STOCHASTIC BEHAVIOR ANALYSIS OF THE GAUSSIAN KERNEL LEAST MEAN SQUARE ALGORITHM

Wemerson D. Parreira(1) *José C. M. Bermudez*(1) *Cédric Richard*(2) *Jean-Yves Tourneret*(3)

(1) Federal University of Santa Catarina, Brazil, wdparreira@gmail.com, j.bermudez@ieee.org
(2) Universit de Nice Sophia-Antipolis, CNRS (UMR 6525), OCA, France, cedric.richard@unice.fr
(3) University of Toulouse, IRIT-ENSEEIHT, France, Jean-Yves.Tourneret@enseeiht.fr

ABSTRACT

Like its linear counterpart, the Kernel Least Mean Square (KLMS) algorithm is also becoming popular in nonlinear adaptive filtering due to its simplicity and robustness. The "kernelization" of the linear adaptive filters modifies the statistics of the input signals, which now depends on the parameters of the used kernel. A Gaussian KLMS has two design parameters; the step size and the kernel bandwidth. Thus, new analytical models are required to predict the kernel-based algorithm behavior as a function of the design parameters. This paper studies the stochastic behavior of the Gaussian KLMS algorithm for white Gaussian input signals. The resulting model accurately predicts the algorithm behavior and can be used for choosing the algorithm parameters in order to achieve a prescribed performance.

Index Terms— Adaptive filtering, KLMS, convergence analysis, nonlinear system, reproducing kernel

1. INTRODUCTION

Most existing dynamic system modeling approaches focus on linear models due to their inherent simplicity. However, many practical applications (e.g., in communications and bioengineering) require nonlinear signal processing. Unlike linear systems which can be uniquely identified by their impulse response, nonlinear systems can be characterized by representations ranging from higher-order statistics to series expansion methods [1, 2].

Since the pioneering works of Aronszajin [3], Kimeldorf and Wahba [4], and Duttweiler and Kailath [5], nonlinear system identification methods based on reproducing kernel Hilbert spaces (RKHS) have gained wide popularity. Recent developments in kernel-based methods related to dynamic system identification include, most prominently, transcriptions of state-of-the-art adaptive filtering techniques into a kernel-based formalism. Kernel adaptive filtering is an appealing solution to the nonlinear adaptive filtering problem. Developing adaptive filters in RKHS allows the use of linear structures to solve nonlinear estimation problems. The kernel least-mean-square algorithm (KLMS) was proposed in [6]. The kernel recursive-least-square algorithm (KRLS) was described in [7]. The kernel-based normalized least-mean-square algorithm (KNLMS) and affine projection (KAPA) algorithms were studied in [8, 9]. A monograph on kernel adaptive filtering was also recently published [10].

Our work brings a new contribution to the discussion about kernel-based adaptive filtering by providing the first convergence analysis of the KLMS. Here we explore the behavior of the coefficients of the KLMS associated with a Gaussian kernel, applied to nonlinear stationary system identification. Our goal is to determine an analytic model for the mean and mean-squared behavior of the estimation error. In a future contribution, this will allow us to evaluate how the convergence speed and quality of the estimate can be controlled by the algorithm setup.

THIS WORK WAS PARTIALLY SUPPORTED BY CNPQ UNDER GRANTS NO. 473123/2009-6, 305377/2009-4 AND 140672/2007-9.

2. FINITE-ORDER KERNEL-BASED ADAPTIVE FILTERS

Let $\mathcal{U}$ be a compact subspace of $\mathbb{R}^q$, $\kappa\colon \mathcal{U}\times\mathcal{U}\to\mathbb{R}$ a reproducing kernel, and $(\mathcal{H},\langle\cdot,\cdot\rangle_{\mathcal{H}})$ the induced RKHS with its inner product. The reproducing property states that any function $\psi(\cdot)$ of $\mathcal{H}$ can be evaluated at any $\boldsymbol{u}(n)$ of $\mathcal{U}$ using $\psi(\boldsymbol{u}(n)) = \langle\psi(\cdot),\kappa(\cdot,\boldsymbol{u}(n))\rangle_{\mathcal{H}}$, where $\kappa(\cdot,\boldsymbol{u}(n))$ is a positive definite kernel. By setting $\mathcal{H}$ as the hypothesis space, we consider the squared error between the model outputs $\psi(\boldsymbol{u}(n))$ and the desired responses $d(n)$, that is,

