# Online kernel adaptive algorithms with dictionary adaptation for MIMO models

Chafic SAIDE, *Student Member, IEEE,* Régis LENGELLE, Paul HONEINE, *Member, IEEE,* and Roger ACHKAR, *Member, IEEE*

**Abstract**

Nonlinear system identification has always been a challenging problem. The use of kernel methods to solve such problems becomes more prevalent. However, the complexity of these methods increases with time which makes them unsuitable for online identification. This drawback can be solved with the introduction of the coherence criterion. Furthermore, dictionary adaptation using a stochastic gradient method proved its efficiency. Mostly, all approaches are used to identify Single Output models which form a particular case of real problems. In this paper we investigate online kernel adaptive algorithms to identify Multiple Inputs Multiple Outputs model as well as the possibility of dictionary adaptation for such models.

**Index Terms**

Nonlinear adaptive filters, machine learning, nonlinear systems, kernel methods.

## I. Introduction

System identification methods based on reproducing kernel Hilbert spaces (RKHS) are very important in kernel-based regression methods such as support vector regression [1], [2]. The derived models are essentially for Single Output systems. In real life, many applications are naturally described by Multiple Inputs Multiple Outputs models (MIMO) [3], which identification



C. SAIDE: Institut Charles Delaunay (UMR CNRS 6279), LM2S, Université de technologie de Troyes, 10010 Troyes, France. Phone: + 33 3 25 71 85 35 e-mail: chafic.saide@utt.fr).

R. LENGELLE: Institut Charles Delaunay (UMR CNRS 6279), LM2S, Université de technologie de Troyes, 10010 Troyes, France. Phone: + 33 3 25 71 56 81 e-mail: regis.lengelle@utt.fr

P. HONEINE: Institut Charles Delaunay (UMR CNRS 6279), LM2S, Université de technologie de Troyes, 10010 Troyes, France. Phone: + 33 3 25 71 56 25 e-mail: paul.honeine@utt.fr

R. ACHKAR: American University of Science and Technology (AUST), Beirut, Lebanon. Phone: + 961 1 21 87 16, e-mail: rachkar@aust.edu.lb

is achieved using several methods such as neural networks [4]. Online algorithms, e.g. Kernel Affine Projection Algorithm (KAPA) and Kernel Recursive Least Squares (KRLS) along with the application of a sparsification criterion [5], [6], made a major step toward online identification by reducing the computational burden [7], [8].

The coherence criterion allows the selection of a subset of past input samples, called the *dictionary*, which contributes the most to the prediction model [9]. Using the coherence criterion it has been shown that the size of the dictionary remains finite with time. Moreover, without dictionary adaptation, elements remain unchanged even if they become less relevant in the model. This is the reason why in [10], and in the MISO case, we proposed an heuristic to adapt the dictionary elements coupled with the coherence criterion for the KAPA algorithm.

In this paper, we present online MIMO kernel adaptive algorithms (KAPA and KRLS) coupled with dictionary adaptation.

## II. Single Output versus Multiple Outputs model

### A. Multiple Inputs Single Output model

Consider an online identification problem and let $\boldsymbol{u}_n \in \mathcal{U} \subset \mathbb{R}^l$ be the input vector at time step $n$, and $d_n \in \mathbb{R}$ be the corresponding desired output. Let $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ be a kernel function and $\mathcal{H}$ the RKHS associated with it. Considering $\psi_n(\cdot)$, a real-valued function that corresponds to the output of the model, for some positive scalar $\eta$ and monotonic increasing function $\Psi(\cdot)$, the solution of the following optimization problem. Then,

$$\psi_n = \arg\min_{\psi} \sum_{i=1}^{n} |d_i - \psi(\boldsymbol{u}_i)|^2 + \eta\, \Psi(\|\psi\|^2_{\mathcal{H}})$$

is, according to the representer theorem [11], [12], given by

$$y_n = \psi_n(\boldsymbol{u}_n) = \sum_{i=1}^{n} \alpha_{n,i}\, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_i).$$

