A Better Metric in Kernel Adaptive Filtering

Airi Takeuchi Masahiro Yukawa
Dept. Electronics and Electrical Engineering,
Keio University, Japan

Klaus-Robert Müller
Dept. Computer Science, Technische Universität Berlin, Germany
Dept. Brain and Cognitive Engineering, Korea University, Korea

***Abstract*—The metric in the reproducing kernel Hilbert space (RKHS) is known to be given by the Gram matrix (which is also called the kernel matrix). It has been reported that the metric leads to a decorrelation of the kernelized input vector because its autocorrelation matrix can be approximated by the (down scaled) squared Gram matrix subject to some condition. In this paper, we derive a better metric (a best one under the condition) based on the approximation, and present an adaptive algorithm using the metric. Although the algorithm has quadratic complexity, we present its linear-complexity version based on a selective updating strategy. Numerical examples validate the approximation in a practical scenario, and show that the proposed metric yields fast convergence and tracking performance.**

I. Introduction

Nonlinear adaptive filtering plays an important role in many applications, including system identification and acoustic echo cancellation. Among several others, a kernel adaptive filter has attracted significant attention, and a number of powerful kernel-based computational methods have been proposed [1]–[8]. The existing kernel adaptive filtering algorithms are classified into two categories from the projection viewpoint [8]: (i) the parameter-space approach and (ii) the functional-space approach. The kernel normalized least mean square (KNLMS) algorithm [9] is a typical example of the parameter-space approach, and it updates the coefficient vector by using the projection onto the zero instantaneous-error hyperplane in a parameter space. The hyperplane projection along affine subspace (HYPASS) algorithm [10] is a typical example of the functional-space approach, and it operates the projection in a functional space. The KNLMS and the HYPASS algorithms can be regarded as normalized versions of Least Mean Square (LMS) algorithm [11], [12] for certain different input-output pairs. The Cartesian HYPASS (CHYPASS) algorithm [6], [7] is a functional-space approach for multikernel adaptive filtering.

As the error contours of the mean squared error (MSE) depend on the autocorrelation matrix of the input vectors in general, the eigenvalue spread governs the convergence speed of adaptive algorithms [13]. The previous study [14] elucidated the mechanism for the reduction of the eigenvalue spread coming from the HYPASS algorithm. The metric of HYPASS is naturally induced by the metric in a reproducing kernel Hilbert space (RKHS). Nevertheless, it is not an ideal one from the aspect of whitening. The key fact here is that the autocorrelation matrix of the kernelized input vector can be approximated by the square of the Gram matrix of the dictionary subject to a certain condition [14]. A predetermined dictionary has been used therein to verify the validity of the theoretical analysis to explain the reduction of the eigenvalue spread. Although the dictionary of kernel adaptive filters is constructed online in practice, numerical verifications for this practical case have not yet been reported.

In this paper, we show the validity of the analysis on the eigenvalue-spread reduction in a practical situation, and propose a better metric from the viewpoint of decorrelation of the kernelized input vectors. We present a kernel adaptive filtering algorithm that employs the proposed metric. The proposed metric reduces the eigenvalue spread more than the one of HYPASS, leading to faster convergence. To reduce the computational complexity, a low-complexity version of the proposed algorithm is derived with selective updating. This significantly reduces the complexity at the expense of slight performance degradations. Numerical examples show that the proposed algorithm attains fast convergence and tracking performance compared to the existing kernel adaptive filtering algorithms.

II. Preliminaries

A. Definitions and System Model

Let $\mathbb{R}, \mathbb{N}$, and $\mathbb{N}^*$ denote the sets of real numbers, nonnegative integers, and positive integers, respectively. We first define the inner product in the N dimensional Euclidean space $\mathbb{R}^N$ and the metric projection onto a closed convex subset of a Hilbert space.

Definition 1: Let $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ be a positive definite matrix. the $\boldsymbol{Q}$-inner product is defined as $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\boldsymbol{Q}} := \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{y}$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$, where $(\cdot)^\top$ denotes a vector (matrix) transpose.

Definition 2: Let $\mathcal{X}$ be a real Hilbert space equipped with a norm $\|\cdot\|_{\mathcal{X}}$. Then, the metric projection of a point $x \in \mathcal{X}$ onto a nonempty closed convex set $\mathcal{K} \subset \mathcal{X}$ is defined as

$$P_{\mathcal{K}}(x) := \operatorname{argmin}_{y\in\mathcal{K}} \|x - y\|_{\mathcal{X}} . \tag{1}$$

We consider an adaptive estimation problem of a nonlinear system $\psi : \mathcal{U} \to \mathbb{R}$, where $\mathcal{U} \subset \mathbb{R}^L$ is the input space. Its noisy output is given by

$$d_n := \psi(\boldsymbol{u}_n) + v_n,\ n \in \mathbb{N}, \tag{2}$$

where $(\boldsymbol{u}_n)_{n\in\mathbb{N}}$ is a sequence of input vectors and $(v_n)_{n\in\mathbb{N}}$ is a sequence of additive noises. The unknown function ψ is modeled as an element of the reproducing kernel Hilbert space

