A Design Methodology for the Gaussian KLMS Algorithm

P. Pedrosa, J. Bermudez, Cédric Richard

A Design Methodology for the Gaussian KLMS Algorithm

P. Pedrosa[1,2], J. C. M. Bermudez[2], C. Richard[3]
[1]Instituto de Telecomunicações, Lisbon, Portugal
[2]Universidade Federal de Santa Catarina, Florinanópolis, SC, Brazil
[3]Université de Nice Sophia-Antipolis, CNRS, France
ppedrosa@lx.it.pt j.bermudez@ieee.org cedric.richard@unice.fr

***Abstract*—The Gaussian kernel least-mean-square (Gaussian KLMS) algorithm has been studied under different implementation conditions. Though analytical models that predict its behavior are available, methodologies for determining the algorithm parameter values to satisfy given design criteria is still missing from the literature. In this paper we propose, test, and validate a methodology for the design of the Gaussian KLMS algorithm. Designing the algorithm consists in selecting adequate values for its free parameters from available theoretical performance models. These parameters comprise the filter length, the adaptive step-size, and the kernel bandwidth. The objective is to achieve specific design objectives, e.g., fast convergence time, good steady-state performance and/or reduced computational load. These goals are quantified in terms of different performance measures. Particularly, the time to convergence, the residual mean-squared-error (MSE), and the filter order.**

I. Introduction

Nonlinear adaptive filtering in reproducing kernel Hilbert spaces (RKHS) has been shown to be an effective technique to solve important nonlinear estimation problems. It has been applied to different domains such as system identification, noise cancellation, echo control, among others [1].

Linear adaptive filters are simple to implement. However, their design can be quite complex as they are time-variant, stochastic in nature, and have weight updates that are nonlinear in the input signal [2]. That is why deriving good analytical models for the behavior of adaptive algorithms under different application environments is so important. The design challenge becomes significantly more complex for nonlinear adaptive filters, as the dependence of the performance measures on the filter parameters tends to be highly nonlinear even for analytical models derived under several approximations. Hence, having good analytical models for nonlinear adaptive filter behavior does not imply having good design methodologies. This is generally true for any type of nonlinear adaptive filter. One type of solution that has been attracting a lot of interest in the recent years is the kernel-based nonlinear adaptive algorithm.

The simplest and more popular kernel-based adaptive algorithm is certainly the Gaussian KLMS. Its behavior has been studied in the last few years under different operating conditions. For instance, the KLMS asymptotic behavior has been studied in [3] for small step-sizes. A statistical analysis was performed in [4], resulting in recursive expressions for the mean weight error vector and the mean-square error (MSE) for Gaussian inputs. In [5] a similar analysis has been carried out considering a pre-tuned dictionary, a condition that is attractive to reduce the on-line computational complexity. This analysis has led to accurate analytical models for algorithm performance, conditioned on the dictionary. In [6], the need for an on-line dictionary update was studied in the event of a non-stationary environment.

One aspect that has not been considered in these works is how to choose the parameters of the KLMS algorithm to obtain a desired performance. Considering the complexity of the KLMS algorithm behavior, this question has no simple response.

The implementation in the RKHS requires the choice of an additional parameter (the kernel bandwidth). Moreover, the relationships between the performance measures and the design parameters become highly nonlinear, making analytical solutions virtually impossible. Hence, the design methodology must rely on a set of design curves relating the performance measures with the different parameters for the problem at hand.

This work addresses the design of the KLMS algorithm using a pre-tuned dictionary and a Gaussian kernel based on the analytical model derived in [5]. The parameters to be specified are the filter length, the adaptation step-size, and the kernel bandwidth, whereas the performance measures are the convergence speed, and the residual excess mean square estimation error. Design examples are presented to illustrate the application of the proposed methodology, which is then validated against simulation results obtained through Monte Carlo techniques.

This work is organized as follows. After the introduction in Sec. I, we present the KLMS algorithm in Sec. II. The evaluation of the algorithm performance with respect to its design parameters is conducted in Sec. III, and the design methodology proposed, tested, and validated in Sec. IV. Sec. V concludes the paper.

This work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under the projects 400566/2013-3 and 307071/2013-8. P. Pedrosa would like to acknowledge the support of Fundação para a Ciência e a Tecnologia (FCT) under project UID/EEA/50008/2013.

