On Adaptivity of Online Model Selection Method Based on Multikernel Adaptive Filtering

Masahiro Yukawa * and Ryu-ichiro Ishii †
*Department of Electronics and Electrical Engineering, Keio University, Japan
†Department of Electrical and Electronics Engineering, Niigata University, Japan

***Abstract*— We investigate adaptivity of the online model selection method which has been proposed recently within the multikernel adaptive filtering framework. Specifically, we consider a situation in which the nonlinear system under study changes during adaptation and an appropriate kernel also does accordingly. Our time-varying cost functions involve three regularizers: the ℓ_1 norm and two block ℓ_1 norms which promote sparsity both in the kernel and data groups. The block ℓ_1 regularizers are approximated by their Moreau envelopes, and the adaptive proximal forward-backward splitting (APFBS) method is applied to the approximated cost function. Numerical examples show that the proposed algorithm can adaptively estimate a reasonable model.**

I. Introduction

Kernel adaptive filtering is an attractive way of extending linear adaptive filtering algorithms to the nonlinear case. The early study of kernel adaptive filtering has been motivated by the success of kernel methods in batch settings, such as support vector machine, Gaussian processes, and regularization networks [1]. In the classical linear adaptive filtering, the system is modeled as a Euclidean vector, and the ith element h_i of the vector (filter) represents a coefficient of a standard basis vector $\boldsymbol{e}_i$ (which has one at the ith position and zeros elsewhere); i.e.,

$$\boldsymbol{h} := [h_1, h_2, \cdots, h_N]^{\mathsf{T}} = \sum_{i=1}^{N} h_i \boldsymbol{e}_i. \tag{1}$$

In the kernel adaptive filtering, on the other hand, the system is modeled as an element of a functional space $\mathcal{H}$ called *a reproducing kernel Hilbert space (RKHS)*, which is a Hilbert space equipped with a reproducing kernel as well as an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ [2,3]. Specifically, an element of a RKHS can be written as

$$\varphi(\boldsymbol{x}) = \sum_{i=1}^{N} h_i \kappa(\boldsymbol{x}, \boldsymbol{u}_i), \ \ \boldsymbol{x} \in \mathcal{U} \subset \mathbb{R}^L, \tag{2}$$

where $\mathcal{U}$ is a compact subset of $\mathbb{R}^L$ (which is reffered to as the input space), $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a reproducing kernel which has the reproducing property $\varphi(\boldsymbol{x}) = \langle \varphi, \kappa(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$ for any $\boldsymbol{x} \in \mathcal{U}$. A popular example of reproducing kernel is the Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) := e^{-\alpha(\boldsymbol{x}-\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x}-\boldsymbol{y})}$, where $\alpha > 0$ is the kernel parameter. In the case of Gaussian kernel, $\kappa(\boldsymbol{x}, \boldsymbol{u}_i)$, given a vector $\boldsymbol{u}_i \in \mathcal{U}$, is a Gaussian function of $\boldsymbol{x}$ centered at $\boldsymbol{u}_i$, and the coefficients h_is determine the heights of the Gaussian functions centered at different points $\boldsymbol{u}_i$s. Comparing (1) and (2), it is seen that $\kappa(\boldsymbol{x}, \boldsymbol{u}_i)$s serve as basis vectors like $\boldsymbol{e}_i$s. The set of $\kappa(\boldsymbol{x}, \boldsymbol{u}_i)$s is called *dictionary*, and it is constructed based on observed data during adaptation in practice.

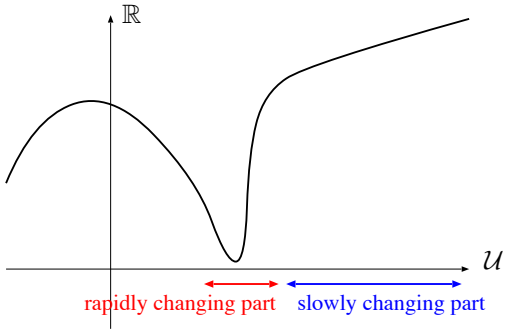


Fig. 1. An illustration of nonlinear system.

How can we choose the kernel? This would be a natural question and has been the main theme of the author's prior work in [4–6]. Modeling is always an important fraction of science and engineering. Under an improper model, there is no chance to obtain a good result. There is no perfect model probably in real world applications, and what people can do is to choose a reasonable model under, for instance, the principle of parsimony also known as Occam's Razor. Although a Gaussian kernel with a large kernel parameter α (i.e., use of a 'narrow' Gaussian) has a capability to express rapidly changing parts in the nonlinear system (see Fig. 1), it requires a large number of center points $\boldsymbol{u}_i$ (i.e., N has to be quite large in (2)) to cover the input space $\mathcal{U}$. (One needs to take many center points to express slowly changing parts in the nonlinear function.) This increases computational costs unreasonably and should be prevented in online settings. In contrast, it is obvious that a Gaussian kernel with a small α (i.e., use of a 'wide' Gaussian) cannot express rapidly changing parts in the system. The availability of a Gaussian kernel that fits well with the system therefore seems to be a strong assumption in some practical scenarios. Nevertheless, this has mostly been a common premise in the literature of kernel adaptive filtering [1,7–12]. In [6], the author has proposed multikernel adaptive filtering which employs multiple different kernels, e.g., 'wide' and 'narrow' Gaussian kernels, linear and Gaussian kernels, etc. The multikernel adaptive filtering provides a systematic way to allocate an appropriate kernel to each center point. In order to make this approach more attractive, one may employ

many kernels, say fifty or even hundreds of kernels. In this case, the model becomes very complex and may cause an overfitting issue. In [13], the authors have addressed this issue and developed an adaptive algorithm which can systematically suppress inappropriate kernels and select appropriate ones in online fashion, thereby preventing the overfitting issue.