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

(RKHS) $\mathcal{H}$ associated with a reproducing kernel $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ [15], [16]. A kernel adaptive filter can be expressed as

$$\varphi_n := \sum_{j \in \mathcal{J}_n} h_j^{(n)} \kappa(\cdot, \boldsymbol{u}_j), \ n \in \mathbb{N}, \tag{3}$$

where $h_j^{(n)} \in \mathbb{R}$ and $\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j \in \mathcal{J}_n}$ is called the dictionary indicated by $\mathcal{J}_n = \{j_1^{(n)}, j_2^{(n)}, \cdots, j_{r_n}^{(n)}\} \subset \{0, 1, 2, \cdots, n\}$. The dictionary is constructed in an incremental fashion based on the coherence [9]

$$c(\boldsymbol{u}, \boldsymbol{v}) := \frac{|\kappa(\boldsymbol{u}, \boldsymbol{v})|}{\sqrt{\kappa(\boldsymbol{u}, \boldsymbol{u})}\sqrt{\kappa(\boldsymbol{v}, \boldsymbol{v})}} \in [0, 1], \ \forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{U}. \tag{4}$$

Let $\mathcal{J}_{-1} := \emptyset$. The dictionary updating rule is given, for $n \in \{-1, 0, 1, 2, \cdots\}$, by

$$\mathcal{J}_{n+1} := \begin{cases} \mathcal{J}_n \cup \{n\} & \text{if } \max_{j \in \mathcal{J}_n} c(\boldsymbol{u}_n, \boldsymbol{u}_j) \le \delta, \\ \mathcal{J}_n & \text{otherwise,} \end{cases} \tag{5}$$

where $\delta \in (0, 1)$. We assume that the dictionary is a linearly independent set; this is guaranteed automatically when a Gaussian kernel is employed. The dictionary subspace is defined as

$$\mathcal{M}_n := \operatorname{span}\{\kappa(\cdot, \boldsymbol{u}_j)\}_{j \in \mathcal{J}_n}. \tag{6}$$

B. Existing Kernel Adaptive Filtering Algorithm

We describe the existing algorithms in the case of $\mathcal{J}_n = \mathcal{J}_{n+1}$ for simplicity. Let $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the inner product and the norm in a RKHS $\mathcal{H}$. It is desired to find a function f that minimizes the distance from the current filter $\varphi_n \in \mathcal{H}$ under the constraint $f(\boldsymbol{u}_n) = \langle f, \kappa(\cdot, \boldsymbol{u}_n) \rangle = d_n$, where the first equality holds due to the reproducing property [15]–[18]. The update equation of the HYPASS algorithm is given as follows:

$$\varphi_{n+1} = \varphi_n + \lambda_n (P_{\Pi_n}(\varphi_n) - \varphi_n), \tag{7}$$

where $\lambda_n \in (0, 2)$ is the stepsize and

$$\Pi_n := \{g \in \mathcal{M}_n : g(\boldsymbol{u}_n) = \langle g, \kappa(\cdot, \boldsymbol{u}_n) \rangle = d_n\}. \tag{8}$$

We will express HYPASS in a parameter space. The hyperplane Π_n can be expressed in the r_n-dimensional Euclidean space $\mathbb{R}^{r_n}$ as [19]

$$H_n := \{\boldsymbol{h} \in \mathbb{R}^{r_n} : \varphi(\boldsymbol{u}_n) = \boldsymbol{\kappa}_n^\top \boldsymbol{h} = d_n\}, \tag{9}$$

where $\boldsymbol{\kappa}_n := [\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_1^{(n)}}), \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_2^{(n)}}), \cdots, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{j_{r_n}^{(n)}})]^\top$ and $\boldsymbol{G}_n \in \mathbb{R}^{r_n \times r_n}$ is the Gram matrix whose (p, q) component is given by $\kappa(\boldsymbol{u}_{j_p^{(n)}}, \boldsymbol{u}_{j_q^{(n)}})$.

Note that the functional subspace $(\mathcal{M}_n, \langle \cdot, \cdot \rangle)$ is an isomorphic Hilbert space of the parameter space $(\mathbb{R}^{r_n}, \langle \cdot, \cdot \rangle_{\boldsymbol{G}_n})$. In $(\mathbb{R}^{r_n}, \langle \cdot, \cdot \rangle_{\boldsymbol{G}_n})$, a normal vector of H_n is given by $\boldsymbol{G}_n^{-1} \boldsymbol{\kappa}_n$ since $\boldsymbol{\kappa}_n^\top \boldsymbol{h} = \langle \boldsymbol{G}_n^{-1} \boldsymbol{\kappa}_n, \boldsymbol{h} \rangle_{\boldsymbol{G}_n}$. Using the $\boldsymbol{G}_n$-projection $P_{H_n}^{\boldsymbol{G}_n}(\boldsymbol{h}_n) := \operatorname{argmin}_{\boldsymbol{h} \in H_n} \|\boldsymbol{h} - \boldsymbol{h}_n\|_{\boldsymbol{G}_n}$, HYPASS can be therefore expressed in $(\mathbb{R}^{r_n}, \langle \cdot, \cdot \rangle_{\boldsymbol{G}_n})$ as follows [19]:

$$\begin{aligned} \boldsymbol{h}_{n+1} &= \boldsymbol{h}_n + \lambda_n (P_{H_n}^{\boldsymbol{G}_n}(\boldsymbol{h}_n) - \boldsymbol{h}_n) \\ &= \boldsymbol{h}_n + \lambda_n \frac{e_n}{\boldsymbol{\kappa}_n^\top \boldsymbol{G}_n^{-1} \boldsymbol{\kappa}_n} \boldsymbol{G}_n^{-1} \boldsymbol{\kappa}_n, \end{aligned} \tag{10}$$

where $\lambda_n \in (0, 2)$ is the stepsize and $e_n := d_n - \boldsymbol{h}_n^\top \boldsymbol{\kappa}_n$ is the instantaneous error.

