

CONVEX COMBINATIONS OF KERNEL ADAPTIVE FILTERS

Wei Gao^{†‡} Cédric Richard[†] Jose-Carlos M. Bermudez^{*} Jianguo Huang[‡]

[†]Université de Nice Sophia-Antipolis, CNRS, Observatoire de la Côte d’Azur, France

[‡]Northwestern Polytechnical University, Xi’an, China

^{*}Federal University of Santa Catarina, Florianópolis, SC, Brazil

gao_wei@mail.nwpu.edu.cn cedric.richard@unice.fr j.bermudez@ieee.org jghuang@nwpu.edu.cn

ABSTRACT

Multikernel adaptive filtering has recently attracted significant research interest due to its enhanced flexibility and adaptation performance over single-kernel methods. In this paper, we focus on convex combinations of two single-kernel adaptive filters, characterized by different convergence speeds and steady-state performances, in order to get the best of both. We consider online estimation using single-kernel adaptive filters that may use different algorithms and kernels. Simulation results illustrate the efficiency of our approach.

Index Terms— Kernel adaptive filtering, convex combination, multikernel method, tracking

1. INTRODUCTION

Single-kernel adaptive filtering is now perceived as an appealing solution for nonlinear system identification and tracking [1–4]. Developing adaptive filters in reproducing kernel Hilbert spaces (RKHS) allows the use of linear structures to solve nonlinear estimation problems. The kernel recursive least-squares (KRLS) algorithm was introduced in [1]. The sliding-window KRLS and the extended KRLS algorithms were derived in [5, 6]. The kernel affine projection algorithm (KAPA) and, as a particular case, the kernel normalized LMS algorithm (KNLMS), were independently introduced in [2, 3, 7, 8]. The kernel least-mean-square algorithm (KLMS), proposed in [9, 10], has attracted much attention in recent years because of its simplicity and robustness. Along the same line, the quantized KLMS algorithm (QKLMS) was recently proposed in [11]. Although many kernel-based algorithms have been recently proposed, few theoretical studies have investigated their convergence behavior. This situation is partly due to technical difficulties stemming from filter nonlinearity with respect to input samples. An analysis of the stochastic behavior of the KLMS algorithm with Gaussian kernel was proposed in [12], and a closed-form condition for convergence was introduced in [13]. The stability of this algorithm with ℓ_1 -norm regularization was studied in [14, 15]. Finally, in [16], a theoretical analysis of the KLMS algorithm with Gaussian kernel was proposed in the case where the dictionary used by the algorithm is considered as part of the filter parameters.

Multikernel models have been proposed in order to enhance the model flexibility and the adaptation performance. The multikernel least-mean-square algorithm (MKLMS) was independently proposed in [17–19] and [20, 21], with distinct functional frameworks and implementations. The former uses block ℓ_1 -norm regularization

to promote sparsity of multikernel models and discards obsolete kernel functions in real time. The latter introduces an original vector-valued functional framework, which allows to address multiple-input multiple-output (MIMO) nonlinear systems. A multikernel normalized LMS algorithm operating in a direct sum of reproducing kernel Hilbert spaces (RKHS), with independently tuned dictionaries, is considered in [22]. In [23], the authors introduce the mixture KLMS algorithm. This approach uses a gate model to adapt the combination weights of single-kernel LMS algorithms. The individual KLMS algorithms in the existing multikernel approaches use the same error signal to adjust the coefficients of all the filters, which is a significant difference compared with our work.

Convex and affine combinations of two linear LMS with distinct step-sizes have been extensively studied in the literature [24–28]. Their performance is well-established now, and analyses have shown that combining two filters potentially allows to obtain the best transient and steady-state performance of both. In the spirit of these works, we focus on convex combinations of two single-kernel adaptive filters with distinct settings including kernel definition and parameterization, step-size, and sparsification criterion. We also consider the case of two distinct algorithms to estimate filter coefficients. Our work differs from [23] in that we use the vector-valued functional framework introduced in [21], and, to eliminate potential side effects due to coupling, the individual KLMS algorithms do not use the same error signal to adapt. This paper is organized as follows. First, we introduce an appropriate framework to derive multikernel LMS. Then, we present the convex combination strategy. Finally, we illustrate our approach with simulation examples.

