ONLINE LEARNING BASED ON ITERATIVE PROJECTIONS IN SUM SPACE OF LINEAR AND GAUSSIAN REPRODUCING KERNEL HILBERT SPACES

*Masahiro Yukawa**

Department of Electronics and Electrical Engineering, Keio University, Japan

ABSTRACT

We propose a novel multikernel adaptive filtering algorithm based on the iterative projections in the sum space of reproducing kernel Hilbert spaces. We employ linear and Gaussian kernels, envisioning an application to partially-linear-system identification/estimation. The algorithm is derived by reformulating the hyperplane projection along affine subspace (HYPASS) algorithm in the sum space. The projection is computable by virtue of Minh's theorem proved in 2010 as long as the input space has nonempty interior. Numerical examples show the efficacy of the proposed algorithm.

Index Terms— reproducing kernel Hilbert space, multikernel adaptive filtering, sum space, orthogonal projection

1. INTRODUCTION

Kernel adaptive filtering has attracted remarkable interests in signal processing, machine learning, and neural networks [1–12]. In its early stage, the sparsification of the *dictionary* was one of the central issues because the expansion length increases unlimitedly as time goes by unlike the case of linear adaptive filters [13, 14]. The sparsification techniques can be classified into the growing and pruning strategies. The growing strategy selectively adds a new datum into the dictionary based on some novelty criterion such as (i) Platt's criterion [15], (ii) approximate linear dependency [2], and (iii) coherence [8] etc. The pruning strategy removes obsolete data from the dictionary, including (i) the simple truncation rule [1], (ii) the fixed budget approaches [4, 5, 11, 12], and (iii) the shrinkage approaches based on ℓ_1 regularization [16–20].

Recently, it has been shown that the use of multiple kernels for online learning yields better performance than the conventional single-kernel approaches [16, 17, 21–24]. Yukawa has proposed *multikernel adaptive filtering* [16, 17, 22]. Its basic algorithm named the multikernel normalized least mean square (MKNLMS) algorithm is a simple extension of the kernel normalized least mean square (KNLMS) algorithm proposed in [8]. Indeed, both KNLMS and MKNLMS project the current estimate onto a zero-instantaneous-error hyperplane in a Euclidean space (a parameter space). The difference is the dimension of the space (the number of parameters). A vector in the Euclidean space for KNLMS consists of expansion coefficients for a single kernel, while that for MKNLMS consists of expansion coefficients for multiple kernels. Tobar, Kung, and Mandic have proposed the multikernel least mean square (MKLMS) algorithm [24] which is closely related to MKNLMS but is applicable to 'vector-valued' functions. Pokharel, Píncipe, and Seth have proposed the mixture kernel least mean square formulation [21]. In this approach, individual nonlinear filters are computed by the LMS algorithm in multiple reproducing kernel Hilbert spaces (RKHSs) simultaneously, and the weights of the combination of the individual filters are learned also in online fashion. Gao, Richard, Bermudez, and Huang have proposed the convex combinations of kernel adaptive filters [23] which is related to the mixture kernel least mean square approach.

*This work was partially supported by KDDI Foundation and JSPS Grants-in-Aid (24760292). The author would like to thank Prof. Johan A. K. Suykens and Dr. Philippe Dreesen for offering the Silverbox data set used in the experiments. Contact Email: yukawa@elec.keio.ac.jp.

Let us turn our attention to *partially linear models* which have been studied considerably in statistics over the last few decades [25]. A partially linear model is defined as a superposition of linear and nonlinear (typically smooth) functions. For batch processing, a significant amount of researches have been done under this model. In particular, partially linear regression has been studied in automatic control with the use of reproducing kernels in [26, 27]. On the other hand, adaptive signal processing under partially linear models would still have plenty of room for investigation. (A convex combination of linear and Gaussian kernels has been used in [28] for nonlinear acoustic echo cancellation; the convex combination coefficients are tuned manually.)

In this paper, we propose an efficient multikernel adaptive filtering algorithm to estimate/track partially linear systems. The proposed algorithm is based on the iterative projections in the sum space of RKHSs associated with linear and Gaussian kernels.[1] The difference from the MKNLMS algorithm is that the projection is operated in a *functional space (the sum space)* rather than a Euclidean space. The algorithm is derived by reformulating the hyperplane projection along affine subspace (HYPASS) algorithm [30] in the sum space. Thanks to Minh's theorem [31], we obtain a closed-form expression of the inner product in the sum space under the practical assumption that the input space has nonempty interior. This allows us to compute the projection in the sum space. We also present a selective updating strategy to reduce the computational costs. Numerical examples show the advantages of the proposed algorithm in performance and complexity for adaptive estimation of the real-life nonlinear dynamical system.