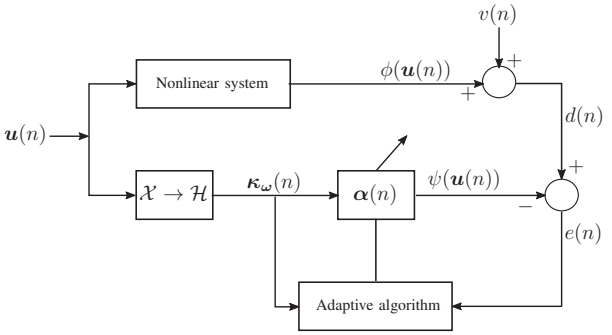

Fig. 1. Nonlinear adaptive filtering using reproducing kernels in Hilbert spaces.

II. The Kernel LMS Algorithm

Consider the nonlinear estimation problem depicted in Fig. 1. The input vectors $\boldsymbol{u}(n)$ are assumed zero-mean, stationary and Gaussian with autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{uu}}$. Vector $\boldsymbol{\kappa}_{\omega}(n) = [\kappa(\boldsymbol{u}(n), \boldsymbol{u}_{\omega_1}) \cdots \kappa(\boldsymbol{u}(n), \boldsymbol{u}_{\omega_m})]^{\top}$ is the finite dimension mapping of the input vector $\boldsymbol{u}(n)$ into the RKHS with kernel $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$. The set $\{\boldsymbol{u}_{\omega_i} : i = 1, \ldots, m\}$ contains the elements of the so-called dictionary [5]. The desired signal is given by $d(n) = \phi(\boldsymbol{u}(n)) + v(n)$, with $v(n)$ being and additive zero-mean Gaussian observation noise with variance σ_v^2. The nonlinear adaptive filter output is given by $\psi(\boldsymbol{u}(n)) = \boldsymbol{\kappa}_{\omega}^{\top}(n)\boldsymbol{\alpha} = \sum_{i=1}^{m} \alpha_i \kappa(\boldsymbol{u}(n), \boldsymbol{u}_{\omega_i})$, where $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_m]^{\top}$ is the adaptive filter weight vector. The estimation error is $e(n) = d(n) - \psi(\boldsymbol{u}(n))$.

From this description we see that the nonlinear estimation in RKHS with a finite dimensional dictionary depicted in Fig. 1 can be interpreted as a linear signal estimation problem in which the input signal vector $\boldsymbol{u}(n)$ has been mapped onto a RKHS, which is characterized by the chosen kernel and by the dictionary. We consider in this work the Gaussian kernel

$$\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp\left(-\frac{\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2}{2\xi^2}\right) \tag{1}$$

where ξ is the kernel bandwidth. The input vector to the adaptive filter is $\boldsymbol{\kappa}_{\omega}(n)$. The weight update equation for KLMS with step-size η is given by [4]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta e(n)\boldsymbol{\kappa}_{\omega}(n) \tag{2}$$

and the properties of the nonlinear mapping from $\boldsymbol{u}(n)$ into $\boldsymbol{\kappa}_{\omega}(n)$ and its effects on the algorithm performance must be taken into account for design.

A. Optimal Solution

Conditioned on the adaptive weights, the estimation error at instant n is given by

$$\begin{aligned} e(n) &= d(n) - \psi(\boldsymbol{u}(n)) \\ &= d(n) - \sum_{i=1}^{m} \alpha_i \kappa(\boldsymbol{u}(n), \boldsymbol{u}_{\omega_i}) \\ &= d(n) - \boldsymbol{\alpha}^{\top}\boldsymbol{\kappa}_{\omega}(n). \end{aligned} \tag{3}$$

Squaring both sides of (3), and taking the expected value leads to the MSE,

$$\begin{aligned} J_{\mathrm{MSE}}(n) &= \mathrm{E}\{e^2(n)\} \\ &= \mathrm{E}\{d^2(n)\} - 2\boldsymbol{p}_{d\kappa}^{\top}\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\top}\boldsymbol{R}_{\kappa\kappa}\boldsymbol{\alpha} \end{aligned} \tag{4}$$

where $\boldsymbol{p}_{d\kappa} = \mathrm{E}\{d(n)\boldsymbol{\kappa}_{\omega}(n)\}$ is the cross-correlation vector between the desired output and the kernelized input vector $\boldsymbol{\kappa}_{\omega}(n)$, and $\boldsymbol{R}_{\kappa\kappa} = \mathrm{E}\{\boldsymbol{\kappa}_{\omega}(n)\boldsymbol{\kappa}_{\omega}^{\top}(n)\}$ is the input autocorrelation matrix conditioned on the chosen dictionary.