2. MULTIKERNEL LMS

We shall now introduce the so-called multikernel LMS (MKLMS) algorithm. It differs from existing solutions by the filter structure. See, e.g., [17, 21, 23] for an overview of existing techniques.

2.1. A preliminary framework

Let \mathcal{H} denote a Hilbert space of real-valued functions $\psi(\cdot)$ on a compact subspace $\mathcal{U} \subset \mathbb{R}^L$, and let $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ be the inner product in the space \mathcal{H} . Suppose that the functional evaluation $\mathcal{L}_{\mathbf{u}}$ defined by $\mathcal{L}_{\mathbf{u}}[\psi] = \psi(\mathbf{u})$ is linear with respect to function $\psi(\cdot)$ and bounded, for all $\mathbf{u} \in \mathcal{U}$. By virtue of Riesz representation theorem, it exists a unique positive definite function $\mathbf{u} \mapsto \kappa(\mathbf{u}, \mathbf{u}')$ in \mathcal{H} , denoted by $\kappa(\cdot, \mathbf{u}')$ and called representer of evaluation, which satisfies [29]

$$\psi(\mathbf{u}') = \langle \psi(\cdot), \kappa(\cdot, \mathbf{u}') \rangle_{\mathcal{H}}, \quad \forall \psi \in \mathcal{H} \quad (1)$$

This work was partially supported by the National Natural Science Foundation of China (61271415).

for every fixed $\mathbf{u}' \in \mathcal{U}$. A proof of this can be found in [30]. Replacing $\psi(\cdot)$ by $\kappa(\cdot, \mathbf{u})$ in (1) yields

$$\kappa(\mathbf{u}, \mathbf{u}') = \langle \kappa(\cdot, \mathbf{u}), \kappa(\cdot, \mathbf{u}') \rangle_{\mathcal{H}} \quad (2)$$

for all $\mathbf{u}, \mathbf{u}' \in \mathcal{U}$. Equation (2) is the origin of the now generic term reproducing kernel to refer to $\kappa(\cdot, \cdot)$, and \mathcal{H} is the RKHS associated to this kernel. Denoting by φ the map that assigns the kernel function $\kappa(\cdot, \mathbf{u}')$ to each input data \mathbf{u}' , property (2) immediately implies that $\kappa(\mathbf{u}, \mathbf{u}') = \langle \varphi(\mathbf{u}), \varphi(\mathbf{u}') \rangle_{\mathcal{H}}$. The kernel thus evaluates the inner product of any pair of elements of \mathcal{U} mapped to \mathcal{H} without any explicit knowledge of $\varphi(\cdot)$ and \mathcal{H} . This key idea is known as the kernel trick. Many kernel functions have been proposed in the literature. In this paper, without any loss of generality, we focus on the Gaussian kernel $\kappa(\mathbf{u}, \mathbf{u}') = \exp(-\|\mathbf{u} - \mathbf{u}'\|_2^2 / 2\xi^2)$ with $\xi > 0$ the kernel bandwidth.

2.2. Problem statement

Let $\{\kappa_k\}_{k=1}^K$ be the family of candidate kernels, and \mathcal{H}_k the RKHS defined by the kernel function κ_k . Consider the multidimensional mapping

$$\begin{aligned} \Phi: \mathcal{U} &\rightarrow \mathcal{M} \\ \mathbf{u} &\mapsto \Phi(\mathbf{u}) = [\varphi_1(\mathbf{u}), \dots, \varphi_K(\mathbf{u})]^\top \end{aligned} \quad (3)$$

with $\varphi_k \in \mathcal{H}_k$, and let $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ be the inner product in \mathcal{M} defined as

$$\langle \Phi, \Phi' \rangle_{\mathcal{M}} = \sum_{k=1}^K \langle \varphi_k, \varphi'_k \rangle_{\mathcal{H}_k}. \quad (4)$$