2. SUM SPACE MODEL

We denote by $\mathbb{R}$ and $\mathbb{N}$ the sets of all real numbers and nonnegative integers, respectively. Vectors and matrices are denoted by lower-case and upper-case letters in bold-face, respectively. The identity matrix is denoted by $\boldsymbol{I}$ and the transposition of a vector/matrix is denoted by $(\cdot)^{\mathsf{T}}$. Let $\mathcal{U} \subset \mathbb{R}^L$ be the input space which is assumed to have nonempty interior, and $\mathbb{R}$ the output space.[2] We consider a problem of estimating/tracking a nonlinear unknown function $\psi : \mathcal{U} \to \mathbb{R}$ by means of sequentially arriving input-output measurements. We focus on the so-called *partially linear* case where ψ is given as a superposition of linear and nonlinear smooth functions [25–27]. The linear and Gaussian kernels are presented below among many other cerebrated examples of reproducing kernel [32, 33].

1. Linear kernel: Given $c \geq 0$,

$$\kappa_{\mathrm{L}}(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} + c,\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}.$$

[1] A more general framework is available online [29].

[2] The interior assumption is required for deriving the proposed algorithm through the sum-space formulation, but does *not* restrict its applicability. Indeed, if $\mathcal{U}$ has no interior (or if, more in general, two RKHSs have common elements other than the null vector), essentially the same algorithm can be derived based on the Cartesian product of RKHSs [29].

2. Gaussian kernel: Given $\sigma > 0$,

$$\kappa_{\mathrm{G}}(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_{\mathbb{R}^L}^2}{2\sigma^2}\right),\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U},$$

where $\|\cdot\|_{\mathbb{R}^L}$ denotes the Euclidean norm in $\mathbb{R}^L$.

We denote by $\mathcal{H}_{\mathrm{L}}$ and $\mathcal{H}_{\mathrm{G}}$ the RKHSs, over the input space $\mathcal{U}$, associated with κ_{L} and κ_{G}, respectively. The inner products in $\mathcal{H}_{\mathrm{L}}$ and $\mathcal{H}_{\mathrm{G}}$ are denoted by $\langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{L}}}$ and $\langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathrm{G}}}$, respectively, and the induced norms by $\|\cdot\|_{\mathcal{H}_{\mathrm{L}}}$ and $\|\cdot\|_{\mathcal{H}_{\mathrm{G}}}$, respectively. The partially linear system ψ is well modeled as an element of the sum space

$$\mathcal{H}^+ := \mathcal{H}_{\mathrm{L}} + \mathcal{H}_{\mathrm{G}} := \{f_{\mathrm{L}} + f_{\mathrm{G}} : f_{\mathrm{L}} \in \mathcal{H}_{\mathrm{L}},\ f_{\mathrm{G}} \in \mathcal{H}_{\mathrm{G}}\}.$$

Theorem 1 (Reproducing kernel of sum space $\mathcal{H}^+$ [34]) *If κ_i is the reproducing kernel of the class $\mathcal{H}_i$ with the norm $\|\cdot\|_{\mathcal{H}_i}$, then $\kappa := \kappa_1 + \kappa_2$ is the reproducing kernel of the class $\mathcal{H}^+$ of all functions $f = f_1 + f_2$ with $f_i \in \mathcal{H}_i$, and with the norm defined by*

$$\|f\|_{\mathcal{H}^+}^2 := \min\left\{\|f_1\|_{\mathcal{H}_1}^2 + \|f_2\|_{\mathcal{H}_2}^2 \mid f = f_1 + f_2,\ f_i \in \mathcal{H}_i\right\}. \tag{1}$$

The following theorem proved by Minh allows us to compute the projection in the sum space $\mathcal{H}^+$.

Theorem 2 ([31]) *Let $\mathcal{U} \subset \mathbb{R}^L$ be any set with nonempty interior and $\mathcal{H}_{\mathrm{G}}$ the RKHS associated with a Gaussian kernel $\kappa_{\mathrm{G}}(\boldsymbol{x}, \boldsymbol{y})$ for an arbitrary $\sigma > 0$ together with the input space $\mathcal{U}$. Then, and $\mathcal{H}_{\mathrm{G}}$ does not contain any polynomial on $\mathcal{U}$, including the nonzero constant function.*

The following corollary is obtained as a direct consequence of Theorem 2.

Corollary 1 (Linear and Gaussian RKHSs) *Assume that the input space $\mathcal{U}$ has nonempty interior. Then,*

$$\mathcal{H}_{\mathrm{L}} \cap \mathcal{H}_{\mathrm{G}} = \{0\}, \tag{2}$$

and thus (1) *is reduced to* [34]

$$\|f\|_{\mathcal{H}^+}^2 = \|f_{\mathrm{L}}\|_{\mathcal{H}_{\mathrm{L}}}^2 + \|f_{\mathrm{G}}\|_{\mathcal{H}_{\mathrm{G}}}^2. \tag{3}$$

The inner product between $f = f_{\mathrm{L}} + f_{\mathrm{G}} \in \mathcal{H}^+$ and $g = g_{\mathrm{L}} + g_{\mathrm{G}} \in \mathcal{H}^+$ is given by

$$\langle f, g\rangle_{\mathcal{H}^+} := \langle f_{\mathrm{L}}, g_{\mathrm{L}}\rangle_{\mathcal{H}_{\mathrm{L}}} + \langle f_{\mathrm{G}}, g_{\mathrm{G}}\rangle_{\mathcal{H}_{\mathrm{G}}}. \tag{4}$$

Theorem 3 *Let $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ be the reproducing kernel of a real Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$. Then, given an arbitrary $w > 0$, $\kappa_w(\boldsymbol{u}, \boldsymbol{v}) := w\kappa(\boldsymbol{u}, \boldsymbol{v})$, $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{U}$, is the reproducing kernel of the RKHS $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H},w})$ with the inner product $\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\mathcal{H},w} := w^{-1}\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\mathcal{H}}$, $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{U}$.*

Proof: See [29].

