$.

This assumption, called modified independence assumption (MIA), is justified in detail in [24]. It has been successfully employed in several adaptive filter analyses, and has been shown in [24] to be less restrictive than the well known independence assumption (IA) [25]. It is called here for further reference *conditioned* MIA, or CMIA, to distinguish it from the MIA used in [19].

3.2. Mean weight error analysis

The estimation error $e_\omega(n)$ can be expressed using $\boldsymbol{v}_\omega(n)$ by

$$e_\omega(n) = y(n) - \boldsymbol{\kappa}_\omega^\top(n)\,\boldsymbol{v}_\omega(n) - \boldsymbol{\kappa}_\omega^\top(n)\,\boldsymbol{\alpha}_\omega^* \qquad (19)$$

Replacing this expression into (6) and using the definition of $\boldsymbol{v}_\omega(n)$, we obtain the following recursive expression for $\boldsymbol{v}_\omega(n)$:

$$\boldsymbol{v}_\omega(n+1) = \boldsymbol{v}_\omega(n) + \eta\, y(n)\boldsymbol{\kappa}_\omega(n)$$
$$- \eta\,\boldsymbol{\kappa}_\omega^\top(n)\boldsymbol{v}(n)\boldsymbol{\kappa}_\omega(n) - \eta\,\boldsymbol{\kappa}_\omega^\top(n)\boldsymbol{\alpha}_\omega^*\boldsymbol{\kappa}_\omega(n) \qquad (20)$$

Taking expected values of both sides of (20), using CMIA and (12) we have the mean weight error model

$$E\{\boldsymbol{v}_\omega(n+1)\} = (\boldsymbol{I} - \eta\boldsymbol{R}_{\kappa\kappa,\omega})\,E\{\boldsymbol{v}_\omega(n)\} \qquad (21)$$

With the Gaussian kernel, the components of $\boldsymbol{R}_{\kappa\kappa,\omega}$ are given by

$$[\boldsymbol{R}_{\kappa\kappa,\omega}]_{ij}$$
$$= E_X\left\{\exp\left(-\tfrac{1}{2\sigma^2}\left[\|\boldsymbol{x}(n) - \boldsymbol{x}_{\omega_i}\|^2 + \|\boldsymbol{x}(n) - \boldsymbol{x}_{\omega_j}\|^2\right]\right)\right\}$$
$$= \exp\left(-\tfrac{1}{2\sigma^2}\left[\|\boldsymbol{x}_{\omega_i}\|^2 + \|\boldsymbol{x}_{\omega_j}\|^2\right]\right) \qquad (22)$$
$$\cdot E_X\left\{\exp\left(-\tfrac{1}{\sigma^2}\left[\|\boldsymbol{x}(n)\|^2 - (\boldsymbol{x}_{\omega_i} + \boldsymbol{x}_{\omega_j})^\top\boldsymbol{x}(n)\right]\right)\right\}$$

Let $\bar{\boldsymbol{x}}_{\omega_{ij}} = \boldsymbol{x}_{\omega_i} + \boldsymbol{x}_{\omega_j}$ and $\|\bar{\boldsymbol{x}}_{\omega_{ij}}\|^{(2)} = \|\boldsymbol{x}_{\omega_i}\|^2 + \|\boldsymbol{x}_{\omega_j}\|^2$. Comparing the second term on the RHS of (22) with (17) with $\boldsymbol{H} = \boldsymbol{I}$, $\boldsymbol{b} = -\bar{\boldsymbol{x}}_{\omega_{ij}}$ and $s = -\tfrac{1}{\sigma^2}$ we get

$$[\boldsymbol{R}_{\kappa\kappa,\omega}]_{ij} = \exp\left(-\tfrac{1}{2\sigma^2}\|\bar{\boldsymbol{x}}_{\omega_{ij}}\|^{(2)}\right)\left|\boldsymbol{I} + \tfrac{2}{\sigma^2}\boldsymbol{R}_{xx}\right|^{-\frac{1}{2}}$$
$$\cdot \exp\left(\tfrac{1}{2\sigma^4}\bar{\boldsymbol{x}}_{\omega_{ij}}^\top\boldsymbol{R}_{xx}(\boldsymbol{I} + \tfrac{2}{\sigma^2}\boldsymbol{R}_{xx})^{-1}\bar{\boldsymbol{x}}_{\omega_{ij}}\right) \qquad (23)$$

