AFFINE PROJECTION ALGORITHM APPLIED TO NONLINEAR ADAPTIVE FILTERING

Cédric Richard$^{(1)}$ *and José Carlos M. Bermudez*$^{(2)}$

$^{(1)}$Institut Charles Delaunay (ICD, FRE CNRS 2848), Université de technologie de Troyes, BP 2060, 10010 Troyes cedex, France
$^{(2)}$Department of Electrical Engineering, Federal University of Santa Catarina 88040-900, Florianópolis, SC - Brazil *

ABSTRACT

In this paper, we present a framework for nonlinear adaptive filtering. It employs the formalism of reproducing kernel Hilbert spaces to incorporate nonlinearity into the classical affine projection algorithm. A nonlinear normalized LMS (NLMS) algorithm with kernels is also derived as a particular case. We propose a sparsification strategy that employs a coherence parameter to control the model order increase typical of kernel-based methods. The resulting algorithms are suitable for real-time applications. Experimental results validate our approach.

Index terms – Adaptive filters, nonlinear systems

1. INTRODUCTION

Adaptive filtering has become a topic of keen interest over the past three decades to help cope with time variations of system parameters and lack of *a priori* statistical information [1, 2]. Linear models are still routinely used because of their inherent simplicity from conceptual and implementational point of view. In many practical situations, however, nonlinear signal processing is needed. Examples are nonlinear system identification, prediction and control. Following the pioneering works [3, 4], there has been recent progress in function approximation methods based on reproducing kernel Hilbert spaces (RKHS), which include support vector regression. A common characteristic of classical kernel-based methods is that they employ models whose order equals the size of the training set. Such methods are then unsuitable for most real time applications. Several algorithms have been proposed to circumvent this computational burden in time series prediction problems [5, 6]. Nevertheless, they usually require excessively elaborate and costly operations such as matrix inversion.

This paper develops a new kernel-based nonlinear adaptive algorithm that leads to reduced model orders. The algorithm follows the affine projection (AP) approach and is derived using the RKHS formalism. The NLMS algorithm follows as a particular case. The increase in the model order is controlled by the coherence parameter, a fundamental quantity usually employed to characterize the behavior of dictionaries of functions in sparse approximation problems [7]. The paper is organized as follows. We first introduce some basic principles of kernel-based optimal filtering in RKHS. Next we present our nonlinear adaptive filtering method. Finally, we present simulation examples that illustrate the performances of both the AP and NLMS nonlinear algorithms.

2. BASIC PRINCIPLES OF NONLINEAR OPTIMAL FILTERING IN RKHS

Let $\mathcal{H}$ be a RKHS of real-valued functions ψ on a compact $\mathcal{U} \subset \mathbb{R}^p$, and let $\langle \cdot\,, \cdot \rangle_{\mathcal{H}}$ be the inner product in this space. Let $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ denote the reproducing kernel of $\mathcal{H}$, which means that $\psi(\boldsymbol{u}_i) = \langle \psi(\cdot), \kappa(\cdot, \boldsymbol{u}_i) \rangle_{\mathcal{H}}$ for all $\psi(\cdot) \in \mathcal{H}$ and every fixed $\boldsymbol{u}_i \in \mathcal{U}$. The problem is to determine a function $\psi(\cdot)$ of $\mathcal{H}$ that minimizes the sum of squared errors between n samples d_i of the desired response and the corresponding model output samples $\psi(\boldsymbol{u}_i)$, namely,

$$\min_{\psi \in \mathcal{H}} \sum_{i=1}^{n} |d_i - \psi(\boldsymbol{u}_i)|^2. \tag{1}$$

By virtue of the representer theorem [4], the solution to this problem can be expressed as a kernel expansion in terms of available data $\boldsymbol{u}_j$, $j = 1, \ldots, n$, that is,