The key word in this paper is *adaptivity in online model selection*. We consider a situation in which the nonlinear system changes during adaptation and an appropriate kernel also does accordingly. Our multikernel adaptive filtering algorithm has three regularizers: the ℓ_1 norm and two block ℓ_1 norms. One of the block ℓ_1 norms is for kernel groups, contributing to nulling the coefficients of such kernels that are unsuitable for the learning task. A proper model is thus selected, alleviating the overfitting problem. The other one is for data groups, contributing to nulling the coefficients of such dictionary data that are less relevant to the learning task than the others. The dictionary data are thus updated in an adaptive manner. Our algorithm is based on the adaptive proximal forward-backward splitting (APFBS) method [14]. To apply it, we approximate the block ℓ_1 regularizers by their *Moreau envelopes*. The resultant cost function contains smooth convex functions and a single nonsmooth convex function, to which APFBS can be applied directly. Numerical examples show that the proposed algorithm can adaptively estimate a reasonable model.

II. Background

Throughout the paper, let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}^*$ denote the sets of all real numbers, nonnegative integers, and positive integers, respectively. We consider online scenarios in which input vectors $(\boldsymbol{u}_n)_{n\in\mathbb{N}} \subset \mathcal{U}$ arrive sequentially and the response $d_n \in \mathbb{R}$, $n \in \mathbb{N}$, is a nonlinear function of the input vector $\boldsymbol{u}_n$. The task of nonlinear adaptive filtering is to find and/or track the time-variable nonlinear function (the *estimandum*) in an online fashion with the sequentially arriving measurements $(\boldsymbol{u}_n, d_n)_{n\in\mathbb{N}}$.

We consider the case that a proper model for the *estimandum* is unknown. A practical approach in this case is to use many possible kernels under the multikernel adaptive filtering framework [6]. Let $\kappa_m : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$, $m \in \mathcal{M} := \{1, 2, \cdots, M\}$, denote the set of positive definite kernels to be used. Let $\{\kappa_m(\cdot, \boldsymbol{u}_j)\}_{m\in\mathcal{M}, j\in\mathcal{J}_n}$ be the dictionary indicated by the dictionary index set $\mathcal{J}_n := \{j_1^{(n)}, j_2^{(n)}, \cdots, j_{r_n}^{(n)}\} \subset \{0, 1, \cdots, n-1\}$, where $r_n \in \mathbb{N}^*$ is the size of the dictionary index set $\mathcal{J}_n$. A multikernel adaptive filter is then given by

$$\phi_n(\boldsymbol{u}) := \underbrace{\sum_{m\in\mathcal{M}} \sum_{j\in\mathcal{J}_n} h_{j,n}^{(m)} \kappa_m(\boldsymbol{u}, \boldsymbol{u}_j)}_{\text{the } m\text{th model}}, \ \boldsymbol{u} \in \mathcal{U} \tag{3}$$

where $h_{j,n}^{(m)} \in \mathbb{R}, m \in \mathcal{M}, j \in \mathcal{J}_n$. Define an inner product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle := \mathrm{tr}(\boldsymbol{A}^\mathsf{T}\boldsymbol{B})$, where $(\cdot)^\mathsf{T}$ and $\mathrm{tr}(\cdot)$ stand for *transpose* and *trace*, respectively. Its induced norm is defined as $\|\boldsymbol{A}\| := \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle}$ for any matrix $\boldsymbol{A}$. Then, an estimate of d_n can be expressed simply in a matrix form as follows:

$$\widehat{d}_n := \phi_n(\boldsymbol{u}_n) = \langle \boldsymbol{H}_n, \boldsymbol{K}_n \rangle \tag{4}$$

where

$$\begin{aligned}
\boldsymbol{H}_n &:= \left[\boldsymbol{h}_{j_1^{(n)},n}\ \boldsymbol{h}_{j_2^{(n)},n} \cdots \boldsymbol{h}_{j_{r_n}^{(n)},n}\right] \in \mathbb{R}^{M\times r_n} \\
\boldsymbol{K}_n &:= \left[\boldsymbol{k}_{j_1^{(n)},n}\ \boldsymbol{k}_{j_2^{(n)},n} \cdots \boldsymbol{k}_{j_{r_n}^{(n)},n}\right] \in \mathbb{R}^{M\times r_n} \\
\boldsymbol{h}_{j,n} &:= \left[h_{j,n}^{(1)}, h_{j,n}^{(2)}, \cdots, h_{j,n}^{(M)}\right]^\mathsf{T} \in \mathbb{R}^M \\
\boldsymbol{k}_{j,n} &:= \left[\kappa_1(\boldsymbol{u}_n, \boldsymbol{u}_j), \kappa_2(\boldsymbol{u}_n, \boldsymbol{u}_j), \cdots, \kappa_M(\boldsymbol{u}_n, \boldsymbol{u}_j)\right]^\mathsf{T} \in \mathbb{R}^M.
\end{aligned}$$

Those readers who are not familiar with convex analysis may refer to the appendix before proceeding to the following section.