$$\sum_{n=1}^{N}[d(n)-\psi(\boldsymbol{u}(n))]^2. \tag{1}$$

The representer theorem [4] states that $\psi(\cdot)$ that minimizes (1) can be written as a kernel expansion in terms of available training data:

$$\psi(\cdot)=\sum_{n=1}^{N}\alpha_n\,\kappa(\cdot,\boldsymbol{u}(n)). \tag{2}$$

This reduces the optimization problem to determining vector $\boldsymbol{\alpha}=[\alpha_1,\dots,\alpha_N]^\top$ that minimizes $\|\boldsymbol{d}-\boldsymbol{K}\boldsymbol{\alpha}\|^2$, where $\boldsymbol{K}$ is the Gram matrix with (n,ℓ)-th entry $\kappa(\boldsymbol{u}(n),\boldsymbol{u}(\ell))$, and $\boldsymbol{d}=[d_1,\dots,d_N]^\top$. Since the order of the model is equal to the number N of available data $\boldsymbol{u}(n)$, this approach cannot be considered for online applications. To overcome this barrier, one can focus on finite-order models

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

where the M kernel functions $\kappa(\cdot,\boldsymbol{u}(\omega_j))$ form the *dictionary*. A possible technique to select the kernel functions in (3) is the approximate linear dependence (ALD) criterion [7]. It consists of including a kernel function $\kappa(\cdot,\boldsymbol{u}(\ell))$ in the dictionary if it satisfies

$$\min_{\boldsymbol{\gamma}}\|\kappa(\cdot,\boldsymbol{u}(\ell))-\sum_j\gamma_j\,\kappa(\cdot,\boldsymbol{u}(\omega_j))\|^2_{\mathcal{H}}>\epsilon_0, \tag{4}$$

where ϵ_0 is a parameter determining the level of sparsity of the model. To control the model order with reduced computational complexity, the coherence-based sparsification rule has also been considered [8, 9]. The kernel $\kappa(\cdot, \boldsymbol{u}(\ell))$ is inserted into the dictionary if

$$\max_j |\kappa(\boldsymbol{u}(\ell), \boldsymbol{u}(\omega_j))| \leq \varepsilon_0 \tag{5}$$

with ε_0 a parameter determining the coherence of dictionary. It was shown in [8] that the dimension of dictionaries determined under rule (5) remains finite. For the rest of the paper, we shall assume that the dictionary is known and that its size M is finite.

The theory outlined above shows that, with a proper sparsification rule, nonlinear adaptive filtering problems can be formulated as finite-order linear ones where the input signal in $\mathcal{U}$ has been nonlinearly mapped to a Hilbert space $\mathcal{H}$. Figure 1 shows a block diagram illustrating a kernel-based adaptive filter. The function $\psi(\boldsymbol{u}(n))$ represents a nonlinear mapping $\psi : \mathcal{U} \rightarrow \mathcal{H}$, and $z(n)$ is a zero-mean additive noise uncorrelated with any other signal.

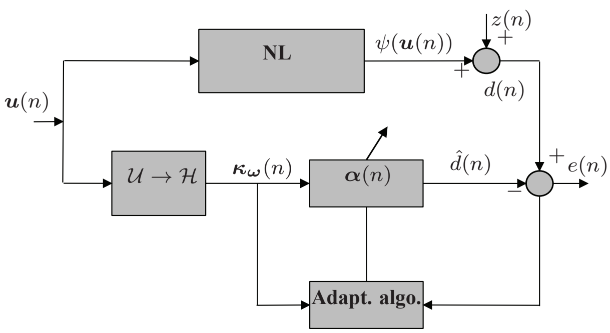

Fig. 1: Kernel-based adaptive system identification.