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

Substituting (2) into (1), elementary algebra shows that problem (1) reduces to

$$\min_{\boldsymbol{\alpha}} \; \|\boldsymbol{d} - \boldsymbol{K}\boldsymbol{\alpha}\|^2 \tag{3}$$

where $\boldsymbol{K}$ denotes the n-by-n Gram matrix whose (i, j)-th entry is $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j)$. Assuming $\boldsymbol{K}^{-1}$ exists, the solution to (3) is $\boldsymbol{\alpha} = \boldsymbol{K}^{-1}\boldsymbol{d}$. Adaptive filtering raises the question of how to process an increasing amount of observations as new data is collected. Clearly the optimal approach outlined above cannot be used, since the model order and the dimension of $\boldsymbol{K}$ increase with n. Attempts have recently been made to circumvent this drawback [5, 6]. Consider the m-th order model at any given time instant n

$$\psi_n(\cdot) = \sum_{j=1}^{m} \alpha_j \, \kappa(\cdot, \boldsymbol{u}_{\omega_j}), \tag{4}$$

where the ω_j's form an m-element subset $\mathcal{J}_n$ of $\{1, \ldots, n\}$. We call $\{\kappa(\cdot, \boldsymbol{u}_{\omega_j})\}_{j=1}^{m}$ the dictionary. These approaches rely on a two-stage process at each iteration: a model order selection step, and a parameter update step. In the first step, the kernel function $\kappa(\cdot, \boldsymbol{u}_n)$ is inserted into the dictionary at time instant n if it cannot be reasonably well represented by a combination of the other kernel functions of the dictionary. This condition usually takes the form

$$\min_{\boldsymbol{\gamma}} \|\kappa(\cdot, \boldsymbol{u}_n) \;-\; \sum_{\omega_j \in \mathcal{J}_{n-1}} \gamma_j \, \kappa(\cdot, \boldsymbol{u}_{\omega_j})\|^2_{\mathcal{H}} > \eta_0, \tag{5}$$

where κ is a unit-norm kernel[1]. The threshold η_0 determines the level of sparsity of the model. These approaches, while accurate, are computationally prohibitive.

*This work has been supported in part by CNPq under grant 308095/2003-0

[1] This means that $\kappa(\boldsymbol{u}_k, \boldsymbol{u}_k) = 1$ for every $\boldsymbol{u}_k \in \mathcal{U}$. Otherwise, substitute $\kappa(\cdot, \boldsymbol{u}_k)/\sqrt{\kappa(\boldsymbol{u}_k, \boldsymbol{u}_k)}$ for $\kappa(\cdot, \boldsymbol{u}_k)$ in (5).

3. MODEL REDUCTION WITH COHERENCE

The coherence is a fundamental parameter to characterize a dictionary in sparse approximation problems [7]. In our kernel-based context, we define the coherence parameter ν as

$$\nu = \max_{i \neq j} |\langle \kappa(\cdot, \boldsymbol{u}_{\omega_i}), \kappa(\cdot, \boldsymbol{u}_{\omega_j}) \rangle_{\mathcal{H}}| = \max_{i \neq j} |\kappa(\boldsymbol{u}_{\omega_i}, \boldsymbol{u}_{\omega_j})|$$

where κ is a unit-norm kernel. It reflects the most extreme correlations in the dictionary, and is equal to zero for every orthonormal basis. Rather than solving a problem of the form (5), we suggest inserting $\kappa(\cdot, \boldsymbol{u}_n)$ into the dictionary if the coherence of the increased dictionary remains below a given threshold ν_0, namely,

$$\max_{\omega_j \in \mathcal{J}_{n-1}} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \nu_0, \tag{6}$$

where $\nu_0 \in [0, 1[$ determines both the sparsity level and the coherence of the dictionary. The motivation for using this test is twofold. First, it is easy to calculate and its computational complexity is only linear in the dictionary size. Second, it offers several attractive properties that can be exploited to assess novelty of input kernel functions. In particular, we have proved in [8] that:

- If $\mathcal{U}$ is compact, the dictionary of kernel functions determined under (6) is finite. Thus, the order of the asymptotic model $\psi_\infty(\cdot)$ is finite and depends on ν_0.
- If $(m-1) < 1/\nu_0$, the elements of the dictionary $\{\kappa(\cdot, \boldsymbol{u}_{\omega_j})\}_{j=1}^m$ are linearly independent.
- If $(m-1) < 1/2\nu_0$, the left-hand side of (5) is lower bounded by $1 - \frac{(m-1)\nu_0^2}{(1-(m-1)\nu_0)}$, which establishes a connection between η_0 and ν_0.

We now describe the parameter update step, whose purpose is to solve problem (3) recursively.