Now, let $\langle \cdot, \cdot \rangle_{\mathbb{R}^{r_n}}$ and $\|\cdot\|_{\mathbb{R}^{r_n}}$ denote the canonical inner product (defined as $\langle \cdot, \cdot \rangle_{\mathbb{R}^{r_n}} := \langle \cdot, \cdot \rangle_{\boldsymbol{I}}$ with the identity matrix $\boldsymbol{I}$) and its induced norm, respectively. We finally introduce the KNLMS algorithm [9], a parameter-space approach that operates the metric projection $P_{H_n}(\boldsymbol{h}_n) := \operatorname{argmin}_{\boldsymbol{h} \in H_n} \|\boldsymbol{h} - \boldsymbol{h}_n\|_{\mathbb{R}^{r_n}}$ in $(\mathbb{R}^{r_n}, \langle \cdot, \cdot \rangle_{\mathbb{R}^{r_n}})$. The update equation is given by

$$\begin{aligned} \boldsymbol{h}_{n+1} &= \boldsymbol{h}_n + \lambda_n (P_{H_n}(\boldsymbol{h}_n) - \boldsymbol{h}_n) \\ &= \boldsymbol{h}_n + \lambda_n \frac{e_n}{\boldsymbol{\kappa}_n^\top \boldsymbol{\kappa}_n} \boldsymbol{\kappa}_n. \end{aligned} \tag{11}$$

In the case of $\mathcal{J}_n \neq \mathcal{J}_{n+1}$, the vector $\boldsymbol{h}_n \in \mathbb{R}^{r_n}$ in (10) and (11) needs to be replaced by $[\boldsymbol{h}_n^\top \ 0]^\top \in \mathbb{R}^{r_{n+1}}$. This also applies to the proposed algorithm presented in Section III-B.

III. Proposed Metric and Algorithm

A. Derivation of Proposed Metric

The autocorrelation matrix of the kernelized input vector $\boldsymbol{\kappa}_n$ can be approximated as [14]

$$\boldsymbol{R} := E(\boldsymbol{\kappa}_n \boldsymbol{\kappa}_n^\top) \approx \frac{1}{r_n} \boldsymbol{G}_n^2 \tag{12}$$

subject to the following covariance-matrix condition:

$$E(\kappa(\cdot, \boldsymbol{u}_n) \kappa(\cdot, \boldsymbol{u}_n)^\top) \approx \frac{1}{r_n} \sum_{j \in \mathcal{J}_n} \kappa(\cdot, \boldsymbol{u}_j) \kappa(\cdot, \boldsymbol{u}_j)^\top. \tag{13}$$

Intuitively, the condition (13) implies that the dictionary data can be considered as randomly picked samples from the input-data distribution. KNLMS can be regarded as the normalized LMS (NLMS) algorithm for the input-output pair $(\boldsymbol{\kappa}_n, d_n)$, and its corresponding MSE function is given by

$$J_{\boldsymbol{h}}(\boldsymbol{h}) := E(e_n^2(\boldsymbol{h})) = \boldsymbol{h}^\top \boldsymbol{R} \boldsymbol{h} - 2\boldsymbol{p}^\top \boldsymbol{h} + E(d_n^2), \tag{14}$$

where $\boldsymbol{p} := E(d_n \boldsymbol{\kappa}_n)$ is the cross-correlation vector between d_n and $\boldsymbol{\kappa}_n$. Fixing the dictionary and letting $\boldsymbol{G}_n := \boldsymbol{G}$, $\forall n \in \mathbb{N}$, HYPASS can be regarded as the NLMS algorithm for the input-output pair $(\tilde{\boldsymbol{\kappa}}_n, d_n)$, and its corresponding MSE function is given by [19], [20]

$$J_{\tilde{\boldsymbol{h}}}(\tilde{\boldsymbol{h}}) := E(e_n^2(\tilde{\boldsymbol{h}})) = \tilde{\boldsymbol{h}}^\top \tilde{\boldsymbol{R}} \tilde{\boldsymbol{h}} - 2\tilde{\boldsymbol{p}}^\top \tilde{\boldsymbol{h}} + E(d_n^2), \tag{15}$$

