STEADY-STATE BEHAVIOR AND DESIGN OF THE GAUSSIAN KLMS 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) Université de Nice Sophia-Antipolis, IUF, CNRS, France, cedric.richard@unice.fr
(3) University of Toulouse, IRIT-ENSEEIHT, CNRS, France, jean-yves.tourneret@enseeiht.fr

ABSTRACT

The Kernel Least Mean Square (KLMS) algorithm is a popular algorithm in nonlinear adaptive filtering due to its simplicity and robustness. In kernel adaptive filters, the statistics of the input to the linear filter depends on the parameters of the kernel employed. A Gaussian KLMS has two design parameters; the step size and the kernel bandwidth. Thus, its design requires analytical models for the algorithm behavior as a function of these two parameters. This paper studies the steady-state behavior and the stability limits of the Gaussian KLMS algorithm for Gaussian inputs. Design guidelines for the choice of the step size and the kernel bandwidth are then proposed based on the analysis results. A design example is presented which validates the theoretical analysis and illustrates its application.

1. INTRODUCTION

Many practical applications (e.g., in communications and bioengineering) require nonlinear signal processing. Nonlinear systems can be characterized by representations ranging from higher-order statistics to series expansion methods [1]. Nonlinear system identification methods based on reproducing kernel Hilbert spaces (RKHS) have gained popularity over the last decades [2, 3]. More recently, kernel adaptive filtering has been recognized an appealing solution to the nonlinear adaptive filtering problem, as working in RKHS allows the use of linear structures to solve nonlinear estimation problems [4]. Algorithms developed using these ideas include the kernel least-mean-square (KLMS) algorithm [5], the kernel recursive-least-square (KRLS) algorithm [6], the kernel-based normalized least-mean-square (KNLMS) algorithm and the affine projection (KAPA) algorithm [7, 8].

In addition to the choice of the usual linear adaptive filter parameters, designing kernel adaptive filters requires the choice of the kernel parameters. The choice of these parameters to achieve a prescribed performance is still an open issue. An analysis of the stochastic behavior of the Gaussian KLMS algorithm, i.e., KLMS with Gaussian kernel, for Gaussian inputs has been presented in [9]. Recursive expressions have been derived for the mean and mean-square adaptive weight behavior.

Building on the results obtained in [9], this paper studies the steady-state behavior and the stability of the Gaussian KLMS algorithm for Gaussian inputs. New expressions are derived for the moments of the linear filter input signal which facilitate the new analysis. A new formulation is proposed for the evolution of the weight second order moments which leads to a closed form expression for the steady-state mean-square error and allows the numerical determination of stability limits. Based on these results, design guidelines are proposed for the choice of the algorithm parameters in order to achieve a prescribed performance.

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

The block diagram of a kernel-based adaptive system identification problem is shown in Figure 1. Here, $\mathcal{U}$ is a compact subspace of $\mathbb{R}^q$, $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a reproducing kernel, $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ is the induced RKHS with its inner product and $z(n)$ is a zero-mean additive noise uncorrelated with any other signal. The representer theorem [2] states that $\psi(\cdot)$ that minimizes the squared estimation error $\sum_{n=1}^{N}[d(n) - \psi(\boldsymbol{u}(n))]^2$ given N input vectors can be written as the kernel expansion $\psi(\cdot) = \sum_{n=1}^{N} \alpha_n \, \kappa(\cdot, \boldsymbol{u}(n))$. For real-time applications, a finite order model

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

must be used, where M is finite and the M kernel functions $\kappa(\cdot, \boldsymbol{u}(\omega_j))$ form the *dictionary*. The model order can be controlled with reduced computational complexity using, for instance, a coherence-based sparsification rule [7, 8] that inserts the kernel $\kappa(\cdot, \boldsymbol{u}(\ell))$ into the dictionary if

$$\max_j |\kappa(\boldsymbol{u}(\ell), \boldsymbol{u}(\omega_j))| \leq \varepsilon_0 \tag{2}$$

where ε_0 determines the coherence of the dictionary. Other criteria and sparsification rules were listed in [4]. In the following, we assume that the dictionary is known and that its size M is finite[1].