The following holds directly by Theorems 1-3 and Corollary 1.

Corollary 2 (Weighted norm and reproducing kernel) *Given any $w_{\mathrm{L}}, w_{\mathrm{G}} > 0$, $\kappa_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v}) := w_{\mathrm{L}}\kappa_{\mathrm{L}}(\boldsymbol{u}, \boldsymbol{v}) + w_{\mathrm{G}}\kappa_{\mathrm{G}}(\boldsymbol{u}, \boldsymbol{v})$, $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{U}$, is the reproducing kernel of the sum space $\mathcal{H}^+$ equipped with the inner product*

$$\langle f, g\rangle_{\mathcal{H}^+,\boldsymbol{w}} := w_{\mathrm{L}}^{-1}\langle f_{\mathrm{L}}, g_{\mathrm{L}}\rangle_{\mathcal{H}_{\mathrm{L}}} + w_{\mathrm{G}}^{-1}\langle f_{\mathrm{G}}, g_{\mathrm{G}}\rangle_{\mathcal{H}_{\mathrm{G}}}. \tag{5}$$

The induced norm is given by

$$\|f\|_{\mathcal{H}^+,\boldsymbol{w}}^2 = w_{\mathrm{L}}^{-1}\|f_{\mathrm{L}}\|_{\mathcal{H}_{\mathrm{L}}}^2 + w_{\mathrm{G}}^{-1}\|f_{\mathrm{G}}\|_{\mathcal{H}_{\mathrm{G}}}^2. \tag{6}$$

Without loss of generality, we let $w_{\mathrm{L}} = w_{\mathrm{G}} = 1$ in Section 3.

3. ONLINE LEARNING IN SUM SPACE $\mathcal{H}^+$

3.1. Dictionary Design

Due to the interior assumption on the input space $\mathcal{U}$, it is seen that the dimension of $\mathcal{H}_{\mathrm{L}}$ is $L+1$. It is clear that $\kappa_{\mathrm{L}}(\cdot, \boldsymbol{0}) = c$ and $\kappa_{\mathrm{L}}(\cdot, \boldsymbol{e}_j) - \kappa_{\mathrm{L}}(\cdot, \boldsymbol{0}) = \boldsymbol{e}_j^{\mathsf{T}}(\cdot)$, where $\boldsymbol{e}_j \in \mathbb{R}^L$ is the unit vector having one at the jth entry and zeros elsewhere. Based on this observation, one can see that

$$\mathcal{D}_{\mathrm{L}} := \{\kappa_{\mathrm{L}}(\cdot, \boldsymbol{e}_j) - \kappa_{\mathrm{L}}(\cdot, \boldsymbol{0})\}_{j=1}^L \cup \{\kappa_{\mathrm{L}}(\cdot, \boldsymbol{0})\} \tag{7}$$

gives a basis of the $L+1$ dimensional space $\mathcal{H}_{\mathrm{L}}$. A typical choice for the parameter c of the linear kernel is $c = 1$. If one knows that the linear component of ψ is zero-passing, one can simply let $c = 0$ and remove $\{\kappa_{\mathrm{L}}(\cdot, \boldsymbol{0})\}$ from the dictionary.

The dictionary $\mathcal{D}_{\mathrm{G},n}$ for the Gaussian kernel should be *time-dependent* in general and needs to be constructed in online fashion. A growing strategy is given as follows: (i) start with $\mathcal{D}_{\mathrm{G},-1} := \emptyset$, and (ii) add a new candidate $\kappa_{\mathrm{G}}(\cdot, \boldsymbol{u}_n)$ into the dictionary at each time $n \in \mathbb{N}$ only when it is sufficiently novel. In this case, the dictionary can be expressed as $\mathcal{D}_{\mathrm{G},n} = \{\kappa_{\mathrm{G}}(\cdot, \boldsymbol{u}_j)\}_{j\in\mathcal{J}_n}$ for some $\mathcal{J}_n := \{j_1^{(n)}, j_2^{(n)}, \cdots, j_{r_n}^{(n)}\} \subset \{0, 1, 2, \cdots, n\}$, where r_n is the size of the Gaussian dictionary. Our novelty criterion is based on Platt's criterion [15]: $\kappa_{\mathrm{G}}(\cdot, \boldsymbol{u}_n)$ is regarded to be novel if $\max_{\boldsymbol{u}\in\mathcal{D}_{\mathrm{G},n}} \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{u}_n\|_{\mathbb{R}^L}^2}{2\sigma^2}\right) < \delta$ for some prespecified threshold $\delta \in (0, 1)$ and if $|d_n - \varphi_n(\boldsymbol{u}_n)|^2 > \varepsilon\,|\varphi_n(\boldsymbol{u}_n)|^2$ for some prespecified error bound $\varepsilon > 0$. Here, $\varphi_n(\boldsymbol{u}_n)$ is the nonlinear filter output for the input vector $\boldsymbol{u}_n$; it is described better in Section 3.2. In the present study, we consider no pruning strategy for clarity of presentation; in practice, one may adopt some pruning strategy (see, e.g., [1, 4, 5, 11, 12, 16–20]).