As can be seen, the model complexity increases with time. To tackle this problem, we briefly introduce the use of the coherence criterion. At time step $n$, for a dictionary $\{\boldsymbol{u}_{w_1}, \cdots, \boldsymbol{u}_{w_m}\}_{m \ll n}$ and any unit-norm[1] kernel $\kappa$, the coherence $\mu$ parameter is defined as:

$$\mu = \max_{i \neq j} |\kappa(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_j})|$$

A candidate input $\boldsymbol{u}_n$ is introduced into the dictionary if the following condition is satisfied:

$$\max_{j=1\cdots m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})| \leq \mu_0$$

where $\mu_0 \in [0, 1[$ is a threshold parameter determining the level of sparsity and the coherence of the dictionary. Maintaining $\mu < \mu_0$ implies that the dimension of the dictionary remains finite as time $n \to \infty$ [9]. Using the coherence criterion to control

[1]Otherwise, normalize the kernel, by replacing $\kappa(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_j})$ with $\kappa(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_j}) / (\kappa(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_i}) \kappa(\boldsymbol{u}_{w_j}, \boldsymbol{u}_{w_j}))^{\frac{1}{2}}$.

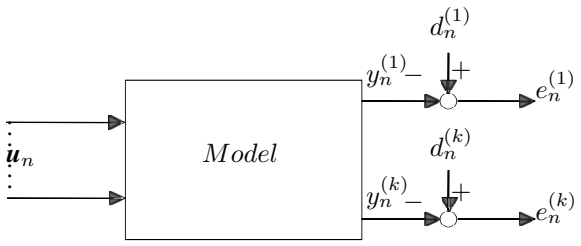


Fig. 1. Multiple Inputs Multiple Outputs model.

the size of the dictionary we get:

$$y_n = \psi_n(\boldsymbol{u}_n) = \sum_{i=1}^{m} \alpha_{n,i}\, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_i}) \tag{1}$$

where $m \ll n$. The estimation of the model parameters results from an optimization problem.

### *B. Application to the single-output KAPA problem*

We first consider the case of the KAPA algorithm (studied in the linear case in [13] and in the Kernelized APA [14], [15]), $\boldsymbol{\alpha}_n = [\alpha_{n,1} \ \cdots \ \alpha_{n,m}]^t$ is the solution of

$$\begin{aligned} \boldsymbol{\alpha}_n &= \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_{n-1}\|^2 \\ \text{subject to} \qquad \boldsymbol{d}_n &= \boldsymbol{H}_n \boldsymbol{\alpha}_n \end{aligned} \tag{2}$$

where $\boldsymbol{d}_n = [d_n \ \cdots \ d_{n-p+1}]^t$ is the desired output, $p$ is the width of the considered sliding window, and $\boldsymbol{H}_n$ is a matrix whose $(i,j)^{th}$ element is $\kappa(\boldsymbol{u}_{n-i+1}, \boldsymbol{u}_{w_j})$ with $i = 1 \cdots p$ and $j = 1 \cdots m$. When $p = 1$, the Kernel Affine projection Algorithm is equivalent to the Kernel Normalized Least squares Algorithm (KNLMS) (see for instance [16]).

### *C. Derivation of the MIMO KAPA problem*

The main difference between several MISO models in parallel and a single MIMO model is that in MIMO models all outputs share the same dictionary. The MIMO model, with $l$ inputs and $k$ outputs, is illustrated in Fig. 1. Let us consider $l$ time series $\{u_{(n)}^{(r)}\}_{r=1 \cdots l}$ as inputs and let $\boldsymbol{u}_n = [u_n^{(1)} \cdots u_n^{(l)}]^t$ be the input vector at time step $n$. Hence, $y_n^{(j)}$ is the $j^{th}$ output of the model, $\boldsymbol{d}_n^{(j)} = [d_n^{(j)} \ \cdots \ d_{n-p+1}^{(j)}]^t$ represents the $j^{th}$ desired output vector and the $j^{th}$ component of the error is $e_n^{(j)} = d_n^{(j)} - y_n^{(j)}$. Given that all the model outputs share the same dictionary, the constraints in (2) become:

