IMPLEMENTATION METHOD OF KERNEL ADAPTIVE FILTER AS AN ADD-ON FOR A LINEAR ADAPTIVE FILTER

Kiyoshi Nishikawa

Department of System Design
Tokyo Metropolitan University, Japan
knishikawa@m.ieice.org

Felix Albu

Valahia University of Targoviste
Targoviste, Romania
felix.albu@gmail.com

Index Terms— Kernel adaptive filter, non-linear system identification, normalized LMS algorithm

ABSTRACT

In this paper, we propose a novel structure for implementing a kernel adaptive filter as an add-on component for a linear adaptive filter. The kernel adaptive filter has been proposed as a solution to non-linear adaptive problems and their effectiveness has been demonstrated. However, it is not intended for replacing the linear adaptive filters at all, rather, we expect it to complement the performance of linear ones in non-linear environments. We, therefore, consider a novel structure which enables us to implement a kernel adaptive filter as an add-on for a linear adaptive filter. The proposed structure performs as a linear adaptive filter in the linear-dominant environments, however, in non-linear environments, we can add a kernel adaptive filter without any modification on the operation of the linear one. The effectiveness of the proposed method is confirmed through the computer simulations.

1. INTRODUCTION

The kernel adaptive filter is one of the emerging techniques for non-linear signal processing [1]. Several algorithms had been proposed so far [1–4]. The kernel adaptive filters are shown to be effective for non-linear problems so that they are expected to complement the linear adaptive filters.

Although the kernel filters are effective for non-linear problems, we believe that the importance of the linear adaptive filters will not affected because of their well-analyzed convergence behavior and acceptable computational complexity. Besides, there still exist implementation problems of the kernel adaptive filters in actual applications.

Of those problems, in this paper, we deal with the two problems, namely, (i) the selection of the value of the kernel parameter; and (ii) the performance degradation in linear-dominant environments. These problems are described as below. The Gaussian kernel is widely used in the kernel adaptive algorithms and which has a parameter, called the kernel parameter, or the bandwidth parameter. It is known that the selection of the parameter affects the convergence characteristics of the filter, however, there are no standard method for selecting the optimum value (problem (i)). Also, when the target environment is near-linear, the performance of the kernel filter is degraded compared with that of the linear filters (problem (ii)).

For the problem (i), the multi-kernel adaptive filter [5, 6] is one of the possible solutions. It uses multiple kernels with different values of kernel parameters simultaneously to relax the selection problem. However, the selection of an optimum value with an acceptable computational complexity remain unsolved [6]. For the problem (ii), [7] proposes to use a linear kernel as one of the multi-kernel structure, and it is shown to be effective to acoustic echo canceling application. However, in this structure, the linear kernel has some dependency on the other non-linear kernels, e.g., they share the same dictionary, effect of sparcification of the input signal, etc.

In this paper, we propose a novel structure of the kernel adaptive filter in order to solve the above mentioned two problems. In the proposed structure, the kernel adaptive filter acts like an add-on to the linear adaptive filter. We can attach or detach the kernel filter according to the environments where the adaptive filter is applied without affecting the operation of the linear filter. The structure is based on the mixture configuration of a kernel and a linear adaptive filters which we proposed in [8]. We confirmed the effectiveness of the proposed structure through computer simulations.

We note that there exist several structures which use linear and non-linear adaptive filters simultaneously, e.g., [9–11]. In those structures, each filter is updated in parallel using its local error signal and then the outputs are adaptively mixed to generate the global output. On the other hand, in the proposed structure, the kernel filters are updated after the update of liner filter. In that process, the a posteriori error of the linear filter is used as the error signal of the kernel filter. This formulation enables the kernel filter to be used as an attachment to the linear one without any modification. Besides, it can reduce the effect of linear component in the error signal so that the structure can improve the convergence property in specific environments as shown in the simulation results.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

2. PREPARATION

Here, the kernel normalized least mean square (KNLMS) algorithm [2] and its multi-kernel implementation [5] are briefly reviewed. In the following, a matrix or a vector is indicated by bold letter, e.g., $\mathbf{R}$, or $\mathbf{r}$. The transpose of $\mathbf{R}$ is indicated as $\mathbf{R}^{\mathrm{T}}$, and variable at time n is expressed as $\mathbf{R}(n)$.