The space \mathcal{M} of vector-valued functions equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ is a Hilbert space as $(\mathcal{H}_k, \langle \cdot, \cdot \rangle_{\mathcal{H}_k})$ is a Hilbert space for all k . We can then define the vector-valued representer of evaluation $\kappa_{\mathcal{M}}(\cdot, \mathbf{u})$ such that

$$\begin{aligned} \Phi(\mathbf{u}) &= [\Phi, \kappa_{\mathcal{M}}(\cdot, \mathbf{u})] \\ &= [\langle \varphi_1, \kappa_1(\cdot, \mathbf{u}) \rangle_{\mathcal{H}_1}, \dots, \langle \varphi_K, \kappa_K(\cdot, \mathbf{u}) \rangle_{\mathcal{H}_K}]^\top \end{aligned} \quad (5)$$

with $\kappa_{\mathcal{M}}(\cdot, \mathbf{u}) = [\kappa_1(\cdot, \mathbf{u}), \dots, \kappa_K(\cdot, \mathbf{u})]^\top$ and $[\cdot, \cdot]$ the entrywise inner product. This yields the following reproducing property

$$\kappa_{\mathcal{M}}(\mathbf{u}, \mathbf{u}') = [\kappa_{\mathcal{M}}(\cdot, \mathbf{u}), \kappa_{\mathcal{M}}(\cdot, \mathbf{u}')]. \quad (6)$$

Let $\Psi = [\psi_1, \dots, \psi_K]^\top$ be a vector-valued function in space \mathcal{M} , and let $\psi = \sum_{k=1}^K \psi_k$ be the scalar-valued function that sums the entries of Ψ , namely, $\psi = \mathbf{1}_K^\top \Psi$.

Given a set $\{(\mathbf{u}(n), d(n))\}_{n=1}^N$ of pairs of input vectors and desired output scalars, we aim at estimating a multidimensional function Ψ in \mathcal{M} that minimizes the regularized mean-square error

$$\begin{aligned} \min_{\Psi \in \mathcal{M}} J(\Psi) &= \frac{1}{K} \sum_{n=1}^N \sum_{k=1}^K [d(n) - \psi_k(\mathbf{u}(n))]^2 + \lambda \sum_{k=1}^K \|\psi_k\|_{\mathcal{H}_k}^2 \\ &= \frac{1}{K} \sum_{n=1}^N \|\mathbf{d}(n) - \Psi(\mathbf{u}(n))\|_2^2 + \lambda \|\Psi\|_{\mathcal{M}}^2 \end{aligned} \quad (7)$$

with $\psi_k \in \mathcal{H}_k$ and $\mathbf{d}(n) = d(n) \mathbf{1}_K \in \mathbb{R}^K$, and λ a nonnegative regularization constant. Calculating the directional Fréchet derivative of $J(\Psi)$ with respect to Ψ , and equating it to zero, yields

$$\Psi(\cdot) = \left[\sum_{n=1}^N \alpha_{n,1} \kappa_1(\cdot, \mathbf{u}(n)), \dots, \sum_{n=1}^N \alpha_{n,K} \kappa_K(\cdot, \mathbf{u}(n)) \right]^\top \quad (8)$$

where $\boldsymbol{\alpha}_k = [\alpha_{1,k}, \dots, \alpha_{N,k}]^\top$ for all k , is the k -th solution of each linear system in \mathbb{R}^N :

$$(\mathbf{K}_k + \lambda \mathbf{I}_N) \boldsymbol{\alpha}_k = \mathbf{d}. \quad (9)$$

The matrix \mathbf{K}_k is the so-called Gram matrix with (i, j) -th entry defined by $\kappa_k(\mathbf{u}(i), \mathbf{u}(j))$, and \mathbf{I}_N is the identity matrix. Finally, the optimal function $\psi(\mathbf{u}) = \sum_{k=1}^K \psi_k(\mathbf{u})$ is given by

$$\psi(\cdot) = \sum_{n=1}^N \sum_{k=1}^K \alpha_{n,k} \kappa_k(\cdot, \mathbf{u}(n)). \quad (10)$$

2.3. Multikernel LMS algorithm

We start by defining the cost function $J(\Psi) = \frac{1}{K} \sum_{k=1}^K E\{e_k^2(n)\}$ with $e_k(n) = d(n) - \psi_k(\mathbf{u}(n))$ the k -th estimation error. For simplicity, without loss of generality, we have omitted the regularization term in problem (7). Applying steepest descent means to calculate the gradient of $J(\Psi)$ with respect to Ψ . Its k -th entry is

$$\partial_{\psi_k} J(\Psi) = -\frac{2}{K} E\{e_k(n) \kappa_k(\cdot, \mathbf{u}(n))\}. \quad (11)$$