$$\boldsymbol{d}_n^{(1)} = \boldsymbol{H}_n \boldsymbol{\alpha}_n^{(1)} \ \cdots \ \boldsymbol{d}_n^{(k)} = \boldsymbol{H}_n \boldsymbol{\alpha}_n^{(k)}$$

where $\boldsymbol{\alpha}^{(j)} = [\alpha_{n,1}^{(j)} \ \cdots \ \alpha_{n,m}^{(j)}]^t$ is the $j^{th}$ solution vector and $j = 1 \cdots k$.

Let, at time step $n$, $\boldsymbol{D}_n = [\boldsymbol{d}_n^{(1)}\ \boldsymbol{d}_n^{(2)} \cdots \boldsymbol{d}_n^{(k)}]$ be the desired output matrix, $\boldsymbol{E}_n = [\boldsymbol{e}_n^{(1)}\ \boldsymbol{e}_n^{(2)} \cdots \boldsymbol{e}_n^{(k)}]$ the error matrix and $\boldsymbol{A}_n = [\boldsymbol{\alpha}_n^{(1)}\ \boldsymbol{\alpha}_n^{(2)} \cdots \boldsymbol{\alpha}_n^{(k)}]$ the solution matrix. By analogy to (2), the optimization problem becomes:

$$\begin{aligned} \boldsymbol{A}_n &= \underset{\boldsymbol{A}}{\operatorname{argmin}} \|\boldsymbol{A} - \boldsymbol{A}_{n-1}\|_F^2 \\ \text{subject to} \qquad & \boldsymbol{D}_n = \boldsymbol{H}_n \boldsymbol{A}_n. \end{aligned} \tag{3}$$

Note that $\|\cdot\|_F^2$ is the Frobenius norm.

## III. Multiple Outputs Kernel Affine Projection Algorithm (MOKAPA)

Our objective is to find the optimal solution matrix $\boldsymbol{A}_n$ with online adaptive algorithms. Using the coherence criterion presented above, at instant $n$, when a new input $\boldsymbol{u}_n$ is fed to the model, one of the following two cases occurs:

- $\max_{j=1,\ldots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})| > \mu_0$

  In this case $\boldsymbol{u}_n$ is not introduced into the dictionary and the Lagrangian is:

  $$J(\boldsymbol{A}, \boldsymbol{\Lambda}) = \|\boldsymbol{A} - \boldsymbol{A}_{n-1}\|_F^2 + \mathbf{1}_p^t (\boldsymbol{\Lambda} \odot (\boldsymbol{D}_n - \boldsymbol{H}_n \boldsymbol{A})) \mathbf{1}_k$$

  where $\odot$ is the Hadamard product, $\boldsymbol{\Lambda}$ is the matrix of Lagrange multipliers, and $\mathbf{1}_n$ a vector of $n$ ones. Finding the derivatives of the above cost function with respect to $\boldsymbol{\Lambda}$ and $\boldsymbol{A}$ and setting these derivatives to zero at $\boldsymbol{\Lambda}_n$ and $\boldsymbol{A}_n$ yields the following two expressions:

  $$2(\boldsymbol{A}_n - \boldsymbol{A}_{n-1}) = \boldsymbol{H}_n^t \boldsymbol{\Lambda}_n \qquad \text{and} \qquad \boldsymbol{D}_n = \boldsymbol{H}_n \boldsymbol{A}_n$$

  Assuming that $\boldsymbol{H}_n \boldsymbol{H}_n^t$ is not singular, we get:

  $$\boldsymbol{\Lambda}_n = 2(\boldsymbol{H}_n \boldsymbol{H}_n^t)^{-1} (\boldsymbol{D}_n - \boldsymbol{H}_n \boldsymbol{A}_{n-1})$$

  The solution matrix is then updated according to:

  $$\boldsymbol{A}_n = \boldsymbol{A}_{n-1} + \boldsymbol{H}_n^t (\epsilon \boldsymbol{I} + \boldsymbol{H}_n \boldsymbol{H}_n^t)^{-1} (\boldsymbol{D}_n - \boldsymbol{H}_n \boldsymbol{A}_{n-1})$$

  where we introduced $\epsilon \boldsymbol{I}$ as a regularization factor.