The optimal weights $\boldsymbol{\alpha}^o$ are given by [4], [5]

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} J_{\mathrm{MSE}}(n) = \boldsymbol{R}_{\kappa\kappa}^{-1}\boldsymbol{p}_{d\kappa}. \tag{5}$$

Using (5) in (4) results in the minimum MSE (MMSE)

$$J_{\min} = \mathrm{E}\{d^2(n)\} - \boldsymbol{p}_{d\kappa}^{\top}\boldsymbol{R}_{\kappa\kappa}^{-1}\boldsymbol{p}_{d\kappa}. \tag{6}$$

B. Mean-squared Error Analysis

The expression of $\boldsymbol{R}_{\kappa\kappa} = \mathrm{E}\{\boldsymbol{\kappa}_{\omega}(n)\boldsymbol{\kappa}_{\omega}^{\top}(n)\}$ has been determined in [5], and its (i,j)th entry is given by

$$\begin{aligned} [\boldsymbol{R}_{\kappa\kappa}]_{ij} &= \left|\boldsymbol{I} + \tfrac{2}{\xi^2}\boldsymbol{R}_{uu}\right|^{-\frac{1}{2}} \times \\ &\exp\left(-\tfrac{1}{4\xi^2}\left[2\|\bar{\boldsymbol{u}}_{\omega_{ij}}\|^{(2)} - \|\bar{\boldsymbol{u}}_{\omega_{ij}}\|^2_{\left(\boldsymbol{I}-\xi^2\boldsymbol{R}_{uu}^{-1}/2\right)^{-1}}\right]\right). \end{aligned} \tag{7}$$

where $\bar{\boldsymbol{u}}_{\omega_{ij}} \triangleq \boldsymbol{u}_{\omega_i} + \boldsymbol{u}_{\omega_j}$ and $\|\bar{\boldsymbol{u}}_{\omega_{ij}}\|^{(2)} \triangleq \|\boldsymbol{u}_{\omega_i}\|^2 + \|\boldsymbol{u}_{\omega_j}\|^2$. Defining the weight error vector $\boldsymbol{v}(n) = \boldsymbol{\alpha} - \boldsymbol{\alpha}^o$, a recursive update for the lexicographic representation of its correlation matrix $\boldsymbol{C}_v(n) = \mathrm{E}\{\boldsymbol{v}(n)\boldsymbol{v}^{\top}(n)\}$ was obtained in [5] as

$$\boldsymbol{c}_v(n+1) = \boldsymbol{G}\boldsymbol{c}_v(n) + \eta^2 J_{\min}\boldsymbol{r}_{\kappa\kappa} \tag{8}$$

where $\boldsymbol{r}_{\kappa\kappa}$ is the lexicographic representation of $\boldsymbol{R}_{\kappa\kappa}$, and matrix $\boldsymbol{G} = \boldsymbol{I} - \eta(\boldsymbol{G}_1 + \boldsymbol{G}_2) + \eta^2\boldsymbol{G}_3$, where $\boldsymbol{I}$ is the $(m^2 \times m^2)$ identity matrix, $\boldsymbol{G}_1 = \boldsymbol{I} \otimes \boldsymbol{R}_{\kappa\kappa}$, with $\otimes$ denoting the Kronecker product, $\boldsymbol{G}_2 = \boldsymbol{R}_{\kappa\kappa} \otimes \boldsymbol{I}$, and $\boldsymbol{G}_3$ with entries $[\boldsymbol{G}_3]_{i+(j-1)m,l+(p-1)m} = \left[\boldsymbol{K}^{(ij)}\right]_{lp}$ for $i,l,j,p = \{1,2,\ldots,m\}$ and matrix $\boldsymbol{K}^{(ij)}$ has its (l,p)th entry given by $[\boldsymbol{K}^{(ij)}]_{lp} = \mathrm{E}\{\kappa_{\omega_i}(n)\kappa_{\omega_j}(n)\kappa_{\omega_l}(n)\kappa_{\omega_p}(n)\}$ such that [5]

$$\begin{aligned} [\boldsymbol{K}^{(ij)}]_{lp} &= \left|\boldsymbol{I} + \tfrac{4}{\xi^2}\boldsymbol{R}_{uu}\right|^{-\frac{1}{2}} \times \\ &\exp\left(-\tfrac{1}{8\xi^2}\left[4\|\bar{\boldsymbol{u}}_{\omega_{ijlp}}\|^{(2)} - \|\bar{\boldsymbol{u}}_{\omega_{ijlp}}\|^2_{\left(\boldsymbol{I}-\xi^2\boldsymbol{R}_{uu}^{-1}/4\right)^{-1}}\right]\right). \end{aligned} \tag{9}$$