where $\tilde{\boldsymbol{h}} := \boldsymbol{G}^{\frac{1}{2}} \boldsymbol{h}$, $\tilde{\boldsymbol{p}} := E(d_n \tilde{\boldsymbol{\kappa}}_n) = \boldsymbol{G}^{-\frac{1}{2}} \boldsymbol{p}$, and $\tilde{\boldsymbol{R}} := E(\tilde{\boldsymbol{\kappa}}_n \tilde{\boldsymbol{\kappa}}_n^\top) = \boldsymbol{G}^{-\frac{1}{2}} \boldsymbol{R} \boldsymbol{G}^{-\frac{1}{2}}$ with the modified kernelized input vector $\tilde{\boldsymbol{\kappa}}_n := \boldsymbol{G}^{-\frac{1}{2}} \boldsymbol{\kappa}_n$. The approximation (12) immediately implies $\tilde{\boldsymbol{R}} \approx \frac{1}{r} \boldsymbol{G}$ ($r_n := r$, $\forall n \in \mathbb{N}$), and hence it follows that [14]

$$\operatorname{cond}_2(\tilde{\boldsymbol{R}}) \approx \sqrt{\operatorname{cond}_2(\boldsymbol{R})}, \tag{16}$$

where $\operatorname{cond}_2(\boldsymbol{R}) := \|\boldsymbol{R}\|_2 \|\boldsymbol{R}^{-1}\|_2$ with the spectral norm $\|\cdot\|_2$. This implies that $\tilde{\boldsymbol{R}}$ is better conditioned than $\boldsymbol{R}$, and that the error contours of HYPASS is better shaped than those of KNLMS.

We consider the specific metric $\langle \cdot, \cdot \rangle_{\boldsymbol{G}_n^2}$ so that its corresponding autocorrelation matrix has its condition number equal to the unity. Recall that HYPASS uses the metric

$\langle\cdot,\cdot\rangle_{\boldsymbol{G}_n}$. As in the arguments in the previous paragraph, we fix the dictionary and let $\boldsymbol{G}_n := \boldsymbol{G}$, $\forall n \in \mathbb{N}$. The use of the metric $\langle\cdot,\cdot\rangle_{\boldsymbol{G}^2}$ modifies the MSE function (15) into (see the proof of Proposition 1 in Section III-B)

$$J_{\hat{\boldsymbol{h}}}(\hat{\boldsymbol{h}}) := E(e_n^2(\hat{\boldsymbol{h}})) = \hat{\boldsymbol{h}}^\top \hat{\boldsymbol{R}}\hat{\boldsymbol{h}} - 2\hat{\boldsymbol{p}}^\top \hat{\boldsymbol{h}} + E(d_n^2) \tag{17}$$

of $\hat{\boldsymbol{h}} := \boldsymbol{G}\boldsymbol{h}$, where $\hat{\boldsymbol{p}} := E(d_n\hat{\boldsymbol{\kappa}}_n) = \boldsymbol{G}^{-1}\boldsymbol{p}$, $\hat{\boldsymbol{R}} := E(\hat{\boldsymbol{\kappa}}_n\hat{\boldsymbol{\kappa}}_n^\top) = \boldsymbol{G}^{-1}\hat{\boldsymbol{R}}\boldsymbol{G}^{-1}$ with $\hat{\boldsymbol{\kappa}}_n := \boldsymbol{G}^{-1}\boldsymbol{\kappa}_n$. By using the approximation (12), it can be seen that $\hat{\boldsymbol{R}} \approx \frac{1}{r}\boldsymbol{I}$, and hence

$$\mathrm{cond}_2(\hat{\boldsymbol{R}}) \approx 1. \tag{18}$$

The metric $\langle\cdot,\cdot\rangle_{\boldsymbol{G}_n^2}$ is therefore ideal from the viewpoint of whitening under the condition (13).

We will illustrate the reduction of the eigenvalue spread yielded by the use of the metrics $\langle\cdot,\cdot\rangle_{\boldsymbol{G}_n}$ and $\langle\cdot,\cdot\rangle_{\boldsymbol{G}_n^2}$. We use the Gaussian kernel $\kappa(\boldsymbol{u},\boldsymbol{v}) := \exp(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|_{\mathbb{R}^L}^2}{2\sigma^2})$, $\boldsymbol{u},\boldsymbol{v} \in \mathcal{U}$, whose scale parameter is set to $\sigma = 0.03$. The dictionary size is fixed to $r = 25$, and the data points of the dictionary are placed uniformly between -0.5 and 0.5. The autocorrelation matrices are computed with 2000 samples of the input data drawn from the uniform distribution within the interval $[-0.5, 0.5]$. Fig. 1(a) depicts the contours of the surface of

$$f(h_1,h_2) := \begin{bmatrix} h_1, & h_2 \end{bmatrix} \begin{bmatrix} \mu_{\max} & 0 \\ 0 & \mu_{\min} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \; h_1, h_2 \in \mathbb{R}, \tag{19}$$

where $\mu_{\max}$ (or $\mu_{\min}$) is the maximum (or minimum) eigenvalue of $\boldsymbol{R}$. Likewise, Figs. 1(b), (c) depict those for $\tilde{\boldsymbol{R}}$ and $\hat{\boldsymbol{R}}$, respectively. It can be seen that the distortion of the contours is alleviated from Fig. 1(a) to Fig. 1(b) and from Fig. 1(b) to Fig. 1(c). The eigenvalue spreads are $\mathrm{cond}_2(\boldsymbol{R}) \approx 85.2$, $\mathrm{cond}_2(\tilde{\boldsymbol{R}}) \approx 14.6$, and $\mathrm{cond}_2(\hat{\boldsymbol{R}}) \approx 4.0$. Hence, the proposed metric is expected to yield fast convergence; this will be demonstrated in Section IV.