Approximating the expectation in the above equation by the instantaneous estimate $\partial_{\psi_k} J(\Psi) \approx -\frac{2}{K} e_k(n) \kappa_k(\cdot, \mathbf{u}(n))$, we obtain the recursive update equation of the multikernel algorithm:

$$\Psi_{n+1}(\cdot) = \Psi_n(\cdot) + \boldsymbol{\eta} \odot \begin{bmatrix} e_1(n) \kappa_1(\cdot, \mathbf{u}(n)) \\ \vdots \\ e_K(n) \kappa_K(\cdot, \mathbf{u}(n)) \end{bmatrix} \quad (12)$$

with $\boldsymbol{\eta} = [\eta_1, \dots, \eta_K]^\top$ a vector of step-sizes, and \odot the Hadamard product. It can be observed from (12) that the function Ψ is iteratively updated with K different step-sizes and estimation errors. For each sub-filter, we have

$$\begin{aligned} \psi_{n+1,k}(\cdot) &= \psi_{n,k}(\cdot) + \eta_k e_k(n) \kappa_k(\cdot, \mathbf{u}(n)) \\ &= \sum_{j=1}^n \eta_k e_k(j) \kappa_k(\cdot, \mathbf{u}(j)). \end{aligned} \quad (13)$$

In order to overcome the problem of the increasing amount n of observations in an online context, a fixed-size model is usually adopted

$$\begin{aligned} \psi(\cdot) &= \sum_{k=1}^K \sum_{m=1}^{M_k} \alpha_{m,k} \kappa_k(\cdot, \mathbf{u}(\omega_{m,k})) \\ &= \sum_{k=1}^K \boldsymbol{\alpha}_k^\top \boldsymbol{\kappa}_{\omega,k}(\cdot) \end{aligned} \quad (14)$$

with $\mathcal{D}_k = \{\kappa_k(\cdot, \mathbf{u}(\omega_{m,k}))\}_{m=1}^{M_k}$ the so-called dictionary of the filter ψ_k , M_k its length, and $\boldsymbol{\kappa}_{\omega,k}(\cdot)$ the $M_k \times 1$ vector with m -th entry $\kappa_k(\cdot, \mathbf{u}(\omega_{m,k}))$. For simplicity, in this paper, the construction strategy of individual dictionaries for each KLMS filter is based the coherence sparsification criterion [3]

$$\max_{m=1, \dots, M_k} |\kappa_k(\mathbf{u}(n), \mathbf{u}(\omega_{m,k}))| \leq \mu_k \quad (15)$$

with μ_k in $[0, 1]$ a threshold that determines both the level of sparsity and the coherence of the k -th dictionary. See [3] for details. There are two cases for each KLMS filter depending on whether the condition (15) is met. At each time instant n , and each filter ψ_k , upon the arrival of $(\mathbf{u}(n), d(n))$, repeat:

- **Case 1:** $\max_{m=1, \dots, M_k} |\kappa_k(\mathbf{u}(n), \mathbf{u}(\omega_{m,k}))| > \mu_k$

$$\alpha_k(n) = \alpha_k(n-1) + \eta_k e_k(n) \kappa_{\omega,k}(\mathbf{u}(n)) \quad (16)$$

- **Case 2:** $\max_{m=1, \dots, M_k} |\kappa_k(\mathbf{u}(n), \mathbf{u}(\omega_{m,k}))| \leq \mu_k$

$$\begin{aligned} \mathcal{D}_k &= \mathcal{D}_k \cup \{\kappa_k(\cdot, \mathbf{u}(n))\} \\ \alpha_k(n) &= \begin{bmatrix} \alpha_k(n-1) \\ 0 \end{bmatrix} + \eta_k e_k(n) \kappa_{\omega,k}(\mathbf{u}(n)) \end{aligned} \quad (17)$$

In Case 2, $\kappa_k(\cdot, \mathbf{u}(n))$ becomes the $(M_k + 1)$ -th element of the dictionary. Accordingly, it is denoted by $\kappa_k(\cdot, \mathbf{u}(\omega_{M_k+1,k}))$.

