EFFICIENT CONSTRUCTION OF DICTIONARIES FOR KERNEL ADAPTIVE FILTERING IN A DYNAMIC ENVIRONMENT

Taichi Ishida and Toshihisa Tanaka

Department of Electrical and Electronic Engineering, Tokyo University of Agriculture and Technology
2–14–16 Nakacho, Koganei-shi, Tokyo, 184–8588, Japan
Email: tishida@sip.tuat.ac.jp, tanakat@cc.tuat.ac.jp

ABSTRACT

One of the major challenges in kernel adaptive filtering is how to construct an efficient dictionary of observed input signals. In this paper, we propose novel dictionary adaptation rules for kernel adaptive filtering. The first algorithm can efficiently "move" elements of the dictionary to increase the approximation performance. The second algorithm mainly focuses on a nonstationary system, which can yield the increase of the dictionary size. The proposed method can eliminate unnecessary elements in the dictionary. Numerical examples support the efficacy of the proposed methods.

Index Terms— nonlinear adaptive filtering, kernel methods, reproducing kernel Hilbert space, dictionary learning

1. INTRODUCTION

Filters that approximate or track unknown systems that change from time to time are known as adaptive filters [1]. Adaptive filtering is a challenging technique in a wide range of signal processing and machine learning applications such as system identification, noise or echo cancellation, and signal prediction [2, 3]. The input-output relation of adaptive filters can be constructed by linear or nonlinear models. Most traditional methods assume that unknown systems are linear. However, there are many situations that require nonlinear adaptive filters, since many systems in the real environment are modeled nonlinear. To this end, a number of research results regarding nonlinear adaptive filters have been reported. A well-known one is the adaptive Volterra filter [4–6]. In recent years, the efficiency of kernel adaptive filters has also become known effective [7–17].

The kernel adaptive filter is a nonlinear adaptive filter that exploits kernel methods, which are one of the techniques to construct effective nonlinear systems with a reproducing kernel Hilbert space (RKHS) induced from a positive definite kernel [18]. The filter is an element in the RKHS and output of the system is modeled as the inner product of the filter with a nonlinear map of the input signal [7].

This work is supported in part by JSPS KAKENHI Grant Number 26280054.

Since a kernel adaptive filter is represented by the linear sum of kernels corresponding to observed input signals, the adaptive algorithm is intended to estimate coupling coefficients of kernels. Hence, as the number of observed input signals increases, the computational load increases due to linearly growing dimension of the subspace. Furthermore, the filter will be prone to overadaptation (or overfitting) due to increasing model dimension. Therefore, a reduction method of the number of kernels to design the filter has been proposed by constructing a set of input signals called a dictionary, using a coherence threshold [11]. This method judges whether to add the kernel corresponding to an observed input signal to the dictionary at each time instant. The observed input signal similar to a signal in the dictionary is discarded. This method makes it possible to prevent overadaptation and to reduce the computation time in updating the filter. In addition to the coherence-based method, reduction methods using approximate linear dependency (ALD) [12] and surprise criteria [19] are proposed. Moreover, l_1 regularization based methods have been proposed to construct a dictionary [8, 20–22].

The difficulty of constructing an efficient dictionary is that some of the observed signals can deteriorate performance in adaptation. A recently proposed solution is to "move" observed samples in the dictionary to increase the performance [23, 24]. In this method, elements in the dictionary constructed by coherence sparsification are updated to minimize the mean square error (MSE). Although the dictionary adaptation is an efficient strategy for increasing performance of kernel adaptive filtering, the dictionary cannot still catch up with the change of unknown systems. Moreover, the adjustment of the step size at each time instance in the dictionary adaptation has been proposed to prevent an increase of similar elements after the update [23, 24], however, it has inferior approximation.