where $\bar{\boldsymbol{u}}_{\omega_{ijlp}} \triangleq \boldsymbol{u}_{\omega_i} + \boldsymbol{u}_{\omega_j} + \boldsymbol{u}_{\omega_l} + \boldsymbol{u}_{\omega_p}$ and $\|\bar{\boldsymbol{u}}_{\omega_{ijlp}}\|^{(2)} \triangleq \|\boldsymbol{u}_{\omega_i}\|^2 + \|\boldsymbol{u}_{\omega_j}\|^2 + \|\boldsymbol{u}_{\omega_l}\|^2 + \|\boldsymbol{u}_{\omega_p}\|^2$.

The MSE can be written as

$$J_{\mathrm{MSE}}(n) = J_{\min} + \mathrm{Trace}\{\boldsymbol{R}_{\kappa\kappa}\boldsymbol{C}_v(n)\}. \tag{10}$$

The last term in (10) is the excess MSE (EMSE).

Observing that $\boldsymbol{c}_v(n) = \boldsymbol{c}_v(n+1) = \boldsymbol{c}_v(\infty)$ in steady-state, and solving (8) for $\boldsymbol{c}_v(\infty)$ yields

$$\boldsymbol{c}_v(\infty) = \eta^2 J_{\min}(\boldsymbol{I} - \boldsymbol{G})^{-1}\boldsymbol{r}_{\kappa\kappa}. \tag{11}$$

Using the matrix form of (11) in (10) yields the residual MSE,

$$J_{\mathrm{MSE}}(\infty) = J_{\min} + \mathrm{Trace}\{\boldsymbol{R}_{\kappa\kappa}\boldsymbol{C}_v(\infty)\}. \tag{12}$$

C. Algorithm stability

The stability limit $\eta_{\max}$ can be obtained by imposing the condition that the eigenvalues of matrix $\boldsymbol{G}$ in (8) be within the unit circle. For a fixed dictionary, the stability limit of the KLMS algorithm can be well approximated by [2]

$$\eta_{\max} = \frac{2}{\mathrm{tr}\{\boldsymbol{R}_{\kappa\kappa}\}}. \tag{13}$$

III. Algorithm Performance Evaluation

To illustrate the impact of each design parameter on the filter performance, we evaluate a set of performance measures with respect to different design parameters using the analytical KLMS model in [5] on a classical nonlinear signal estimation problem.

A. Design Parameters and Performance Measures

The typical design parameters for linear LMS are the filter length and the step-size. Things become more complex in KLMS design since the properties of the input vector correlation matrix depend on the kernel bandwidth due to the mapping to the RKHS. Hence, besides the filter length m (or dictionary size, in the KLMS context) and the step-size η, one needs to choose the kernel bandwidth ξ. Moreover, the relationships connecting these parameters and the performance measures become highly nonlinear in KLMS.

The performance measures used in design may vary depending on the application. However, most of the times one is concerned about steady-state MSE and convergence time. Hence, we approach design with the objective of obtaining a residual MSE below a specified value in a reduced number of iterations. To consider the latter requirement, we define the objective measure of *time to convergence* as the number of iterations n_ε given by

$$n_\varepsilon \triangleq \min_n \{J_{\mathrm{MSE}}(n) \leq (1+\varepsilon) J_{\mathrm{MSE}}(\infty)\}. \tag{14}$$

B. Nonlinear Signal Estimation Problem

Consider an input sequence

$$u(n) = \rho u(n-1) + \sigma_u \sqrt{1-\rho^2} w(n), \tag{15}$$

with $w(n)$ i.i.d. zero-mean Gaussian with variance $\sigma_w^2 = 1$, and a nonlinear system defined as

$$\begin{cases} y(n) = 0.5u(n) - 0.3u(n-1) \\ d(n) = y(n) - 0.5y(n)^2 + 0.1y(n)^3 + v(n) \end{cases} \tag{16}$$

with $v(n)$ a zero-mean Gaussian noise with variance $\sigma_v^2 = 2.5 \times 10^{-3}$. The input vector is $\boldsymbol{u}(n) = [u(n)\ u(n-1)]^\top$, and the reference signal is $d(n)$. We set $\sigma_u = 0.5$ and $\rho = 0.5$.