B. Proposed Algorithm

We consider the Hilbert space $\mathbb{R}^{r_n}$ equipped with the inner product $\langle\cdot,\cdot\rangle_{\boldsymbol{G}_n^2}$. As in Section II-B, we present the proposed algorithm in the case of $\mathcal{J}_n = \mathcal{J}_{n+1}$. Using the $\boldsymbol{G}_n^2$-projection $P_{H_n}^{\boldsymbol{G}_n^2}(\boldsymbol{h}_n) := \mathrm{argmin}_{\boldsymbol{h}\in H_n} \|\boldsymbol{h} - \boldsymbol{h}_n\|_{\boldsymbol{G}_n^2}$, the filter updating rule in $(\mathbb{R}^{r_n}, \langle\cdot,\cdot\rangle_{\boldsymbol{G}_n^2})$ is given as follows:

$$\boldsymbol{h}_{n+1} = \boldsymbol{h}_n + \lambda_n (P_{H_n}^{\boldsymbol{G}_n^2}(\boldsymbol{h}_n) - \boldsymbol{h}_n), \tag{20}$$

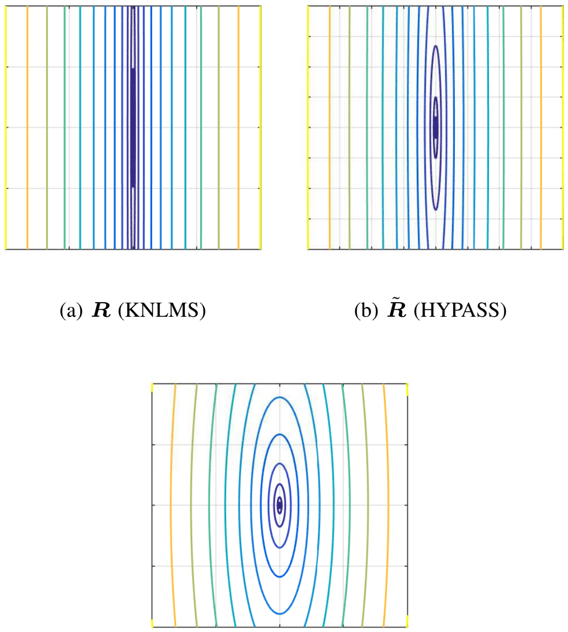
where $\lambda_n \in (0,2)$ is the stepsize. Observing that $H_n = \{\boldsymbol{h} \in \mathbb{R}^{r_n} : \langle \boldsymbol{G}_n^{-2}\boldsymbol{\kappa}_n, \boldsymbol{h}\rangle_{\boldsymbol{G}_n^2} = d_n\}$, we obtain

$$P_{H_n}^{\boldsymbol{G}_n^2}(\boldsymbol{h}_n) = \boldsymbol{h}_n - \lambda_n \frac{\langle \boldsymbol{G}_n^{-2}\boldsymbol{\kappa}_n, \boldsymbol{h}_n\rangle_{\boldsymbol{G}_n^2} - d_n}{\left\|\boldsymbol{G}_n^{-2}\boldsymbol{\kappa}_n\right\|_{\boldsymbol{G}_n^2}^2} \boldsymbol{G}_n^{-2}\boldsymbol{\kappa}_n. \tag{21}$$

Substituting (21) into (20) yields

$$\boldsymbol{h}_{n+1} = \boldsymbol{h}_n + \lambda_n \frac{e_n}{\boldsymbol{\kappa}_n^\top \boldsymbol{G}_n^{-2}\boldsymbol{\kappa}_n} \boldsymbol{G}_n^{-2}\boldsymbol{\kappa}_n. \tag{22}$$

The following proposition can be verified.


(a) $\boldsymbol{R}$ (KNLMS)

(b) $\tilde{\boldsymbol{R}}$ (HYPASS)

(c) $\hat{\boldsymbol{R}}$ (PROPOSED)

Fig. 1. Contours of $f(h_1,h_2)$ for $\boldsymbol{R}, \tilde{\boldsymbol{R}}$, and $\hat{\boldsymbol{R}}$.

Proposition 1: Fix the dictionary and let $\boldsymbol{G}_n := \boldsymbol{G}$ and $r_n := r$, $\forall n \in \mathbb{N}$. The error contours for the proposed algorithm (22) are governed by the autocorrelation matrix $\hat{\boldsymbol{R}} := \boldsymbol{G}^{-1}\boldsymbol{R}\boldsymbol{G}^{-1} \approx \frac{1}{r}\boldsymbol{I}$ of the whitened kernelized input vector $\hat{\boldsymbol{\kappa}} := \boldsymbol{G}^{-1}\boldsymbol{\kappa}_n$.