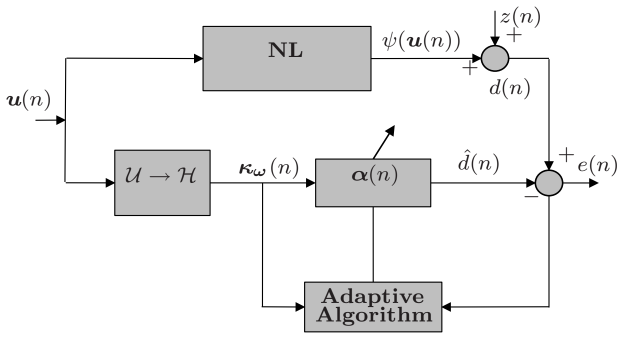


Figure 1: Kernel-based adaptive system identification.

3. MEAN SQUARE ERROR

This paper studies the kernel-based nonlinear adaptive system identification problem illustrated in Figure 1 for a stationary environment, zero-mean independent and identically

[1]It was shown in [7] that M determined under rule (2) is finite.

distributed (i.i.d.) Gaussian $(q \times 1)$ input vectors $\boldsymbol{u}(n)$ so that one has $E\{\boldsymbol{u}(n-i)\boldsymbol{u}^\top(n-j)\} = \boldsymbol{0}$ for $i \neq j$, and for the Gaussian kernel

$$\kappa(\boldsymbol{u}, \boldsymbol{u}') = \exp\left\{\frac{-\|\boldsymbol{u} - \boldsymbol{u}'\|^2}{2\xi^2}\right\} \tag{3}$$

where ξ is the kernel bandwidth [4]. The components of the input vector $\boldsymbol{u}(n)$ can be correlated. The stationarity assumption holds when $\psi(\boldsymbol{u}(n))$ is stationary for $\boldsymbol{u}(n)$ stationary. This is satisfied by several nonlinear systems used to model practical situations, such as memoryless, Wiener and Hammerstein systems.

Let $\boldsymbol{\kappa_\omega}(n)$ be the vector of kernels at time n, that is,

$$\boldsymbol{\kappa_\omega}(n) = [\kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_1)), \ldots, \kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_M))]^\top, \tag{4}$$

where the $\boldsymbol{u}(\omega_i)$, $i = 1, \ldots, M$ denotes input vectors chosen to build the dictionary. From Figure 1 and (1),

$$\hat{d}(n) = \boldsymbol{\alpha}^\top(n)\, \boldsymbol{\kappa_\omega}(n), \tag{5}$$

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

$$e(n) = d(n) - \hat{d}(n). \tag{6}$$

Squaring both sides of (6) and taking the expected value yields the mean-square error (MSE) $J_{ms}(n) = E[e^2(n)]$

$$J_{ms}(n) = E[d^2(n)] - 2\boldsymbol{p}_{\kappa d}^\top \boldsymbol{\alpha}(n) + \boldsymbol{\alpha}^\top(n)\, \boldsymbol{R}_{\kappa\kappa}\, \boldsymbol{\alpha}(n) \tag{7}$$

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

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

$$\boldsymbol{\alpha}_{\text{opt}} = \boldsymbol{R}_{\kappa\kappa}^{-1} \boldsymbol{p}_{\kappa d} \tag{8}$$

and the minimum MSE is given by

$$J_{\min} = E\left[d^2(n)\right] - \boldsymbol{p}_{\kappa d}^\top \boldsymbol{R}_{\kappa\kappa}^{-1} \boldsymbol{p}_{\kappa d}. \tag{9}$$

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, determining the optimum $\boldsymbol{\alpha}_{\text{opt}}$ requires the determination of $\boldsymbol{R}_{\kappa\kappa}$ given the statistical properties of $\boldsymbol{u}(n)$ and the kernel function.

3.1 Input kernel vector correlation matrix

Let us introduce the following notations

$$\|\boldsymbol{u}(n) - \boldsymbol{u}(\omega_\ell)\|^2 = \boldsymbol{y}_2^\top\, \boldsymbol{O}_2\, \boldsymbol{y}_2$$
$$\|\boldsymbol{u}(n) - \boldsymbol{u}(\omega_\ell)\|^2 + \|\boldsymbol{u}(n) - \boldsymbol{u}(\omega_p)\|^2 = \boldsymbol{y}_3^\top\, \boldsymbol{O}_3\, \boldsymbol{y}_3, \quad \ell \neq p$$

with

$$\boldsymbol{y}_2 = [\boldsymbol{u}^\top(n)\ \boldsymbol{u}^\top(\omega_\ell)]^\top$$
$$\boldsymbol{y}_3 = [\boldsymbol{u}^\top(n)\ \boldsymbol{u}^\top(\omega_\ell)\ \boldsymbol{u}^\top(\omega_p)]^\top$$