In this paper, we propose two novel dictionary adaptation algorithms. The first algorithm is a modified version of [23, 24] that has superior approximation in the adjustment of the step size. The second algorithm is a novel dictionary adaptation algorithm for nonstationary systems. In the proposed method, we exploit a novel method for preventing the increase of similar elements, which removes updated dictionary elements that do not satisfy a coherence criteria. Furthermore, we apply the dictionary adaptation to a coherence-based dictionary with l_1 sparsity [22]. As a result, the proposed method can "move" and adaptively "eliminate" dictionary elements. Numerical examples support the efficacy of the proposed methods.

2. KERNEL ADAPTIVE FILTERS AND DICTIONARY CONSTRUCTION

2.1. Kernel Nonlinear Filtering Model

Let $\mathcal{U} \subset \mathbb{R}^L$, $\boldsymbol{u} \in \mathcal{U}$, and $d \in \mathbb{R}$ denote the input space, the input signal, and the desired signal respectively. Also, $\kappa(\cdot,\cdot) : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ and $\mathcal{H}$ denote the kernel and the corresponding RKHS. In kernel adaptive filtering, suppose that the output of the system is modeled as the inner product $f(\boldsymbol{u}) = \langle \Omega, \phi(\boldsymbol{u}) \rangle$ of the filter $\Omega \in \mathcal{H}$ with a nonlinear mapping of an input signal $\phi(\boldsymbol{u}) \in \mathcal{H}$. We consider the problem of adaptively estimating filter Ω. Figure 1 shows a conceptual diagram of the kernel adaptive filter. By representer theorem [11], Ω at time instant n can be written as

$$\Omega_n = \sum_{j \in \mathcal{J}_n} h_{j,n} \kappa(\cdot, \boldsymbol{u}_j), \tag{1}$$

where $h_{j,n} \in \mathbb{R}$ is a weight of the kernel. From this, it is seen that estimating Ω_n is essentially equivalent to estimating $h_{j,n} \in \mathbb{R}$. Here, $\{\boldsymbol{u}_j\}_{j \in \mathcal{J}_n}$ is called a dictionary. Define the index set of dictionary elements and the dictionary size as $\mathcal{J}_n := \{j_1^{(n)}, j_2^{(n)}, \ldots, j_{r_n}^{(n)}\} \subset \{0, 1, \ldots, n-1\}$ and $r_n = |\mathcal{J}_n|$ respectively. The filter output is represented as

$$y_n = \langle \phi(\boldsymbol{u}_n), \Omega_n \rangle = \sum_{j \in \mathcal{J}_n} h_{j,n} \kappa(\boldsymbol{u}_n, \boldsymbol{u}_j) = \boldsymbol{h}_n^\top \boldsymbol{\kappa}_n, \tag{2}$$

where

$$\boldsymbol{h}_n := [h_{j_1^{(n)},n}, h_{j_2^{(n)},n}, \ldots, h_{j_{r_n}^{(n)},n}]^\top \in \mathbb{R}^{r_n}, \tag{3}$$

$$\boldsymbol{\kappa}_n := [\kappa(\boldsymbol{u}_{j_1^{(n)}}, \boldsymbol{u}_n), \kappa(\boldsymbol{u}_{j_2^{(n)}}, \boldsymbol{u}_n), \ldots, \kappa(\boldsymbol{u}_{j_{r_n}^{(n)}}, \boldsymbol{u}_n)]^\top \in \mathbb{R}^{r_n}. \tag{4}$$

2.2. Kernel NLMS With Coherence Criteria

It is crucial to reduce the number of samples for representing Ω_n in (1) without deteriorating the performance in adaptation. In the following, we review a well-known dictionary update method based on the coherence criteria for kernel normalized least square (KNLMS) algorithm [11].