In order to express this formula in a more compact manner, we use the identity $(\boldsymbol{I}+\boldsymbol{A}^{-1})^{-1} = \boldsymbol{A}(\boldsymbol{A}+\boldsymbol{I})^{-1}$ with $\boldsymbol{R}_{xx}(\boldsymbol{I} + \tfrac{2}{\sigma^2}\boldsymbol{R}_{xx})^{-1}$. This yields

$$[\boldsymbol{R}_{\kappa\kappa,\omega}]_{ij} = \left|\boldsymbol{I} + \tfrac{2}{\sigma^2}\boldsymbol{R}_{xx}\right|^{-\frac{1}{2}}$$
$$\cdot \exp\left(-\tfrac{1}{4\sigma^2}\left[2\|\bar{\boldsymbol{x}}_{\omega_{ij}}\|^{(2)} - \|\bar{\boldsymbol{x}}_{\omega_{ij}}\|^2_{(\boldsymbol{I}+\sigma^2\boldsymbol{R}_{xx}^{-1}/2)^{-1}}\right]\right) \qquad (24)$$

Theorem 3.1 *(Stability in the mean) Assume CMIA holds. Then, for any initial condition, given a dictionary $\boldsymbol{\omega}$, the Gaussian KLMS algorithm (6) asymptotically converges in the mean if the step size is chosen to satisfy*

$$0 < \eta < \frac{2}{\lambda_{\max}(\boldsymbol{R}_{\kappa\kappa,\omega})} \qquad (25)$$

where $\lambda_{max}(\cdot)$ denotes the maximum eigenvalue of the matrix. The entries of $\boldsymbol{R}_{\kappa\kappa,\omega}$ are given by (24).

3.3. Mean-square error analysis

The second-order moments of the weights are related to the MSE through [19]

$$J_{\text{MSE},\omega}(n) = J_{\min,\omega} + \text{trace}\{\boldsymbol{R}_{\kappa\kappa,\omega}\boldsymbol{C}_{v,\omega}(n)\} \qquad (26)$$

where $\boldsymbol{C}_{v,\omega}(n)$ denotes the correlation matrix of the weight error vector $\boldsymbol{v}_\omega(n)$, i.e.,

$$\boldsymbol{C}_{v,\omega}(n) = E\{\boldsymbol{v}_\omega(n)\,\boldsymbol{v}_\omega^\top(n)\}, \qquad (27)$$

and $J_{\min,\omega}$ is the minimum MSE given by (13). The second term on the RHS of (26) is the excess mean-square error (EMSE), denoted by $J_{\text{EMSE},\omega}(n)$. Estimating $J_{\text{MSE},\omega}(n)$ requires a model for $\boldsymbol{C}_{v,\omega}(n)$.

Post-multiplying (20) by its transpose, taking the expectation, and using CMIA, we obtain the following recursion

$$\boldsymbol{C}_{v,\omega}(n+1) \approx \boldsymbol{C}_{v,\omega}(n) + \eta^2\,\boldsymbol{T}_\omega + \eta^2\,\boldsymbol{R}_{\kappa\kappa,\omega}J_{\min,\omega}$$
$$- \eta\,(\boldsymbol{R}_{\kappa\kappa,\omega}\boldsymbol{C}_{v,\omega}(n) + \boldsymbol{C}_{v,\omega}(n)\boldsymbol{R}_{\kappa\kappa,\omega}) \qquad (28)$$

with

$$\boldsymbol{T}_\omega = E\{\boldsymbol{\kappa}_\omega(n)\,\boldsymbol{\kappa}_\omega^\top(n)\,\boldsymbol{v}_\omega(n)\,\boldsymbol{v}_\omega^\top(n)\,\boldsymbol{\kappa}_\omega(n)\,\boldsymbol{\kappa}_\omega^\top(n)\}. \qquad (29)$$