and

$$\boldsymbol{O}_2 = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \qquad \boldsymbol{O}_3 = \begin{bmatrix} 2\boldsymbol{I} & -\boldsymbol{I} & -\boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

where $\boldsymbol{I}$ is the $(q \times q)$ identity matrix and $\boldsymbol{0}$ is the null matrix. Then, the (i,j)-th element of $\boldsymbol{R}_{\kappa\kappa}$ can be determined using the results in [10][2]

$$[\boldsymbol{R}_{\kappa\kappa}]_{ij} = \begin{cases} \det\left(\boldsymbol{I}_2 + 2\,\boldsymbol{O}_2\, \boldsymbol{R}_2/\xi^2\right)^{-1/2}, & i = j \\ \det\left(\boldsymbol{I}_3 + \boldsymbol{O}_3\, \boldsymbol{R}_3/\xi^2\right)^{-1/2}, & i \neq j \end{cases} \tag{10}$$

with $\boldsymbol{R}_k$ the $(kq \times kq)$ correlation matrix of the vector $\boldsymbol{y}_k$, $\boldsymbol{I}_k$ the $(kq \times kq)$ identity matrix, and $\det(\cdot)$ the matrix determinant.

4. SECOND-ORDER MOMENT ANALYSIS

The KLMS weight update equation for the system presented in Figure 1 is [4]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\, e(n)\, \boldsymbol{\kappa_\omega}(n). \tag{11}$$

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

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n)\ + \eta\, e(n)\, \boldsymbol{\kappa_\omega}(n). \tag{12}$$

The error equation is given by

$$e(n) = d(n) - \boldsymbol{\kappa_\omega}^\top(n)\, \boldsymbol{v}(n) - \boldsymbol{\kappa_\omega}^\top(n)\, \boldsymbol{\alpha}_{\text{opt}} \tag{13}$$

and the optimal estimation error is

$$e_0(n) = d(n) - \boldsymbol{\kappa_\omega}^\top(n)\, \boldsymbol{\alpha}_{\text{opt}}. \tag{14}$$

Substituting equation (13) into equation (12) yields

$$\begin{aligned} \boldsymbol{v}(n+1) &= \boldsymbol{v}(n) + \eta\, d(n)\, \boldsymbol{\kappa_\omega}(n) \\ &\quad - \eta\, \boldsymbol{\kappa_\omega}^\top(n)\, \boldsymbol{v}(n)\, \boldsymbol{\kappa_\omega}(n) - \eta\, \boldsymbol{\kappa_\omega}^\top(n)\, \boldsymbol{\alpha}_{\text{opt}}\, \boldsymbol{\kappa_\omega}(n). \end{aligned} \tag{15}$$

Using (13) and the independence assumption (IA), [11], we obtain the expression for the MSE

$$J_{ms}(n) = J_{\min} + \text{tr}\{\boldsymbol{R}_{\kappa\kappa}\, \boldsymbol{C}_v(n)\} \tag{16}$$

where $\boldsymbol{C}_v(n) = E[\boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)]$ is the autocorrelation matrix of $\boldsymbol{v}(n)$ and $J_{\min} = E[e_0^2(n)]$ is the minimum MSE.

Using IA and assuming $e_0(n)$ to be sufficiently close to the optimal solution of the infinite order model so that $E[e_0(n)] \approx 0$, the following recursion has already been derived in [9] for the weight-error correlation matrix:

$$\begin{aligned} \boldsymbol{C}_v(n+1) =&\, \boldsymbol{C}_v(n) - \eta\, [\boldsymbol{R}_{\kappa\kappa}\, \boldsymbol{C}_v(n) - \boldsymbol{C}_v(n)\, \boldsymbol{R}_{\kappa\kappa}] \\ &+ \eta^2\, \boldsymbol{T}(n) + \eta^2\, \boldsymbol{R}_{\kappa\kappa}\, J_{\min} \end{aligned} \tag{17a}$$

with

$$\boldsymbol{T}(n) = E\left[\boldsymbol{\kappa_\omega}(n)\, \boldsymbol{\kappa_\omega}^\top(n)\, \boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\, \boldsymbol{\kappa_\omega}(n)\, \boldsymbol{\kappa_\omega}^\top(n)\right]. \tag{17b}$$