- $\max_{j=1,\ldots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})| \leq \mu_0$

  In this case $\boldsymbol{u}_n$ is introduced into the dictionary and $\boldsymbol{u}_{w_{m+1}} = \boldsymbol{u}_n$. The size of the matrix $\boldsymbol{H}_n$ is increased by concatenating the column $[\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_{m+1}}) \cdots \kappa(\boldsymbol{u}_{n-p+1}, \boldsymbol{u}_{w_{m+1}})]^t$ and the solution matrix is updated using the following expression:

  $$\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{n-1} \\ \mathbf{0} \end{bmatrix} + \boldsymbol{H}_n^t (\epsilon \boldsymbol{I} + \boldsymbol{H}_n \boldsymbol{H}_n^t)^{-1} \Big( \boldsymbol{D}_n - \boldsymbol{H}_n \begin{bmatrix} \boldsymbol{A}_{n-1} \\ \mathbf{0} \end{bmatrix} \Big)$$

We call these last expressions Multiple Outputs Kernel Affine Projection Algorithm (MOKAPA). Note that when choosing $p = 1$, the MOKAPA is called the Multiple Outputs Kernel Normalized Least Mean squares algorithm (MOKNLMS).

## IV. Multiple Outputs Kernel Recursive Least squares Algorithm (MOKRLS)

In this section, we briefly present the KRLS algorithm extended to the multiple outputs case, and coupled with the coherence criterion to perform selection of the dictionary elements. In the single output case, the KRLS algorithm takes into consideration all the previous $n$ inputs to the model [17], [18], and minimizes

$$\boldsymbol{\alpha}_n = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{d}_n - \boldsymbol{H}_n\boldsymbol{\alpha}\|^2 + \xi\boldsymbol{\alpha}^t\boldsymbol{K}_n\boldsymbol{\alpha}.$$

$\boldsymbol{H}_n$ becomes a $n \times m$ matrix where the $(i,j)^{th}$ entry is $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_{w_j})$ and the $(i,j)^{th}$ entry of the $m \times m$ Gram matrix $\boldsymbol{K}_n$ is $\kappa(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_j})$.

In the multiple output case, and as in the previous section, all outputs share the same dictionary, then $\boldsymbol{H}_n$ and $\boldsymbol{K}_n$ are the same for all outputs. Assuming that $\boldsymbol{P}_n = (\boldsymbol{H}_n^t\boldsymbol{H}_n + \xi\boldsymbol{K}_n)^{-1}$ exists, it is also shared by all outputs ($\xi > 0$ is a regularization coefficient). At iteration $n$, when a new input vector is fed to the model, one of the two following cases occurs:

- $\max_{j=1,\dots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})| > \mu_0$

  In this case, $\boldsymbol{u}_n$ is not introduced into the dictionary, and a new line $\boldsymbol{h}_n = [\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_1}) \cdots \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_m})]$ is introduced into the matrix $\boldsymbol{H}_{n-1}$. It can be shown that the solution matrix $\boldsymbol{A}_n$ can be updated using the following expressions:

$$\boldsymbol{A}_n = \boldsymbol{A}_{n-1} + \frac{\boldsymbol{P}_{n-1}\boldsymbol{h}_n^t}{1+\varrho}\left([d_n^{(1)} \cdots d_n^{(k)}] - \boldsymbol{h}_n\boldsymbol{A}_{n-1}\right)$$
$$\boldsymbol{P}_n = \boldsymbol{P}_{n-1} - \frac{\boldsymbol{P}_{n-1}\boldsymbol{h}_n^t\boldsymbol{h}_n\boldsymbol{P}_{n-1}}{1+\varrho}$$

  where $\varrho = \boldsymbol{h}_n\boldsymbol{P}_{n-1}\boldsymbol{h}_n^t$.