III. Proposed Adaptive Algorithm

The size and associated data indices of the coefficient matrix $\boldsymbol{H}_n \in \mathbb{R}^{M\times r_n}$ depend on the dictionary index set $\mathcal{J}_n$ and are therefore time dependent. The cost function to be considered is thus a function of a matrix in $\mathbb{R}^{M\times r_{n+1}}$ (not in $\mathbb{R}^{M\times r_n}$). We define the following cost function:

$$\Theta_n(\boldsymbol{X}) := \underbrace{\varphi_n(\boldsymbol{X})}_{\text{smooth}} + \underbrace{\psi_n^{(1)}(\boldsymbol{X}) + \psi_n^{(2)}(\boldsymbol{X}) + \psi_n^{(3)}(\boldsymbol{X})}_{\text{proximable}},$$

$$\begin{aligned}
\boldsymbol{X} &:= \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,r_{n+1}} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,r_{n+1}} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M,1} & x_{M,2} & \cdots & x_{M,r_{n+1}} \end{bmatrix} =: \begin{bmatrix} \boldsymbol{\xi}_1^\mathsf{T} \\ \boldsymbol{\xi}_2^\mathsf{T} \\ \vdots \\ \boldsymbol{\xi}_M^\mathsf{T} \end{bmatrix} \\
&=: \left[\boldsymbol{x}_1\ \boldsymbol{x}_2 \cdots \boldsymbol{x}_{r_{n+1}}\right] \in \mathbb{R}^{M\times r_{n+1}}
\end{aligned}$$

where

$$\begin{aligned}
\varphi_n(\boldsymbol{X}) &:= \frac{1}{2} d^2(\boldsymbol{X}, C_n) \\
\psi_n^{(1)}(\boldsymbol{X}) &:= \lambda_1 \sum_{i=1}^{r_{n+1}} w_i^{(n)} \|\boldsymbol{x}_i\| \\
\psi_n^{(2)}(\boldsymbol{X}) &:= \lambda_2 \sum_{m=1}^{M} \nu_m^{(n)} \|\boldsymbol{\xi}_m\| \\
\psi_n^{(3)}(\boldsymbol{X}) &:= \lambda_3 \sum_{i=1}^{r_{n+1}} \sum_{m=1}^{M} \mu_{m,i}^{(n)} |x_{m,i}|.
\end{aligned}$$

Here, $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are the regularization parameters, $w_i^{(n)}, \nu_m^{(n)}, \mu_{m,i}^{(n)} > 0$ are the weights, and

$$d(\boldsymbol{X}, C_n) := \min_{\boldsymbol{Y}\in C_n} \|\boldsymbol{X} - \boldsymbol{Y}\| \tag{5}$$

is the metric distance between a point $\boldsymbol{X} \in \mathbb{R}^{M\times r_{n+1}}$ and the set

$$C_n := \left\{\boldsymbol{X} \in \mathbb{R}^{M\times r_{n+1}} : |\varepsilon_n(\boldsymbol{X})| \leq \rho\right\}, \tag{6}$$

where $\rho \geq 0$ and $\varepsilon_n(\boldsymbol{X}) := \left\langle \boldsymbol{X}, \widetilde{\boldsymbol{K}}_n \right\rangle - d_n$. Here, $\widetilde{\boldsymbol{K}}_n \in \mathbb{R}^{M\times r_{n+1}}$ consists of (i) those column vectors of $\boldsymbol{K}_n$ whose associated indices are included in the new dictionary index set

$\mathcal{J}_{n+1}$ and (ii) $\boldsymbol{k}_{n,n}$ at the rightmost column if $n \in \mathcal{J}_{n+1}$. (The formal definition of $\widetilde{\boldsymbol{K}}_n$ is given later on.) Each term of the cost function plays the following role.

(a) $\varphi_n(\boldsymbol{X})$ contributes to reducing empirical risks.
(b) $\psi_n^{(1)}$ is the block ℓ_1 norm for data groups, promoting *column-wise sparsity* to select relevant data.
(c) $\psi_n^{(2)}$ is the block ℓ_1 norm for kernel groups, promoting *row-wise sparsity* to select relevant kernels. This term is particularly important in terms of model selection.
(d) $\psi_n^{(3)}$ is the weighted ℓ_1 norm which promotes sparsity of the coefficient matrix. In particular, it will lead to selecting relevant kernels for each data point.