3. MEAN SQUARE ERROR ANALYSIS

The performance of kernel adaptive filters is a function of the model order, the kernel used, the adaptive algorithm employed and the properties of the signal operating environment (SOE). This paper studies the kernel-based optimal nonlinear filtering problem for stationary SOE, zero-mean white Gaussian input signal $\boldsymbol{u}(n)$, and the so-called Gaussian kernel defined by

$$\kappa(\boldsymbol{u}, \boldsymbol{u}') = \exp\left\{\frac{-\|\boldsymbol{u} - \boldsymbol{u}'\|^2}{2\xi^2}\right\} \tag{6}$$

where ξ is the kernel bandwidth [10]. The stationary SOE assumption holds when a stationary input vector $\boldsymbol{u}(n)$ yields a stationary vector $\psi(\boldsymbol{u}(n))$ at the output of the nonlinearity. This requirement is satisfied by several nonlinear systems used to model practical situations, such as Wiener and Hammerstein systems with memoryless nonlinearities. The following sections study the transient behavior of KLMS under these conditions and for $\{\boldsymbol{u}(n)\}$ a sequence of independent and identically distributed (i.i.d.) Gaussian vectors with possibly correlated components.

Let us denote by $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)$ the vector of kernels at time n, that is,

$$\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n) = [\kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_1)), \ldots, \kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_M))]^\top. \tag{7}$$

where the $\boldsymbol{u}(\omega_i)$'s denote input vectors chosen to build the dictionary. From Fig. 1, the estimated desired response is given by

$$\hat{d}(n) = \boldsymbol{\alpha}^\top(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n), \tag{8}$$

where $\boldsymbol{\alpha}(n) = [\alpha_1(n), \ldots, \alpha_M(n)]^\top$. The estimation error is

$$e(n) = d(n) - \hat{d}(n). \tag{9}$$

Squaring both sides of (9) and taking the expected value yields the mean-square error (MSE)

$$J_{ms}(n) = E[d^2(n)] - 2\boldsymbol{p}_{\kappa d}\boldsymbol{\alpha}(n) + \boldsymbol{\alpha}^\top(n)\boldsymbol{R}_{\kappa\kappa}\boldsymbol{\alpha}(n) \tag{10}$$

where $\boldsymbol{R}_{\kappa\kappa} = E\left[\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\right]$ is the correlation matrix of the input kernel $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)$, and $\boldsymbol{p}_{\kappa d} = E\,[d(n)\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)]$ the cross-correlation vector between $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)$ and $d(n)$.

Assuming $\boldsymbol{R}_{\kappa\kappa}$ positive definite, the optimum weight vector is

$$\boldsymbol{\alpha}_{\text{opt}} = \boldsymbol{R}_{\kappa\kappa}^{-1}\boldsymbol{p}_{\kappa d} \tag{11}$$

and the minimum MSE is given by

$$J_{ms_{\min}} = E\left[d^2(n)\right] - \boldsymbol{p}_{\kappa d}^\top\boldsymbol{R}_{\kappa\kappa}^{-1}\boldsymbol{p}_{\kappa d}. \tag{12}$$

These are the well-known expressions of the Wiener solution and minimum J_{ms}, where the input signal vector has been replaced by the input kernel vector. Thus, determination of $\boldsymbol{\alpha}_{\text{opt}}$ requires the determination of $\boldsymbol{R}_{\kappa\kappa}$ given the statistical properties of the input vector $\boldsymbol{u}(n)$ and the kernel function.