Evaluating (29) is a challenging step in the analysis. In [19], for independent Gaussian-distributed dictionary elements, this leads to extensive calculations of up to eighth-order moments of $\boldsymbol{x}(n)$. In [18] we provided a greatly simplified alternative to this. However both situations do not match the framework developed this paper, since dictionary elements are now considered as known. In order to determine the expected value of $\boldsymbol{T}_\omega$, assuming CMIA holds, the (i, j)-th entry of $\boldsymbol{T}_\omega$ writes:

$$[\boldsymbol{T}_\omega]_{ij} \approx \sum_{\ell=1}^{M}\sum_{p=1}^{M} E_X\{\kappa_{\omega,i}(n)\,\kappa_{\omega,j}(n)\kappa_{\omega,\ell}(n)\,\kappa_{\omega,p}(n)\}$$
$$\cdot [\boldsymbol{C}_{v,\omega}(n)]_{\ell p}. \qquad (30)$$

where $\kappa_{\omega,i}(n) = \kappa(\boldsymbol{x}(n), \boldsymbol{x}_{\omega_i})$. Let us define the matrix $\boldsymbol{K}_\omega^{(i,j)}$ with (ℓ, p)-th entry

$$[\boldsymbol{K}_\omega^{(i,j)}]_{\ell p} = E_X\{\kappa_{\omega,i}(n)\kappa_{\omega,j}(n)\kappa_{\omega,\ell}(n)\kappa_{\omega,p}(n)\}. \qquad (31)$$

Expression (30) can be rewritten as

$$[\boldsymbol{T}_\omega(n)]_{ij} \approx \text{trace}\{\boldsymbol{K}_\omega^{(i,j)}\boldsymbol{C}_{v,\omega}(n)\} \qquad (32)$$

In order to determine (32), we need to evaluate the expected values in $[\boldsymbol{K}_\omega^{(i,j)}]_{\ell p}$. Let us introduce further the notations

$$\bar{\boldsymbol{x}}_{\omega_{ij\ell p}} = \boldsymbol{x}_{\omega_i} + \boldsymbol{x}_{\omega_j} + \boldsymbol{x}_{\omega_\ell} + \boldsymbol{x}_{\omega_p}$$
$$\|\bar{\boldsymbol{x}}_{\omega_{ij\ell p}}\|^{(2)} = \|\boldsymbol{x}_{\omega_i}\|^2 + \|\boldsymbol{x}_{\omega_j}\|^2 + \|\boldsymbol{x}_{\omega_\ell}\|^2 + \|\boldsymbol{x}_{\omega_p}\|^2 \qquad (33)$$

We have

$$[\boldsymbol{K}_\omega^{(i,j)}]_{\ell p}$$
$$= E_X\{\kappa_{\omega,i}(n)\,\kappa_{\omega,j}(n)\,\kappa_{\omega,\ell}(n)\,\kappa_{\omega,p}(n)\}$$
$$= E_X\left\{\exp\left(-\tfrac{1}{2\sigma^2}\sum_{k=\{i,j,\ell,p\}}\|\boldsymbol{x}(n) - \boldsymbol{x}_{\omega_k}\|^2\right)\right\} \qquad (34)$$
$$= \exp\left(-\tfrac{1}{2\sigma^2}\|\bar{\boldsymbol{x}}_{\omega_{ij\ell p}}\|^{(2)}\right)$$
$$\cdot E_X\left\{\exp\left(-\tfrac{1}{\sigma^2}[2\|\boldsymbol{x}(n)\|^2 - \bar{\boldsymbol{x}}_{\omega_{ij\ell p}}^\top\,\boldsymbol{x}(n)]\right)\right\}$$