3.2. Adaptive Learning Algorithm

Our nonlinear adaptive filter takes the following form:

$$\varphi_n := \varphi_{\mathrm{L},n} + \varphi_{\mathrm{G},n} \in \mathcal{M}_{n-1}^+,\ n \in \mathbb{N}, \tag{8}$$

where $\varphi_{\mathrm{L},n} \in \mathcal{H}_{\mathrm{L}}$, $\varphi_{\mathrm{G},n} \in \mathcal{M}_{\mathrm{G},n-1} := \mathrm{span}\mathcal{D}_{\mathrm{G},n-1} \subset \mathcal{H}_{\mathrm{G}}$, and

$$\mathcal{M}_n^+ := \mathcal{H}_{\mathrm{L}} + \mathcal{M}_{\mathrm{G},n} \subset \mathcal{H}^+, \quad n \in \mathbb{N}. \tag{9}$$

We assume that

$$\varphi_n \in \mathcal{M}_n^+ \cap \mathcal{M}_{n-1}^+, \tag{10}$$

which means that 'active' elements in $\mathcal{D}_{\mathrm{G},n-1}$ remain in the updated dictionary $\mathcal{D}_{\mathrm{G},n}$. At every time instant $n \in \mathbb{N}$, a new measurement $\boldsymbol{u}_n$ and d_n arrives, and φ_n is updated to $\varphi_{n+1} \in \mathcal{M}_n^+$ based on the new measurement. We define the following subset of the dictionary subspace $\mathcal{M}_n^+$:

$$\Pi_n := \left\{f \in \mathcal{M}_n^+ \mid f(\boldsymbol{u}_n) = \langle f, \kappa(\cdot, \boldsymbol{u}_n)\rangle_{\mathcal{H}^+} = d_n\right\},$$

which contains those vectors which nullify the instantaneous error. Note here that Π_n can also be represented as

$$\Pi_n := \mathcal{M}_n^+ \cap \Pi_{n,\mathcal{H}^+}, \tag{11}$$

where $\Pi_{n,\mathcal{H}^+} := \left\{f \in \mathcal{H}^+ \mid f(\boldsymbol{u}_n) = \langle f, \kappa(\cdot, \boldsymbol{u}_n)\rangle_{\mathcal{H}^+} = d_n\right\}$ is a hyperplane in the whole space $\mathcal{H}^+$. For an initial filter $\varphi_0 \in \mathcal{H}^+$ ($\varphi_0 := 0$ without any a priori information), our kernel adaptive filter is updated by the following simple recursion:

$$\varphi_{n+1} := \varphi_n + \lambda_n(P_{\Pi_n}(\varphi_n) - \varphi_n) \in \mathcal{M}_n^+,\ n \in \mathbb{N}, \tag{12}$$

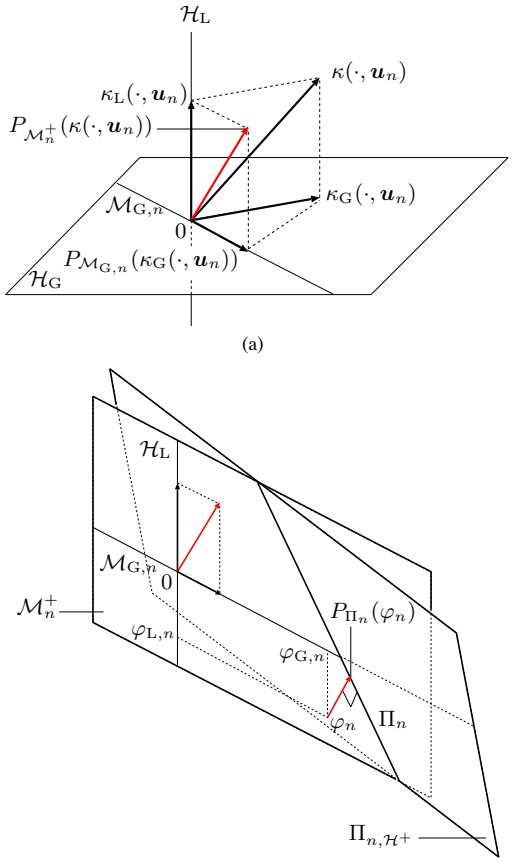

Fig. 1. A geometric interpretation of $P_{\mathcal{M}_n^+}(\kappa(\cdot,\boldsymbol{u}_n))$ and $P_{\Pi_n}(\varphi_n)$.