- $\max_{j=1,\dots,m} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})| \leq \mu_0$

  In this case, $\boldsymbol{u}_n$ is introduced into the dictionary. Hence, the size of the dictionary is increased by one to become $m+1$. The updates of the solution matrix and of the matrix $\boldsymbol{P}_n$ are achieved in two phases. In the first phase, we apply the same expression as above to obtain $\widetilde{\boldsymbol{A}}_n$ and $\widetilde{\boldsymbol{P}}_n$. In the second phase, the updates are obtained as follows :

$$\boldsymbol{A}_n = \begin{bmatrix} \widetilde{\boldsymbol{A}}_n \\ \boldsymbol{0}_k \end{bmatrix} + \begin{bmatrix} -\widetilde{\boldsymbol{P}}_n\boldsymbol{h}_n^t \\ 1/h_0 \end{bmatrix} \frac{[d_n^{(1)} \cdots d_n^{(k)}] - \boldsymbol{h}_n\widetilde{\boldsymbol{A}}_n}{1-\widetilde{\varrho}}$$
$$\boldsymbol{P}_n = \begin{bmatrix} \widetilde{\boldsymbol{P}}_n & \boldsymbol{0}_k^t \\ \boldsymbol{0}_k & 0 \end{bmatrix} + \frac{1}{1-\widetilde{\varrho}} \begin{bmatrix} -\widetilde{\boldsymbol{P}}_n\boldsymbol{h}_n^t \\ 1/h_0 \end{bmatrix} [-(\widetilde{\boldsymbol{P}}_n\boldsymbol{h}_n^t)^t \quad 1/h_0]$$

  where $\widetilde{\varrho} = h_n\widetilde{\boldsymbol{P}}_n\boldsymbol{h}_n^t$ and $h_0 = \kappa(\boldsymbol{u}_n, \boldsymbol{u}_n)$.

TABLE I

EXPERIMENTAL SETUP AND PERFORMANCE, WITH $\mu_0 = 0.3$ AND $\nu_0 = 0.05$

| | | Output#1 | Output#2 | Output#3 | Output#4 | Output#5 | Output#6 | Output#7 | Output#8 |
|---|---|---|---|---|---|---|---|---|---|
| MOKAPA | NMSE (Without adaptation) | 0.21691 | 0.11548 | 0.62603 | 0.03242 | 1.00880 | 0.69114 | 1.03240 | 1.20270 |
| | NMSE (With adaptation) | 0.10505 | 0.07601 | 0.25562 | 0.022164 | 0.50270 | 0.44475 | 0.52830 | 0.69226 |
| | Decrease | 51.57% | 34.18% | 59.17% | 31.63% | 50.17% | 35.65% | 48.83% | 42.44% |
| MOKRLS | NMSE (Without adaptation) | 0.39397 | 0.43833 | 0.26720 | 0.70404 | 0.077164 | 0.10241 | 0.07745 | 0.12580 |
| | NMSE (With adaptation) | 0.19617 | 0.25243 | 0.05901 | 0.48677 | 0.03004 | 0.03999 | 0.029872 | 0.03654 |
| | Decrease | 50.21% | 42.41% | 77.91% | 30.86% | 58.07% | 60.95% | 61.43% | 70.96% |

The previous expressions used to update the solution matrix are called the Multiple Outputs Kernel Recursive Least Squares algorithm (MOKRLS), and correspond to the extension to the MIMO case of equations presented in [18].

## V. DICTIONARY ADAPTATION FOR MIMO MODELS

Without adaptation, a dictionary element, once introduced into the dictionary, remains unchanged even if it becomes less relevant in the model. In a previous work and in the MISO case, [10], the importance of dictionary adaptation was revealed and the gain using adaptation was obvious. We now consider the dictionary elements as model parameters to be adapted jointly with the matrix $\boldsymbol{A}_n$.