Now setting $\boldsymbol{H} = 2\boldsymbol{I}$, $\boldsymbol{b} = -\bar{\boldsymbol{x}}_{\omega_{ij\ell p}}$ and $s = -\tfrac{1}{\sigma^2}$ in (17), we get

$$[\boldsymbol{K}_\omega^{(i,j)}]_{\ell p} = \left|\boldsymbol{I} + \tfrac{4}{\sigma^2}\boldsymbol{R}_{xx}\right|^{-\frac{1}{2}}$$
$$\cdot \exp\left(-\tfrac{1}{8\sigma^2}\left[4\|\bar{\boldsymbol{x}}_{\omega_{ij\ell p}}\|^{(2)} - \|\bar{\boldsymbol{x}}_{\omega_{ij\ell p}}\|^2_{(\boldsymbol{I}+\sigma^2\boldsymbol{R}_{xx}^{-1}/4)^{-1}}\right]\right) \qquad (35)$$
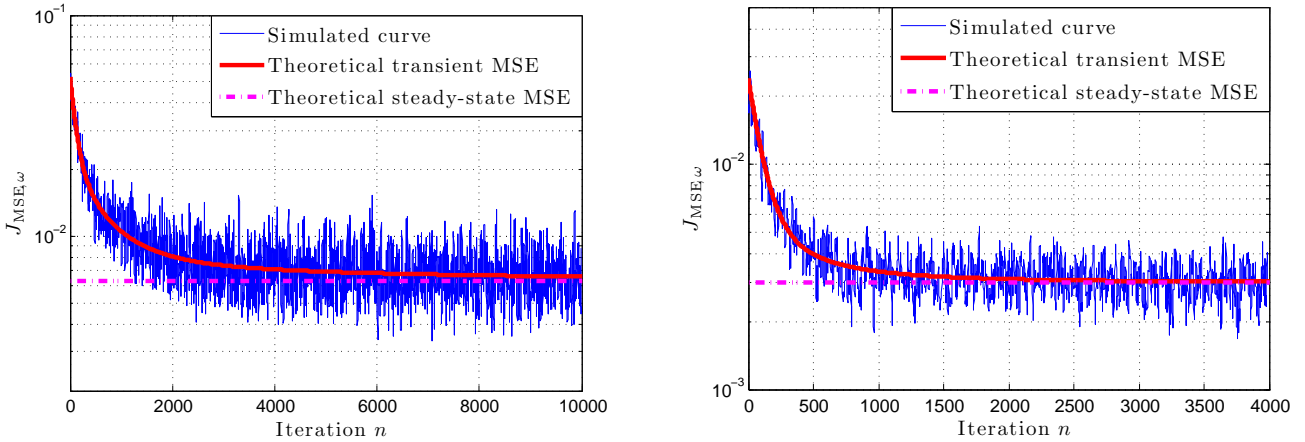
With this expression, recursion (28) can readily be evaluated. In lexicographic form, i.e., the columns of each matrix are stacked on top of each other into a vector, equation (28) becomes

$$\boldsymbol{c}_{v,\omega}(n+1) = \boldsymbol{G}_\omega\,\boldsymbol{c}_{v,\omega}(n) + \eta^2 J_{\min,\omega}\,\boldsymbol{r}_{\kappa\kappa,\omega} \qquad (36)$$

with

$$\boldsymbol{G}_\omega = \boldsymbol{I} - \eta(\boldsymbol{G}_{\omega,1} + \boldsymbol{G}_{\omega,2}) + \eta^2\boldsymbol{G}_{\omega,3} \qquad (37)$$

where $\boldsymbol{c}_{v,\omega}(n)$ and $\boldsymbol{r}_{\kappa\kappa,\omega}$ are the lexicographic representations of matrices $\boldsymbol{C}_{v,\omega}(n)$ and $\boldsymbol{R}_{\kappa\kappa,\omega}$, respectively. Matrix $\boldsymbol{G}_\omega$ is given by:

Fig. 1. Simulation results (Left: learning curves for system 1. Right: Learning curves for system 2).

- $\boldsymbol{I}$ is the identity matrix of dimension $M^2 \times M^2$;
- $\boldsymbol{G}_{\omega,1} = \boldsymbol{I} \otimes \boldsymbol{R}_{\kappa\kappa,\omega}$, where $\otimes$ denotes the Kronecker product;
- $\boldsymbol{G}_{\omega,2} = \boldsymbol{R}_{\kappa\kappa,\omega} \otimes \boldsymbol{I}$;
- $\boldsymbol{G}_{\omega,3}$ entries are: $[\boldsymbol{G}_{\omega,3}]_{i+(j-1)M,\ell+(p-1)M} = [\boldsymbol{K}_{\omega}^{(i,j)}]_{\ell,p}$ with $1 \leq i,j,\ell,p \leq M$.