where $\lambda_n \in (0,2)$. Here, $P_{\Pi_n}(\varphi_n) := \operatorname{argmin}_{f\in\Pi_n}\|f-\varphi_n\|_{\mathcal{H}^+}$ is the orthogonal projection of φ_n onto Π_n [35] and can be computed as follows [29, Proposition 1 and Lemma 1]:

$$P_{\Pi_n}(\varphi_n) = \varphi_n + \frac{d_n - \varphi_n(\boldsymbol{u}_n)}{\left\|P_{\mathcal{M}_n^+}(\kappa(\cdot,\boldsymbol{u}_n))\right\|^2_{\mathcal{H}^+}} P_{\mathcal{M}_n^+}(\kappa(\cdot,\boldsymbol{u}_n)). \quad (13)$$

Here,

$$P_{\mathcal{M}_n^+}(\kappa(\cdot,\boldsymbol{u}_n)) = \kappa_{\mathrm{L}}(\cdot,\boldsymbol{u}_n) + P_{\mathcal{M}_{\mathrm{G},n}}(\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_n)), \quad (14)$$

where

$$P_{\mathcal{M}_{\mathrm{G},n}}(\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_n)) = \sum_{j\in\mathcal{J}_n}\alpha_j\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_j) \quad (15)$$

with $\boldsymbol{\alpha} := \left[\alpha_{j_1^{(n)}}, \alpha_{j_2^{(n)}}\cdots,\alpha_{j_{r_n}^{(n)}}\right]^{\mathsf{T}}$ satisfying the following normal equation: $\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{b}$. Here, $\boldsymbol{K} \in \mathbb{R}^{r_n\times r_n}$ is the Gram matrix (also called the kernel matrix) whose (p,q) entry is $K_{pq} := \kappa_{\mathrm{G}}\left(\boldsymbol{u}_{j_p^{(n)}}, \boldsymbol{u}_{j_q^{(n)}}\right)$ and $\boldsymbol{b} := \left[\kappa_{\mathrm{G}}\left(\boldsymbol{u}_{j_1^{(n)}},\boldsymbol{u}_n\right), \kappa_{\mathrm{G}}\left(\boldsymbol{u}_{j_2^{(n)}},\boldsymbol{u}_n\right), \cdots, \kappa_{\mathrm{G}}\left(\boldsymbol{u}_{j_{r_n}^{(n)}},\boldsymbol{u}_n\right)\right]^{\mathsf{T}} \in \mathbb{R}^{r_n}$. Fig. 1(a) presents a geometric interpretation of $P_{\mathcal{M}_n^+}(\kappa(\cdot,\boldsymbol{u}_n))$ in (14), and Fig. 1(b) presents that of $P_{\Pi_n}(\varphi_n)$ in (13); see also (8), (9), and (11).

3.3. Complexity Issue and Practical Remedy

The computation of $P_{\mathcal{M}_{\mathrm{G},n}}(\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_n))$ in (15) would involve the inversion of the $r_n \times r_n$ kernel matrix $\boldsymbol{K}$ (if invertible) as well as the multiplication of the inverse matrix with a vector. A practical remedy to reduce the complexity is the selective update: modify the hyperplane Π_n into

$$\tilde{\Pi}_n := \left\{f \in \mathcal{V}_n^+ \mid f(\boldsymbol{u}_n) = \langle f, \kappa(\cdot,\boldsymbol{u}_n)\rangle_{\mathcal{H}^+} = d_n\right\}, \quad (16)$$

where $\mathcal{V}_n^+ := \mathcal{H}_{\mathrm{L}} + \mathcal{V}_{\mathrm{G},n}$ with

$$\mathcal{V}_{\mathrm{G},n} := \tilde{\mathcal{M}}_{\mathrm{G},n} + \varphi_{\mathrm{G},n} := \operatorname{span}\tilde{\mathcal{D}}_{\mathrm{G},n} + \varphi_{\mathrm{G},n} \subseteq \mathcal{M}_{\mathrm{G},n}. \quad (17)$$

Here, $\tilde{\mathcal{D}}_{\mathrm{G},n} := \{\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_j)\}_{j\in\tilde{\mathcal{J}}_n}$ for $\tilde{\mathcal{J}}_n := \{\iota_1^{(n)}, \iota_2^{(n)}, \cdots, \iota_{s_n}^{(n)}\} \subset \mathcal{J}_n$ is a size-s_n selected subset of the dictionary $\mathcal{D}_{\mathrm{G},n}$ ($s_n \le r_n$), containing a few $\kappa_{\mathrm{G},n}(\cdot,\boldsymbol{u}_j)$s in $\mathcal{D}_{\mathrm{G},n}$ that are most coherent to $\kappa_{\mathrm{G},n}(\cdot,\boldsymbol{u}_n)$. More precisely, choose $\tilde{\mathcal{J}}_n$ so that $\kappa_{\mathrm{G},n}(\boldsymbol{u}_\iota,\boldsymbol{u}_n) \ge \kappa_{\mathrm{G},n}(\boldsymbol{u}_j,\boldsymbol{u}_n)$ for any $\iota \in \tilde{\mathcal{J}}_n$, $j \in \mathcal{J}_n \setminus \tilde{\mathcal{J}}_n$ [30], or equivalently, $\|\boldsymbol{u}_\iota - \boldsymbol{u}_n\|_{\mathbb{R}^L} \le \|\boldsymbol{u}_j - \boldsymbol{u}_n\|_{\mathbb{R}^L}$; i.e., collect s_n neighbors of $\boldsymbol{u}_n$. The validity of this selection strategy will be shown in Section 4. The update equation (12) is modified into