If the initial value of weight vector is $\boldsymbol{h}_0 := 0$ and the coherence threshold [11] is $\delta > 0$, the update rule of KNLMS is then given as follows:

1. If $\max_{j \in \mathcal{J}_n} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_j)| > \delta$, then $\mathcal{J}_{n+1} = \mathcal{J}_n$ and

$$\boldsymbol{h}_{n+1} = \boldsymbol{h}_n + \frac{\mu}{\rho + \|\boldsymbol{\kappa}_n\|^2}(d_n - \boldsymbol{h}_n^\top \boldsymbol{\kappa}_n)\boldsymbol{\kappa}_n, \tag{5}$$

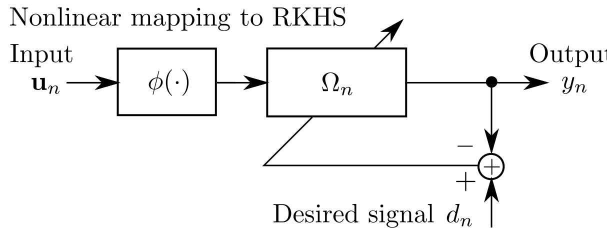

Fig. 1. Conceptual diagram of kernel adaptive filters

2. If $\max_{j \in \mathcal{J}_n} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_j)| \le \delta$, then $\mathcal{J}_{n+1} = \mathcal{J}_n \cup \{n\}$ and

$$\boldsymbol{h}_{n+1} = \bar{\boldsymbol{h}}_n + \frac{\mu}{\rho + \|\bar{\boldsymbol{\kappa}}_n\|^2}(d_n - \bar{\boldsymbol{h}}_n^\top \bar{\boldsymbol{\kappa}}_n)\bar{\boldsymbol{\kappa}}_n, \tag{6}$$

where μ and ρ are a step size parameter and a stabilization parameter respectively. Besides, $\bar{\boldsymbol{\kappa}}_n := [\boldsymbol{\kappa}_n^\top, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_n)]^\top$ and $\bar{\boldsymbol{h}}_n := [\boldsymbol{h}_n^\top, 0]^\top$. Namely, this rule with the coherence threshold judges whether to add the observed input signal to the dictionary at each time instant. The observed input signal is added to the dictionary if necessary, or discarded if not needed. Consequently, the filter is an element in the subspace that is spanned by the dictionary.

2.3. Dictionary Adaptation

The dictionary constructed in the KNLMS algorithm is a set of observed input signals selected by a coherence threshold. Hence, adaptive performance deteriorates if suitable signals to represent filters cannot be observed. In [23,24], dictionary adaptation is proposed. This is summarized as follows:

The underlying idea is to update elements of the dictionary to minimize MSE at time instance n:

$$e_n^2 = [d_n - y_n]^2 = \left[d_n - \sum_{j \in \mathcal{J}_n} h_{j,n} \kappa(\boldsymbol{u}_n, \boldsymbol{u}_j)\right]^2. \tag{7}$$

For the Gaussian kernel given as

$$\kappa(\boldsymbol{u}_n, \boldsymbol{u}_j) = \exp(-\zeta \|\boldsymbol{u}_n - \boldsymbol{u}_j\|^2), \tag{8}$$

(7) can be rewritten as

$$e_n^2 = \left[d_n - \sum_{j \in \mathcal{J}_n} h_{j,n} \exp(-\zeta \|\boldsymbol{u}_n - \boldsymbol{u}_j\|^2)\right]^2. \tag{9}$$

By using LMS algorithm [1], the dictionary at n $\{\boldsymbol{u}_j\}_{j \in \mathcal{J}_n}$ is updated as follows:

$$\boldsymbol{u}_{j,n+1} = \boldsymbol{u}_{j,n} - \eta_n \boldsymbol{g}_{j,n}, \tag{10}$$

$$\boldsymbol{g}_{j,n} = -2 e_n h_{j,n} \nabla_{\boldsymbol{u}_j} \exp(-\zeta \|\boldsymbol{u}_n - \boldsymbol{u}_{j,n}\|^2), \tag{11}$$

where η_n is a step size, which should be chosen such that updated dictionary elements $\boldsymbol{u}_{j,n+1}$ satisfy the coherence condition, that is,

$$|\kappa(\boldsymbol{u}_{i,n+1}, \boldsymbol{u}_{j,n+1})| \leq \delta. \quad (\forall \boldsymbol{u}_{i,n+1}, \forall \boldsymbol{u}_{j,n+1}, i \neq j) \tag{12}$$