Note that $\boldsymbol{G}_{\omega,1}$ to $\boldsymbol{G}_{\omega,3}$ are symmetric matrices, which implies that the matrix $\boldsymbol{G}_{\omega}$ is also symmetric. The following results directly come from (36)–(37):

Theorem 3.2 *(Mean-square stability) Assume CMIA holds. For any initial condition, given a dictionary $\boldsymbol{\omega}$, the Gaussian KLMS algorithm* (6) *is mean-square stable if the matrix $\boldsymbol{G}_{\omega}$ is stable.*

Theorem 3.3 *(Mean-squared error) Consider a sufficiently small step size η, which ensures mean and mean-square stability. The steady-state MSE is given by* (26) *with the lexicographic representation of $\boldsymbol{C}_{v,\omega}(\infty)$ given by*

$$\boldsymbol{c}_{v,\omega}(\infty) = \eta^2 J_{\min,\omega} (\boldsymbol{I} - \boldsymbol{G}_{\omega})^{-1} \boldsymbol{r}_{\kappa\kappa,\omega}. \tag{38}$$

4. EXPERIMENT

In this section, we consider two problems of nonlinear system identification with KLMS. We shall compare simulated learning curves and analytical models to validate our approach.

In the first experiment, we considered the input sequence

$$x(n) = \rho\, x(n-1) + \sigma_x \sqrt{1 - \rho^2}\, w(n) \tag{39}$$

with $w(n)$ a noise following the i.i.d. standard normal distribution. The nonlinear system was defined as follows:

$$\begin{cases} u(n) = 0.5\,x(n) - 0.3\,x(n-1) \\ y(n) = u(n) - 0.5\,u^2(n) + 0.1\,u^3(n) + v(n) \end{cases} \tag{40}$$

where $v(n)$ is an additive zero-mean Gaussian noise with standard deviation $\sigma_v = 0.05$. At each instant, the KLMS algorithm was updated based on the input vector $\boldsymbol{x}(n) = [x(n), x(n-1)]^\top$ and the reference signal $y(n)$. We set $\sigma_x = 0.5$ and $\rho = 0.5$. The Gaussian

kernel with bandwidth $\sigma = 0.25$ was used. Twenty-five samples on a uniform grid defined on $[-1, 1] \times [-1, 1]$ were randomly selected to be the dictionary elements $\boldsymbol{x}_{\omega_i}$, $i = 1, \ldots, 25$. The step size was set to $\eta = 0.05$. The learning curves of the algorithm are depicted in Fig. 1 (left). The simulated curves were obtained by averaging over 100 Monte-Carlo runs. Theoretical MSE evolution was estimated by (26), and $\boldsymbol{C}_{v,\omega}(n)$ was recursively evaluated with expression (36). The steady-state MSE was calculated by Theorem 3. It can be noticed that although inputs $\boldsymbol{x}(n)$ are correlated in time, theoretical results derived with CMIA accurately describe the behavior of the KLMS algorithm.

In the second experiment, we considered the fluid-flow control problem studied in [26, 27]. The input signal was also generated with (39) with $\sigma_x = 0.25$ and $\rho = 0.5$. The nonlinear system was defined by

$$\begin{cases} u(n) = 0.1044\,x(n) + 0.0883\,x(n-1) \\ \qquad\quad + 1.4138\,y(n-1) - 0.6065\,y(n-2) \\ y(n) = 0.3163\,u(n)/\sqrt{0.10 + 0.90\,u^2(n)} + v(n) \end{cases} \tag{41}$$

where $v(n)$ is an additive zero-mean Gaussian noise with standard deviation $\sigma_v = 0.05$. A Gaussian kernel with bandwidth $\sigma = 0.15$ was used. Thirty-seven dictionary elements were pre-selected with the *coherence criterion* [6]. The step size was set to $\eta = 0.05$. We ran the KLMS algorithm over 100 Monte-Carlo to empirically estimate its performance, and we evaluated the theoretical model. This led us to the learning curves presented in Fig. 1 (right). This simulation also confirms the validity of our theoretical analysis.