Using IA, the element (i,j) of $\boldsymbol{T}(n)$ is given by

$$\begin{aligned} &[\boldsymbol{T}(n)]_{ij} \approx \\ &\sum_{\ell=1}^{M} \sum_{p=1}^{M} E\{\kappa_{\omega_i}(n)\, \kappa_{\omega_\ell}(n)\, \kappa_{\omega_p}(n)\, \kappa_{\omega_j}(n)\}\, [\boldsymbol{C}_v(n)]_{\ell p} \end{aligned} \tag{18}$$

where $\kappa_{\omega_q}(n) = \kappa(\boldsymbol{u}(n), \boldsymbol{u}(\omega_q))$. Depending on i, j, ℓ and p, we have [10, p. 100]:

[2] Note that as $\boldsymbol{u}(\omega_i)$ and $\boldsymbol{u}(\omega_j)$ are i.i.d., $[\boldsymbol{R}_{\kappa\kappa}]_{ii} = [\boldsymbol{R}_{\kappa\kappa}]_{jj}$ for all i, j and $[\boldsymbol{R}_{\kappa\kappa}]_{ik} = [\boldsymbol{R}_{\kappa\kappa}]_{jk}$ for all $i \neq k$ and $j \neq k$.

$\mu_1 := E\{\kappa_{\omega_i}(n)\,\kappa_{\omega_\ell}(n)\,\kappa_{\omega_p}(n)\,\kappa_{\omega_j}(n)\}$ with $i=j=p=\ell$.
Denoting $\boldsymbol{y}_2 = [\boldsymbol{u}^\top(n)\,\boldsymbol{u}^\top(\omega_i)]^\top$, yields

$$\mu_1 = \left[\det(\boldsymbol{I}_2 + 4\,\boldsymbol{O}_2\,\boldsymbol{R}_2/\xi^2)\right]^{-1/2} \tag{19}$$

$\mu_2 := E\{\kappa_{\omega_i}(n)\,\kappa_{\omega_\ell}(n)\,\kappa_{\omega_p}(n)\,\kappa_{\omega_j}(n)\}$ with $i \neq j = p = \ell$.
Denoting $\boldsymbol{y}_3 = [\boldsymbol{u}^\top(n)\,\boldsymbol{u}^\top(\omega_i)\,\boldsymbol{u}^\top(\omega_j)]^\top$, yields

$$\mu_2 = [\det(\boldsymbol{I}_3 + \boldsymbol{O}_{3'}\,\boldsymbol{R}_3/\xi^2)]^{-1/2} \tag{20}$$

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

$\mu_3 := E\{\kappa_{\omega_i}(n)\,\kappa_{\omega_\ell}(n)\,\kappa_{\omega_p}(n)\,\kappa_{\omega_j}(n)\}$ with $i = j \neq p = \ell$.
Denoting $\boldsymbol{y}_3 = [\boldsymbol{u}^\top(n)\,\boldsymbol{u}^\top(\omega_i)\,\boldsymbol{u}^\top(\omega_p)]^\top$, yields

$$\mu_3 = [\det(\boldsymbol{I}_3 + 2\,\boldsymbol{O}_3\,\boldsymbol{R}_3)/\xi^2]^{-1/2} \tag{21}$$

$\mu_4 := E\{\kappa_{\omega_i}(n)\,\kappa_{\omega_\ell}(n)\,\kappa_{\omega_p}(n)\,\kappa_{\omega_j}(n)\}$ with $i = j \neq p \neq \ell$.
Denoting $\boldsymbol{y}_4 = [\boldsymbol{u}^\top(n)\,\boldsymbol{u}^\top(\omega_i)\,\boldsymbol{u}^\top(\omega_\ell)\,\boldsymbol{u}^\top(\omega_p)]^\top$, yields

$$\mu_4 = [\det(\boldsymbol{I}_4 + \boldsymbol{O}_4\,\boldsymbol{R}_4)/\xi^2]^{-1/2} \tag{22}$$

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

$\mu_5 := E\{\kappa_{\omega_i}(n)\,\kappa_{\omega_\ell}(n)\,\kappa_{\omega_p}(n)\,\kappa_{\omega_j}(n)\}$ with $i \neq j \neq p \neq l$.
Denoting $\boldsymbol{y}_5 = [\boldsymbol{u}^\top(n)\,\boldsymbol{u}^\top(\omega_i)\,\boldsymbol{u}^\top(\omega_j)\,\boldsymbol{u}^\top(\omega_\ell)\,\boldsymbol{u}^\top(\omega_p)]^\top$,

$$\mu_5 = [\det(\boldsymbol{I}_5 + \boldsymbol{O}_5\,\boldsymbol{R}_5)/\xi^2]^{-1/2} \tag{23}$$