4. A KERNEL-BASED AFFINE PROJECTION ALGORITHM WITH ORDER-UPDATE MECHANISM

We now describe the procedure to use (6) in a sparsification strategy to control the order of the model. Let $\psi_n(\cdot)$ denote the model at time instant n,

$$\psi_n(\cdot) = \sum_{j=1}^{m} \alpha_{n,j}\, \kappa(\cdot, \boldsymbol{u}_{\omega_j}), \tag{7}$$

supposed to be of order $m \leq n$, where the $\kappa(\cdot, \boldsymbol{u}_{\omega_j})$'s form a ν_0-coherent dictionary determined under (6). In accordance with problem (3), the optimal parameter vector $\boldsymbol{\alpha}_n$ solves

$$\min_{\boldsymbol{\alpha}} \; \|\boldsymbol{d}_n - \boldsymbol{H}_n \boldsymbol{\alpha}\|^2, \tag{8}$$

where $\boldsymbol{H}_n$ denotes the n-by-m matrix whose (i,j)-th entry is $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_{\omega_j})$. Assuming that $(\boldsymbol{H}_n^t \boldsymbol{H}_n)^{-1}$ exists,

$$\boldsymbol{\alpha}_n = (\boldsymbol{H}_n^t \boldsymbol{H}_n)^{-1} \boldsymbol{H}_n^t \boldsymbol{d}_n. \tag{9}$$

In [8], we developed a recursive algorithm to compute the updated estimate $\boldsymbol{\alpha}_n$ from the least-squares solution $\boldsymbol{\alpha}_{n-1}$ upon the arrival of $\boldsymbol{u}_n$.

4.1. The Affine Projection Algorithm

The RLS approach of [8] uses two distinct recursion formulas, depending on whether the coherence-based rule (6) is satisfied or not. We now introduce a simpler stochastic-gradient method to solve (8), where the required gradient vectors are replaced with suitable approximations. At each iteration n, only the p most recent inputs $\{\boldsymbol{u}_n, \ldots, \boldsymbol{u}_{n-p+1}\}$ and observations $\{d_n, \ldots, d_{n-p+1}\}$ are used. The new method trades convergence speed for a reduced computational complexity.

From now, $\boldsymbol{H}_n$ denotes the p-by-m matrix whose (i,j)-th entry is $\kappa(\boldsymbol{u}_{n-i+1}, \boldsymbol{u}_{\omega_j})$, and $\boldsymbol{d}_n$ is the p-by-1 vector whose i-th element is d_{n-i+1}. Our approach starts with the affine projection problem [2]

$$\min_{\boldsymbol{\alpha}_n} \; \|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_{n-1}\|^2 \quad \text{subject to} \quad \boldsymbol{d}_n = \boldsymbol{H}_n \boldsymbol{\alpha}_n \tag{10}$$

with the *a priori* error defined as $\boldsymbol{e}_{a,n} = \boldsymbol{d}_n - \boldsymbol{H}_n \boldsymbol{\alpha}_{n-1}$. In other words, $\boldsymbol{\alpha}_n$ is obtained by projecting $\boldsymbol{\alpha}_{n-1}$ onto the intersection of the p affine subspaces $\mathcal{A}_i$ defined as

$$\mathcal{A}_i = \{\boldsymbol{\alpha}_n \in \mathrm{I\!R}^m : \; \boldsymbol{h}_{n-i+1}^t \boldsymbol{\alpha}_n - d_{n-i+1} = 0\}, \; i = 1, \ldots, p$$

with $\boldsymbol{h}_{n-i+1} = [\kappa(\boldsymbol{u}_{n-i+1}, \boldsymbol{u}_{\omega_1}) \ldots \kappa(\boldsymbol{u}_{n-i+1}, \boldsymbol{u}_{\omega_m})]^t$. At time instant n, upon the arrival of new data, one of the following alternatives holds. If $\kappa(\cdot, \boldsymbol{u}_n)$ satisfies the coherence-based rule (6), it is inserted into the dictionary. In this case, the number of columns of $\boldsymbol{H}_n$ is increased by one, relative to $\boldsymbol{H}_{n-1}$, and one more entry is added to the vector $\boldsymbol{\alpha}_n$. The model order m is then increased by 1. The new column of $\boldsymbol{H}_n$ will be $[\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_{m+1}}) \ldots \kappa(\boldsymbol{u}_{n-p+1}, \boldsymbol{u}_{\omega_{m+1}})]^t$. If $\kappa(\cdot, \boldsymbol{u}_n)$ does not satisfy (6), dictionary remains unchanged.