3.1. Kernel-based input correlation matrix

The entries of the correlation matrix $\boldsymbol{R}_{\kappa\kappa}$ are given by

$$[\boldsymbol{R}_{\kappa\kappa}]_{ij} = \begin{cases} E[\kappa^2(\boldsymbol{u}(n), \boldsymbol{u}(\omega_i))], & i = j \\ E[\kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_i))\, \kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_j))], & i \neq j \end{cases} \tag{13}$$

with $1 \leq i, j \leq M$. As $\boldsymbol{u}(\omega_i)$ and $\boldsymbol{u}(\omega_j)$ are input vectors chosen to build the dictionary, they are also i.i.d. Gaussian. Using the results in [11, p.100], it can be shown that the $\boldsymbol{R}_{\kappa\kappa}$ entries are given by

$$[\boldsymbol{R}_{\kappa\kappa}]_{ij} = \begin{cases} \dfrac{\xi^q}{\det(4\boldsymbol{\Sigma}_0 + \xi^2\boldsymbol{I})^{1/2}}, & i = j \\ \dfrac{\xi^{2q}}{\det((\boldsymbol{\Sigma}_0 + \xi^2\boldsymbol{I})(3\boldsymbol{\Sigma}_0 + \xi^2\boldsymbol{I}))^{1/2}}, & i \neq j \end{cases} \tag{14}$$

where q is dimension of $\boldsymbol{u}(n)$, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{\Sigma}_0$ is the autocorrelation matrix of $\boldsymbol{u}(n)$, and $\det(\cdot)$ denotes matrix determinant.

4. GAUSSIAN KERNEL KLMS ANALYSIS

The KLMS weight update equation for the system in Fig. 1 is [10]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\, e(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n). \tag{15}$$

Defining the weight-error vector $\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}_{\text{opt}}$ leads to the weight-error vector update equation

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta\, e(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n). \tag{16}$$

The error equation is given by

$$e(n) = d(n) - \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{v}(n) - \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{\alpha}_{\text{opt}} \tag{17}$$

and, as a consequence, the optimal estimation error is

$$e_0(n) = d(n) - \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{\alpha}_{\text{opt}}. \tag{18}$$

In the following we use the independence assumption (IA) [12] to neglect the statistical dependency of $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)$ and $\boldsymbol{\alpha}(n)$.

4.1. Mean weight behavior

Using (17) in (16) yields

$$\begin{aligned}\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta\, d(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n) - \eta\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{v}(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n) \\ - \eta\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{\alpha}_{\text{opt}}\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n).\end{aligned} \tag{19}$$

Taking the expected value of both sides and using IA yields

$$E[\boldsymbol{v}(n+1)] = (\boldsymbol{I} - \eta \boldsymbol{R}_{\kappa\kappa})\, E[\boldsymbol{v}(n)] \tag{20}$$

which is the LMS mean-weight behavior for an input vector $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)$.

4.2. Mean square error

Using (17) and the IA, the second-order moments of the weights are related to the MSE through [12]

$$J_{ms}(n) = J_{ms_{\min}} + \text{tr}\,\{\boldsymbol{R}_{\kappa\kappa} \boldsymbol{C}_v(n)\} \tag{21}$$

where $\boldsymbol{C}_v(n) = E[\boldsymbol{v}(n)\boldsymbol{v}^\top(n)]$ is the autocorrelation matrix of $\boldsymbol{v}(n)$ and $J_{ms_{\min}} = E[e_0^2(n)]$ is the minimum MSE. The study of the KLMS MSE behavior requires a model for $\boldsymbol{C}_v(n)$. This model is highly affected by the transformation imposed on the input signal $\boldsymbol{u}(n)$ by the kernel. An analytical model for the behavior of $\boldsymbol{C}_v(n)$ is derived in the next section.

4.3. Weight-error correlation matrix behavior

Using (18) and (19), the weight-error vector update becomes

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta\, e_0(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n) - \eta\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{v}(n) \tag{22}$$

Post-multiplying (22) by its transpose and taking the expected value, neglecting the statistical dependence of $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)\boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)$ and $\boldsymbol{v}(n)$, and assuming that $e_0(n)$ is sufficiently close to the optimal solution of the infinite order model so that $E[e_0(n)] \approx 0$, yields

$$\begin{aligned}\boldsymbol{C}_v(n+1) \approx \boldsymbol{C}_v(n) - \eta\, [\boldsymbol{R}_{\kappa\kappa} \boldsymbol{C}_v(n) - \boldsymbol{C}_v(n) \boldsymbol{R}_{\kappa\kappa}] \\ + \eta^2 \boldsymbol{T}(n) + \eta^2 \boldsymbol{R}_{\kappa\kappa} J_{ms_{\min}}\end{aligned} \tag{23a}$$

with

$$\boldsymbol{T}(n) = E\left[\boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)\boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\, \boldsymbol{v}(n)\boldsymbol{v}^\top(n)\, \boldsymbol{\kappa}_{\boldsymbol{\omega}}(n)\boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(n)\right]. \tag{23b}$$