The first term φ_n is a differentiable convex function having a Lipschitz continuous gradient. On the other hand, the terms $\psi_n^{(1)}$, $\psi_n^{(2)}$, and $\psi_n^{(3)}$ are nondifferentiable but *convex and proximable*. Here, *proximable* means that the proximity operator can be computed easily (see the appendix). The adaptive proximal forward-backward splitting algorithm [14] can suppress a sequence of functions each of which is a sum of a smooth function and a single proximable function. We thus approximate $\psi_n^{(1)}$ and $\psi_n^{(2)}$ and consider the following cost function:

$$\widetilde{\Theta}_n(\boldsymbol{X}) := \underbrace{\varphi_n(\boldsymbol{X}) + {}^{\gamma_1}\psi_n^{(1)}(\boldsymbol{X}) + {}^{\gamma_2}\psi_n^{(2)}(\boldsymbol{X})}_{\text{smooth}} + \underbrace{\psi_n^{(3)}(\boldsymbol{X})}_{\text{proximable}},$$

where $\gamma_1, \gamma_2 \in (0, \infty)$. The gradient of

$$g_n(\boldsymbol{X}) := \varphi_n(\boldsymbol{X}) + {}^{\gamma_1}\psi_n^{(1)}(\boldsymbol{X}) + {}^{\gamma_2}\psi_n^{(2)}(\boldsymbol{X}) \tag{7}$$

is β-Lipschitz continuous with

$$\beta := 1 + \frac{1}{\gamma_1} + \frac{1}{\gamma_2} > 1. \tag{8}$$

Now we show how to suppress the sequence of cost functions $(\widetilde{\Theta}_n)_{n\in\mathbb{N}}$. We define the modified matrices $\widetilde{\boldsymbol{H}}_n \in \mathbb{R}^{M\times r_{n+1}}$ and $\widetilde{\boldsymbol{K}}_n \in \mathbb{R}^{M\times r_{n+1}}$ with their (m,i) entries given respectively by

$$[\widetilde{\boldsymbol{H}}_n]_{m,i} := h^{(m)}_{j_i^{(n+1)},n} \tag{9}$$

$$[\widetilde{\boldsymbol{K}}_n]_{m,i} := \kappa_m\left(\boldsymbol{u}_n, \boldsymbol{u}_{j_i^{(n+1)}}\right). \tag{10}$$

The modified matrix $\widetilde{\boldsymbol{H}}_n$ consists of a submatrix of $\boldsymbol{H}_n$ eliminating some columns with minor contributions and possibly a new entry $\boldsymbol{h}_{n,n} := \boldsymbol{0}$ at the rightmost column if $n \in \mathcal{J}_{n+1}$. The dictionary is initialized as $\mathcal{J}_0 := \{0\}$. Let $\widetilde{\boldsymbol{H}}_0 := \boldsymbol{h}_{0,0} = \boldsymbol{0}$. The proposed algorithm is then given by

$$\boldsymbol{H}_{n+1} := \operatorname{prox}_{\eta\psi_n^{(3)}}\left[\widetilde{\boldsymbol{H}}_n - \eta\nabla g_n(\widetilde{\boldsymbol{H}}_n)\right],\ n \in \mathbb{N}, \tag{11}$$

where $\eta \in (0, 2/\beta) \subset (0, 2)$ is the step size and

$$\nabla\varphi_n(\widetilde{\boldsymbol{H}}_n) = \widetilde{\boldsymbol{H}}_n - P_{C_n}(\widetilde{\boldsymbol{H}}_n) \tag{12}$$

with

$$P_{C_n}(\widetilde{\boldsymbol{H}}_n) = \widetilde{\boldsymbol{H}}_n - \operatorname{sign}(\varepsilon_n(\widetilde{\boldsymbol{H}}_n))\frac{\max\{|\varepsilon_n(\widetilde{\boldsymbol{H}}_n)| - \rho, 0\}}{\|\widetilde{\boldsymbol{K}}_n\|^2}\widetilde{\boldsymbol{K}}_n.$$

Finally, the proximity operators are given by

$$\operatorname{prox}_{\gamma_1\psi_n^{(1)}}(\boldsymbol{X}) = \sum_{i=1}^{r_{n+1}} \max\left\{1 - \frac{\lambda_1\gamma_1 w_i^{(n)}}{\|\boldsymbol{x}_i\|}, 0\right\}\boldsymbol{x}_i\boldsymbol{e}_{i,r_{n+1}}^{\mathsf{T}},$$

$$\operatorname{prox}_{\gamma_2\psi_n^{(2)}}(\boldsymbol{X}) = \sum_{m=1}^{M} \max\left\{1 - \frac{\lambda_2\gamma_2\nu_m^{(n)}}{\|\boldsymbol{\xi}_m\|}, 0\right\}\boldsymbol{e}_{m,M}\boldsymbol{\xi}_m^{\mathsf{T}},$$

$$\operatorname{prox}_{\eta\psi_n^{(3)}}(\boldsymbol{X}) = \sum_{i=1}^{r_{n+1}}\sum_{m=1}^{M} \max\left\{1 - \frac{\lambda_3\eta\mu_{m,i}^{(n)}}{|x_{m,i}|}, 0\right\}\boldsymbol{E}_{m,i}x_{m,i}.$$

Here, $\boldsymbol{e}_{p,q}$, $p, q \in \mathbb{N}^*$, is a length-q unit vector that has one at its pth entry and zeros elsewhere, and $\boldsymbol{E}_{m,i}$ is an $M \times r_{n+1}$ matrix that has one at its (m,i) entry and zeros elsewhere.