4.2. First case study: $\max_{j=1,\ldots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| > \nu_0$

In this case $\kappa(\cdot, \boldsymbol{u}_n)$ does not need to be inserted into the dictionary, as the former can be reasonably well represented by the kernel functions already in the latter. Solution of (10) can be determined by minimizing the Lagrangian function

$$J(\boldsymbol{\alpha}_n, \boldsymbol{\lambda}) = \|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_{n-1}\|^2 + \boldsymbol{\lambda}^t(\boldsymbol{d}_n - \boldsymbol{H}_n \boldsymbol{\alpha}_n) \tag{11}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Differentiating this expression with respect to $\boldsymbol{\alpha}_n$ and $\boldsymbol{\lambda}$, and setting the derivatives to zero, yields

$$2\,(\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_{n-1}) = \boldsymbol{H}_n^t\, \boldsymbol{\lambda} \tag{12}$$

$$\boldsymbol{H}_n \boldsymbol{\alpha}_n = \boldsymbol{d}_n \tag{13}$$

Assuming $\boldsymbol{H}_n \boldsymbol{H}_n^t$ nonsingular, these equations lead to $\boldsymbol{\lambda} = 2\,(\boldsymbol{H}_n \boldsymbol{H}_n^t)^{-1}(\boldsymbol{d}_n - \boldsymbol{H}_n \boldsymbol{\alpha}_{n-1})$. Substituting into (12), we obtain a recursive update equation for $\boldsymbol{\alpha}_n$:

$$\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_{n-1} + \mu\, \boldsymbol{H}_n^t (\epsilon \mathbf{I} + \boldsymbol{H}_n \boldsymbol{H}_n^t)^{-1} (\boldsymbol{d}_n - \boldsymbol{H}_n \boldsymbol{\alpha}_{n-1}), \tag{14}$$

where we have introduced the step-size control parameter μ, and the regularization factor $\epsilon\mathbf{I}$ [2]. At each iteration n, (14) requires inverting the usually small p-by-p matrix $(\epsilon\mathbf{I} + \boldsymbol{H}_n \boldsymbol{H}_n^t)$. Note that the i-th element of the *a priori* error $\boldsymbol{e}_{a,n}$ can also be expressed as $d_{n-i+1} - \psi_{n-1}(\boldsymbol{u}_{n-i+1})$.

4.3. Second case study: $\max_{j=1,\ldots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \nu_0$

In this case $\kappa(\cdot, \boldsymbol{u}_n)$ cannot be represented by the kernel functions in the dictionary. Then, it is inserted into the dictionary and will henceforth be denoted by $\kappa(\cdot, \boldsymbol{u}_{\omega_{m+1}})$. The order m of (7) is increased by one, and $\boldsymbol{H}_n$ is updated to a p-by-$(m+1)$ matrix[2]. To accommodate the new element in $\boldsymbol{\alpha}_n$, we modify (10) as follows

$$\begin{aligned}\min_{\boldsymbol{\alpha}_n} \; & \|\boldsymbol{\alpha}_{n,m} - \boldsymbol{\alpha}_{n-1}\|^2 + \alpha_{n,m+1}^2 \\ \text{subject to} \quad & \boldsymbol{d}_n = \boldsymbol{H}_n \boldsymbol{\alpha}_n,\end{aligned} \tag{15}$$

where $\boldsymbol{\alpha}_{n,m}$ denotes the first m elements of the vector $\boldsymbol{\alpha}_n$. Note that the $(m+1)$-th element, $\alpha_{n,m+1}$, is incorporated as a regularizing term of the objective function. Similar arguments as above lead us to the following recursion

$$\begin{aligned}\boldsymbol{\alpha}_n = & \begin{bmatrix}\boldsymbol{\alpha}_{n-1}\\ 0\end{bmatrix} + \\ & \mu\, \boldsymbol{H}_n^t(\epsilon \mathbf{I} + \boldsymbol{H}_n\boldsymbol{H}_n^t)^{-1}\left(\boldsymbol{d}_n - \boldsymbol{H}_n \begin{bmatrix}\boldsymbol{\alpha}_{n-1}\\ 0\end{bmatrix}\right).\end{aligned} \tag{16}$$

We call the set of recursions (14) and (16) the kernel-based AP or KAPA(μ, ϵ) algorithm. The value of p is termed the order of the algorithm. Next, we explore the idea of using instantaneous approximations for the gradient vectors.