5. CONCLUSION AND PERSPECTIVES

Designing a KLMS-type adaptive filter requires to select a dictionary, which has a significant impact on performance and thus requires careful consideration. In this paper, we derived a theoretical model to characterize the convergence behavior of the KLMS algorithm with Gaussian kernel. This model depends on dictionary setting, which can now be considered as part of the filter parameters to be set. In future work, we will exploit this additional flexibility to design application-dependent dictionaries.

6. REFERENCES

[1] W. Liu, J. C. Príncipe, and S.H Haykin, *Kernel Adaptive Filtering*, Wiley, 2011.

[2] 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.

[3] S. Van Vaerenbergh, J. Vía, and I. Santamaría, "A sliding-window kernel RLS algorithm and its application to nonlinear channel identification," in *Proc. IEEE ICASSP*, Toulouse, France, May 2006, pp. 789–792.

[4] W. Liu, I. M. Park, Y. Wang, and J. C. Príncipe, "Extended kernel recursive least squares algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3801–3814, Oct. 2009.

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

[6] 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.

[7] S. Slavakis and S. Theodoridis, "Sliding window generalized kernel affine projection algorithm using projection mappings," *EURASIP J. Adv. Sig. Pr.*, vol. 2008:735351, Apr. 2008.

[8] W. Liu and J. C. Príncipe, "Kernel affine projection algorithms," *EURASIP J. Adv. Sig. Pr.*, vol. 2008:784292, Mar. 2008.

[9] C. Richard, "Filtrage adaptatif non-linéaire par méthodes de gradient stochastique court-terme à noyau," in *Proc. GRETSI*, Louvain-la-Neuve, Belgium, Sep. 2005, pp. 1–4.

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

[11] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 22–32, Jan. 2012.

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

[13] W. Liu, I. M. Park, and J. C. Príncipe, "An information theoretic approach of designing sparse kernel adaptive filters," *IEEE Trans. Signal Process.*, vol. 20, no. 12, pp. 1950–1961, Dec. 2009.

[14] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Online learning in reproducing kernel Hilbert spaces," in *E-Reference Signal Process.*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2013.

[15] B. Chen, S. Zhao, S. Seth, and J. C. Príncipe, "Online efficient learning with quantized KLMS and L1 regularization," in *Proc. IJCNN*, Brisbane, Australia, Jun. 2012, pp. 1–6.

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

[17] W. Gao, J. Chen, C. Richard, J. Huang, and R. Flamary, "Kernel LMS algorithm with forward-backward splitting for dictionary learning," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 5735–5739.

[18] W. Gao, J. Chen, C. Richard, and J. Huang, "Online dictionary learning for kernel LMS: Analysis and forward-backward splitting algorithm," *IEEE Trans. Signal Process.*, 2013 (submitted). Also available as arXiv:1306.5310v1 [stat.ML], Jun. 2012.

[19] 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.

[20] 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.

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

[22] S. Van Vaerenbergh, J. Via, and I. Santamaría, "Nonlinear system identification using a new sliding-window kernel RLS algorithm," *J. Commun.*, vol. 2, no. 3, pp. 1–8, May 2007.

[23] J. Omura and T. Kailath, "Some useful probability distributions," Tech. Rep. 7050-6, Stanford Electronics Laboratories, Stanford University, Stanford, California, USA, 1965.

[24] J. Minkoff, "Comment: On the unnecessary assumption of statistical independence between reference signal and fillter weights in feedforward adaptive systems," *IEEE Trans. Signal Process.*, vol. 49, no. 5, pp. 1109, May 2001.

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

[26] H. Al-Duwaish, M. N. Karim, and V. Chandrasekar, "Use of multilayer feedfoorward neural networks in identification and control of wiener model," in *Proc. Control Theory Appl.*, May 1996, vol. 143.

[27] J. Vörös, "Modeling and identification of Wiener systems with two-segment nonlinearities," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 2, pp. 253–257, Mar. 2003.