Proof: Left-multiplying both sides of (22) by $\boldsymbol{G}_n$ and letting $\boldsymbol{G}_n := \boldsymbol{G}$ yields

$$\hat{\boldsymbol{h}}_{n+1} = \hat{\boldsymbol{h}}_n + \lambda_n \frac{e_n}{\hat{\boldsymbol{\kappa}}_n^\top \hat{\boldsymbol{\kappa}}_n} \hat{\boldsymbol{\kappa}}_n. \tag{23}$$

The proposed algorithm (23) can be regarded as a normalized version of the following LMS algorithm:

$$\hat{\boldsymbol{h}}_{n+1} = \hat{\boldsymbol{h}}_n + \eta_n e_n \hat{\boldsymbol{\kappa}}_n, \tag{24}$$

where $\eta_n > 0$ and the instantaneous error can be rewritten as $e_n = d_n - \langle \hat{\boldsymbol{h}}_n, \hat{\boldsymbol{\kappa}}_n\rangle_{\mathbb{R}^{r_n}}$. This implies that the MSE function is given by (17), and hence the error contours of MSE for the proposed algorithm are governed by $\hat{\boldsymbol{R}}$. Also, from (24), the proposed algorithm can be regarded as the NLMS algorithm for $(\hat{\boldsymbol{\kappa}}_n, d_n)$. □

C. Coherence-Based Selective Updating Strategy

To reduce the computational complexity, we employ the coherence-based selective updating strategy [9]. The basic idea is to select, from the dictionary $\{\kappa(\cdot,\boldsymbol{u}_j)\}_{j\in\mathcal{J}_n}$, a small number of elements that are most coherent to $\kappa(\cdot,\boldsymbol{u}_\iota)$. The coherence is defined as (4).

For each $\iota \in \mathcal{I}_n$, $n \in \mathbb{N}$, define the selected dictionary index subset $\check{\mathcal{J}}_n \subset \mathcal{J}_n$ of cardinality $s \in \mathbb{N}^*$ such that

$$c(\boldsymbol{u}_\iota, \boldsymbol{u}_j) \ge c(\boldsymbol{u}_\iota, \boldsymbol{u}_k), \quad \forall j \in \check{\mathcal{J}}_n, \forall k \in \mathcal{J}_n \backslash \check{\mathcal{J}}_n. \tag{25}$$

TABLE I
COMPUTATIONAL COMPLEXITY

	Computational Complexity
PROPOSED (full)	$2r_n^2 + v_{\rm inv}(r_n) + O(r_n)$
PROPOSED (selective)	$(L+5)r_n + \frac{1}{2}s^3 + \frac{L+4}{2}s^2 + v_{\rm inv}(s) + O(s)$
HYPASS	$(L+5)r_n + \frac{L+5}{2}s^2 + v_{\rm inv}(s) + O(s)$
KRLS tracker	$2r_n^2 + v_{\rm inv}(r_n) + O(r_n)$

In other words, we select a dictionary subset $\{\kappa(\cdot,\boldsymbol{u}_j)\}_{j\in\check{\mathcal{J}}}$ that are maximally coherent to $\kappa(\cdot,\boldsymbol{u}_\iota)$. For the selected dictionary indicated by $\check{\mathcal{J}}_n = \{\iota_1^{(n)},\iota_2^{(n)},\cdots,\iota_s^{(n)}\}$, its associated Gram matrix is denoted by $\check{\boldsymbol{G}}_{n,\iota} \in \mathbb{R}^{s\times s}$ whose (p,q) component is given by $\kappa(\boldsymbol{u}_{\iota_p^{(n)}},\boldsymbol{u}_{\iota_q^{(n)}})$ and the denoted kernelized input vector is given by $\check{\boldsymbol{\kappa}}_{n,\iota} := [\kappa(\boldsymbol{u}_n,\boldsymbol{u}_{\iota_1^{(n)}}),\kappa(\boldsymbol{u}_n,\boldsymbol{u}_{\iota_2^{(n)}}),\cdots,\kappa(\boldsymbol{u}_n,\boldsymbol{u}_{\iota_s^{(n)}})]$. Using the vector $\check{\boldsymbol{h}}_{n,\iota} := [h_{n,\iota_1^{(n)}},h_{n,\iota_2^{(n)}},\cdots,h_{n,\iota_s^{(n)}}]^\top$ of the selected coefficients together with $\check{\boldsymbol{G}}_{n,\iota}$ and $\check{\boldsymbol{\kappa}}_{n,\iota}$, the proposed algorithm with the selective updating strategy is given by

$$\check{\boldsymbol{h}}_{n+1,\iota} = \check{\boldsymbol{h}}_{n,\iota} + \lambda_n \frac{e_n}{\check{\boldsymbol{\kappa}}_{n,\iota}^\top \check{\boldsymbol{G}}_{n,\iota}^{-2} \check{\boldsymbol{\kappa}}_{n,\iota}} \check{\boldsymbol{G}}_{n,\iota}^{-2} \check{\boldsymbol{\kappa}}_{n,\iota}. \qquad (26)$$

D. *Computational Complexity*

We discuss the computational complexity of the proposed algorithm. Suppose that the Gaussian kernel is employed. The computational complexity of a kernel adaptive filter is given in terms of the dictionary size r_n, $n\in\mathbb{N}$, and the dimension L of the input space $\mathcal{U}$. Table I summarizes the overall per-iteration complexity of the proposed fully updating algorithm, the proposed selective updating algorithm, HYPASS, and the kernel recursive least squares (KRLS) tracker algorithm [5]. In the table, $v_{\rm inv}(r_n)$ represents the complexity for $\boldsymbol{G}_n^{-1}$, and $v_{\rm inv}(s)$ represents the complexity for the inverse of the $s\times s$ submatrix $\check{\boldsymbol{G}}_{n,\iota}$ of $\boldsymbol{G}_n$. The complexity of the proposed selective updating algorithm and HYPASS using the selective updating strategy, depends on the number s of the selected coefficients. The proposed fully updating algorithm requires the $O(r_n^2)$ complexity, while the proposed selective updating algorithm requires the $O(r_n)$ complexity, provided that $r_n \gg s$.