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

Finally, the elements of $\boldsymbol{T}(n)$ are given by

$$\begin{aligned}[\boldsymbol{T}(n)]_{ii} =& \mu_1[\boldsymbol{C}_v(n)]_{ii} + \sum_{\substack{\ell=1\\ \ell\neq i}}^{M}\Big\{2\mu_2[\boldsymbol{C}_v(n)]_{\ell i} + \mu_3[\boldsymbol{C}_v(n)]_{\ell\ell} \\ &+ \mu_4 \sum_{\substack{p=1\\ p\neq\{\ell,i\}}}^{M} [\boldsymbol{C}_v(n)]_{\ell p}\Big\}\end{aligned} \tag{24}$$

and, for $j \neq i$,

$$\begin{aligned}[\boldsymbol{T}(n)]_{ij} =& \mu_2\left([\boldsymbol{C}_v(n)]_{jj} + [\boldsymbol{C}_v(n)]_{ii}\right) + 2\mu_3[\boldsymbol{C}_v(n)]_{ij} \\ &+ \sum_{\substack{\ell=1\\ \ell\neq\{j,i\}}}^{M}\Big\{2\mu_4[\boldsymbol{C}_v(n)]_{j\ell} + 2\mu_4[\boldsymbol{C}_v(n)]_{i\ell} \\ &+ \mu_4[\boldsymbol{C}_v(n)]_{\ell\ell} + \mu_5 \sum_{\substack{p=1\\ p\neq\{i,j,\ell\}}}^{M} [\boldsymbol{C}_v(n)]_{\ell p}\Big\}\end{aligned} \tag{25}$$

which completes the evaluation of $\boldsymbol{T}(n)$ in (17) [3].

[3] The details on how μ_i are determined can be found in [9].

5. STEADY-STATE BEHAVIOR

Let $\boldsymbol{c}_v(n)$ be the lexicographic representation of $\boldsymbol{C}_v(n)$, i.e., the matrix is stacked column-wise into a single vector. Consider the family of $(M \times M)$ matrices $\boldsymbol{H}^{ij}$, $1 \leq i, j \leq M$, whose elements are given by

$$(i=j): \begin{cases} [\boldsymbol{H}^{ii}]_{ii} = 1 - 2\eta r_{\mathrm{md}} + \eta^2\mu_1, & \\ [\boldsymbol{H}^{ii}]_{pp} = \eta^2\mu_3, & p \neq i \\ [\boldsymbol{H}^{ii}]_{ip} = \eta^2\mu_2 - \eta r_{\mathrm{od}} = [\boldsymbol{H}^{ii}]_{pi}, & p \neq i \\ [\boldsymbol{H}^{ii}]_{pl} = \eta^2\mu_4, & \text{otherwise} \end{cases} \tag{26}$$

$$(i \neq j): \begin{cases} [\boldsymbol{H}^{ij}]_{ij} = [\boldsymbol{H}^{ij}]_{ji} = \frac{1}{2}(1 - 2\eta r_{\mathrm{md}} + 2\eta^2\mu_3) \\ [\boldsymbol{H}^{ij}]_{pp} = \eta^2\mu_4, \quad p \neq i,j \\ [\boldsymbol{H}^{ij}]_{ii} = [\boldsymbol{H}^{ij}]_{jj} = \eta^2\mu_2 - \eta r_{\mathrm{od}}, \\ [\boldsymbol{H}^{ij}]_{ip} = [\boldsymbol{H}^{ij}]_{pi} = \frac{1}{2}(2\eta^2\mu_4 - \eta r_{\mathrm{od}}), \quad p \neq i,j \\ [\boldsymbol{H}^{ij}]_{pj} = [\boldsymbol{H}^{ij}]_{jp} = \frac{1}{2}(2\eta^2\mu_4 - \eta r_{\mathrm{od}}), \quad p \neq i,j \\ [\boldsymbol{H}^{ij}]_{p\ell} = \eta^2\mu_5, \quad \text{otherwise} \end{cases} \tag{27}$$

where $r_{\mathrm{md}} = [\boldsymbol{R}_{\kappa\kappa}]_{ii}$ and $r_{\mathrm{od}} = [\boldsymbol{R}_{\kappa\kappa}]_{ij}$. Finally, we define the $(M^2 \times M^2)$ symmetric matrix $\boldsymbol{G}$ given by

$$\boldsymbol{G} = \left[\boldsymbol{h}^{11}\ \boldsymbol{h}^{12}\ \ldots\ \boldsymbol{h}^{1M}\ \ldots\ \boldsymbol{h}^{MM}\right] \tag{28}$$

with $\boldsymbol{h}^{\ell p}$ the $(M^2 \times 1)$ lexicographic representation of $\boldsymbol{H}^{\ell p}$.