The adaptation technique is based on the stochastic gradient of the instantaneous quadratic error w.r.t. the dictionary elements. In the case of a MIMO model, the error at time $n$ is a vector $\boldsymbol{e}_n = [e_n^{(1)} \quad e_n^{(2)} \cdots e_n^{(k)}]^t$. Dictionary adaptation is achieved using the gradient of the $\ell_2$ norm of $\boldsymbol{e}_n$ with respect to the dictionary elements so as to reduce the instantaneous quadratic error. Each dictionary element $\boldsymbol{u}_{w_i}$ is modified to become $\boldsymbol{u}_{w_i}^A$ according to:

$$\boldsymbol{u}_{w_i}^A = \boldsymbol{u}_{w_i} - \nu_n \boldsymbol{g}_{w_i} \tag{4}$$

where $\boldsymbol{g}_{w_i} = \nabla_{\boldsymbol{u}_{w_i}} \|\boldsymbol{e}_n\|^2$ and $\nu_n$ is the gradient step size that should be chosen to adapt all the dictionary elements without violating the coherence criterion. In other words, a coherent dictionary must remain coherent after the adaptation of its elements. Since $\|\boldsymbol{e}_n\|^2 = (d_n^{(1)} - y_n^{(1)})^2 + \cdots + (d_n^{(k)} - y_n^{(k)})^2$, then $\boldsymbol{g}_{w_i} = -2\big(e_n^{(1)}\alpha_{n,i}^{(1)} + \cdots + e_n^{(k)}\alpha_{n,i}^{(k)}\big)\frac{\partial \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_i})}{\partial \boldsymbol{u}_{w_i}}$.

When we consider a Gaussian kernel function of the form $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = f(\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2/2\sigma^2)$, $\sigma$ its bandwidth, then $\nabla_{\boldsymbol{u}_{w_i}} \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_i}) = \frac{1}{\sigma^2}(\boldsymbol{u}_n - \boldsymbol{u}_{w_i})\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_i})$, which yields

$$\boldsymbol{g}_{w_i} = \frac{-2}{\sigma^2}\big(e_n^{(1)}\alpha_{n,j}^{(1)} + \cdots + e_n^{(k)}\alpha_{n,i}^{(k)}\big)\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_i})(\boldsymbol{u}_n - \boldsymbol{u}_{w_i}).$$

Adaptation must be done while satisfying the coherence criterion and this could be achieved by choosing the appropriate $\nu_n$ to adapt all the dictionary elements[2]. Thus, after adaptation, the whole dictionary must satisfy:

$$\max_{i \neq j} |\kappa(\boldsymbol{u}^A_{w_i}, \boldsymbol{u}^A_{w_j})| \leq \mu_0. \tag{5}$$

Considering, $\delta\boldsymbol{u} = \boldsymbol{u}_{w_i} - \boldsymbol{u}_{w_j}$ and $\delta\mathbf{g} = \mathbf{g}_{w_i} - \mathbf{g}_{w_j}$, the expressions (4) and (5) lead to:

$$f(\|\delta\boldsymbol{u} - \nu_n\, \delta\mathbf{g}\|^2) \leq \mu_0. \tag{6}$$

Approximating the previous expression with a Taylor series around $\nu_n \sim 0$, we obtain the following inequality:

$$-\big(2\|\delta\mathbf{g}\|^2\nu_n^2 - 2\nu_n\delta\boldsymbol{u}^t\delta\mathbf{g})f^{(1)}(\|\delta\boldsymbol{u}\|^2\big) + \mu_0 - f(\|\delta\boldsymbol{u}\|^2) \geq 0,$$

where $f^{(1)}(\cdot)$ is negative for the Gaussian kernel. If the discriminant $\Delta < 0$, there is no constraint on the choice of $\nu_n$. If $\Delta \geq 0$, then the roots $\{\nu_{i,j-}, \nu_{i,j+}\}$ of the corresponding equality define intervals of admissible values of $\nu_n$ such that $\nu_n \in\ ]-\infty, \nu_{i,j-}] \cup [\nu_{i,j+}, +\infty[$. The adaptation requires the resolution of $m(m-1)/2$ quadratic equations at each iteration, with a low computational cost.