Let us now determine the fourth-order moments in (23b) for the Gaussian kernel. For $\boldsymbol{y} = [\boldsymbol{u}(n)\ \boldsymbol{u}(\omega_i)]^\top$, we have $\|\boldsymbol{u}(n) - \boldsymbol{u}(\omega_i)\|^2 = \boldsymbol{y}^\top \boldsymbol{O} \boldsymbol{y}$, with

$$\boldsymbol{O} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix}, \tag{24}$$

and $\boldsymbol{I}$ the identity matrix. Thus, $E[\boldsymbol{y}] = \boldsymbol{0}$ and

$$\boldsymbol{R}_{\boldsymbol{y}} = E[\boldsymbol{y}\boldsymbol{y}^\top] = \begin{bmatrix} \boldsymbol{\Sigma}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_0 \end{bmatrix}. \tag{25}$$

From [11, p.100], the characteristic function of $\boldsymbol{y}^\top \boldsymbol{O} \boldsymbol{y}$ is given by

$$\psi_z(\beta) = E\{e^{j\beta z}\} = \det(\boldsymbol{I} - 2j\beta \boldsymbol{O} \boldsymbol{R}_y)^{-1/2}. \tag{26}$$

which can be used to obtain the following result

$$E\left\{\kappa(\boldsymbol{u}(\omega_i), \boldsymbol{u}(n))\right\} = E\left\{e^{-\frac{\|\boldsymbol{u}(n)-\boldsymbol{u}(\omega_i)\|^2}{2\xi^2}}\right\} \tag{27}$$

$$= \det(\boldsymbol{I} + \boldsymbol{O}\boldsymbol{R}_y/\xi^2)^{-1/2}. \tag{28}$$

Using IA to determine the element (i,j) of $\boldsymbol{T}(n)$ in (23b) yields

$$[\boldsymbol{T}(n)]_{ij} \approx \sum_{\ell=1}^{M}\sum_{p=1}^{M} E\{\kappa_{\boldsymbol{\omega}_i}(n)\kappa_{\boldsymbol{\omega}_\ell}(n)\kappa_{\boldsymbol{\omega}_p}(n)\kappa_{\boldsymbol{\omega}_j}(n)\}[\boldsymbol{C}_v(n)]_{\ell p} \tag{29}$$

where $\kappa_{\omega_q}(n) = \kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_q))$. Depending on i, j, ℓ and p values we have [11, p.100]:

$\mu_1 := E\{\kappa_{\boldsymbol{\omega}_i}(n)\kappa_{\boldsymbol{\omega}_\ell}(n)\kappa_{\boldsymbol{\omega}_p}(n)\kappa_{\boldsymbol{\omega}_j}(n)\}$ with $i = j = p = \ell$
Denoting $\boldsymbol{y} = [\boldsymbol{u}(n), \boldsymbol{u}(\omega_i)]^\top$, yields

$$\mu_1 = \left[\det(\boldsymbol{I}_2 + \boldsymbol{O}_1 \boldsymbol{R}_{\boldsymbol{y}}/\xi^2)\right]^{-1/2} \tag{30}$$

where $\boldsymbol{O}_1 = \begin{bmatrix} 4\boldsymbol{I} & -4\boldsymbol{I} \\ -4\boldsymbol{I} & 4\boldsymbol{I} \end{bmatrix}$.