Using these definitions, it can be shown that the lexicographic representation of (17) can be written as

$$\boldsymbol{c}_v(n+1) = \boldsymbol{G}\,\boldsymbol{c}_v(n) + \eta^2 J_{\min}\,\boldsymbol{r}_{\kappa\kappa} \tag{29}$$

where $\boldsymbol{r}_{\kappa\kappa}$ is the lexicographic representation of $\boldsymbol{R}_{\kappa\kappa}$. The closed-form solution of (29) is then [12]

$$\boldsymbol{c}_v(n) = \boldsymbol{G}^n\ [\boldsymbol{c}_v(0) - \boldsymbol{c}_v(\infty)] + \boldsymbol{c}_v(\infty) \tag{30}$$

where the steady-state value of $\boldsymbol{c}_v(n)$ is given by

$$\boldsymbol{c}_v(\infty) = \eta^2 J_{\min}(\boldsymbol{I} - \boldsymbol{G})^{-1}\boldsymbol{r}_{\kappa\kappa}. \tag{31}$$

From (31) and (16), we know now that the steady-state of the MSE is given by

$$J_{ms}(\infty) = J_{\min} + \mathrm{tr}\{\boldsymbol{R}_{\kappa\kappa}\,\boldsymbol{C}_v(\infty)\} \tag{32}$$

where $\mathrm{tr}\{\boldsymbol{R}_{\kappa\kappa}\,\boldsymbol{C}_v(\infty)\}$ is the steady-state of the excess MSE, denoted by $J_{ex}(\infty)$.

Note that matrix $\boldsymbol{G}$ is symmetric, which implies that it can be diagonalized and all its eigenvalues are real-valued. Consequently, a necessary and sufficient condition for asymptotic stability of (29) is that all the eigenvalues of $\boldsymbol{G}$ lie inside the interval $]-1, 1[$. The stability limit for η can thus be numerically determined for given values of M and ξ.

6. TIME FOR CONVERGENCE

Assuming convergence, we define time for convergence as the number N_ϵ of iterations required for (29) to reach the condition

$$||\boldsymbol{c}_v(\infty) - \boldsymbol{c}_v(N_\epsilon)|| \leq \epsilon \tag{33}$$

where ϵ is a design parameter, here used equal to 10^{-2}.

7. DESIGN GUIDELINES

The analysis results are now used to establish design guidelines. Suppose one wishes to obtain a MSE which is less than a specified value $J_{\max}$. The following procedure can be used:

- For the given input signal, run a set of tests for different values of ξ in which dictionaries are built for different coherence values ε_0. Then, choose ε_0 that yields a suitable range of values of M for a wide range of ξ values. From the test results, using the chosen ε_0, combine each value of ξ with the value of M obtained from simulation in pairs (ξ, M).
- Using $d(n)$ measurements, estimate $E[d^2(n)]$ and $\boldsymbol{p}_{\kappa d}$. Then, use (9) to obtain an estimate of $J_{\min}$.
- From the (ξ, M) pairs, keep only those that satisfy the constraint $J_{\min} < J_{\max}$.
- For each remaining pair (ξ, M), compute the eigenvalues of $\boldsymbol{G}$ for different values of η and keep the maximum value of η, denoted as $\eta_{\max}(\xi, M)$, ensuring these eigenvalues belong to]-1;1[.
- Using suitable (ξ, M) pairs and (32), search for values of η so that $\eta < \eta_{\max}(\xi, M)$, $J_{ms}(\infty) \leq J_{\max}$ and the convergence time N_ϵ satisfies (33) for a suitable value of ϵ. Choose among the possible solutions using some additional criterion if necessary.

8. SIMULATION RESULTS

A design example is now presented to verify the theory and illustrate the design procedure. The input signal was a sequence of statistically independent vectors

$$\boldsymbol{u}(n) = [u_1(n) \; u_2(n)]^\top \tag{34}$$

with correlated samples satisfying $u_1(n) = 0.65u_2(n) + \eta_u(n)$, with $u_2(n)$ white Gaussian with variance $\sigma_{u_2}^2 = 1$ and $\eta_u(n)$ white Gaussian so that $\sigma_{u_1}^2 = 1$. The nonlinear system to be identified was defined by

$$\psi\left(\boldsymbol{u}(n)\right) = \sum_{i=0}^{3} a^{i+1} \exp\left\{ \frac{-||\boldsymbol{u}(n-i) - \boldsymbol{b}_i||^2}{s_i^2} \right\} \tag{35}$$