To keep the dictionary size bounded by some constant $r_{\max} \in \mathbb{N}$, we use the following sparsification strategy: (i) add each new datum into the dictionary as long as the dictionary size is smaller than $r_{\max}$, and (ii) discard those data which have minor contributions to estimation. To be precise, the dictionary index set is updated as follows:

$$\mathcal{J}_{n+1} := \begin{cases} \mathcal{J}^n_{\neq\boldsymbol{0}} \cup \{n\}, & \text{if } \left|\mathcal{J}^n_{\neq\boldsymbol{0}}\right| < r_{\max}, \\ \mathcal{J}^n_{\neq\boldsymbol{0}}, & \text{if } \left|\mathcal{J}^n_{\neq\boldsymbol{0}}\right| = r_{\max}, \end{cases} \quad n \in \mathbb{N}, \tag{13}$$

where

$$\mathcal{J}^n_{\neq\boldsymbol{0}} := \{j \in \mathcal{J}_n :\ \boldsymbol{h}_{j,n} \neq \boldsymbol{0}\}.$$

One may also use a sparsification strategy based on the coherence criterion for an admission test as in [9,12,13]. However, to determine the coherence threshold reasonably in the sense of achieving good performance as well as keeping the dictionary size bounded, some information about the range of input data would be required. An advantage of the present strategy is that, without such information, it yields good performance while keeping the dictionary size bounded by $r_{\max}$.

The proximity operators $\operatorname{prox}_{\gamma_1\psi_n^{(1)}}$ and $\operatorname{prox}_{\gamma_2\psi_n^{(2)}}$ shrink those column and row vectors of $\widetilde{\boldsymbol{H}}_n$ which have minor contributions in estimation. However, it is not ensured that the gradient operation in (11) *completely* nullify such column and row vectors. Note here that $\eta\nabla^{\gamma_1}\psi_n^{(1)}(\widetilde{\boldsymbol{H}}_n) = \frac{\eta}{\gamma_1}\left(\widetilde{\boldsymbol{H}}_n - \operatorname{prox}_{\gamma_1\psi_n^{(1)}}(\widetilde{\boldsymbol{H}}_n)\right)$ and $\frac{\eta}{\gamma_1} < \frac{2}{\beta\gamma_1} < 2$; the same applies to ${}^{\gamma_2}\psi_n^{(2)}$. Nevertheless, the final operation $\operatorname{prox}_{\eta\psi_n^{(3)}}$ attracts nearly-zero components to zero and this assists minor row and column vectors to vanish completely.

IV. Numerical Examples

We conduct simulations in an estimation task of nonlinear function with an abrupt change for $L = 1$ to show the efficacy of the proposed algorithm. We test 100 independent trials and, at each trial $t = 1, 2, \cdots, 100$, the data is generated as $d_n^{(t)} := \psi_n(\boldsymbol{u}_n^{(t)}) + v_n^{(t)}$, $n \in \mathbb{N}$, with $\psi_n(\boldsymbol{x}) := \exp\left(-2\|\boldsymbol{x} - 0.2\|^2\right)$ for $n \le 20{,}000$ $\psi_n(\boldsymbol{x}) := -\exp\left(-20\|\boldsymbol{x} - 0.1\|^2\right) - 2\exp\left(-20\|\boldsymbol{x} - 0.8\|^2\right)$ for $n >$

$20,000$, Here, each component of the input vector $\boldsymbol{u}_n^{(t)}$ obeys the i.i.d. uniform distribution between 0 and 1. It is supposed that the data are contaminated by impulsive noise, at iterations $10,000$ and $30,000$, of amplitude 100, and by Gaussian noise obeying $\mathcal{N}(0, 0.1)$ at the other iterations. Totally $M = 45$ Gaussian kernels are employed with the kernel parameters $a \times 10^b$, $a \in \{1, 2, \cdots, 9\}$, $b \in \{-2, -1, 0, 1, 2\}$. To be precise, $\kappa_m(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-\zeta_m \|\boldsymbol{x} - \boldsymbol{y}\|^2\right)$, $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$, $m \in \mathcal{M}$, where $\zeta_1 = 0.01$, $\zeta_2 = 0.02$, $\cdots$, $\zeta_{45} = 900$.