Fig. 2 illustrates the complexity of each algorithm as a function of the dictionary size for $L=2$ and $s=7$. Here, $v_{\rm inv}(s)$ is counted as s^3, and $v_{\rm inv}(r_n)$ is counted as $(r_n-1)^2$ since we use the matrix inversion lemma [21].

IV. NUMERICAL EXAMPLES

We first show the validity of the approximation (12), and verify the effect of the whitening of the proposed metric in a practical scenario when the dictionary grows over time under a novelty criterion. We then show the advantages of the proposed metric in its application to online prediction of time-series data.

A. *Experiment A: Eigenvalue Spread*

The input data are randomly generated from a uniform distribution over the interval $[-0.5,0.5]$. We use the Gaussian kernel whose scale parameter is set to $\sigma = 0.03$. The dictionary is constructed with the coherence criterion [9] with the threshold $\delta = 0.9$. The eigenvalue spreads, averaged over 300 independent runs, of $\boldsymbol{R}, \tilde{\boldsymbol{R}}$ and $\hat{\boldsymbol{R}}$ are plotted in Fig. 3. One can see that the eigenvalue spreads of $\hat{\boldsymbol{R}}$ are much smaller than those of $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$, and that the approximation ${\rm cond}_2(\tilde{\boldsymbol{R}}) \approx \sqrt{{\rm cond}_2(\boldsymbol{R})}$ is valid to a certain extent. The slight gap between ${\rm cond}_2(\tilde{\boldsymbol{R}})$ and $\sqrt{{\rm cond}_2(\boldsymbol{R})}$ is due to the fact that, under the coherence criterion, the dictionary data distribute non-uniformly in a strict sense.

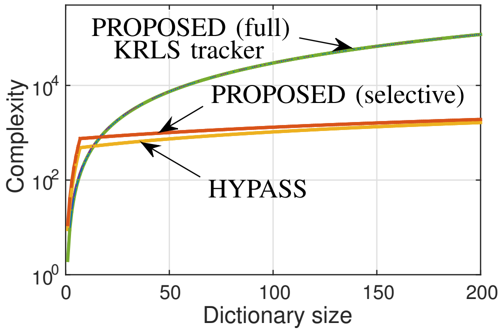


Fig. 2. Computational complexity for $L=2, s=7$.

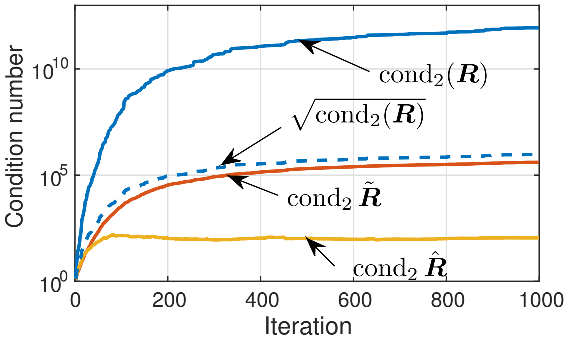
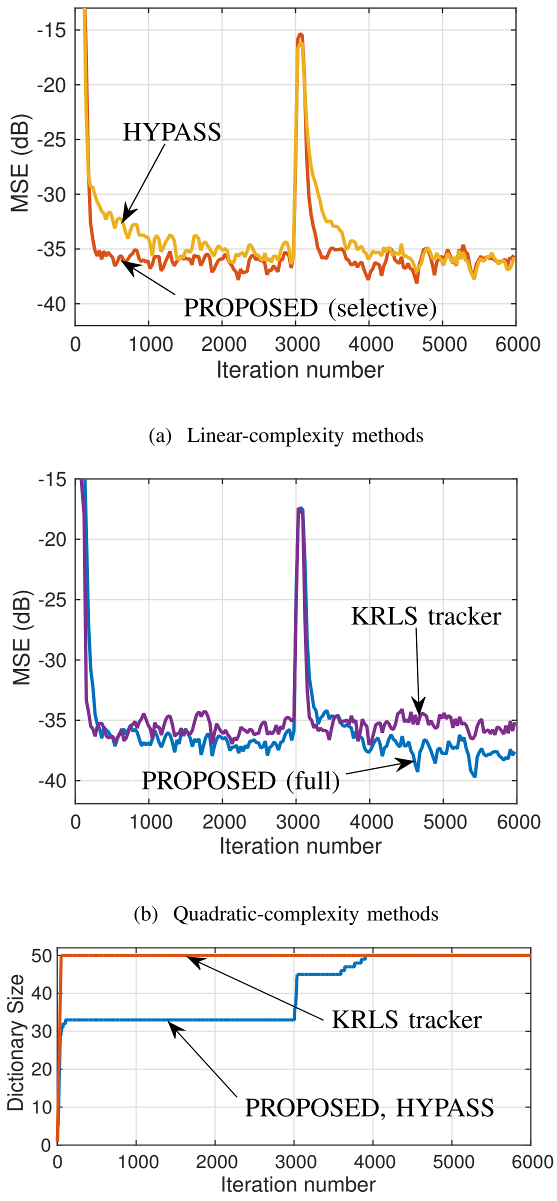


Fig. 3. Eigenvalue spreads of $\boldsymbol{R}$, $\tilde{\boldsymbol{R}}$, and $\hat{\boldsymbol{R}}$ (proposed).