where $a = 0.5$ and

$$\boldsymbol{b}_0 = [-0.1454 \quad -0.3862]^\top \qquad \boldsymbol{b}_1 = [1.3162 \quad -0.7965]^\top$$
$$\boldsymbol{b}_2 = [0.1354 \;\; 0.4178]^\top \qquad \boldsymbol{b}_3 = [0.8199 \quad -0.8544]^\top$$

$$\boldsymbol{s} = [0.8063 \;\; 0.9873 \;\; 0.2756 \;\; 0.7662]$$

with s_i in (35) being the i-th entry of $\boldsymbol{s}$. The nonlinear system output was corrupted by a zero-mean white Gaussian noise $z(n)$, with variance $\sigma_z^2 = 10^{-6}$. The required $J_{\max}$ was set to -16.8 dB.

After several tests, a coherence level $\varepsilon_0 = 10^{-3}$ has been chosen. Kernel bandwidth ξ was varied from 0.1 to 50 in increments of 0.01. For each value of ξ, dictionary dimensions M_i were determined for $i = 1, \ldots, 1000$ realizations of the input process. Each realization used 500 input vectors $\boldsymbol{u}(n)$. Each M_i was determined as the minimum value of M required to achieve the coherence level ε_0. The value $M(\xi)$ was determined as the average of all M_i, $i = 1, \ldots, 1000$.

$J_{\min}(\xi)$ was determined from (9) for each pair (ξ, M). Figure 2 shows the pairs (ξ, M) satisfying the design objective $J_{\min}(\xi) < -16.8$ dB. The corresponding values of $J_{\min}(\xi, M)$ are shown in Figure 3. Interpolation was used to facilitate the visualization.

Table 1 illustrates three possible designs, for $\xi = 0.78, 1.04$ and 1.33. For each pair (ξ, M), the step-size η was chosen so that the algorithm was stable (η less than $\eta_{\max}(\xi, M)$ determined from the eigenvalues of $\boldsymbol{G}$) and $J_{ms}(\infty) < -16.8$ dB. The values of $J_{ms}(\infty)$ and $J_{ex}(\infty)$ were determined from (32) and N_ϵ was obtained from (33) for $\epsilon = 10^{-2}$. From these three cases, using the pair $(M, \xi, \eta) = (2, 1.33, 0.0601)$ seems to yield a good design choice, as it leads to $J_{\min} = -16.894$ dB $< J_{\max}$ and $J_{ms}(\infty) = -16.803$dB with convergence time $N_\epsilon = 317$ iterations.

Other design criteria could be accommodated using the results derived in this paper. Figures 4 and 5 show, respectively, the achievable MSE and excess MSE determined from (32) for the chosen range of values for ξ. Figure 6 shows the corresponding convergence times for $\epsilon = 10^{-2}$ in (33). All plots are interpolated for easier visualization. From these figures, it is clear that different design choices could be accommodated.

Finally, Figure 7 shows Monte Carlo simulation results to illustrate the accuracy of the analytical model derived for the three cases shown in Table 1. Figure 7(a) shows excellent matching between simulations, averaged over 500 runs, and the theoretical predictions from (16) and (30). Figures 7(b)-(d) compare simulated steady-state results with theoretical predictions using (31). The matching is clearly excellent.

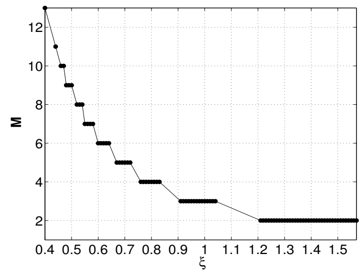


Figure 2: Average dictionary length with $J_{\min}(\xi) < J_{\max}$.

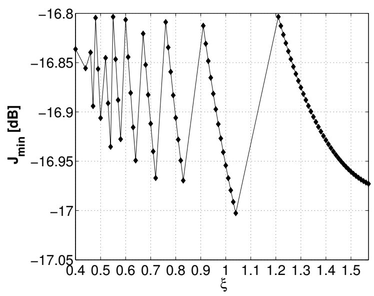


Figure 3: $J_{\min}(\xi)$ such that $J_{\min}(\xi) < J_{\max}$.