$\mu_2 := E\{\kappa_{\boldsymbol{\omega}_i}(n)\kappa_{\boldsymbol{\omega}_\ell}(n)\kappa_{\boldsymbol{\omega}_p}(n)\kappa_{\boldsymbol{\omega}_j}(n)\}$ with $i = j = p \neq \ell$
Denoting $\boldsymbol{y} = [\boldsymbol{u}(n), \boldsymbol{u}(\omega_i), \boldsymbol{u}(\omega_\ell)]^\top$, yields

$$\mu_2 = [\det(\boldsymbol{I}_3 + \boldsymbol{O}_2 \boldsymbol{R}_{\boldsymbol{y}}/\xi^2)]^{-1/2} \tag{31}$$

where $\boldsymbol{O}_2 = \begin{bmatrix} 4\boldsymbol{I} & -3\boldsymbol{I} & -1\boldsymbol{I} \\ -3\boldsymbol{I} & 3\boldsymbol{I} & 0\boldsymbol{I} \\ -1\boldsymbol{I} & 0\boldsymbol{I} & 1\boldsymbol{I} \end{bmatrix}$.

$\mu_3 := E\{\kappa_{\boldsymbol{\omega}_i}(n)\kappa_{\boldsymbol{\omega}_\ell}(n)\kappa_{\boldsymbol{\omega}_p}(n)\kappa_{\boldsymbol{\omega}_j}(n)\}$ with $i = j \neq p = \ell$
Denoting $\boldsymbol{y} = [\boldsymbol{u}(n), \boldsymbol{u}(\omega_i), \boldsymbol{u}(\omega_p)]^\top$, yields

$$\mu_3 = [\det(\boldsymbol{I}_4 + \boldsymbol{O}_3 \boldsymbol{R}_{\boldsymbol{y}})/\xi^2]^{-1/2} \tag{32}$$

where $\boldsymbol{O}_3 = \begin{bmatrix} 4\boldsymbol{I} & -2\boldsymbol{I} & -2\boldsymbol{I} \\ -2\boldsymbol{I} & 2\boldsymbol{I} & 0\boldsymbol{I} \\ -2\boldsymbol{I} & 0\boldsymbol{I} & 2\boldsymbol{I} \end{bmatrix}$.

$\mu_4 := E\{\kappa_{\boldsymbol{\omega}_i}(n)\kappa_{\boldsymbol{\omega}_\ell}(n)\kappa_{\boldsymbol{\omega}_p}(n)\kappa_{\boldsymbol{\omega}_j}(n)\}$ with $i = j \neq p \neq \ell$
Denoting $\boldsymbol{y} = [\boldsymbol{u}(n), \boldsymbol{u}(\omega_i), \boldsymbol{u}(\omega_\ell), \boldsymbol{u}(\omega_p)]^\top$, yields

$$\mu_4 = [\det(\boldsymbol{I}_5 + \boldsymbol{O}_4 \boldsymbol{R}_{\boldsymbol{y}})/\xi^2]^{-1/2} \tag{33}$$

where $\boldsymbol{O}_4 = \begin{bmatrix} 4\boldsymbol{I} & -2\boldsymbol{I} & -1\boldsymbol{I} & -1\boldsymbol{I} \\ -2\boldsymbol{I} & 2\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} \\ -1\boldsymbol{I} & 0\boldsymbol{I} & 1\boldsymbol{I} & 0\boldsymbol{I} \\ -1\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} & 1\boldsymbol{I} \end{bmatrix}$.

$\mu_5 := E\{\kappa_{\boldsymbol{\omega}_i}(n)\kappa_{\boldsymbol{\omega}_\ell}(n)\kappa_{\boldsymbol{\omega}_p}(n)\kappa_{\boldsymbol{\omega}_j}(n)\}$ with $i \neq j \neq p \neq l$
Denoting $\boldsymbol{y} = [\boldsymbol{u}(n), \boldsymbol{u}(\omega_i), \boldsymbol{u}(\omega_j)\boldsymbol{u}(\omega_\ell), \boldsymbol{u}(\omega_p)]^\top$, yields

$$\mu_5 = [\det(\boldsymbol{I}_6 + \boldsymbol{O}_5 \boldsymbol{R}_{\boldsymbol{y}})/\xi^2]^{-1/2} \tag{34}$$

where $\boldsymbol{O}_5 = \begin{bmatrix} 4\boldsymbol{I} & -1\boldsymbol{I} & -1\boldsymbol{I} & -1\boldsymbol{I} & -1\boldsymbol{I} \\ -1\boldsymbol{I} & 1\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} \\ -1\boldsymbol{I} & 0\boldsymbol{I} & 1\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} \\ -1\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} & 1\boldsymbol{I} & 0\boldsymbol{I} \\ -1\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} & 0\boldsymbol{I} & 1\boldsymbol{I} \end{bmatrix}$.