The parameters for the proposed algorithm are set to $\rho = 0$, $\eta = 0.1$, and $r_{\max} = 25$. The regularization parameters λ_1, λ_2, and λ_3 are controlled adaptively as $\lambda_1 = \lambda_2 = \lambda_n$ and $\lambda_3 = 0.1\lambda_n$ with $\lambda_n := \max_{m,j} \left|h_{j,n}^{(m)}\right|$, $n \in \mathbb{N}$. The index γ_1 and γ_2 of the Moreau envelopes are chosen as $\gamma_1 := \gamma_2 := 2(2/(\eta + \epsilon_\gamma) - 1)^{-1} > 0$, where $\epsilon_\gamma := 10^{-5} < 2 - \eta$ and it is automatically guaranteed that $\eta \in (0, 2/\beta)$. The weight design is based on the idea of the iteratively reweighted least squares (IRLS) [15]. Specifically, $w_i^{(n)} = \dfrac{r_{n+1}\hat{w}_i^{(n)}}{\sum_{\iota=1}^{r_{n+1}} \hat{w}_\iota^{(n)}}$, $i = 1, 2, \cdots, r_{n+1}$, $n \in \mathbb{N}$, where $\hat{w}_i^{(n)} = \dfrac{1}{\check{h}_{i,n}^{1-p} + \epsilon}$ with $\epsilon = 10^{-6}$, $p = 0.5$, and $\check{h}_{i,n} = \max_{m \in \mathcal{M}} \left|h_{j_i^{(n+1)},n}^{(m)}\right|$. Analogously, $\nu_m^{(n)} = \dfrac{M\hat{\nu}_m^{(n)}}{\sum_{l \in \mathcal{M}} \hat{\nu}_l^{(n)}}$, $m \in \mathcal{M}$, $n \in \mathbb{N}$, where $\hat{\nu}_m^{(n)} = \dfrac{1}{\acute{h}_{m,n}^{1-p} + \epsilon}$ with $\acute{h}_{m,n} = \max_{i=1,2,\cdots,r_{n+1}} \left|h_{j_i^{(n+1)},n}^{(m)}\right|$, and $\mu_{m,i}^{(n)} = \dfrac{r_{n+1}M\hat{\mu}_{m,i}^{(n)}}{\sum_{l \in \mathcal{M}} \sum_{\iota=1}^{r_{n+1}} \hat{\mu}_{l,\iota}^{(n)}}$, where $\hat{\mu}_{m,i}^{(n)} = \dfrac{1}{\left|h_{j_i^{(n+1)},n}^{(m)}\right|^{1-p} + \epsilon}$. We compare the proposed algorithm with the KNLMS-BT (kernel normalized least mean square with block soft-thresholding sparsification) algorithm, a single kernel method, for the parameters $\eta = 0.1$, $\zeta = 3$ (a kernel parameter which gives the best performance during the first 20,000 iterations (before the abrupt change of the nonlinear system) in this experiment), $\lambda = 0.05$, $r_{\max} = 25$, $r_{\min} = 23$, $\rho = 0$, and $\epsilon_w = 10^{-5}$.

The MSE learning curves are plotted in Fig. 2. It is seen that the proposed algorithm outperforms the single kernel method after the abrupt change of the system. It should be mention that the observed poor performance of the single kernel method is due to the mismatch between the employed kernel and the nonlinear system. To show that the proposed algorithm adapts the model to the system change, we show the norms of each row vectors of $\boldsymbol{H}_n$ at (a) $n = 20,000$ (right before the system change) and (b) $n = 40,000$ (at the end of the adaptation) in Fig. 3. Comparing Fig. 3(a) and Fig. 3(b), it is seen that the mean of the distribution shifts to the right. Indeed, the nonlinear system for $n \leq 20,000$ is a single Gaussian function with its variance corresponding to the 20th kernel $\zeta_{20} = 2.0$. On the other hand, the nonlinear system for $n > 20,000$ is composed of two Gaussian functions with their variances corresponding to the 29th kernel $\zeta_{29} = 20$. This means that an appropriate kernel should be around the 20th kernel for $n \leq 20,000$ and around the 29th kernel for $n > 20,000$. We can see that the model selected by the proposed algorithm is more or less reasonable.

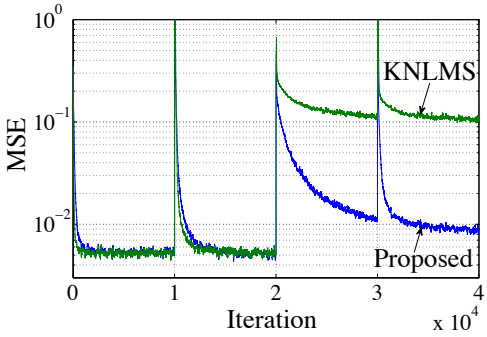
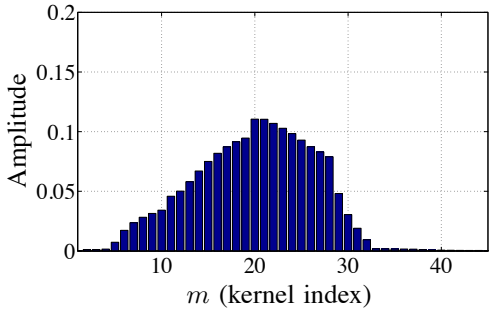
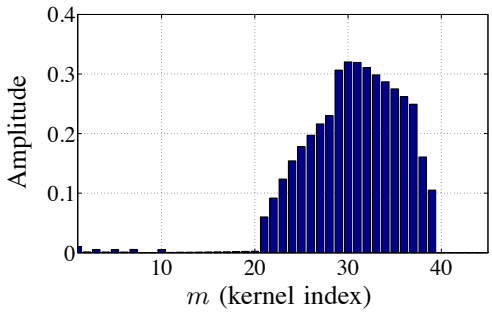


Fig. 2. MSE learning curves. The nonlinear system changes at the $20,000$th iteration and impulsive noise is added at the $10,000$th and $30,000$th iterations.