First, the condition (12) is rewritten as

$$|\kappa(\Delta \boldsymbol{u}, \eta_n \Delta \boldsymbol{g})| \leq \delta, \quad (\forall \boldsymbol{u}_{i,n+1}, \forall \boldsymbol{u}_{j,n+1}, i \neq j) \tag{13}$$

where $\Delta \boldsymbol{u} = \boldsymbol{u}_{i,n} - \boldsymbol{u}_{j,n}$ and $\Delta \boldsymbol{g} = \boldsymbol{g}_{i,n} - \boldsymbol{g}_{j,n}$. Since it is difficult to analytically find η_n that satisfies (13), in [23, 24], a Taylor series expansion around $\eta_n \sim 0$ is applied to the left hand side[1], so that η_n is adjusted so that η_n is included in intervals $(-\infty, \eta_{n-}] \cup [\eta_{n+}, \infty)$, where

$$\eta_{n\pm} = \frac{\zeta \Delta \boldsymbol{u}^\top \Delta \boldsymbol{g} e^{-\zeta \|\Delta \boldsymbol{u}\|^2} \pm \sqrt{D}}{\zeta \|\Delta \boldsymbol{g}\|^2 e^{-\zeta \|\Delta \boldsymbol{u}\|^2}}, \tag{14}$$

$$D = \left(\zeta \Delta \boldsymbol{u}^\top \Delta \boldsymbol{g} e^{-\zeta \|\Delta \boldsymbol{u}\|^2}\right)^2 - 2\zeta \|\Delta \boldsymbol{g}\|^2 e^{-\zeta \|\Delta \boldsymbol{u}\|^2} (\delta - e^{-\zeta \|\Delta \boldsymbol{u}\|^2}). \tag{15}$$

We name this method for KNLMS dictionary adaptation for KNLMS (AKNLMS).

3. DICTIONARY ADAPTATION FOR KNLMS WITH l_1 REGULARIZATION

Although the dictionary adaptation is an efficient strategy for increasing performance of kernel adaptive filtering, the dictionary cannot still catch up with the change of unknown systems. Besides, there is a doubt on obtaining the step size in (10) with a Taylor expansion. In the following, we construct two types of KNLMS algorithm.

3.1. Modified Dictionary Adaptation

Although a Taylor series expansion around $\eta_n \sim 0$ is applied to (13), this is indeed not the center of the kernel. We suggest that the equation (13) should be approximated with a Taylor series around $\eta_n \sim \frac{\Delta \boldsymbol{u}^\top \Delta \boldsymbol{g}}{\Delta \boldsymbol{g}^\top \Delta \boldsymbol{g}}$ (around the center of Gaussian kernels). When the dimension of input signal is $L = 1$, the approximated condition is written as

$$-1 + \zeta(\Delta u - \Delta g \eta_n)^2 + \delta \geq 0. \quad (\forall \boldsymbol{u}_{i,n+1}, \forall \boldsymbol{u}_{j,n+1}, i \neq j) \tag{16}$$

Since this is rewritten as

$$(\zeta \Delta g^2)\eta_n^2 - (2\zeta \Delta u \Delta g)\eta_n + (\delta - 1 + \zeta \Delta u^2) \geq 0, \tag{17}$$

η_n is adjusted in intervals $(-\infty, \eta_{n-}] \cup [\eta_{n+}, \infty)$,

$$\eta_{n\pm} = \beta \frac{\zeta \Delta u \Delta g \pm \sqrt{(\zeta \Delta u \Delta g)^2 - \zeta \Delta g^2 (\delta - 1 + \zeta \Delta u^2)}}{\zeta \Delta g^2}, \tag{18}$$

where β is a correction parameter of the approximation. β should be empirically determined, but it needs to be small enough ($\beta \ll 1$). We name this algorithm for KNLMS modified dictionary adaption for KNLMS (MAKNLMS).