3. CONVEX COMBINATION OF KERNEL FILTERS

Limiting the number of single-kernel filters to $K = 2$, the adaptive convex combination scheme described in [24–28] applies. The block diagram is depicted in the Fig. 1. The overall weighted function $\psi(\cdot)$ is now defined as

$$\begin{aligned} \psi(\cdot) &= \lambda(n) \psi_1(\cdot) + [1 - \lambda(n)] \psi_2(\cdot) \\ &= \lambda(n) \sum_{m=1}^{M_1} \alpha_{m,1} \kappa_1(\cdot, \mathbf{u}(\omega_{m,1})) \\ &\quad + [1 - \lambda(n)] \sum_{m=1}^{M_2} \alpha_{m,2} \kappa_2(\cdot, \mathbf{u}(\omega_{m,2})) \end{aligned} \quad (18)$$

where ψ_1 and ψ_2 are updated at each time instant n with KLMS update equations (16)–(17). As illustrated in the following, the combination scheme (18) can be generalized by considering other kernel adaptive filtering techniques such as KRLS, KAPA, KNLMS, etc.

We shall now introduce a strategy to adapt the combination weight $\lambda(n)$. Following [24, 26], a possible strategy is to define $\lambda(n)$ as a sigmoid function of the variable $a(n)$:

$$\lambda(n) = \text{sgm}(a(n)) = [1 + \exp(-a(n))]^{-1}. \quad (19)$$

Parameter $a(n)$ is updated via stochastic gradient descent of the estimation error $E\{e^2(n)\}$. We get

$$\begin{aligned} a(n+1) &= a(n) - \frac{\mu}{2} \frac{\partial e^2(n)}{\partial a(n)} \\ &= a(n) + \mu e(n) [\psi_1(\mathbf{u}(n)) - \psi_2(\mathbf{u}(n))] \lambda(n) [1 - \lambda(n)]. \end{aligned} \quad (20)$$

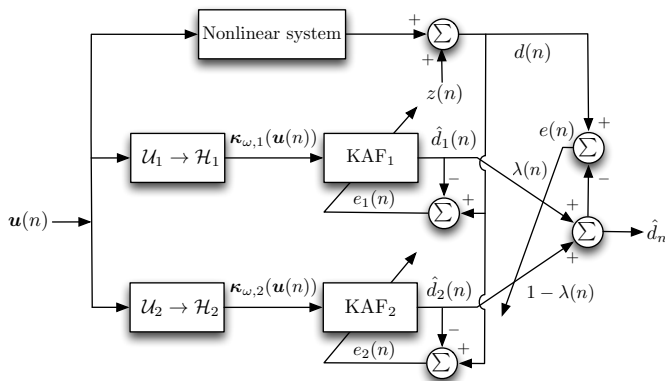


Fig. 1. Adaptive combination of two single-kernel adaptive filters.

The adaptation speed is reduced by a factor of $\lambda(n)[1 - \lambda(n)]$ when $\lambda(n)$ tends to 0 or 1. In order to circumvent this drawback, $a(n)$ can be limited to an interval $[-a_0, a_0]$, restricting $\lambda(n)$ to be in the range $[1 - \lambda_0, \lambda_0]$ with $\lambda_0 = \text{sgm}(a_0)$.

4. EXPERIMENTS

In this section, we consider two nonlinear system identification problems to illustrate the performance of our algorithm.

4.1. Combination of two KLMS filters

Consider first the convex combination of two KLMS algorithms with Gaussian kernels of different bandwidths. We used the nonlinear system described by the difference equation [3]

$$\begin{aligned} d(n) &= [0.8 - 0.5 \exp(-d(n-1)^2)] d(n-1) \\ &\quad - [0.3 + 0.9 \exp(-d(n-1)^2)] d(n-2) \\ &\quad + 0.1 \sin(d(n-1)\pi) \end{aligned} \quad (21)$$

with $d(n)$ the system output, and $\mathbf{u}(n) = [d(n-1), d(n-2)]^\top$ the input. The initial condition was set to $d(0) = d(1) = 0.1$. The output $d(n)$ was corrupted by an additive zero-mean white Gaussian noise $z(n)$ with standard deviation $\sigma = 0.1$. A comparison was performed between the convex combination of two KLMS algorithms, called ComKAF hereafter, the two KLMS algorithms considered separately, the MKLMS algorithm [18, 21], and the MxKLMS algorithm [23]. The coherence criterion was used to construct the filter dictionaries. The coherence thresholds were set in order to get the same filter length for all the filters. The parameter setting for each algorithm is provided in Table 1. The parameters μ and a_0 were set to 80 and 4, respectively. The learning rate used to update the gate function of the MxKLMS algorithm was set to 0.1. The experiments were averaged over 200 Monte Carlo runs.