which completes the evaluation of $\boldsymbol{T}(n)$ in (23). Few iterations of (23) show that $\boldsymbol{T}(n)$ assumes the same form as $\boldsymbol{R}_{\kappa\kappa}$. Identifying the diagonal and off-diagonal entries of $\boldsymbol{T}(n)$ leads to

$$\boldsymbol{T}(n) = \text{tr}\{(\boldsymbol{T}_1 - \boldsymbol{T}_2)\, \boldsymbol{C}_v(n)\}\boldsymbol{I}_M + \text{tr}\{\boldsymbol{T}_2\, \boldsymbol{C}_v(n)\}\boldsymbol{1}\boldsymbol{1}^\top \tag{35}$$

where $\boldsymbol{1}$ is a M-by-1 unit vector, and $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are given by

$$\boldsymbol{T}_{1,2} = \begin{bmatrix} \mu_{1,2} & \mu_{2,3} & \mu_{2,4} & \mu_{2,4} & \cdots & \mu_{2,4} \\ \mu_{2,3} & \mu_{3,2} & \mu_{4,4} & \mu_{4,4} & \cdots & \mu_{4,4} \\ \mu_{2,4} & \mu_{4,4} & \mu_{3,4} & \mu_{4,5} & \cdots & \mu_{4,5} \\ \mu_{2,4} & \mu_{4,4} & \mu_{4,5} & \mu_{3,4} & \cdots & \mu_{4,5} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{2,4} & \mu_{4,4} & \mu_{4,5} & \mu_{4,5} & \cdots & \mu_{3,4} \end{bmatrix} \tag{36}$$

Finally, the recursive equation of the weight-error correlation matrix can be computed as follows

$$\begin{aligned}\boldsymbol{C}_v(n+1) &= \boldsymbol{C}_v(n) - \eta\left[\boldsymbol{R}_{\kappa\kappa}\boldsymbol{C}_v(n) + \boldsymbol{C}_v(n)\boldsymbol{R}_{\kappa\kappa}\right] \\ &+ \eta^2\,\mathrm{tr}\left\{(\boldsymbol{T}_1 - \boldsymbol{T}_2)\,\boldsymbol{C}_v(n)\right\}\boldsymbol{I}_M \\ &+ \eta^2\,\mathrm{tr}\left\{\boldsymbol{T}_2\,\boldsymbol{C}_v(n)\right\}\mathbf{1}\mathbf{1}^\top + \eta^2\,J_{ms_{\min}}\boldsymbol{R}_{\kappa\kappa}.\end{aligned} \quad (37)$$

5. SIMULATION RESULTS

This section presents Monte Carlo simulation results (averaged over 1000 runs) to verify the accuracy of the theoretical model. A coherence level $\varepsilon_0 = 10^{-5}$ has been used with (5) to build the dictionary in all simulations. For a given i-th realization of the statistical simulations, $i = 1, \ldots, 1000$, the dictionary dimension M_i and the input vectors used to initialize the dictionary were determined from the observation of 500 samples of $\boldsymbol{u}(n)$. The dimension M_i was determined as the minimum required to achieve the coherence level ε_0, given the kernel bandwidth ξ. Thus, the dimension of the dictionary was in general different for each realization. Once determined, however, the dictionary dimension M_i remained fixed throughout the entire i-th realization. The value of M used in the theoretical model was equal to the average of all M_i, $i = 1, \ldots, 1000$.