4.4. Instantaneous approximations

Now consider the case $p = 1$. At each iteration n, the algorithm described above then enforces the equality $d_n = \boldsymbol{h}_n^t \boldsymbol{\alpha}_n$ with $\boldsymbol{h}_n = [\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_1}) \ldots \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_m})]^t$. Relations (14) and (16) reduce to
1. if $\max_{j=1,\ldots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| > \nu_0$

$$\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_{n-1} + \frac{\mu}{\epsilon + \|\boldsymbol{h}_n\|^2}\,(\boldsymbol{d}_n - \boldsymbol{h}_n^t \boldsymbol{\alpha}_{n-1})\,\boldsymbol{h}_n, \tag{17}$$

2. if $\max_{j=1,\ldots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{\omega_j})| \leq \nu_0$

$$\boldsymbol{\alpha}_n = \begin{bmatrix}\boldsymbol{\alpha}_{n-1}\\ 0\end{bmatrix} + \frac{\mu}{\epsilon + \|\boldsymbol{h}_n\|^2}\left(\boldsymbol{d}_n - \boldsymbol{h}_n^t \begin{bmatrix}\boldsymbol{\alpha}_{n-1}\\ 0\end{bmatrix}\right)\boldsymbol{h}_n. \tag{18}$$

The form of these recursions is that of the normalized LMS algorithm with kernels, referred to as KNLMS(μ, ϵ) to highlight the presence of μ and ϵ.

5. SIMULATION EXAMPLES

The purpose of this section is to demonstrate the validity of the proposed approach. We compare the performances of the KAPA and KNLMS algorithms with the results obtained in [8] for our kernel-based KRLS algorithm.

5.1. Experiment with the Laplace kernel

As an application of the proposed approach, we consider the discrete-time nonlinear dynamical system

$$\begin{cases} v_n = 1.1\, \exp(-|v_{n-1}|) + u_n \\ d_n = v_n^2 \end{cases} \tag{19}$$

where u_n and d_n are the input and the desired output, respectively. The data were generated from the initial condition $v_0 = 0.5$. The input u_n was sampled from a zero-mean Gaussian distribution with standard deviation 0.25. The system output d_n was corrupted by an additive zero-mean white Gaussian noise with standard deviation equal to 1. This resulted in a signal-to-noise ratio, defined as the ratio of the powers of d_n and the additive noise, of -4.0 dB. Our approach was used to estimate d_n using a model of the form $d_n = \hat{\psi}(\boldsymbol{u}_n)$. The Laplace kernel $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|/\beta_0)$ with $\beta_0 = 0.35$ was used for being the most accurate among several options. The coherence threshold ν_0 was set to 0.3 and the dictionary was initialized with $\kappa(\cdot, \boldsymbol{u}_1)$. Preliminary experiments were conducted for the KAPA algorithm on sequences of 5000 samples to determine the order p, the step size parameter μ and the regularization constant ϵ. Performance was measured using the mean-square estimation error over the last 500 samples of each sequence, and averaged over 50 trials. KAPA orders ranging from 1 to 5 were tested, and $p = 5$ was chosen. Parameters μ and ϵ were set to 9×10^{-4} and 7×10^{-3}. These values were found by performing a simple grid search over the parameter space $(10^{-4} \leq \mu \leq 10^{-1}) \times (10^{-3} \leq \epsilon \leq 10^{-2})$ with (logarithmic) increment 2×10^{-k} within ranges $[10^{-k}, 10^{-k+1}]$. KAPA was tested on fifty 5000-sample test sequences. The average order m of the kernel expansion (7) was 5.4, which confirms the control of the proposed technique over the model order. Steady-state performance of KAPA was measured by the mean-square prediction error over the last 500 samples of each time series, that is,

$$\xi = \frac{1}{500}\sum_{n=4501}^{5000} (d_n - \hat{\psi}_{n-1}(\boldsymbol{u}_n))^2. \tag{20}$$

The average value of $\xi_{\text{KAPA}} = 0.0839$ was obtained for KAPA over the 50 test sequences. KRLS algorithm led to $\xi_{\text{KRLS}} = 0.0711$, but at the cost of a much higher computational complexity. For the KNLMS algorithm, the best steady-state performance was obtained with $\mu = 5 \times 10^{-3}$ and $\epsilon = 9 \times 10^{-3}$. As for KAPA, the average value of the model order m was 5.4. The average value of (20) over the fifty test sequences was $\xi_{\text{KNLMS}} = 0.0896$. As expected, the substantial computational savings gained by using KNLMS came at the cost of a worse steady-state performance.