$$\varphi_{n+1} := \varphi_n + \lambda_n(P_{\tilde{\Pi}_n}(\varphi_n) - \varphi_n) \in \mathcal{M}_n^+,\ n \in \mathbb{N}, \quad (18)$$

where $\lambda_n \in (0,2)$. The algorithm in (18) is a sum-space extension of the HYPASS algorithm proposed in [30], and is a particular case of the Cartesian HYPASS (CHYPASS) algorithm derived in [29] through the product-space formulation.[3] The projection $P_{\tilde{\Pi}_n}(\varphi_n)$ in (18) can be computed as

$$P_{\tilde{\Pi}_n}(\varphi_n) = \varphi_n + \frac{d_n - \varphi_n(\boldsymbol{u}_n)}{\left\|P_{\tilde{\mathcal{M}}_n^+}(\kappa(\cdot,\boldsymbol{u}_n))\right\|^2_{\mathcal{H}^+}} P_{\tilde{\mathcal{M}}_n^+}(\kappa(\cdot,\boldsymbol{u}_n)), \quad (19)$$

where $\tilde{\mathcal{M}}_n^+ := \mathcal{H}_{\mathrm{L}} + \tilde{\mathcal{M}}_{\mathrm{G},n}$. We mention here that $\mathcal{V}_n^+ = \tilde{\mathcal{M}}_n^+ + \varphi_{\mathrm{G},n}$. Finally, $P_{\tilde{\mathcal{M}}_n^+}(\kappa(\cdot,\boldsymbol{u}_n))$ in (19) is given by

$$P_{\tilde{\mathcal{M}}_n^+}(\kappa(\cdot,\boldsymbol{u}_n)) = \kappa_{\mathrm{L}}(\cdot,\boldsymbol{u}_n) + P_{\tilde{\mathcal{M}}_{\mathrm{G},n}}(\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_n)) \quad (20)$$

with

$$P_{\tilde{\mathcal{M}}_{\mathrm{G},n}}(\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_n)) = \sum_{\iota\in\tilde{\mathcal{J}}_n}\beta_\iota\kappa_{\mathrm{G}}(\cdot,\boldsymbol{u}_\iota), \quad (21)$$

where $\boldsymbol{\beta} := \left[\beta_{\iota_1^{(n)}}, \beta_{\iota_2^{(n)}}\cdots,\beta_{\iota_{s_n}^{(n)}}\right]^{\mathsf{T}} \in \mathbb{R}^{s_n}$ satisfies $\tilde{\boldsymbol{K}}\boldsymbol{\beta} = \tilde{\boldsymbol{b}}$ with an $s_n \times s_n$ Gram matrix $\tilde{\boldsymbol{K}}$ whose (p,q) entry is given by $\tilde{K}_{pq} := \kappa_{\mathrm{G}}\left(\boldsymbol{u}_{\iota_p^{(n)}}, \boldsymbol{u}_{\iota_q^{(n)}}\right)$ and a length s_n vector $\tilde{\boldsymbol{b}} := \left[\kappa_{\mathrm{G}}\left(\boldsymbol{u}_{\iota_1^{(n)}},\boldsymbol{u}_n\right), \kappa_{\mathrm{G}}\left(\boldsymbol{u}_{\iota_2^{(n)}},\boldsymbol{u}_n\right), \cdots, \kappa_{\mathrm{G}}\left(\boldsymbol{u}_{\iota_{s_n}^{(n)}},\boldsymbol{u}_n\right)\right]^{\mathsf{T}}$.

Under the use of linear and Gaussian kernels (κ_{L} and κ_{G}) together with the novelty criterion described in Section 3.1, the complexities of the proposed algorithm and MKNLMS [17] are both linear in the dictionary size r_n. To be specific, the complexity of MKNLMS is $(L+5)r_n + 3L + \min\{L, r_n\} + 6$ and that of the proposed algorithm is $(L+3)r_n + 3L + O(s_n^3)$. Here, complexity means the total number of multiplications, divisions, and comparisons for updating the coefficients and dictionary at each iteration; L is the dimension of the input space $\mathcal{U}$; and $O(s_n^3)$ is for the inversion of the matrix $\tilde{\boldsymbol{K}}$ which is small since $s_n \le 5$ typically.

[3] In the present case, the sum space $\mathcal{H}_+$ is isomorphic to the Cartesian product of $\mathcal{H}_{\mathrm{L}}$ and $\mathcal{H}_{\mathrm{G}}$, and therefore the algorithms obtained through the sum-space and product-space formulations are the same essentially.

Table 1. Parameter settings and complexities for the experiment.

	parameter		Test MSE [dB] (complexity)
KNLMS	$\lambda_n = 0.5$	$\delta = 0.92$	-32.4 (1729)
HYPASS	$\sigma = 0.5$	$\delta = 0.92, s_n = 5$	-43.1 (1524)
MKNLMS	$\varepsilon = 0.01$	$\delta = 0.9, c = 1$ $w_{\mathrm{L}} = 0.8, w_{\mathrm{G}} = 0.2$	-64.7 (1278)
CHYPASS		$\delta = 0.9, c = 1$ $w_{\mathrm{L}} = 0.8, w_{\mathrm{G}} = 0.2$ $s_n = 5$	-66.4 (1155)

4. NUMERICAL EXAMPLES

We show the efficacy of the proposed algorithm (CHYPASS) for adaptive estimation of the real-life nonlinear dynamical system lying in the Silverbox data set[4] shown in Fig. 2. The input signal is divided by the maximum value so that its maximum is normalized to one. The 80,000 samples after the 40,000th sample (after the 'head of the arrow') are used for learning. The first 40,000 samples are used as test data. We let $\boldsymbol{u}_n := [x_n, x_{n-1}, x_{n-2}, y_{n-1}, y_{n-2}]^{\mathsf{T}}$ (i.e., $L = 5$) and $d_n := y_n, n = 0, 1, 2, \cdots$, where $x_n \in \mathbb{R}$ and $y_n \in \mathbb{R}$ denotes the input and output signals, respectively (x_0 and y_0 corresponds to the $40,001$th samples of the input and output, respectively). We compare the proposed algorithm with MKNLMS [17], HYPASS [30], and KNLMS [8]. Here, the same linear and Gaussian kernels are employed for MKNLMS. For the design of Gaussian-kernel dictionaries, we employ the same strategy as described in Section 3.1 for all the algorithms. Table 1 lists the set of parameters used in the experiment for each algorithm as well as the Test-MSE/complexities averaged over samples/iterations. For CHYPASS, $c = 1$ is a default choice and the weights can be designed as $w_{\mathrm{L}} + w_{\mathrm{G}} = 1$. Note that the threshold δ for the single kernel methods (KNLMS and HYPASS) is set to a slightly larger value than that for the multikernel methods (MKNLMS and CHYPASS) for demonstrating the efficiency of the multikernel methods.

The results are plotted in Fig. 3. Here, "Dictionary size" for CHYPASS and MKNLMS means an arithmetic average of $|\mathcal{D}_{\mathrm{L}}| + |\mathcal{D}_{\mathrm{G},n}|$ $(= r_n + L + 1)$. The average dictionary size was: KNLMS 172.7, HYPASS 170.2, MKNLMS 125.4, and CHYPASS 121.4. One can observe that (i) CHYPASS significantly outperforms the other algorithms with the lowest complexity, and (ii) the multikernel methods attain lower training/test MSEs with a smaller dictionary size. We emphasize here that the low MSEs of CHYPASS and MKNLMS are due to the exploitation of the partial linearity of the dynamical system and the low complexity of CHYPASS is due to the selective updating strategy presented in Section 3.3.

5. CONCLUDING REMARKS

We have proposed an efficient multikernel adaptive filtering algorithm based on the iterative projections in the sum space of linear and Gaussian RKHSs. The major difference from the existing multikernel adaptive filtering algorithms is that the projection is operated in a *functional space*. The algorithm has been derived by reformulating the HYPASS algorithm in the sum space. The selective updating strategy has also been presented to reduce the complexity. The numerical examples have demonstrated that the proposed algorithm attains better performance with lower complexity than KNLMS, HYPASS, and MKNLMS for the real-life partially-linear dynamical system. We finally remark that the functional-space-projection methods (CHYPASS and HYPASS) have outperformed their Euclidean-space-projection counterparts (MKNLMS and KNLMS) in the present experimental results. Our recent research suggests that this is because the functional-space-projection methods would have a *decorrelation property*, but further investigations would be required to verify this empirical finding.

[4]The data was used in the NOLCOS 2004 Special Session benchmark [26].

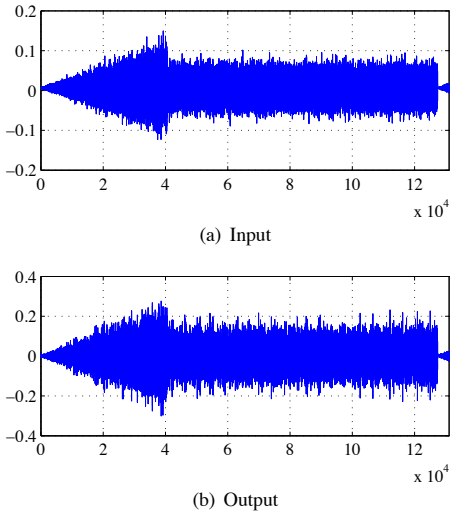

Fig. 2. Signal for the Silverbox data set.

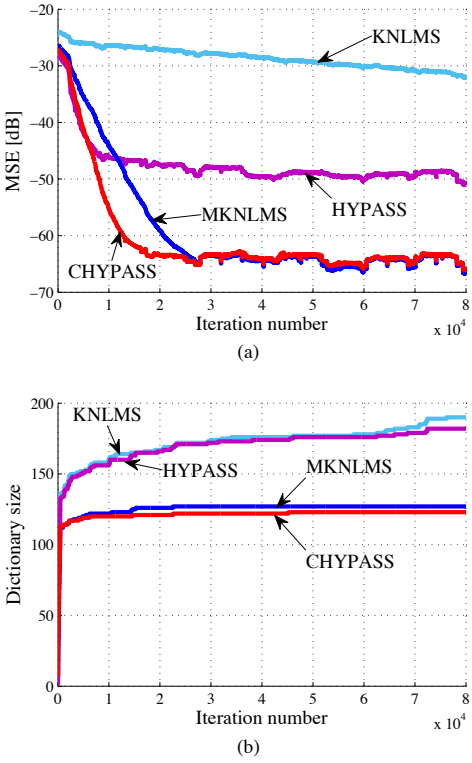

Fig. 3. (a) Learning curves and (b) evolution of dictionary size.