The function $\psi(\boldsymbol{u}(n))$ in Fig. 1 was the Wiener model [13]

$$\psi\left(\boldsymbol{u}(n)\right) = \left[b_1 u(n) + b_2 u(n-1)\right]^2 \quad (38)$$

The input signal was a sequence of statistically independent vectors $\boldsymbol{u}(n) = [u_1(n)\ u_2(n)]^\top$ with correlated samples satisfying $u_1(n) = 0.65 u_2(n) + \eta_u(n)$ so that $\sigma_{u_1}^2 = 1$. The nonlinear system output was corrupted by an i.i.d. noise $z(n) \sim \mathcal{N}(0, \sigma_z^2 = 10^{-6})$.

Table 1 shows a summary of the simulation results for $\xi = 0.51$ and $\xi = 0.65$. Different values of M have been obtained for each case. The values of η were chosen so that both implementations would lead to the same steady-state MSE J_{ms_∞}. N_∞ is the number of iterations required for MSE convergence to within 1dB of $J_{ms_{ex_\infty}}$, where $J_{ms_{ex_\infty}}$ is the steady-state $J_{ms_{ex}}$. Note that the choice of $\xi = 0.65$ leads to faster convergence and smaller computational complexity (smaller value of M). Fig. 2 shows the Monte Carlo simulations and the behaviors predicted by the theoretical model for both cases. Both plots show excellent matching between theory and simulations. These results clearly show that the derived model can be used for design purposes, as it allows the prediction of the algorithm behavior for different choices of parameters ξ and η. Moreover, these examples show the influence of the value of the kernel bandwidth ξ on the algorithm performance, since the step size η is basically the same in both cases.

Table 1: Summary of the Simulation Results.

ξ	M	η	$J_{ms_{\min}}$ [dB]	J_{ms_∞} [dB]	$J_{ms_{ex_\infty}}$ [dB]	N_∞
0.51	5	0.038	-18.435	-18.416	-41.987	3076
0.65	3	0.041	-18.459	-18.439	-41.920	1951

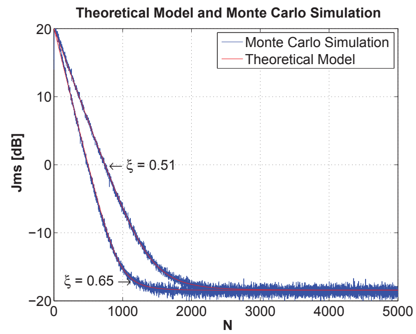

Fig. 2: KLMS behavior for two different kernel bandwidths.

6. CONCLUSIONS

A transient analysis of the KLMS adaptive algorithm implemented using the Gaussian kernel has been presented. The input signal kernel statistics and its effects on the algorithm performance in nonlinear adaptive estimation with white Gaussian inputs have been studied. Analytical models have been derived for the mean and mean-square behaviors of the adaptive weights. Monte Carlo simulation results have illustrated the accuracy of the proposed model, and its applicability for design. It has been verified that the choice of the kernel bandwidth can significantly modify the adaptive filter performance.

7. REFERENCES

[1] M. Schetzen, *The Volterra and Wiener theory of the nonlinear systems*, Wiley, New York, NY, 1980.

[2] N. Wiener, *Nonlinear problems in random theory*, Wiley, New York, NY, 1958.

[3] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, 1950.

[4] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Appl.*, vol. 33, pp. 82–95, 1971.

[5] D. L. Duttweiler and T. Kailath, "An RKHS approach to detection and estimation theory: Some parameter estimation problems (Part V)," *IEEE Trans. Inf. Theory*, vol. 19, no. 1, pp. 29–37, 1973.

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

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

[8] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE ISIT'07*, Nice, France, June 2007, pp. 956–960.

[9] 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, March 2009.

[10] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*, Jonh Wiley & Sons, Inc., 2010.

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

[12] A. H. Sayed, *Fundamentals of adaptive filtering*, John Wiley & Sons, Hoboken, NJ, 2003.

[13] T. Ogunfunmi, *Adaptive Nonlinear System Identification - The Volterra and Weiner Mode Approaches*, Springer, 2007.