Table 1. Parameters setting for experiment 1.

Algorithm	Kernel bandwidth ξ	Step-size η	Coherence threshold μ
KLMS1	0.25	0.05	0.5
KLMS2	1	0.05	0.9576
MKLMS	[0.25; 1]	0.03	[0.5; 0.9576]
MxKLMS	[0.25; 1]	0.15	[0.5; 0.9576]
ComKAF	[0.25; 1]	[0.05; 0.05]	[0.5; 0.9576]

Figure 2 (left) shows that our algorithm got the best of both single-kernel KLMS: it converged as fast as the fastest KLMS, and achieved the lowest steady-state EMSE of both algorithms. This behavior is due to the adaptive gate function $\lambda(n)$ that allows to switch between the two single-kernel algorithms, as illustrated in Figure 2 (right). Figure 2 (middle) confirms that, for fair comparison, the coherence thresholds were set in order to get the same dictionary length for all the algorithms. Regarding method comparison, Figure 2 (left) shows that the three multikernel methods obtained almost the same performance. Figure 2 (right) shows that the gate function of MxKLMS did not converge toward the same value as our approach.

4.2. Combination of KLMS and KRLS filters

Consider now the convex combination of KLMS and KRLS algorithms. We addressed a nonlinear channel equalization problem de-

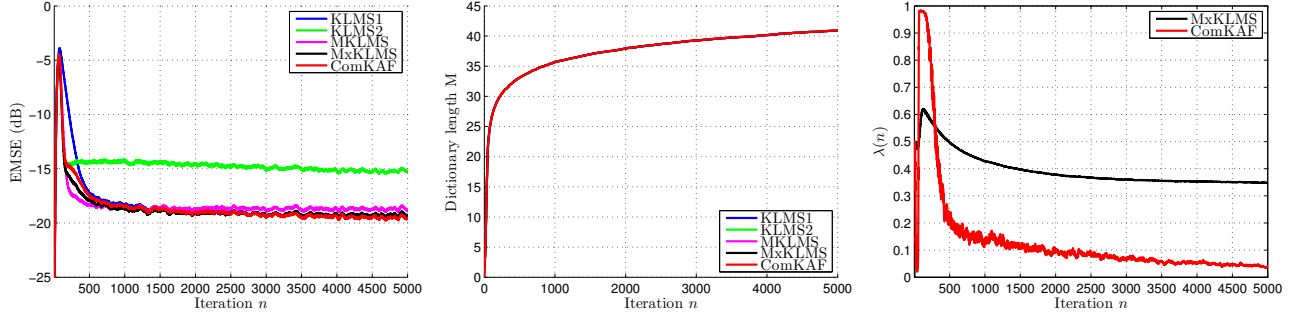


Fig. 2. The results of experiment 1. Ensemble-average EMSE learning curves (left). Ensemble-average evolution of the dictionary length (middle). Averaged curves for the function weight (right).

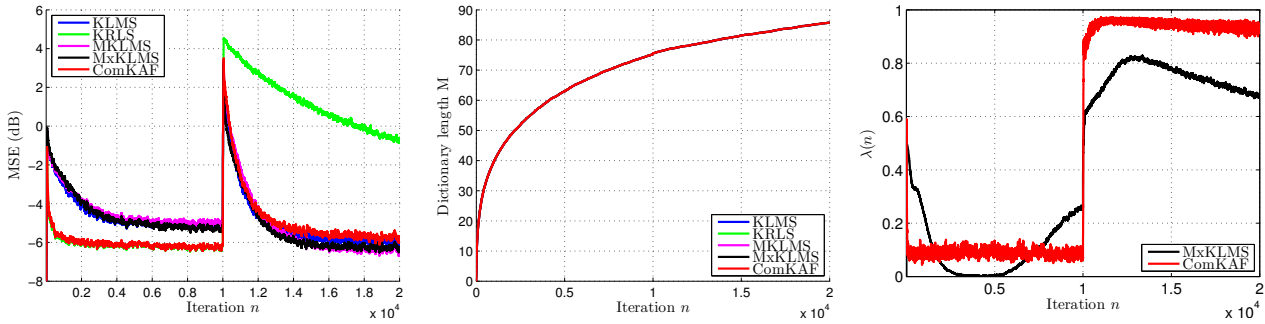


Fig. 3. The results of experiment 2. Ensemble-average MSE learning curves (left). Ensemble-average evolution of the dictionary length (middle). Averaged curves for the function weight (right).

fined as follows. At first, the input signal passes through a channel consisting of the linear part and nonlinear part

$$\begin{cases} x(n) = -0.8u(n) + 0.7u(n-1) - 0.6u(n-2) + 0.1u(n-3) \\ d(n) = x(n) + 0.08x(n)^2 + z(n) \end{cases}$$

with $z(n)$ an additive white Gaussian noise with SNR = 15dB. After 1×10^4 time samples, the channel abruptly changes into the following system

$$\begin{cases} x(n) = 0.6u(n) - 0.9u(n-1) + 0.5u(n-2) - 0.4u(n-3) \\ d(n) = x(n) + 0.1x(n)^2 + z(n). \end{cases}$$

The sample set was defined as $\{(u(n), d(n))\} = \{[u(n), \dots, u(n-\ell+1)]^T, d(n-\delta)\}$, with equalizer length $\ell = 5$ and equalization time delay $\delta = 2$. The coherence criterion was used to construct the filter dictionaries. The coherence thresholds were set in order to get the same filter length for all the filters. The parameter setting for each algorithm is provided in Table 2. The parameters μ and a_0 were set to 40 and 4, respectively. The learning rate used to update the gate function of the MxKLMS algorithm was set to 0.1. The experiments were averaged over 200 Monte Carlo runs.

As shown in Figure 3 (left), the proposed algorithm got the best of KLMS and KRLS algorithm. During the first stage of the experiment, it performed as well as the KRLS algorithm. Nevertheless, the latter is characterized by a poor tracking ability. During the second part of the experiment, the proposed algorithm then switched to the KLMS behavior as illustrated in Figure 3 (right). Figure 3 (middle) confirms that, for fair comparison, the coherence thresholds were set in order to get the same dictionary length for all the algorithms. Regarding method comparison, Figure 3 (left) shows that our method

Table 2. Parameters setting for experiment 2.

Algorithm	Kernel bandwidth ξ	Step-size η	Coherence threshold μ
KLMS	2.5	0.05	0.6457
KRLS	3.5	—	0.8
MKLMS	[2.5; 3.5]	0.025	[0.6457; 0.8]
MxKLMS	[2.5; 3.5]	0.05	[0.6457; 0.8]
ComKAF	[2.5; 3.5]	0.05	[0.6457; 0.8]

outperformed the other multikernel approaches during the first part of the experiment. The three multikernel methods obtained almost the same performance during the second part of the experiment. Figure 3 (right) shows that, unlike our approach, the gate function of MxKLMS did not converged toward a steady-state value. This behavior is confirmed by the experiments depicted in [23].

5. CONCLUSION

As an ongoing extension of convex or affine combinations of two LMS filters with different step-sizes, we proposed a flexible combination of two single-kernel adaptive filters. Simulations illustrated the ability of our mixture of filters to get the best of both single-kernel adaptive filters. In a future work, we shall analyze the convergence behavior of the proposed and consider the combination of more than two filters. To adapt multiple gate coefficients $\lambda_k(n)$ in an online manner, subject to nonnegativity and sum-to-one constraints, the so-called NNLS algorithm described in [31–33] appears as a promising strategy.

6. REFERENCES

- [1] Y. Engel, S. Mannor, and R. Meir, “Kernel recursive least squares,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.
- [2] W. Liu, P. P. Pokharel, and J. C. Príncipe, “The kernel least-mean-square algorithm,” *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.
- [3] C. Richard, J.-C. M. Bermudez, and P. Honeine, “Online prediction of time series data with kernels,” *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009.
- [4] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, John Wiley & Sons, New-York, 2010.
- [5] S. Van Vaerenbergh, J. Vía, and I. Santamaría, “A sliding-window kernel RLS algorithm and its application to nonlinear channel identification,” in *Proc. IEEE ICASSP*, Toulouse, France, May 2006, pp. 789–792.
- [6] W. Liu, I. M. Park, Y. Wang, and J. C. Príncipe, “Extended kernel recursive least squares algorithm,” *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3801–3814, 2009.
- [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] S. Slavakis and S. Theodoridis, “Sliding window generalized kernel affine projection algorithm using projection mappings,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008:735351, Apr. 2008.
- [9] C. Richard, “Filtrage adaptatif non-linéaire par méthodes de gradient stochastique court-terme à noyau,” in *Proc. GRETSI*, Louvain-la-Neuve, Belgium, Sep. 2005, pp. 1–4.
- [10] W. Liu and J. C. Príncipe, “Kernel affine projection algorithms,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 2008:784292, 2008.
- [11] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, “Quantized kernel least mean square algorithm,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.
- [12] W. D. Parreira, J.-C. M. Bermudez, C. Richard, and J.-Y. Tourneret, “Stochastic behavior analysis of the Gaussian kernel-least-mean-square algorithm,” *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2208–2222, 2012.
- [13] C. Richard and J.-C. M. Bermudez, “Closed-form conditions for convergence of the gaussian kernel-least-mean-square algorithm,” in *Proc. Asilomar*, Pacific Grove, CA, USA, Nov. 2012.
- [14] W. Gao, J. Chen, C. Richard, J. Huang, and R. Flamary, “Kernel LMS algorithm with forward-backward splitting for dictionary learning,” in *Proc. IEEE ICASSP*, Vancouver, Canada, 2013, pp. 5735–5739.
- [15] W. Gao, J. Chen, C. Richard, and J. Huang, “Online dictionary learning for kernel LMS,” *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2765–2777, 2014.
- [16] J. Chen, W. Gao, C. Richard, and J.-C. M. Bermudez, “Convergence analysis of kernel LMS algorithm with pre-tuned dictionary,” in *Proc. IEEE ICASSP*, Florence, Italia, May 2014.
- [17] M. Yukawa, “Nonlinear adaptive filtering techniques with multiple kernels,” in *Proc. EUSIPCO*, Barcelona, Spain, 2011, pp. 136–140.
- [18] M. Yukawa, “Multikernel adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4672–4682, 2012.
- [19] M. Yukawa and R. Ishii, “Online model selection and learning by multikernel adaptive filtering,” in *Proc. EUSIPCO*, Marrakech, Morocco, Sept. 2013, pp. 1–5.
- [20] F. A. Tobar and D. P. Mandic, “Multikernel least squares estimation,” in *Proceedings of Sensor Signal Processing for Defence*, London, UK, 2012.
- [21] F. A. Tobar, S.-Y. Kung, and D. P. Mandic, “Multikernel least mean square algorithm,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 265–277, 2014.
- [22] T. Ishida and T. Tanaka, “Multikernel adaptive filters with multiple dictionaries and regularization,” in *Proc. APSIPA*, Kaohsiung, Taiwan, Oct.-Nov. 2013.
- [23] R. Pokharel, S. Seth, and J. Príncipe, “Mixture kernel least mean square,” in *Proc. IEEE IJCNN*, 2013.
- [24] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, “Mean-square performance of a convex combination of two adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, 2006.
- [25] N. J. Bershad, J.-C. M. Bermudez, and J.-Y. Tourneret, “An affine combination of two LMS adaptive filters—transient mean-square analysis,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1853–1864, 2008.
- [26] M. T. M. Silva and V. H. Nascimento, “Improving the tracking capability of adaptive filters via convex combination,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3137–3149, 2008.
- [27] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, “Steady-state mse performance analysis of mixture approaches to adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4050–4063, 2010.
- [28] R. Candido, M. T. M. Silva, and V. H. Nascimento, “Transient and steady-state analysis of the affine combination of two adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4064–4078, 2010.
- [29] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.
- [30] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, “The method of potential functions for the problem of restoring the characteristic of a function converter from randomly observed points,” *Automation and Remote Control*, vol. 25, no. 12, pp. 1546–1556, 1964.
- [31] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine, “Nonnegative least-mean-square algorithm,” *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5225–5235, 2011.
- [32] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine, “Variants of non-negative least-mean-square algorithm and convergence analysis,” Tech. Rep., University of Nice Sophia-Antipolis, France, 2014. Available at <http://www.cedric-richard.fr/Articles/chen2013variants.pdf>.
- [33] J. Chen, J.-C. M. Bermudez, and C. Richard, “Steady-state performance of non-negative least-mean-square algorithm and its variants,” *IEEE Signal Processing Letters*, 2014.