For transient behavior comparison, KAPA was designed to match the KNLMS steady-state performance. This resulted in $p = 3$, $\mu = 3 \times 10^{-3}$ and $\epsilon = 5 \times 10^{-2}$. Figure 1(a) shows the evolution of the mean-square error for the three algorithms, where the convergence speed of KAPA is seen to be between KNLMS and KRLS, as expected.

The experiment reported in Figure 1(a) was repeated for u_n a colored Gaussian process. Signal u_n was obtained by passing the same white Gaussian noise used before through a linear filter with impulse response given by $\boldsymbol{h} = [0.9045\ 1\ 0.9045\ 0]^t$. The order m of the kernel expansion was, on average, 8.64 for the KAPA and the KNLMS. Figure 1(b) show the mean-square error evolution for the three algorithms, and corroborates the results verified in Figure 1(a) for white inputs.

5.2. Experiment with the Gaussian kernel

As a second benchmark problem, we consider the nonlinear system described by the difference equation

$$\begin{aligned} d_n = & \left(0.8 - 0.5\, \exp(-d_{n-1}^2)\right) d_{n-1} \\ & - \left(0.3 + 0.9\, \exp(-d_{n-1}^2)\right) d_{n-2} + 0.1\, \sin(d_{n-1}\pi) \end{aligned} \tag{21}$$

where d_n is the desired output. This highly nonlinear time series has been investigated in [5]. The data were generated by iterating the above equation from the initial condition $(0.1, 0.1)$. Outputs d_n were corrupted by a measurement noise sampled from a zero-mean Gaussian distribution with standard deviation equal to 0.1, corresponding to a signal-to-noise ratio of 17.2 dB. These data were used

[2] Note that any dictionary determined under rule (6) is finite, see [8].

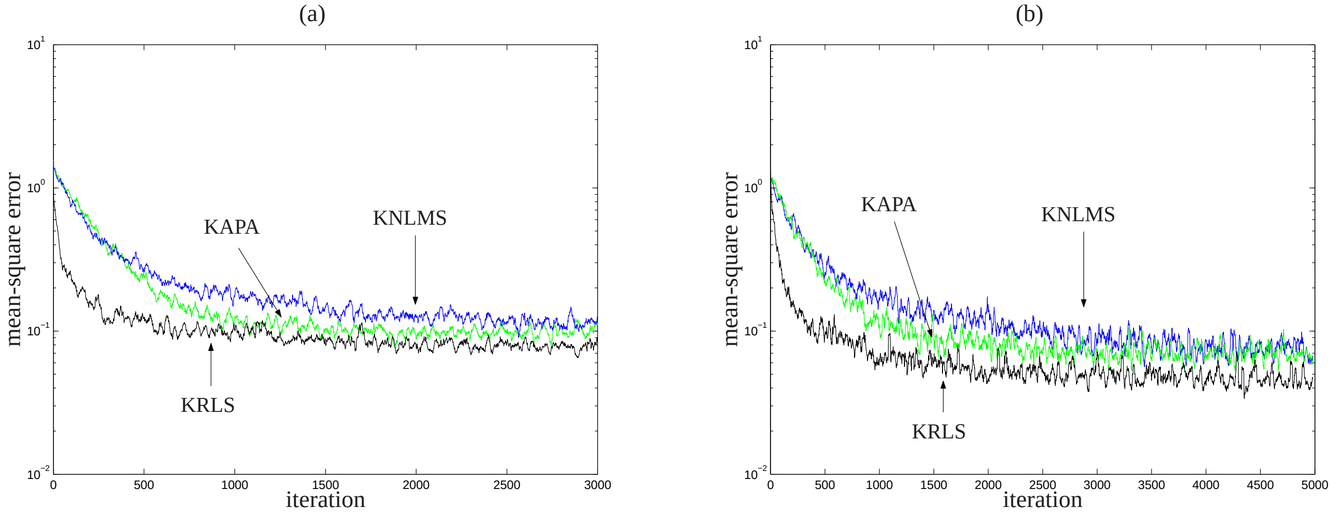

Fig. 1. Convergence behavior of KAPA, KNLMS and KRLS for (a) white and (b) colored inputs. The error curves were smoothed by time averaging over 20 consecutive samples.

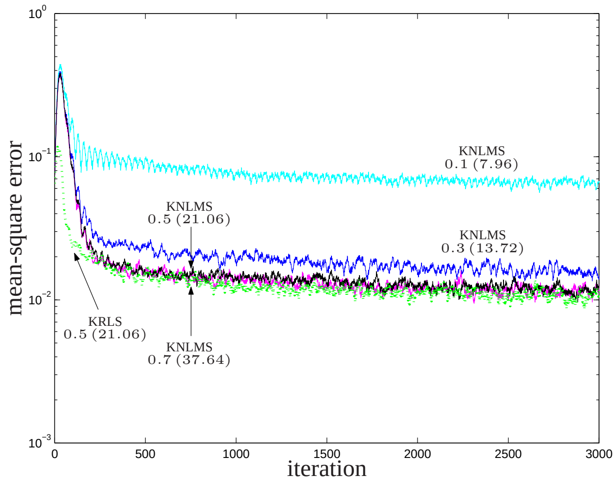

Fig. 2. Convergence behavior of KNLMS and KRLS algorithms for several values of ν_0. Value of ν_0 is outside the parentheses. The mean order of each model is given in parentheses. The error curves were smoothed by time averaging over 20 consecutive samples.

to estimate a nonlinear model of the form $d_n = \psi(d_{n-1}, d_{n-2})$. In identifying the system, we restricted ourselves to KNLMS and the experimental setup described in [5]. In particular, the Gaussian kernel $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp\big(-3.73 \, \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2\big)$ was considered. As in the previous experiment, trial-and-error on fifty 5000-sample sequences was used to determine the threshold ν_0, and parameters μ_0 and ϵ. The dictionary was initialized with $\kappa(\cdot, \boldsymbol{u}_1)$. A practical compromise between the model order and its *a priori* estimation error was reached by setting ν_0 to 0.5. The step-size parameter μ_0 and the regularization coefficient ϵ were fixed to 9×10^{-2}. Figure 2 shows that KNLMS has almost the same steady-state performance as KRLS, but converges slower. However, the computational complexity of KRLS is quadratic in m whereas that of KNLMS is linear. Error curves obtained for KNLMS using other values of ν_0 are also plotted in this figure. No significant performance improvement is observed for $\nu_0 > 0.5$.

6. CONCLUSION

We presented a framework for nonlinear adaptive filtering. Our approach is based on the classical affine projection algorithm, and uses reproducing kernels to incorporate nonlinearity. As a particular case, we also derived a nonlinear normalized LMS algorithm with kernels. A common characteristic in kernel-based methods is that they deal with models whose size increases with the number of input data. We used a sparsification procedure based on the coherence parameter to circumvent this computational burden. Finally, we presented experimental results to validate our approach.

7. REFERENCES

[1] S. Haykin, *Adaptive filter theory*, Prentice Hall, Upper Saddle River, NJ, fourth edition, 2002.

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

[3] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American Mathematical Society*, vol. 68, pp. 337–404, 1950.

[4] G. Kimeldorf and G. Wahba, “Some results on Tchebycheffian spline functions,” *Journal of Mathematical Analysis and Applications*, vol. 33, pp. 82–95, 1971.

[5] T. J. Dodd, V. Kadirkamanathan, and R. F. Harrison, “Function estimation in Hilbert space using sequential projections,” in *Proc. IFAC Conference on Intelligent Control Systems and Signal Processing*, 2003, pp. 113–118.

[6] Y. Engel, S. Mannor, and R. Meir, “Kernel recursive least squares,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[7] J.A. Tropp, “Greed is good: algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[8] C. Richard and J. C. M. Bermudez, “Nonlinear RLS filtering with Mercer kernels and a coherence-based criterion,” *submitted to IEEE Transactions on Signal Processing*, 2006.