(a) 20,000th iteration



(b) 40,000th iteration

Fig. 3. Adaptivity in online model selection.

V. Conclusion

This paper investigated adaptivity of the online model selection method which is based on the multikernel adaptive filtering framework. Specifically, we considered a situation in which the nonlinear system under study changes during adaptation and an appropriate kernel also does accordingly. Our cost function involved three regularizers: the ℓ_1 norm and two block ℓ_1 norms which promote sparsity both in the kernel and data groups. The block ℓ_1 regularizers were approximated by their Moreau envelopes, and the adaptive proximal forward-backward splitting (APFBS) method was applied to the ap-

proximated cost function. Numerical examples showed that the proposed algorithm can adaptively estimate a reasonable model when the nonlinear system changes.

Acknowledgements: This work was supported by KDDI Foundation.

APPENDIX — MATHEMATICAL INGREDIENTS

This appendix provides some mathematical tools and notions coming from convex analysis and fixed point theory of nonexpansive mapping [16,17]. We use the notation $\mathcal{H} := \mathbb{R}^{M\times r_{n+1}}$ since everything can be defined and discussed in the general Hilbert space.

Convex function: A function $f : \mathcal{H} \to (-\infty,\infty] := \mathbb{R}\cup\{\infty\}$ is said to be convex on $\mathcal{H}$ if

$$f(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}) \le \alpha f(\boldsymbol{x})+(1-\alpha)f(\boldsymbol{y}), \qquad (14)$$

$\forall(\boldsymbol{x},\boldsymbol{y},\alpha)\in\mathcal{H}\times\mathcal{H}\times[0,1]$.

Strictly convex function: A function $f : \mathcal{H} \to (-\infty,\infty]$ is said to be strictly convex on $\mathcal{H}$ if (14) holds with strict inequality for any $\boldsymbol{x}\neq\boldsymbol{y}$.

Proper convex function: If a convex function $f : \mathcal{H} \to (-\infty,\infty]$ is said to be proper if there exists an $\boldsymbol{x}\in\mathcal{H}$ such that $f(\boldsymbol{x})<\infty$.

Lower semicontinuous function: A function $f : \mathcal{H} \to (-\infty,\infty]$ is said to be *lower semicontinuous* on $\mathcal{H}$ if the set $\mathrm{lev}_{\le a}f := \{\boldsymbol{x}\in\mathcal{H} : f(\boldsymbol{x})\le a\}$ is closed for any $a\in\mathbb{R}$. Any continuous function is lower semicontinuous.

The set of all proper lower semicontinuous convex functions $\Gamma_0(\mathcal{H})$: The set of all proper lower semicontinuous convex functions from $\mathcal{H}$ to $(-\infty,\infty]$ is denoted commonly by $\Gamma_0(\mathcal{H})$.

Coercive function: A function $f\in\Gamma_0(\mathcal{H})$ is said to be coercive if $\|\boldsymbol{x}\|\to\infty$ implies $f(\boldsymbol{x})\to\infty$. Coercivity guarantees the existence of a minimizer of f.

Closed convex set: A subset C of $\mathcal{H}$ is said to be convex if $\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}\in C$, $\forall(\boldsymbol{x},\boldsymbol{y},\alpha)\in C\times C\times[0,1]$. A set S is said to be open if any point $\boldsymbol{x}\in S$ has its neighbor included by S; i.e., for any $\boldsymbol{x}\in S$, there exists some $\epsilon_{\boldsymbol{x}}>0$ such that $B(\boldsymbol{x},\epsilon_{\boldsymbol{x}}) := \{\boldsymbol{y}\in\mathcal{H} : \|\boldsymbol{x}-\boldsymbol{y}\|<\epsilon_{\boldsymbol{x}}\}\subset S$. A set S is said to be closed if it complement set $\mathcal{H}\setminus S$ is open. If a convex set is closed, it is said to be a closed convex set.

Metric projection: Let $C\subset\mathcal{H}$ be an arbitrary closed convex set. Then, for any point $\boldsymbol{x}\in\mathcal{H}$, its closest point in C exists uniquely, and the unique closest point

$$P_C(\boldsymbol{x}) := \underset{\boldsymbol{y}\in C}{\operatorname{argmin}} \|\boldsymbol{x}-\boldsymbol{y}\| \qquad (15)$$

is called the metric projection of $\boldsymbol{x}$ onto C. It is a generalization of the orthogonal projection.

Lipschitz continuous mapping: A mapping $T : \mathcal{H}\to\mathcal{H}$ is said to be Lipschitz continuous, or α-Lipschitz continuous to be more specific, if

$$\|T(\boldsymbol{x})-T(\boldsymbol{y})\| \le \alpha\|\boldsymbol{x}-\boldsymbol{y}\|,\ \forall\boldsymbol{x},\boldsymbol{y}\in\mathcal{H}, \qquad (16)$$

for some constant $\alpha>0$. Any Lipschitz continuous mapping is continuous.