We initially select a reference step size $\nu_0 > 0$, as commonly done for adaptive algorithms with a fixed step size. By considering all the $(\nu_{i,j-}, \nu_{i,j+})$ pairs between all couples of elements of the dictionary, then $\nu_n$ is selected using the following heuristic:

- if $\max_{i,j} \nu_{i,j+} \leq 0 \Rightarrow \nu_n = \nu_0$
- if $0 \leq \min_{i,j} \nu_{i,j-} \leq \nu_0 \Rightarrow \nu_n = \min_{i,j} \nu_{i,j-}$
- if $0 \leq \nu_0 \leq \min_{i,j} \nu_{i,j-} \Rightarrow \nu_n = \nu_0$
- if $0 \leq \min_{i,j} (\nu_{i,j-})^+ \leq \nu_0 \Rightarrow \nu_n = \min_{i,j} (\nu_{i,j-})^+$
- if $0 \leq \nu_0 \leq \min_{i,j} (\nu_{i,j-})^+ \Rightarrow \nu_n = \nu_0$

where $(\nu_{i,j-})^+$ indicates the positive value of $\nu_{i,j-}$.

## VI. Experimentation

In this section, eight EMG Physical Action Data Sets taken from [19] (subset 1 - normal - running) and used as inputs. We only considered the first 2000 samples of the time series. The Gaussian kernel with $2\sigma^2 = 0.35$ (value selected with a rough grid search) is used and the performance criterion is the Normalized Mean Squared Error (NMSE) estimated over the last 500

[2]One may also consider adapting each element of the dictionary with a different $\nu_n$. However, this approach drastically increases the computational cost, and thus requires the derivation of an efficient heuristic.

samples of each output signal using the formula:

$$\text{NMSE}^{(i)} = \frac{\sum_{n=1501}^{2000}(d_n^{(i)} - y_n^{(i)})^2}{\sum_{n=1501}^{2000}(d_n^{(i)})^2} \tag{7}$$

We selected the parameter settings in such a way that the final sizes of the dictionaries are almost equal and we compared the NMSE to highlight the gain using dictionary adaptation. The selected coherence criterion is $\mu_0 = 0.3$ and the reference step size $\nu_0 = 0.05$. First, MOKAPA ($p = 3, \epsilon = 0.09$) is used to identify a model with eight outputs. With dictionary adaptation, the final size of the dictionary is 148 elements versus 151 elements without adaptation. Finally, MOKRLS is used. With dictionary adaptation, the final size of the dictionary is 376 elements which easily compares to 382 elements without adaptation. The NMSE for all outputs is shown in table I. Due to space limitations, figures 2 and 3 show only the learning curves for output#1 using MOKAPA and MOKRLS.

The MOKRLS is also tested with the approximate linear dependence criterion (ALD) [17] with a sparsification threshold set to $0.856$ leading to a comparable dictionary size of $378$ elements. As illustrated in figure 3, the curves without adaptation are almost superimposed, while the dictionary adaptation along with the coherence criterion shows an obvious gain.

## VII. Conclusion

In this paper, we explored the possibility to apply online kernel adaptive algorithms for MIMO models along with dictionary adaptation. The obtained results reveal the decrease of the NMSE. For future works, we will extend this work to other kernel functions and adaptive algorithms.

## References


[1] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[2] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 97 –123, jan. 2011.

[3] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Adaptive multiregression in reproducing kernel hilbert spaces: The multiaccess mimo channel case," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 260 –276, feb. 2012.

[4] S. Haykin, *"Neural Networks and Learning Machines."*, 3rd ed. Prentice Hall, Nov. 2008.

[5] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, March 2009.

[6] B. Chen, S. Zhao, P. Zhu, and J. Principe, "Quantized kernel least mean square algorithm," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 1, pp. 22 –32, jan. 2012.