B. *Experiment B: MSE Learning Curves*

We consider online prediction of the nonstationary time series data generated by the following equation (cf. [9]): $d_n := (0.8-0.5\exp(-d_{n-2}^2))d_{n-1} - (0.3+0.9\exp(-d_{n-1}^2))d_{n-2} + 0.1\sin(d_{n-1}\pi)$ for $0 \le n \le 3000$ ($d_{-2} := d_{-1} := 0.1$) and $d_n := (0.8-0.5\exp(-d_{n-1}^2))d_{n-1} - (0.3+0.9\exp(-d_{n-1}^2))d_{n-2} + 0.1\sin(d_{n-1}\pi)$ for $n > 3000$. In this experiment, we predict the output d_n with the input vector $\boldsymbol{u}_n := [d_{n-1},d_{n-2}]^\top$. The noise is white Gaussian with the signal to noise ratio (SNR) 40 dB. We use the Gaussian kernel with the scale parameter $\sigma = 0.3$. The proposed algorithm is compared with HYPASS and KRLS tracker [5]. For the proposed selective updating algorithm and HYPASS, the cardinality of the selected dictionary is set to $s = 7$. The dictionaries of the proposed algorithm and HYPASS are chosen by the coherence criterion with the threshold $\delta = 0.8$. The bound of the dictionary size of KRLS tracker is set to $M = 50$.



(a) Linear-complexity methods

(b) Quadratic-complexity methods

(c) Dictionary size

Fig. 4. Results for Experiment B.

Fig. 4(a) depicts the MSE learning curves of the linear-complexity methods: the proposed selective updating algorithm and HYPASS. It can be seen that the proposed algorithm exhibits faster convergence and tracking performance than HYPASS. Fig. 4(b) depicts the results of the quadratic-complexity methods: the proposed full updating algorithm and KRLS tracker. The simulation result shows that the proposed algorithm exhibits the same tracking speed as KRLS tracker, and achieves the better steady-state performance. The evolutions of the dictionary sizes are shown in Fig. 4(c).

V. Conclusion

This paper proposed the ideal metric from the viewpoint of whitening, and presented the kernel adaptive filtering algorithm with this metric. We showed that the proposed metric reduced the eigenvalue spread of the autocorrelation matrix. Since the error contours of the proposed algorithm are governed by the autocorrelation matrix of the whitened kernelized input vectors, the algorithm converges uniformly. Numerical examples validated the approximation $\boldsymbol{R} \approx \frac{1}{r}\boldsymbol{G}^2$ for the growing dictionary, and showed that the proposed algorithm significantly outperformed the existing kernel adaptive filtering algorithms.

Acknowledgment

This work was partially supported by JSPS Grants-in-Aid (15K06081, 15H02757). K.-R. Müller is thankful for partial funding in BK21 through NRF.

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, 2004.

[2] W. Liu, P. P. Pokharel, and J. C. Principe, “The kernel least-mean-square algorithm,” *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, 2008.

[3] W. Liu and J. Príncipe, “Kernel affine projection algorithms,” *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, pp. 1–12, 2008.

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

[5] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, “Kernel recursive least-squares tracker for time-varying regression,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1313–1326, 2012.

[6] M. Yukawa, “Multikernel adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4672–4682, 2012.

[7] M. Yukawa, “Adaptive learning in Cartesian product of reproducing kernel Hilbert spaces,” *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6037–6048, 2015.

[8] M. Takizawa and M. Yukawa, “Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space,” *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4257–4269, 2015.

[9] 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, 2009.

[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] S. Haykin, “Adaptive filter theory,” *2nd. ed., Prentice-Hall, Englewood Cliffs, NJ*, 1991.

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

[13] D. G. Luenberger, *Optimization by vector space methods*, John Wiley & Sons, 1997.

[14] M. Yukawa and K.-R. Müller, “Why does a Hilbertian metric work efficiently in online learning with kernels?,” submitted for publication.

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

[16] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*, Springer Science & Business Media, 2011.

[17] A. J. Smola and B. Schölkopf, *Learning with kernels*, Citeseer, 1998.

[18] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An introduction to kernel-based learning algorithms,” *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, 2001.

[19] M. Takizawa and M. Yukawa, “Efficient dictionary-refining kernel adaptive filter with fundamental insights,” *IEEE Trans. Signal Process.*, 2016, to appear.

[20] M. Takizawa, M. Yukawa, and C. Richard, “A stochastic behavior analysis of stochastic restricted-gradient descent algorithm in reproducing kernel Hilbert spaces,” in *Proc. ICASSP*, 2015, pp. 2001–2005.

[21] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge university press, 2012.