Sum of Lipschitz continuous mappings: Let $T_1 : \mathcal{H}\to\mathcal{H}$ and $T_2 : \mathcal{H}\to\mathcal{H}$ be α_1-Lipschitz continuous and α_2-Lipschitz continuous mappings, respectively, for $\alpha_1>0$ and $\alpha_2>0$. Then, the sum $T := T_1+T_2$ is an $(\alpha_1+\alpha_2)$-Lipschitz continuous mapping. This can readily be verified by the triangular inequality.

Contractive mapping: A mapping $T : \mathcal{H}\to\mathcal{H}$ is said to be contractive if (16) holds for $0<\alpha<1$.

Nonexpansive mapping: A mapping $T : \mathcal{H}\to\mathcal{H}$ is said to be nonexpansive if (16) holds for $\alpha=1$.

Firmly nonexpansive mapping: A mapping $T : \mathcal{H}\to\mathcal{H}$ is said to be α-averaged nonexpansive if there exist a nonexpansive mapping $N : \mathcal{H}\to\mathcal{H}$ and a constant $\alpha\in(0,1)$ such that

$$T = (1-\alpha)I+\alpha N, \qquad (17)$$

where $I : \mathcal{H}\to\mathcal{H}, \boldsymbol{x}\mapsto\boldsymbol{x}$. A $\frac{1}{2}$-averaged nonexpansive mapping is particularly important and is specially called firmly nonexpansive. The metric projection is a typical example of firmly nonexpansive mapping.

Smooth function: A function $f\in\Gamma_0(\mathcal{H})$ is said to be smooth if it is differentiable and its gradient is Lipschitz continuous.

Moreau envelope: For any $f\in\Gamma_0(\mathcal{H})$,

$${}^{\gamma}f(\boldsymbol{x}) := \min_{\boldsymbol{y}\in\mathbb{R}^{M\times r_{n+1}}}\left(f(\boldsymbol{y})+\frac{1}{2\gamma}\|\boldsymbol{x}-\boldsymbol{y}\|^2\right),\ \gamma\in(0,\infty), \qquad (18)$$

is the Moreau envelope of f of index γ. One of the remarkable properties of the Moreau envelope is its smoothness, even though the original function f is not necessarily differentiable and could even be discontinuous.

Proximity operator (Proximal mapping): For any $f\in\Gamma_0(\mathcal{H})$,

$$\mathrm{prox}_{\gamma f}(\boldsymbol{x}) := \underset{\boldsymbol{y}\in\mathbb{R}^{M\times r_{n+1}}}{\operatorname{argmin}}\left(f(\boldsymbol{y})+\frac{1}{2\gamma}\|\boldsymbol{x}-\boldsymbol{y}\|^2\right). \qquad (19)$$

is the proximity operator (or the proximal mapping) of f of index γ. Here, the existence and the uniqueness of the minimizer are guaranteed respectively by the coercivity and strict convexity of the regularized function $f(\boldsymbol{y})+\frac{1}{2\gamma}\|\boldsymbol{x}-\boldsymbol{y}\|^2$. The proximity operator is known to be firmly nonexpansive.

Gradient of Moreau envelope: The Lipschitz continuous gradient of a Moreau envelope ${}^{\gamma}f(\boldsymbol{x})$ is given by

$$\boldsymbol{\nabla}{}^{\gamma}f(\boldsymbol{x}) = \frac{\boldsymbol{x}-\mathrm{prox}_{\gamma f}(\boldsymbol{x})}{\gamma}, \qquad (20)$$

which is $\frac{1}{\gamma}$-Lipschitz continuous.

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

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

[4] M. Yukawa, "On use of multiple kernels in adaptive learning —Extended reproducing kernel Hilbert space with Cartesian product," in *Proc. IEICE Signal Processing Symposium*, Nov. 2010, pp. 59–64.

[5] M. Yukawa, "Nonlinear adaptive filtering techniques with multiple kernels," in *Proc. EUSIPCO*, 2011, pp. 136–140.

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

[7] 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.

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

[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, Mar. 2009.

[10] K. Slavakis, S. Theodoridis, and I. Yamada, "Adaptive constrained learning in reproducing kernel Hilbert spaces: the robust beamforming case," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4744–4764, Dec. 2009.

[11] M. Takizawa and M. Yukawa, "An efficient data-reusing kernel adaptive filtering algorithm based on parallel hyperslab projection along affine subspaces," in *Proc. IEEE ICASSP*, 2013, pp. 3557–3561.

[12] W. Gao, J. Chen, C. Richard, J. Huang, and R. Flamary, "Kernel LMS algorithm with forward-backward splitting for dictionary learning," in *Proc. IEEE ICASSP*, 2013, pp. 5735–5739.

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

[14] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. IEEE ICASSP*, 2010, pp. 3734–3737.

[15] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[16] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York: NY, 1st edition, 2011.

[17] I. Yamada, M. Yukawa, and M. Yamagishi, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, vol. 49 of *Optimization and Its Applications*, chapter 17, pp. 345–390, Springer, New York, 2011.