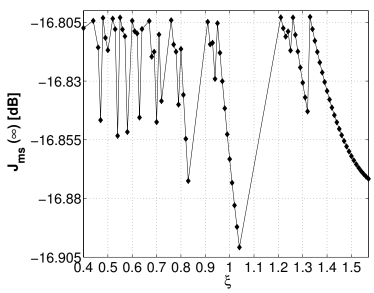


Figure 4: Steady-state of MSE with $J_{ms} < J_{\max}$.

9. CONCLUSIONS

This paper studied the steady-state behavior and the stability of the Gaussian KLMS algorithm for Gaussian inputs. New analytical results were presented to describe the KLMS steady-state performance. Moreover, a new recursive expression was provided for the time evolution of the adaptive

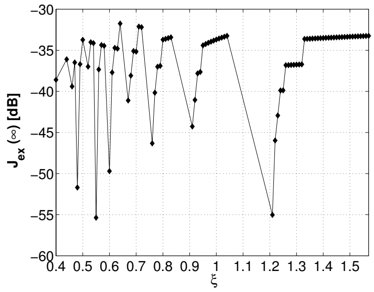


Figure 5: Steady-state of excess MSE with $J_{ms} < J_{\max}$.

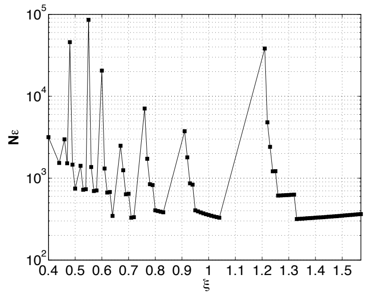
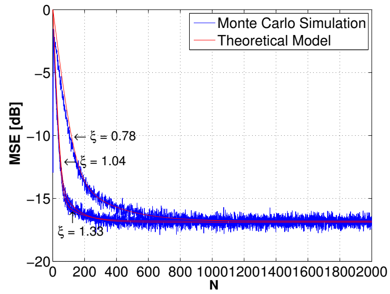


Figure 6: Number of iterations needed for convergence of the filter with $J_{ms} < J_{\max}$.

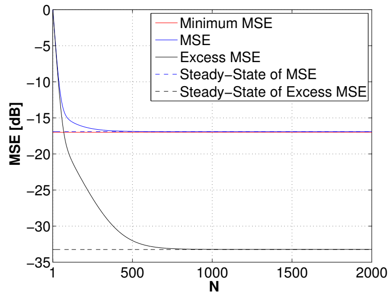
Table 1: Summary of the simulation results.

ξ	M	η	$J_{\min}$ [dB]	$J_{ms}(\infty)$ [dB]	$J_{ex}(\infty)$ [dB]	N_ϵ
0.78	4	0.0298	-16.859	-16.817	-37.005	844
1.04	3	0.0618	-17.001	-16.901	-33.250	329
1.33	2	0.0601	-16.894	-16.803	-33.626	317

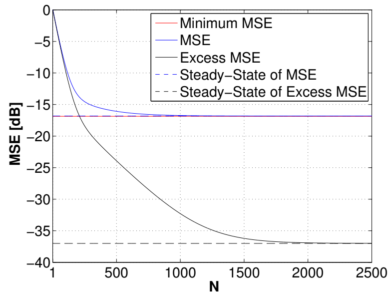
weight-error vector fluctuations which allows the numerical determination of the step-size stability limit for given kernel bandwidth and model order. Based on these original theoretical results, new design guidelines were proposed for the KLMS algorithm. A design example was presented to verify the accuracy of the theory and to illustrate its applicability in design.



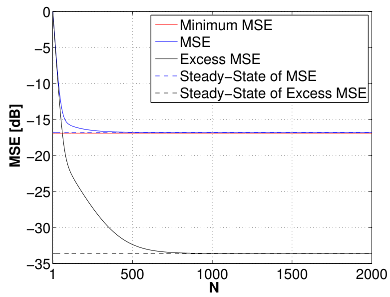
(a) Theoretical model and Monte Carlo simulation.



(b) Theoretical model with $\xi = 0.78$ and $M = 4$.



(c) Theoretical model with $\xi = 1.04$ and $M = 3$.



(d) Theoretical model with $\xi = 1.33$ and $M = 2$.

Figure 7: Behavior of KLMS for different kernel bandwidths.

REFERENCES

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

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

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] W. D. Parreira, J. C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the gaussian kernel least mean square algorithm," in *Proc. IEEE ICASSP'11*, Prague, Czech Republic, May 2011.

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

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

[12] D. G. Luenberger, *Introduction Dynamic Systems: Theory, Models & Applications*, Jonh Wiley & Sons, Inc., Stanford, CA, 1979.