6. REFERENCES

[1] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[2] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[3] A. V. Malipatil, Y.-F. Huang, S. Andra, and K. Bennett, "Kernelized set-membership approach to nonlinear adaptive filtering," in *Proc. IEEE ICASSP*, 2005, pp. 149–152.

[4] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Tracking the best hyperplane with a simple budget perceptron," *Journal of Machine Learning Research*, vol. 69, no. 2-3, pp. 143–167, 2007.

[5] F. Orabona, J. Keshet, and B. Caputo, "The projectron: A bounded kernel-based perceptron," in *Proc. ICML*, 2008, pp. 720–727.

[6] W. Liu and J. Príncipe, "Kernel affine projection algorithms," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–12, 2008, Article ID 784292.

[7] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, July 2008.

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

[9] W. Liu, J. Príncipe, and S. Haykin, *Kernel Adaptive Filtering*, Wiley, New Jersey, 2010.

[10] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections: a unifying framework for linear and nonlinear classification and regression tasks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[11] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, Aug. 2012.

[12] P. Zhao, J. Wang, P. Wu, R. Jin, and S. C. H. Hoi, "Fast bounded online gradient descent algorithms for scalable kernel-based online learning," in *Proc. ICML*, 2012.

[13] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 4th edition, 2002.

[14] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, New Jersey, 2003.

[15] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, 1991.

[16] M. Yukawa, "Nonlinear adaptive filtering techniques with multiple kernels," in *European Signal Processing Conference (EUSIPCO)*, 2011, pp. 136–140.

[17] M. Yukawa, "Multikernel adaptive filtering," *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4672–4682, Sept. 2012.

[18] B. Chen, S. Zhao, S. Seth, and J. C. Príncipe, "Online efficient learning with quantized KLMS and l_1 regularization," in *Int. Joint Conf. Neural Netw.*, 2012.

[19] W. Gao, J. Chen, C. Richard, and J. Huang, "Online dictionary learning for kernel LMS," *IEEE Trans. Signal Processing*, vol. 62, no. 11, pp. 2765–2777, Jun. 2014.

[20] M. Takizawa and M. Yukawa, "An efficient sparse kernel adaptive filtering algorithm based on isomorphism between functional subspace and Euclidean space," in *Proc. IEEE ICASSP*, 2014, pp. 4541–4545.

[21] R. Pokharel, J. Príncipe, and S. Seth, "Mixture kernel least mean square," in *IEEE IJCNN*, 2013.

[22] M. Yukawa and R. Ishii, "Online model selection and learning by multikernel adaptive filtering," in *Proc. EUSIPCO*, 2013.

[23] W. Gao, C. Richard, J.-C. M. Bermudez, and J. Huang, "Convex combinations of kernel adaptive filters," in *IEEE Int. Workshop on MLSP*, 2014.

[24] F. A. Tobar, S.-Y. Kung, and D. P. Mandic, "Multikernel least mean square algorithm," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 265–277, Feb. 2014.

[25] W. Härdle, H. Liang, and J. Gao, *Partially Linear Models*, Physica-Verlag, Heidelberg, Germany, 2000.

[26] M. Espinoza, J. A. K. Suykens, and B. D. Moor, "Kernel based partially linear models and nonlinear identification," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1602–1606, Oct. 2005.

[27] Y.-L. Xu and D.-R. Chen, "Partially-linear least-squares regularized regression for system identification," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2637–2641, Nov. 2009.

[28] J. M. Gil-Cacho, M. Signoretto, T. van Waterschoot, M. Moonen, and S. H. Jensen, "Nonlinear acoustic echo cancellation based on a sliding-window leaky kernel affine projection algorithm," *IEEE Trans. Audio, Speech and Language Processing*, vol. 21, no. 9, pp. 1867–1878, Sept. 2013.

[29] M. Yukawa, "Adaptive learning in Cartesian product of reproducing kernel Hilbert spaces," 2014, submitted for publication (available online arXiv:1408.0853).

[30] M. Yukawa and R. Ishii, "An efficient kernel adaptive filtering algorithm using hyperplane projection along affine subspace," in *Proc. EUSIPCO*, 2012, pp. 2183–2187.

[31] H. Q. Minh, "Some properties of Gaussian reproducing kernel Hilbert spaces and their implications for function approximation and learning theory," *Constr. Approx.*, vol. 32, no. 2, pp. 307–338, Oct. 2010.

[32] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2001.

[33] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic, New York, 4th edition, 2008.

[34] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, May 1950.

[35] D. G. Luenberger, *Optimization by Vector Space Methods*, New York: Wiley, 1969.