[7] F. Liu, T. Zhang, and J. Sun, "Adaptive mimo channel estimation and multiuser detection based on kernel iterative inversion," *IEICE trans. on fundamentals of electronics, communications and computer sciences*, vol. 87, no. 3, pp. 649–655, 2004.

[8] Y. Liu, W. Chen, H. Wang, Z. Gao, and P. Li, "Adaptive control of nonlinear time-varying processes using selective recursive kernel learning method," *Industrial & Engineering Chemistry Research*, vol. 50, no. 5, pp. 2773–2780, 2011.

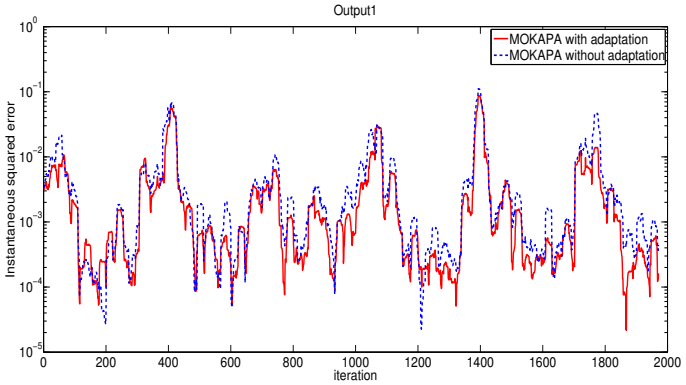


Fig. 2. MOKAPA: learning curves for output#1 with and without adaptation

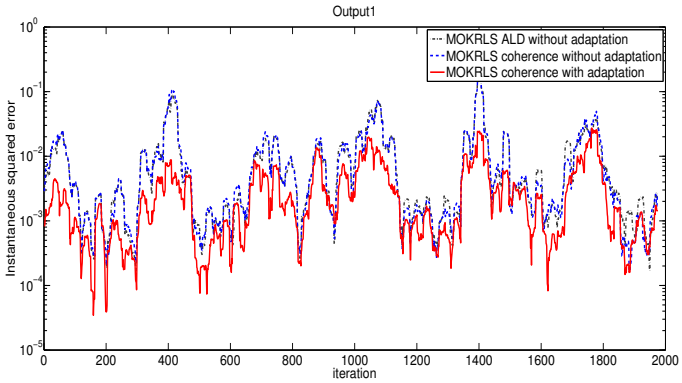


Fig. 3. MOKRLS: learning curves for output#1 with and without adaptation

[9] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE International Symposium on Information Theory*, Nice, France, June 2007, pp. 956–960.

[10] C. Saide, R. Lengelle, P. Honeine, C. Richard, and R. Achkar, "Dictionary adaptation for online prediction of time series data with kernels," in *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, aug. 2012, pp. 604 –607.

[11] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82 – 95, 1971.

[12] B. Schölkopf, R. Herbrich, A. Smola, and R. Williamson, "A generalized representer theorem," NeuroCOLT, Tech. Rep. 81, 2000.

[13] A. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[14] W. Liu and J. Príncipe, "Kernel Affine Projection Algorithms," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 784 292+, 2008.

[15] K. Slavakis and S. Theodoridis, "Sliding window generalized kernel affine projection algorithm using projection mappings." *EURASIP J. Adv. Sig. Proc.*, vol. 2008, 2008.

[16] W. Liu, P. Pokharel, and J. Principe, "The kernel least-mean-square algorithm," *Trans. Sig. Proc.*, vol. 56, no. 2, pp. 543–554, Feb. 2008.

[17] Y. Engel, S. Mannor, and R. Meir, "Kernel recursive least squares," *IEEE Trans. on Signal Processing*, vol. 52, pp. 2275–2285, 2004.

[18] P. Honeine, C. Richard, and J. C. M. Bermudez, "Modélisation parcimonieuse non linéaire en ligne par une méthode á noyau reproduisant et un critère de cohérence," in *Actes du XXI-ème Colloque GRETSI sur le Traitement du Signal et des Images*, Troyes, France, September 2007.

[19] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml