AN EFFICIENT DATA-REUSING KERNEL ADAPTIVE FILTERING ALGORITHM BASED ON PARALLEL HYPERSLAB PROJECTION ALONG AFFINE SUBSPACES

Masa-aki Takizawa and Masahiro Yukawa

Dept. Electrical and Electronic Engineering, Niigata University, Japan

ABSTRACT

We propose a novel kernel adaptive filtering algorithm, dubbed Parallel HYperslab Projection along Affine Sub-Spaces (Φ-PASS), which reuses observed data efficiently. We first derive its fully-updating version that projects the current filter onto multiple hyperslabs in parallel along the dictionary subspace. Each hyperslab accommodates one of the data observed up to the present time instant. The algorithm is derived with the adaptive projected subgradient method (APSM) based on which a convergence analysis is presented. We then generalize the algorithm so that only a few coefficients, whose associated dictionary-data are coherent to the datum of each hyperslab, can be updated selectively for low complexity. This is accomplished by performing the hyperslab projections along affine subspaces defined with the selected dictionary-data. Numerical examples show the efficacy of the proposed algorithm.

Index Terms— kernel adaptive filter, projection algorithms, reproducing kernel Hilbert space, the HYPASS algorithm

1. INTRODUCTION

The power of the theory of reproducing kernel has been proven by the success of support vector machine and its related techniques since 1990's. In the present century, a considerable amount of attention has been paid to *online* algorithms which is of great importance in signal processing; see [1–6] among many others. The notable advantages of a kernel adaptive filter include that it involves no local minima unlike the neural network approach, and a marginal number of parameters unlike a Volterra filter [6]. The design of kernel is a challenging task in online/adaptive scenarios and it has been firstly addressed in [7]. In the present work, an adequate kernel is assumed available, as is also assumed in the literature [1–6]. Relation to prior work will be discussed in Section 3.

The goal of this paper is to develop a computationally-efficient nonlinear adaptive filtering algorithm enjoying high adaptation-capability. We propose a data-reusing kernel adaptive filtering algorithm based on the parallel projection onto multiple hyperslabs along certain affine subspaces in a reproducing kernel Hilbert space (RKHS). Each hyperslab accommodates a pair of input-output data and bounds the instantaneous error for the data pair. By performing the hyperslab projections in parallel, the filter is updated in such a way that the errors for multiple data pairs are suppressed simultaneously, hence enhancing the convergence and tracking performance. We employ the parallel projection technique [8, 9] for data-reusing rather than the affine projection algorithm (APA) which has been reported to suffer from noise sensitivity [8]. We first derive a data-reusing algorithm, dubbed *fully-updating parallel hyperslab projection along affine subspaces (Φ-PASS)*, and present its convergence analysis by using the adaptive projected subgradient method (APSM).

The hyperslab projections are performed along the dictionary subspace for exploiting all measurements to polish the coefficients of dictionary data (see [10] for more details about this idea). We then generalize the algorithm so that each hyperslab projection can be performed along an *affine* subspace which contains the current filter and whose underlying subspace is spanned by an arbitrary number of dictionary element(s). The generalized algorithm is referred to simply as *the Φ-PASS algorithm*. If a single dictionary-datum is selected that is maximally coherent to the datum of each hyperslab, only a few coefficients are updated at each iteration. This coherence-based selective-update strategy brings a remarkably low complexity while taking significant benefits from data-reusing. The numerical examples show clear advantages of the proposed algorithm over the conventional algorithms.

2. FULLY-UPDATING Φ-PASS ALGORITHM

Let ψ be a nonlinear system of which the input $\boldsymbol{u}$ belongs to the compact input space $\mathcal{U} \subset \mathbb{R}^L$ and the output $d := \psi(\boldsymbol{u})$ takes a real value. We consider online scenarios in which the data sequences $(\boldsymbol{u}_n)_{n\in\mathbb{N}} \subset \mathcal{U}$ and $(d_n)_{n\in\mathbb{N}} \subset \mathbb{R}$ arrive sequentially. We model the nonlinear function ψ to be estimated as an element of the RKHS $\mathcal{H}$ associated with a positive definite kernel $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R},\ (\boldsymbol{x}, \boldsymbol{y}) \mapsto \kappa(\boldsymbol{x}, \boldsymbol{y})$ [11]. We denote by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ the inner product and the norm defined in $\mathcal{H}$, respectively. A popular example of positive definite kernel is a Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\zeta \|\boldsymbol{x} - \boldsymbol{y}\|^2_{\mathbb{R}^L}), \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$, for a kernel parameter $\zeta > 0$, where $\|\cdot\|_{\mathbb{N}_L}$ stands for the Euclidian norm. A kernel adaptive filter is given by the following expansion:

$$\varphi_n(\boldsymbol{u}) = \sum_{j\in\mathcal{J}_n} h_{j,n}\kappa\left(\boldsymbol{u}, \boldsymbol{u}_j\right),\ n \in \mathbb{N}, \tag{1}$$

where $\mathcal{J}_n := \{j_1^{(n)}, j_2^{(n)}, \cdots, j_{r_n}^{(n)}\} \subset \{0, 1, \cdots, n\}$ is the dictionary index set and $h_{j,n} \in \mathbb{R}$ is the coefficient of $\kappa(\cdot, \boldsymbol{u}_j)$ at time n. The set of r_n functions $\{\kappa\left(\cdot, \boldsymbol{u}_j\right)\}_{j\in\mathcal{J}_n}$ is called a *dictionary*.

The key of the proposed algorithm is *data-reusing*. Namely, each pair of data $(\boldsymbol{u}_n, d_n)$ is exploited to polish the coefficients at multiple time instants. The set of indices indicating the set of data exploited at time $n \in \mathbb{N}$ is denoted by $\mathcal{I}_n \subset \{n, n-1, \cdots, 0\}$. A simple strategy is to use the p most recent input-output pairs $\left(\boldsymbol{u}_n, d_n\right), \left(\boldsymbol{u}_{n-1}, d_{n-1}\right), \cdots, \left(\boldsymbol{u}_{n-p+1}, d_{n-p+1}\right)$, i.e., $\mathcal{I}_n := \{n, n-1, \cdots, n-p+1\}$, and φ_n is updated in such a way that estimation errors for

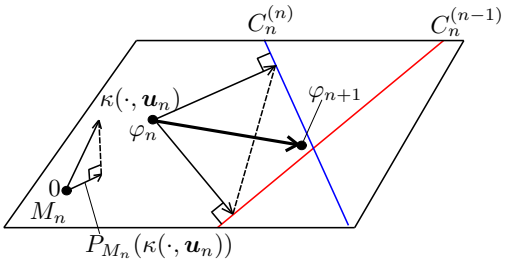

Fig. 1. A geometric interpretation of the fully-updating Φ-PASS algorithm for $\rho = 0$.

those data diminish. Specifically, we employ the metric projections of the current filter φ_n onto the convex sets:

$$C_\iota^{(n)} := \{g \in M_n : (g(\boldsymbol{u}_\iota) - d_\iota)^2 \le \rho\} = S_\iota \cap M_n, \quad n \in \mathbb{N},\ \iota \in \mathcal{I}_n, \quad (2)$$

where $\rho \ge 0$ is the error bound and

$$S_\iota := \{g \in \mathcal{H} : (g(\boldsymbol{u}_\iota) - d_\iota)^2 \le \rho\} \quad \text{(bounded error)},$$
$$M_n := \operatorname{span}\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j \in \mathcal{J}_n} \quad \text{(dictionary subspace)}.$$

Here, S_ι is the bounded-instantaneous-error hyperslabs, and M_n the subspace spanned by the dictionary elements. Since $\varphi_n \in M_n$ from (1), the projection of φ_n onto the convex set $C_\iota^{(n)} = S_\iota \cap M_n$ can be regarded as the projection of φ_n onto the hyperslab S_ι along the subspace M_n (see Fig. 1). Here, in general, the projection of $x \in \mathcal{H}$ onto a nonempty closed convex set $K \subset \mathcal{H}$ is the closest point of x in K and is denoted by $P_K(x) := \operatorname{argmin}_{y \in K} \|x - y\|$.

Now we derive the proposed algorithm which updates the current filter φ_n by projecting it onto the convex sets $C_\iota^{(n)}$, $\iota \in \mathcal{I}_n$, in parallel and then convexly combining the projections (see [9] for a comprehensive tutorial on this parallel-projection strategy). Define the sequence of convex functions $(\Theta_n)_{n \in \mathbb{N}}$ as follows:

$$\Theta_n(g) := \sum_{\iota \in \mathcal{I}_n} \frac{\omega_\iota^{(n)} d(\varphi_n, C_\iota^{(n)})}{\nu_n} d(g, C_\iota^{(n)}),\ g \in \mathcal{H}, \quad (3)$$

where $d(g, C_\iota^{(n)}) := \min_{\phi \in C_\iota^{(n)}} \|g - \phi\|$ is the metric distance function between a point g and the set $C_\iota^{(n)}$, $\nu_n := \sum_{\iota \in \mathcal{I}_n} \omega_\iota^{(n)} d(\varphi_n, C_\iota^{(n)})$, and $\omega_\iota^{(n)} > 0, \forall \iota \in \mathcal{I}_n$, satisfying $\sum_{\iota \in \mathcal{I}_n} \omega_\iota^{(n)} = 1$, is the weight assigned to each set $C_\iota^{(n)}$. The additional weight $d(\varphi_n, C_\iota^{(n)})$ emphasizes those sets which are more distant from the current filter φ_n than the other sets. An application of APSM to the function sequence $(\Theta_n)_{n \in \mathbb{N}}$ yields the following algorithm.

Algorithm 1 *With $\mathcal{J}_{-1} := \emptyset$, the dictionary index set $\mathcal{J}_n$ is defined as*

$$\mathcal{J}_n := \begin{cases} \mathcal{J}_{n-1} \cup \{n\}, & \text{if } \max_{j \in \mathcal{J}_n} c(\boldsymbol{u}_n, \boldsymbol{u}_j) \le \sigma, \\ \mathcal{J}_{n-1}, & \text{otherwise,} \end{cases}$$

where $\sigma > 0$ and $c(\boldsymbol{u}, \boldsymbol{v}) := \frac{|\kappa(\boldsymbol{u}, \boldsymbol{v})|}{\sqrt{\kappa(\boldsymbol{u}, \boldsymbol{u})}\sqrt{\kappa(\boldsymbol{v}, \boldsymbol{v})}}$ is the coherence [5]. For the initial estimate $\varphi_0 := 0$, generate the sequence $(\varphi_n)_{n \in \mathbb{N}}$ of nonlinear filters by

$$\varphi_{n+1} := \varphi_n + \lambda_n \left(\sum_{\iota \in \mathcal{I}_n} \omega_\iota^{(n)} P_{C_\iota^{(n)}}(\varphi_n) - \varphi_n \right),\ n \in \mathbb{N},$$

where $\lambda_n \in (0, 2\mathcal{L}_n)$ is the step size with the extrapolation coefficient

$$\mathcal{L}_n := \frac{\sum_{\iota \in \mathcal{I}_n} \omega_\iota^{(n)} \left\| P_{C_\iota^{(n)}}(\varphi_n) - \varphi_n \right\|^2}{\left\| \sum_{\iota \in \mathcal{I}_n} \omega_\iota^{(n)} P_{C_\iota^{(n)}}(\varphi_n) - \varphi_n \right\|^2} \ge 1. \quad (4)$$

In the exceptional case that the denominator is zero, $\mathcal{L}_n := 1$.

Theorem 1 (Convergence Analysis [12, 13])

(a) **Monotone approximation:** *Assume that $\Omega_n := \operatorname{arginf}_{g \in \mathcal{H}} \Theta_n(g) \ne \emptyset$ and that $\varphi_n \notin \Omega_n$. Then, for any $\lambda_n \in \left(0, 2\left(1 - \frac{\Theta_n^*}{\Theta_n(\varphi_n)}\right)\right)$, where $\Theta_n^* := \inf_{g \in \mathcal{H}} \Theta_n(g)$, $\|\varphi_{n+1} - \varphi^*\| < \|\varphi_n - \varphi^*\|$, $\forall \varphi^* \in \Omega_n$.*
In the following, suppose that (i) $(\lambda_n)_{n \in \mathbb{N}} \subset [\mathcal{L}_n \epsilon_1, \mathcal{L}_n (2 - \epsilon_2)] \subset (0, 2\mathcal{L}_n)$, $\exists \epsilon_1, \epsilon_2 > 0$ and that (ii) there exists $N_0 \in \mathbb{N}$ such that $\bigcap_{\iota \in \tilde{\mathcal{I}}_n, n \ge N_0} C_\iota^{(n)}$ has a relative interior with respect to the subspace $\bigcap_{n \ge N_0} M_n = M_{N_0}$, where $\tilde{\mathcal{I}}_n := \{\iota \in \mathcal{I}_n : \varphi_n \notin C_\iota^{(n)}\}$.

(b) **Convergence and asymptotic optimality:** *The sequence $(\varphi_n)_{n \in \mathbb{N}}$ generated by Algorithm 1 converges to some $\hat{\varphi} \in \mathcal{H}$ and $\lim_{n \to \infty} \Theta_n(\varphi_n) = \lim_{n \to \infty} \Theta_n(\hat{\varphi}) = 0$.*

(c) **Characterization of the limit point:** *Suppose that $\inf_{n \ge N_0, \iota \in \mathcal{I}_n} \omega_\iota^{(n)} > 0$. Then, the limit point is characterized by $\hat{\varphi} \in \liminf_{n \to \infty} \bigcap_{\iota \in \tilde{\mathcal{I}}_n} C_\iota^{(n)}$.*

Proof: The claims are readily verified by [12, Theorem 2, Proposition 3] and [13, Theorem 1].

3. THE Φ-PASS ALGORITHM

We generalize Algorithm 1 for deriving a low-complexity version. The idea for complexity reduction is to select and update only a few coefficients $h_{j,n}$ of $\kappa(\cdot, \boldsymbol{u}_j)$ that are maximally coherent to $\kappa(\cdot, \boldsymbol{u}_\iota)$, $\iota \in \mathcal{I}_n$. To be more specific, at time instant $n \in \mathbb{N}$, each data pair $(\boldsymbol{u}_\iota, d_\iota)$, $\iota \in \mathcal{I}_n$, is exploited to polish only a few, say $s_\iota^{(n)} (\le r_n)$, selected coefficients $\{h_{j,n}\}_{j \in \mathcal{J}_n^{(\iota)}}$ for some $\mathcal{J}_n^{(\iota)} := \{j_1^{(n,\iota)}, j_2^{(n,\iota)}, \cdots, j_{s_\iota^{(n)}}^{(n,\iota)}\} \subset \mathcal{J}_n$.

We constrain the direction vector of the projection onto each hyperslab S_ι to the subspace spanned by the selected dictionary elements that are maximally coherent to $\kappa(\cdot, \boldsymbol{u}_\iota)$:

$$M_\iota^{(n)} := \operatorname{span}\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j \in \mathcal{J}_n^{(\iota)}},\ \iota \in \mathcal{I}_n.$$

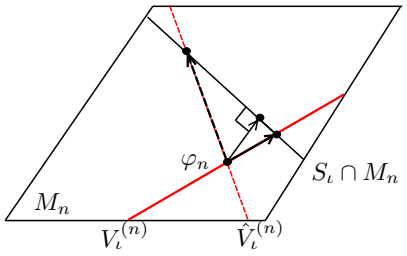

Fig. 2. Design of $V_\iota^{(n)}$ for $\rho = 0$.

This is accomplished by projecting the current filter onto

$$C_\iota^{(n)} := \{g \in V_n^{(\iota)} : (g(\boldsymbol{u}_\iota) - d_\iota)^2 \le \rho\} = S_\iota \cap V_n^{(\iota)}, \quad \iota \in \mathcal{I}_n, \quad (5)$$

where

$$V_\iota^{(n)} := M_\iota^{(n)} + \varphi_n := \{f + \varphi_n : f \in M_\iota^{(n)}\},\ \iota \in \mathcal{I}_n.$$

The Φ-PASS algorithm is given exactly by Algorithm 1 with $C_\iota^{(n)}$ defined by (5), instead of (2). The current filter φ_n is projected onto the convex sets $C_\iota^{(n)}$ in parallel, or equivalently it is projected onto the hyperslabs S_ι along the affine subspaces $V_\iota^{(n)}$; the name of Φ-PASS comes from this geometric property. The projections $P_{C_\iota^{(n)}}(\varphi_n)$ have the following closed-form expression:

$$P_{C_\iota^{(n)}}(\varphi_n) = \varphi_n + \varsigma_\iota^{(n)} \frac{|d_\iota - \varphi_n(\boldsymbol{u}_\iota)| - \rho}{\sum_{j \in \mathcal{J}_n^{(\iota)}} \alpha_j^{(\iota)} \kappa(\boldsymbol{u}_\iota, \boldsymbol{u}_j)} P_{M_\iota^{(n)}}(\kappa(\cdot, \boldsymbol{u}_\iota)),$$

Here, $\varsigma_\iota^{(n)} := \mathrm{sign}(d_\iota - \varphi_n(\boldsymbol{u}_\iota))$, if $|d_\iota - \varphi_n(\boldsymbol{u}_\iota)| > \rho$, and $\varsigma_\iota^{(n)} := 0$, otherwise, and $P_{M_\iota^{(n)}}(\kappa(\cdot, \boldsymbol{u}_\iota)) = \sum_{j \in \mathcal{J}_n^{(\iota)}} \alpha_j^{(\iota)} \kappa(\cdot, \boldsymbol{u}_j)$ with $\left[\alpha_{j_1^{(n,\iota)}}^{(\iota)}, \alpha_{j_2^{(n,\iota)}}^{(\iota)}, \cdots, \alpha_{j_{s_\iota^{(n)}}^{(n,\iota)}}^{(\iota)}\right]^T = \boldsymbol{K}_{\iota,n}^\dagger \boldsymbol{y}_{\iota,n}$, where $\boldsymbol{K}_{\iota,n}^\dagger$ is the Moore-Perose pseudo-inverse of the $s_\iota^{(n)} \times s_\iota^{(n)}$ kernel matrix $\boldsymbol{K}_{\iota,n}$ whose (s,t) entry is given by $[\boldsymbol{K}_{\iota,n}]_{s,t} := \kappa\big(\boldsymbol{u}_{j_s^{(n,\iota)}}, \boldsymbol{u}_{j_t^{(n,\iota)}}\big)$ and $\boldsymbol{y}_{\iota,n}$ is a $s_\iota^{(n)} \times 1$ vector whose sth component is given by $[\boldsymbol{y}_{\iota,n}]_s := \kappa\big(\boldsymbol{u}_{j_s^{(n,\iota)}}, \boldsymbol{u}_\iota\big)$.

Design of $V_\iota^{(n)}$ and geometric interpretation: A simplest example is to select the most coherent one for each $\iota \in \mathcal{I}_n$, i.e., $\mathcal{J}_n^{(\iota)} := \mathrm{argmax}_{j \in \mathcal{J}_n} c(\boldsymbol{u}_\iota, \boldsymbol{u}_j)$, and in this case $s_\iota^{(n)} = 1$, $\forall \iota \in \mathcal{I}_n$, $\forall n \in \mathbb{N}$, with probability one. Geometrically, such a function $\kappa(\cdot, \boldsymbol{u}_j)$ is selected that has the least angle to the normal vector of $S_\iota \cap M_n$ in M_n. See Fig. 2 in which the arrow on $V_\iota^{(n)}$ has a smaller angle with the normal vector than the arrow on another affine subspace $\hat{V}_\iota^{(n)}$. The above interpretation is justified by $\langle \kappa(\cdot, \boldsymbol{u}_\iota), \kappa(\cdot, \boldsymbol{u}_j) \rangle = \langle P_{M_n}(\kappa(\cdot, \boldsymbol{u}_\iota)) + P_{M_n^\perp}(\kappa(\cdot, \boldsymbol{u}_\iota)), \kappa(\cdot, \boldsymbol{u}_j) \rangle = \langle P_{M_n}(\kappa(\cdot, \boldsymbol{u}_\iota)), \kappa(\cdot, \boldsymbol{u}_j) \rangle$. Here, the first equality is due to the orthogonal decomposition ($M_n^\perp$ is the orthogonal complement of M_n) and the second equality is verified by $\kappa(\cdot, \boldsymbol{u}_j) \in M_n$.

Table 1. Computational complexity of the proposed and conventional algorithms.

QKLMS	$r_n L$
HYPASS	$r_n L + O(s^3) + (\frac{L}{2} + 1)s^2 + (2 - \frac{L}{2})s$
HYPASS (full)	$r_n(L+2) + r_n^2 + O((r_n - 1)^2)$
PKLMS	$r_n(L+2) + 2r_n^2 + O((r_n - 1)^2)$
KAP	$r_n(L + p^2 + 2p) + O(p^3) + 2p^2 + p$
Proposed	$r_n L + pO(s^3)$ $+(\frac{L}{2} + 2p)s^2 + (-\frac{L}{2} + 2p + 1)s$
Proposed (full)	$r_n(L + 2p + 1) + 2pr_n^2 + O((r_n - 1)^2)$

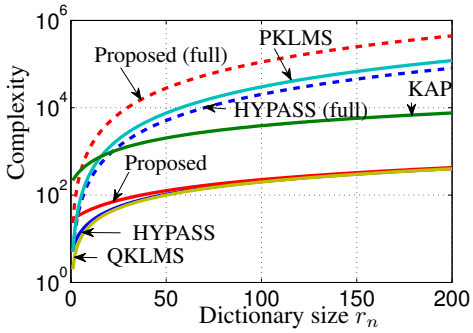

Fig. 3. Comparisons of the proposed and conventional algorithms in computational complexity for $L = 2$ and $p = 5$.

In contrast, the computationally most expensive example is $\mathcal{J}_n^{(\iota)} = \mathcal{J}_n$, $\forall \iota \in \mathcal{I}_n$, $\forall n \in \mathbb{N}$, which reproduces the fully-updating Φ-PASS algorithm presented in the previous section. Among many other possibilities, a reasonable option would be to select the s most coherent ones for each $\boldsymbol{u}_\iota$ unless the coherence values for some of them are below some prespecified threshold.

Relation to prior work: Letting $\mathcal{I}_n := \{n\}$, $\forall n \in \mathbb{N}$, reproduces the HYPASS algorithm [10] which solely exploits the current data at each iteration. Letting in addition, $\mathcal{J}_n^{(n)} := \mathrm{argmax}_{j \in \mathcal{J}_n} c(\boldsymbol{u}_n, \boldsymbol{u}_j)$, we obtain the quantized kernel normalized LMS (QKNLMS) algorithm, which coincides with the quantized kernel LMS (QKLMS) algorithm [14] in the case of Gaussian kernels. Letting $\mathcal{I}_n := \{n\}$, $\forall n \in \mathbb{N}$, $\mathcal{J}_n^{(n)} = \mathcal{J}_n$, and $\lambda_n := \frac{\sum_{j \in \mathcal{J}_n} \alpha_j \kappa(\boldsymbol{u}_n, \boldsymbol{u}_j)}{\kappa(\boldsymbol{u}_n, \boldsymbol{u}_n)}$ reproduces a projected kernel normalized LMS (PKNLMS), which coincides with the sequential projection algorithm (referred to as PKLMS) [15] in the case of Gaussian kernels. Hence, the Φ-PASS algorithm provides a family of algorithms including the HYPASS, QKNLMS, and PKNLMS. In the case of Gaussian kernels, the Φ-PASS algorithm for $s_\iota^{(n)} = 1$ can be viewed as quantizing each data $\boldsymbol{u}_\iota$, $\iota \in \mathcal{I}_n$, to its nearest point in the dictionary data set $\{\boldsymbol{u}_j\}_{j \in \mathcal{J}_n}$.

The existing data-reusing algorithm is based on *affine projection* [16, 17], including the kernel affine projection algorithms (KAPAs) [18] and the kernel affine projection (KAP) algorithm [5]. A related, but not data-reusing, algorithm named the quantized kernel APSM (QKAPSM) has also been proposed [19]. The proposed algorithm, as well as

Table 2. Parameter settings for each algorithm.

QKLMS	$\bar{r} = 121.54,\ \lambda_n = 0.35,\ \sigma = 0.854$
HYPASS	$\bar{r} = 121.54,\ \lambda_n = 0.35,\ \sigma = 0.854$
HYPASS (full)	$\bar{r} = 121.01,\ \lambda_n = 0.35,\ \sigma = 0.854$
PKLMS	$\bar{r} = 121.68,\ \tilde{\sigma} = 1.0 \times 10^{-5}$
KAP	$\bar{r} = 120.75,\ \lambda_n = 0.03,\ \sigma = 0.854$
proposed	$\bar{r} = 120.31,\ \lambda_n = 0.1\mathcal{L}_n,\ \sigma = 0.85$
proposed (full)	$\bar{r} = 120.74,\ \lambda_n = 0.1\mathcal{L}_n,\ \sigma = 0.85$

Table 3. Average MSEs and computational complexity.

	Average MSE [dB]	Complexity
QKLMS	-34.95	480
HYPASS	-34.95	485
HYPASS (full)	-36.67	29281
PKLMS	-31.10	43561
KAP	-36.10	16090
Proposed	-37.32	531
Proposed (full)	-40.00	149000

QKAPSM, is based on *parallel projection* [8, 9, 20]. In the following section, we compare the performance of the proposed algorithm with KAP as well as the HYPASS, QKLMS, and PKLMS algorithms.

Computational complexity: Computational complexity of the proposed and related algorithms is presented in Table 1.[1] HYPASS (full) and Proposed (full) in the table indicate the fully-updating HYPASS algorithm and the fully-updating Φ-PASS algorithm, respectively. For comparisons, Fig. 3 illustrates how the computational complexity increases with increasing dictionary size r_n. It is seen that the proposed algorithm for $s = 1$ requires 700 times less complexity than the PKLMS algorithm for $r_n = 200$. Moreover, the proposed algorithm is comparable to HYPASS for $s = 1$ and QKLMS in complexity, whereas it considerably outperforms them in MSE performance, as shown in the following section.

4. NUMERICAL EXAMPLES

We compare the performance of the proposed algorithm with the conventional algorithms in the application, estimating a function given as [18]

$$d_n = x_n - 0.2d_{n-1} - d_{n-1}x_{n-1} + 0.1x_{n-1} + 0.4d_{n-2},$$

where x_n is the input of the unknown function. This unknown function is predicted by a kernel adaptive filter with its input $\boldsymbol{u}_n := [d_n, d_{n-1}, d_{n-2}, y_{n-1}]^T \in \mathcal{U} \subset \mathbb{R}^L$ $(L = 4)$, where $\hat{d}_{n-1} := \varphi_{n-1}(\boldsymbol{u}_{n-1})$ which is a replica of d_{n-1} accommodating the past data $d_{n-3},\ d_{n-4}, \cdots$. In this example, the input of the unknown function is assumed white and uniformly distributed within the range of $[-0.5, 0.5]$. We employ the Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\zeta\|\boldsymbol{x} - \boldsymbol{y}\|^2_{\mathbb{R}^L})$ for $\zeta = 2.0$.

The p most recent input-output data are used for each update, i.e., $\mathcal{I}_n := \{n, n-1, \cdots, n-p+1\}$, for both the proposed and KAP algorithms, and the data-reusing factor (the number of hyperslabs for the proposed algorithm and the affine order for KAP) is set to $p = 15$. For stability, we add the regularization parameter $\epsilon_{\mathcal{L}} := 10^{-5}$ to the denominator of the extrapolation coefficient $\mathcal{L}_n$. Uniform weights are used; i.e., $\omega_\iota^{(n)} = (\min\{p, n+1\})^{-1}$ for all $\iota \in \mathcal{I}_n$. The error bound is set to $\rho = 0$. The fully-updating versions and $s = 1$ are tested for the proposed and HYPASS algorithms. For $s = 1$, HYPASS is reduced to a normalized version of QKLMS. (In the present case of Gaussian kernel, the normalization factor is one. See Relation to prior work in the previous section.) The set of parameters employed in the experiments is listed in Table 2. The step size λ_n, the coherence threshold σ, and the distance threshold $\tilde{\sigma}$ are chosen so that each algorithm attains the best performance. The regularization parameter for KAP is set to $\epsilon := 3.0 \times 10^{-4}$, which was chosen so as to achieve the best performance.

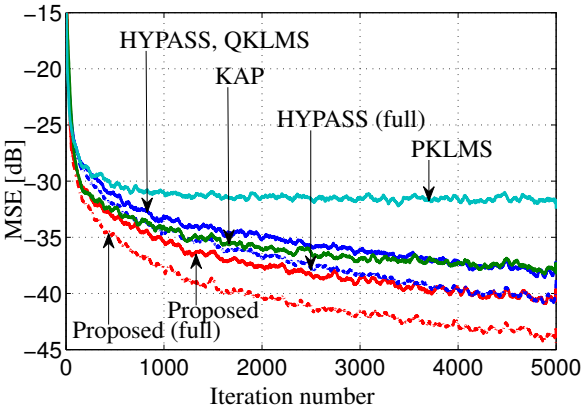

Fig. 4. MSE learning curves of each algorithm.

Fig. 4 depicts the MSE learning curves for each algorithm. One can see that the proposed algorithms outperform the conventional algorithms in the steady state MSE. Moreover, the proposed algorithm and KAP, both of which reuse past data, converges faster than HYPASS, PKLMS and QKLMS. Table 3 shows the average MSEs and computational complexity. Remarkably, the proposed algorithm for $s = 1$ attains approximately $0.65 - 6.22$ [dB] lower than HYPASS, KAP, and PKLMS despite its lower complexity. We finally mentioned that PKLMS is implemented in exactly the same way as in [15] and its step size is identically one. By extending the method in [15] so that the step size can be chosen within the range of $[0, 2]$, PKLMS could attain performance comparable to QKLMS (and HYPASS for $s = 1$).

5. CONCLUSION

This paper has presented the Φ-PASS algorithm, which reuses observed data efficiently by means of parallel hyperslab projection along affine subspaces. The efficiency, in the particular case of one-dimensional affine subspaces, comes from the fact that only one coefficient is updated per datum. The coefficients to be updated are selected based on the simple coherence criterion. Numerical examples have shown that the proposed algorithm outperforms the existing kernel adaptive filtering algorithms with low complexity.

Acknowledgements: This work was supported by KDDI Foundation.

[1] A careful inspection of the matrix $\boldsymbol{K}_{\iota,n}$ suggests that, in the fully-updating Φ-PASS algorithm, $\boldsymbol{K}^\dagger_{\iota,n} \neq \boldsymbol{K}^\dagger_{\iota,n-1}$ only when the dictionary size increases and that $\boldsymbol{K}^\dagger_{\iota,n}$ can be computed with $\boldsymbol{K}^\dagger_{\iota,n-1}$ by the matrix inversion lemma [21] in the $O((r_n - 1)^2)$ complexity.

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] 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.

[5] 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.

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

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

[8] I. Yamada, K. Slavakis, and K. Yamada, "An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques," *IEEE Trans. Signal Processing*, vol. 50, no. 5, pp. 1091–1101, May 2002.

[9] 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 Processing Magazine*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

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

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

[12] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Funct. Anal. Optim.*, vol. 25, no. 7&8, pp. 593–617, 2004.

[13] M. Yukawa and I. Yamada, "A deterministic analysis of linearly constrained adaptive filtering algorithms," in *Proc. EUSIPCO*, 2011, pp. 131–135.

[14] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

[15] T. J. Dodd, V. Kadirkamanathan, and R. F. Harrison, "Function estimation in Hilbert space using sequential projections," in *IFAC Conf. Intell. Control Syst. Signal Process.*, 2003, pp. 113–118.

[16] T. Hinamoto and S. Maekawa, "Extended theory of learning identification," *Trans. IEE Japan*, vol. 95, no. 10, pp. 227–234, 1975, in Japanese.

[17] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *IEICE Trans.*, vol. 67-A, no. 5, pp. 126–132, 1984, in Japanese.

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

[19] K. Slavakis, P. Bouboulis, and S. Theodoridis, *Online Learning in Reproducing Kernel Hilbert Spaces*, Elsevier, to appear in E-Reference Signal Processing.

[20] M. Yukawa, L. G. Cavalcante, and I. Yamada, "Efficient blind mai suppression on ds/cdma sytem by embedded constraint prallel projection techniques.," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E-88A, no. 8, pp. 2062–2071, Aug. 2005.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.