The dictionary elements are the farthest m equidistant points taken from a $[-3\sigma_u, 3\sigma_u] \times [-3\sigma_u, 3\sigma_u]$ grid. We consider values of $m \in \{4, 9, 16, 25, 36\}$. Such fixed grid dictionaries have been verified to yield very reasonable filter performances [5]. More complex dictionary design approaches could also be considered [1], [7], as long as the dictionary is designed offline and remains fixed during operation. Finally, we consider adaptive step-size values $\eta = r\eta_{\max}$ with $r \in \{0.1, 0.25, 0.5\}$.

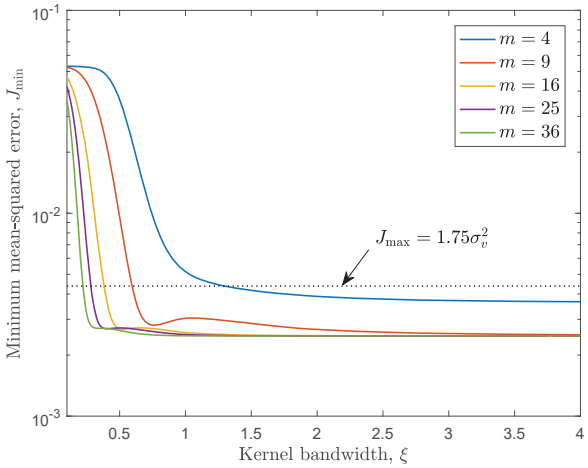

Fig. 2. Minimum MSE *vs.* the kernel bandwidth.

C. Performance Curves

Fig. 2 shows the curves, obtained using (6), for the MMSE $J_{\min}$ as a function of ξ for the different dictionary sizes. The power of $d(n)$ and the cross-correlation vector $\boldsymbol{p}_{d\kappa}$ were estimated from $d(n)$ through Monte Carlo simulations. $\boldsymbol{R}_{\kappa\kappa}$ was determined using (7). Notice that $J_{\min}$ is a function of both m and ξ. We will sometimes express this dependence as $J_{\min}(m, \xi)$ for clarity. Moreover, $J_{\min}$ lower bounds the MSE performance of the algorithm, i.e., $J_{\min} \leq J_{\mathrm{MSE}}(\infty)$. The dashed horizontal line indicates the value of $J_{\max}$, the limiting MSE value specified for the design (see Sec. IV).

Fig. 3 shows the residual MSE, $J_{\mathrm{MSE}}(\infty)$, as a function of ξ for different values of m and η (three values for each m). Each $J_{\mathrm{MSE}}(\infty)$ curve is traced using (6), (9), (11) and (12). Again, this dependency of the performance measure on the design parameters can be made explicit writing $J_{\mathrm{MSE}}(\infty)(m, \xi, \eta)$. Inspecting the figure, we see that larger step-sizes lead to poorer adaptation and that filters with larger dictionaries tend to perform better. Regarding the value of ξ, increasing it beyond a certain value will have no significant impact on $J_{\mathrm{MSE}}(\infty)$. However we will soon see that the time to convergence increases with ξ in that range.

Fig. 4 shows the time to convergence n_ε as a function of ξ for different values of m and η. The curves were plotted using (10), (12), and (14). The dashed horizontal lines indicate examples of limiting values $n_{\max}$ for the intended times to convergence (see Sec. IV). The dependency of n_ε on the design parameter can be explicitly stated by writing $n_\varepsilon(m, \xi, \eta)$. Fig. 4 shows that the n_ε curves typically assume a U-shape that raises or falls inversely to the value of η centered at some value of ξ. Also, the U-shape widens as the value of η increases.

It is important to note that, contrary to the linear case, the conclusions reached above are valid for the nonlinear system (16), but cannot in general be assumed for any other system. Hence the importance of the proposed design methodology based on a set of design curves. The methodology is detailed in the next section.

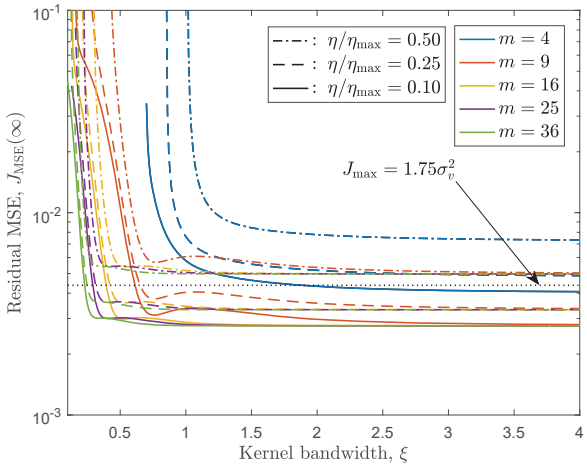

Fig. 3. Residual MSE *vs.* the kernel bandwidth.

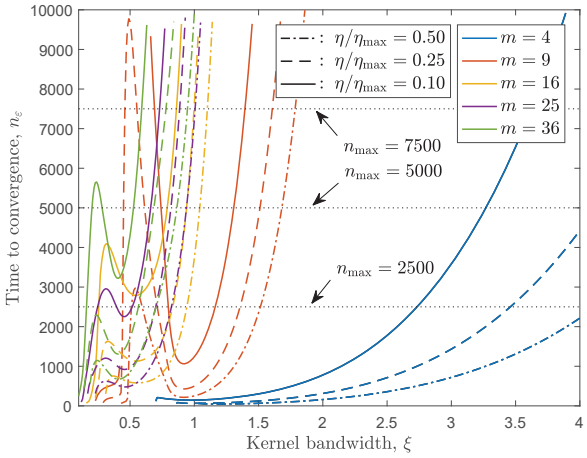

Fig. 4. Time to convergence *vs.* the kernel bandwidth with the steady-state entry-instant parameter $\varepsilon = 0.1$.

IV. Design Methodology

Suppose we want to design KLMS for the problem described in Section III-B with $J_{\mathrm{MSE}}(\infty) < J_{\max}$ and $n_\varepsilon < n_{\max}$. Then, we may determine from Fig. 2 a minimum dictionary size m and a range of values of ξ complying with the MSE constraint. The $J_{\mathrm{MSE}}(\infty)$ curves in Fig. 3 allow the determination of the maximum value of η so that $J_{\mathrm{MSE}}(\infty) < J_{\max}$ for each combination of dictionary size m and kernel bandwidth ξ. Finally, Fig. 4 can be used to determine the constraints on the range of values for ξ as a function of the desired time for convergence η_ε, given choices of values for m and η. Interestingly, increasing the step-size increases also the robustness of the design to the choice of ξ, as it leads to wider U-shaped n_ε curves.

We now detail the proposed methodology for designing the Gaussian KLMS algorithm with speed and adaptation constraints.

A. Speed and Adaptation Constrained Design

Start by setting the following design constraints:

i) $\{J_{\min}, J_{\mathrm{MSE}}(\infty)\} < J_{\max}$;
ii) $n_\varepsilon < n_{\max}$;

and then take the following steps:

1) Choose an $m = m_{\mathrm{initial}}$;
2) Choose an interval $\mathcal{X}_0$ so that $\xi \in \mathcal{X}_0$;
3) Evaluate $J_{\min}(m, \xi)$ (Eqs. (6), and (7));
4) Define $\mathcal{X}_1 = \{\xi | J_{\min}(m, \xi) < J_{\max}\}$. If $\mathcal{X}_1 = \varnothing$, $m = m + 1$ and return to 2);
5) Choose $\eta = \eta_{\mathrm{initial}} < \eta_{\max}$;
6) Evaluate $J_{\mathrm{MSE}}(\infty)(m, \xi, \eta)$ (Eqs. (6), (9), (11), and (12));
7) Define $\mathcal{X}_2 = \{\xi | J_{\mathrm{MSE}}(\infty)(m, \xi, \eta) < J_{\max}\}$. If $\mathcal{X}_2 = \varnothing$, reduce η and return to 6);
8) Evaluate $n_\varepsilon(m, \xi, \eta)$ (Eqs. (10), (12), and (14));
9) Define $\mathcal{X}_3 = \{\xi | n_\varepsilon(m, \xi, \eta) < n_{\max}\}$. If $\mathcal{X}_3 = \varnothing$, relax design constraints or use another algorithm.

B. Test and Validation

Consider the experimental set-up described in Sec. III-B and the corresponding design figures, Fig. 2-Fig. 4. Consider also that our design requires $J_{\max} = 1.75\sigma_v^2$, and the minimum possible time for convergence. Then, Fig. 2 shows that any value of $m \in \{4, 9, 16, 25, 36\}$ can be associated to some $\xi \in [0.1, 4]$ so that $J_{\min}(m, \xi) < J_{\max}$. Note that $m = 4$ would have been excluded if we had specified $J_{\max} = 3 \times 10^{-3}$. Inspecting Fig. 3, we see that the interval $\xi \in \mathcal{X}_2$ can be defined graphically for each pair (m, η) by identifying the minimum value of ξ for which the $J_{\mathrm{MSE}}(\infty)(m, \xi, \eta)$ curve remains below the straight line $J_{\max}$. TABLE I lists the possible design parameters values after Step 4 and after Step 7.

The impact of the step-size value on filter performance is quantified evaluating first $J_{\mathrm{MSE}}(\infty)(m, \xi, \eta)$ and then $n_\varepsilon(m, \xi, \eta)$ (Fig. 3 and Fig. 4, respectively). As an example, we see in Fig. 3 that all curves associated to step-size $\eta = 0.5\eta_{\max}$ stay above $J_{\max} = 1.75\sigma_v^2$ for all $m \in \{4, 9, 16, 25, 36\}$, and $\xi \in [0.1, 4]$. Thus, all tuples (m, ξ, η) with $\eta = 0.5\eta_{\max}$ are discarded. Again, these results are listed in TABLE I.

The final design results are listed in TABLE II for three different values of $n_{\max}$. Intervals $\mathcal{X}_3$ shown in the two rightmost columns were obtained following the proposed design methodology. Column Analysis was obtained using the theoretical model in [5], and column Simulation from Monte Carlo simulation. Comparing the ranges of kernel bandwidth values found analytically and through simulation we see that the mismatch between them are clearly acceptable for design purposes, with the former being slightly conservative. Furthermore, only two valid design modes were not successfully identified, namely, $(m, \eta, \xi) = \{(16, 0.1\eta_{\max}, [0.46, 0.78]); (36, 0.1\eta_{\max}, [0.31, 0.49])\}$.

Interestingly, the smallest value in $\mathcal{X}_3$ results either from evaluating $J_{\mathrm{MSE}}(\infty)(m, \xi, \eta)$ (being thus the same as in $\mathcal{X}_2$), or from evaluating $n_\varepsilon(m, \xi, \eta)$. In the latter case it is the choice of $n_{\max}$ that dictates the smallest value in $\mathcal{X}_3$. The

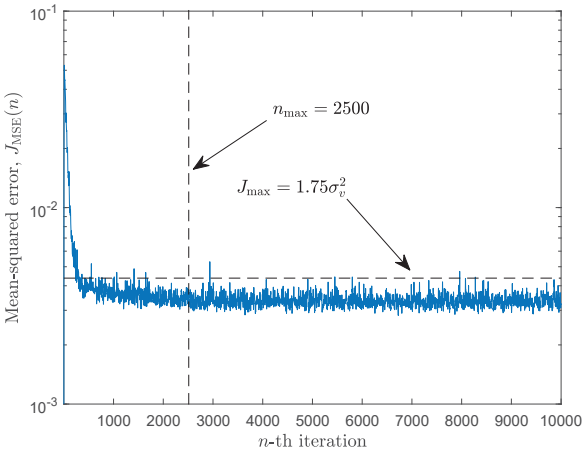

Fig. 5. Analytical and simulated learning curves.

largest value in $\mathcal{X}_3$ results typically from the convergence speed constraint.

Given the information in TABLE II, one may decide which set of parameters will be employed in the design. A typical design decision is to opt for the best possible estimation accuracy, given that the convergence time is within the acceptable range. We assume that convergence must be achieved before 2500 iterations. Hence, choosing for instance $m = 9$ for the dictionary, we may choose $\eta = 0.1\eta_{\max}$, which will lead to a lower residual MSE than $\eta = 0.25\eta_{\max}$, and $\xi = 0.92$ to be somewhere near the center of the available interval. Other design options could be made using the options in TABLE II. Fig. 5 shows the evolution of the MSE (Monte Carlo simulation with 1000 runs) for the chosen parameters, where one can readily verify that the design specifications have been met.

TABLE I
SPEED AND ADAPTATION CONSTRAINED DESIGN AFTER STEP 4 AND STEP 7
$\sigma_v^2 = 2.5 \times 10^{-3}$, $J_{\max} = 1.75\sigma_v^2$, $\mathcal{X}_0 = [0.1, 4]$

m	$\xi \in \mathcal{X}_1$	$\eta/\eta_{\max}$	$\xi \in \mathcal{X}_2$
4	$[1.30, 4]$	0.1	$[1.89, 4]$
9	$[0.60, 4]$	0.25	$[0.66, 4]$
		0.1	$[0.62, 4]$
16	$[0.39, 4]$	0.25	$[0.43, 4]$
		0.1	$[0.40, 4]$
25	$[0.29, 4]$	0.25	$[0.32, 4]$
		0.1	$[0.30, 4]$
36	$[0.23, 4]$	0.25	$[0.25, 4]$
		0.1	$[0.23, 4]$

V. CONCLUSIONS

In this paper we proposed, tested, and validated a novel methodology for the design of the Gaussian KLMS algorithm for specified performance limits. The proposed methodology is based on existing theoretical analytical models for the algorithm performance and on graphical representations of the existing nonlinear dependencies among parameters and performance measures. This methodology determines appropriate values for the design parameters aiming at specific project goals defined by a desired steady-state performance and a maximum convergence time. We have shown that the proposed methodology leads to the identification of appropriate sets of design parameters satisfying the established constraints. A detailed example illustrates the usefulness of the methodology for design purposes.

TABLE II
SPEED AND ADAPTATION CONSTRAINED DESIGN AFTER STEP 9
$\varepsilon = 0.1$, $\sigma_v^2 = 2.5 \times 10^{-3}$, $J_{\max} = 1.75\sigma_v^2$, $\mathcal{X}_0 = [0.1, 4]$

$n_{\max}$	m	$\eta/\eta_{\max}$	$\xi \in \mathcal{X}_3$	
			Analysis	Simulation
7500	4	0.1	$[1.89, 3.62]$	$[1.93, 3.86]$
	9	0.25	$[0.66, 1.60]$	$[0.66, 1.86]$
		0.1	$[0.70, 1.39]$	$[0.65, 1.64]$
	16	0.25	$[0.43, 0.99]$	$[0.43, 1.34]$
		0.1	$[0.40, 0.86]$	$[0.40, 1.09]$
	25	0.25	$[0.32, 0.88]$	$[0.32, 1.13]$
		0.1	$[0.30, 0.72]$	$[0.30, 0.95]$
	36	0.25	$[0.25, 0.77]$	$[0.25, 1.06]$
		0.1	$[0.23, 0.59]$	$[0.24, 0.87]$
5000	4	0.1	$[1.89, 3.26]$	$[1.93, 3.44]$
	9	0.25	$[0.66, 1.51]$	$[0.66, 1.69]$
		0.1	$[0.75, 1.30]$	$[0.69, 1.49]$
	16	0.25	$[0.43, 0.93]$	$[0.43, 1.10]$
		0.1	$[0.40, 0.78]$	$[0.40, 0.96]$
	25	0.25	$[0.32, 0.81]$	$[0.32, 0.99]$
		0.1	$[0.30, 0.66]$	$[0.30, 0.84]$
	36	0.25	$[0.25, 0.68]$	$[0.25, 0.92]$
		0.1	$[0.29, 0.52]$	$[0.24, 0.74]$
2500	4	0.1	$[1.89, 2.73]$	$[1.93, 2.85]$
	9	0.25	$[0.72, 1.35]$	$[0.66, 1.43]$
		0.1	$[0.80, 1.16]$	$[0.74, 1.25]$
	16	0.25	$[0.43, 0.83]$	$[0.43, 0.92]$
		0.1	$[-]$	$[0.46, 0.78]$
	25	0.25	$[0.32, 0.69]$	$[0.32, 0.80]$
		0.1	$[0.40, 0.51]$	$[0.30, 0.59]$
	36	0.25	$[0.25, 0.56]$	$[0.25, 0.69]$
		0.1	$[-]$	$[0.31, 0.49]$

REFERENCES

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

[2] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.

[3] W. Liu, P. P. Pokharel, and J. C. Príncipe, "The Kernel Least-Mean-Square Algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 1633–1642, Feb. 2008.

[4] W. 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.

[5] J. Chen, W. Gao, C. Richard, and J. C. M. Bermudez, "Convergence analysis of kernel LMS algorithm with pre-tuned dictionary," in *Proc. of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Florence, Italy: IEEE, 2014, pp. 7243–7247.

[6] W. Gao, J. Chen, and C. Richard, "Online Dictionary Learning for Kernel LMS," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2765–2777, Feb. 2014.

[7] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online Prediction of Time Series Data With Kernels," *IEEE Trans. on Signal Process.*, vol. 57, no. 3, pp. 1058–1067, 2009.