[1] This approximation could be incorrectly calculated. We discuss this issue in Section 4.

3.2. l_1-regularized KNLMS With Dictionary Adaptation

Another approach is to jointly update and eliminate elements in the dictionary. In the proposed method, we apply dictionary adaptation to KNLMS with l_1 regularization [22]. In the dictionary adaption, we exploit a novel method to satisfy coherence condition for nonstationary systems.

The cost function of KNLMS is added an l_1 regularization term in order to effectively adapt nonstationary systems. The cost function is written as follows:

$$\Theta_n := |d_n - \boldsymbol{h}_n^\top \boldsymbol{\kappa}_n|^2 + \lambda \|\boldsymbol{h}_n\|_1, \tag{19}$$

where λ is a regularization parameter. It is not possible to apply the stochastic gradient approach to the cost function since the l_1 norm is nonsmooth. However, since Θ_n is a convex function, we can apply the forward-backward splitting [25]. The update rule is then given as follows:

1. If $\max_{j \in \mathcal{J}_n} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_j)| > \delta$, then $\mathcal{J}_{n+1} = \mathcal{J}_n$ and

$$\boldsymbol{h}_{n+1} = \mathrm{prox}_{\mu\lambda\|\boldsymbol{h}_n\|_1} \left[\boldsymbol{h}_n + \frac{\mu(d_n - \boldsymbol{h}_n^\top \boldsymbol{\kappa}_n)\boldsymbol{\kappa}_n}{\rho + \|\boldsymbol{\kappa}_n\|^2} \right], \tag{20}$$

2. If $\max_{j \in \mathcal{J}_n} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_j)| \leq \delta$, then $\mathcal{J}_{n+1} = \mathcal{J}_n \cup \{n\}$ and

$$\boldsymbol{h}_{n+1} = \mathrm{prox}_{\mu\lambda\|\boldsymbol{h}_n\|_1} \left[\bar{\boldsymbol{h}}_n + \frac{\mu(d_n - \bar{\boldsymbol{h}}_n^\top \bar{\boldsymbol{\kappa}}_n)\bar{\boldsymbol{\kappa}}_n}{\rho + \|\bar{\boldsymbol{\kappa}}_n\|^2} \right], \tag{21}$$

where $\mathrm{prox}_{\mu\lambda\|\boldsymbol{h}_n\|_1}(\cdot)$ denotes the proximal operator [25] of $\lambda\|\boldsymbol{h}_n\|_1$. This rule promotes the sparsity of $h_{j,n}$, and then some of coefficients becomes $h_{j,n} \approx 0$. This yields the following update rule of the dictionary:

Elimination of elements (1)

For all $j \in \mathcal{J}_{n+1}$, if $h_{j,n} \approx 0$, remove $\boldsymbol{u}_j$, that is,

$$\mathcal{J}_{n+1} \leftarrow \mathcal{J}_{n+1} - \{j\}. \tag{22}$$

Although the dictionary adaptation proposed in [23, 24] and the previous section, elements in the dictionary are prevented from getting close to each other. This yields the increase of the dictionary size especially when the system dynamically changes. Thus, in this section we propose another method that uses a fixed step size instead of the adaptive step size in (10) and that removes an element close to another one. Dictionary elements obtained from the above rule are updated to minimize MSE with an update similar to (10) as

$$\boldsymbol{u}_{j,n+1} = \boldsymbol{u}_{j,n} - \eta \boldsymbol{g}_{j,n}, \tag{23}$$

where, η is a fixed step size. After the update of elements, $\boldsymbol{u}_{j,n+1}$, they are judged whether it should be removed from the dictionary as follows:

Elimination of elements (2)

If $|\kappa(\boldsymbol{u}_{i,n+1}, \boldsymbol{u}_{j,n+1})| > \delta$ and $|h_{i,n}| > |h_{j,n}|$, then remove $\boldsymbol{u}_{j,n+1}$, that is,

$$\mathcal{J}_{n+1} \leftarrow \mathcal{J}_{n+1} - \{j\}. \tag{24}$$

Table 1. Parameters

KNLMS	$\mu = 0.09, \rho = 0.03, \zeta = 1, \delta = 0.5$
AKNLMS	$\mu = 0.09, \rho = 0.03, \zeta = 1$ $\delta = 0.5, \eta_0 = 1.0 \times 10^{-3}$
MAKNLMS	$\mu = 0.09, \rho = 0.03, \zeta = 1, \delta = 0.5$ $\eta_0 = 1.0 \times 10^{-3}, \beta = 1.2 \times 10^{-4}$
AKNLMS-l_1	$\mu = 0.09, \rho = 0.03, \zeta = 1, \delta = 0.5$ $\eta = 1.0 \times 10^{-3}, \lambda = 5.0 \times 10^{-3}$

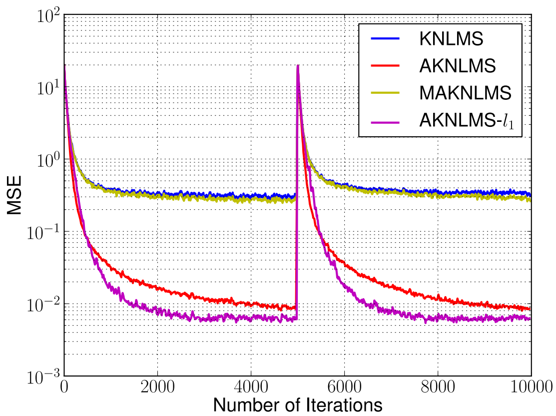

Fig. 2. Learning curves of KNLMS, AKNLMS, MAKNLMS and AKNLMS-l_1. AKNLMS-l_1 shows a rapid adaptation and the smallest MSE. The results are obtained by the average over 100 independent runs.

This update rule removes elements that do not satisfy (12). We call the proposed algorithm developed in this section the dictionary adaptation for KNLMS with l_1 regularization (AKNLMS-l_1).

4. NUMERICAL EXAMPLES

We consider the nonstationary nonlinear system as follows:

- $d_n := 10\{\exp(-(u_n - 3)^2) + \exp(-(u_n - 7)^2)\}$ for $0 \leq n \leq 5000$;
- $d_n := 10\{\exp(-(u_n - 13)^2) + \exp(-(u_n - 17)^2)\}$ for $5000 < n \leq 10000$,

where d_n is corrupted by noise sampled from a zero-mean Gaussian distribution with standard deviation equal to 0.3. Input signals u_n are sampled from uniform distribution on the interval $[0, 10]$ when $0 \leq n \leq 5000$ and the interval $[10, 20]$ when $5000 < n \leq 10000$. In the online prediction of the system, we compare the AKNLMS [23], the proposed

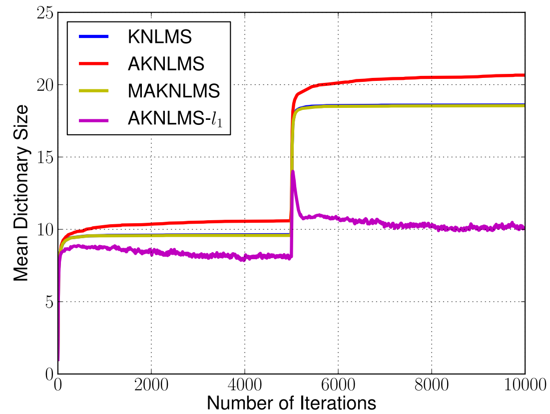

Fig. 3. Mean dictionary sizes of KNLMS, AKNLMS, MAKNLMS and AKNLMS-l_1. The change of the system causes the increased size of the dictionary. However, AKNLMS-l_1 can suppress the increase.

MAKNLMS and AKNLMS-l_1. We adopted MSE as the evaluation criteria. The MSE is calculated by taking an arithmetic average over 100 independent realizations. Parameters of each filter in this experiment are given in Table 1.

Figures 2 and 3 show the MSE and the mean dictionary size of filters at each iteration, respectively. In Fig. 2, AKNLMS shows lower MSE than KNLMS. However, Fig. 3 illustrates that AKNLMS has the large dictionary size compered to the others due to the increase of similar dictionary elements. This may be caused by inferior approximation and the incorrect approximation in [23, 24]. It is moreover observed that MAKNLMS shows lower MSE than KNLMS. However, the dictionary size monotonically increases due to the system change. Finally, it is seen that, thanks to the l_1 regularization and the element removal to satisfy coherence condition, AKNLMS-l_1 shows lower MSE and smaller the dictionary size than the others.

5. CONCLUSION

This paper proposed an efficient update algorithm for dictionary adaptation incorporated with the l_1 regularization promoting sparsity in kernel adaptive filtering. Numerical examples showed that the proposed method has advantages over the previous methods in the decreased number of elements in the dictionary and the high approximation performance in the presence of system changes. Even though the numerical example shown in this paper is a limited case of $L = 1$, we can confirm the efficiency of the proposed methods. Higher dimensional examples and practical applications in a real environment will be reported in near future.

6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, 2002.

[2] B. Widrow *et al.*, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.

[3] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Signal Process.*, vol. 40, no. 8, pp. 1862–1875, 1992.

[4] J. Lee and J. V. Mathews, "A fast recursive least squares adaptive second order Volterra filter and its performance analysis," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1087–1102, 1993.

[5] R. D. Nowak and B. D. Van Veen, "Random and pseudorandom inputs for Volterra filter identification," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 2124–2135, 1994.

[6] ——, "Volterra filter equalization: A fixed point approach," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 377–388, 1997.

[7] W. Liu, J. Principe, and S. Haykin, *Kernel Adaptive Filtering*. Hoboken, NJ: Wiley, 2010.

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

[9] T. Ishida and T. Tanaka, "Multikernel adaptive filters with multiple dictionaries and regularization," in *Proc. of 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA 2013)*, 2013, pp. 1–6.

[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, 2009.

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

[12] 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.

[13] 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.

[14] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 964–978, 2011.

[15] J. Gil-Cacho, T. van Waterschoot, M. Moonen, and S. Jensen, "Nonlinear acoustic echo cancellation based on a parallel-cascade kernel affine projection algorithm," in *Proc. of 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, 2012, pp. 33–36.

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

[17] F. Tobar, S.-Y. Kung, and D. Mandic, "Multikernel least mean square algorithm," *IEEE Trans. Neural Netw.*, vol. 25, no. 2, pp. 265–277, 2014.

[18] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 9, pp. 337–404, 1950.

[19] W. Liu, I. Park, and J. Principe, "An information theoretic approach of designing sparse kernel adaptive filters," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1950–1961, 2009.

[20] Y. Nakajima and M. Yukawa, "Nonlinear channel equalization by multi-kernel adaptive filter," in *Proc. of 2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2012)*, 2012, pp. 384–388.

[21] M. Yukawa and R. Ishii, "Online model selection and learning by multikernel adaptive filtering," in *Proc. of the 21st European Signal Processing Conference (EUSIPCO)*, 2013, pp. 1–5.

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

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

[24] C. Saide, R. Lengelle, P. Honeine, and R. Achkar, "Online kernel adaptive algorithms with dictionary adaptation for MIMO models," *IEEE Signal Process. Lett.*, vol. 20, no. 5, pp. 535–538, 2013.

[25] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. of 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, 2010, pp. 3734–3737.