2.1. Kernel adaptive filters

The concept of a kernel adaptive filter is derived by applying the kernel method to the linear adaptive filter [1]. To apply the kernel method, the input signal $\{x(n) \mid n = 0, 1, 2, \ldots\}$ is mapped onto a higher order characteristics space. We express this mapping as $\phi(\mathbf{x}(n))$ where $\mathbf{x}(n)$ shows the input vector of the filter at time n, whose length is denoted by S.

Then, we expand and approximate the coefficient vector of the kernel filter $\mathbf{w}(n)$ using $\{\phi(\mathbf{x}(m)) \mid m = 0, \ldots, n-1\}$ as $\mathbf{w}'(n)$:

$$\mathbf{w}'(n) = h_0\phi(\mathbf{x}(0)) + \cdots + h_{n-1}\phi(\mathbf{x}(n-1)) \quad (1)$$

where $\{h_i \mid i = 0, \ldots, n-1\}$ are the weights to be determined. The length of $\mathbf{w}(n)$ and $\mathbf{w}'(n)$ are same as that of $\mathbf{x}(n)$, or S. Using the kernel trick [1], the output signal $y(n)$ is expressed as

$$\begin{aligned} y(n) = & \quad \phi(\mathbf{x}(n))^{\mathrm{T}}\mathbf{w}'(n) \\ = & \quad h_0\kappa(\mathbf{x}(n), \mathbf{x}(0)) + \cdots + h_n\kappa(\mathbf{x}(n), \mathbf{x}(n-1)) \end{aligned} \quad (2)$$

where $\kappa(\cdot, \cdot)$ shows the kernel function. By defining the vectors $\mathcal{X}(n)$ and $\boldsymbol{h}(n)$ as

$$\begin{aligned} \mathcal{X}(n) &= [\kappa(\mathbf{x}(n), \mathbf{x}(0)) \quad \cdots \quad \kappa(\mathbf{x}(n), \mathbf{x}(n-1))]^{\mathrm{T}} \\ \boldsymbol{h}(n) &= [h_0 \quad \ldots \quad h_{n-1}]^{\mathrm{T}} \end{aligned} \quad (3)$$

and substituting into (2), we have a simpler expression of $y(n)$

$$y(n) = \mathcal{X}^{\mathrm{T}}(n)\boldsymbol{h}(n). \quad (4)$$

This equation show the input-output relation of the kernel adaptive filter $\boldsymbol{h}(n)$. For updating $\boldsymbol{h}(n)$, we use the kernel adaptive algorithms [1].

As the kernel function, the Gaussian kernel of the following form is generally used in kernel adaptive filtering, i.e.,

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\zeta||\mathbf{x} - \mathbf{y}||^2\right) \quad (5)$$

where ζ is a parameter called the kernel parameter or the kernel bandwidth. The selection of the value of ζ affects the convergence characteristics (problem (i)).

2.2. KNLMS algorithm

The KNLMS algorithm [2] is the kernel version of the linear NLMS algorithm.The filter coefficients $\boldsymbol{h}(n)$ are updated by

$$\boldsymbol{h}(n+1) = \boldsymbol{h}(n) + \eta\frac{e(n)\mathcal{X}(n)}{\epsilon + \mathcal{X}^{\mathrm{T}}(n)\mathcal{X}(n)} \quad (6)$$

$$e(n) = d(n) - \mathcal{X}^{\mathrm{T}}(n)\boldsymbol{h}(n-1) \quad (7)$$

where η and ϵ are a step-size and a stabilization parameters respectively. The past input signal vectors $\{\mathbf{x}(j) \mid j = 0, 1, \cdots, n-1\}$ are stored as the dictionary for learning and its size increases as time advances. Hence, the computational complexity is time varying. To maintain the applicable amount of computation, several sparsification methods of the input signal were proposed [1, 2, 12]. In the following, we assume to use the method of [2] due to its simple structure. In that, the condition below is examined at each time

$$\max|\mathcal{X}(n)|_{j=1,\cdots,J} < \mathcal{T}_0 \quad (8)$$

where $\mathcal{T}_0$ is the predefined threshold value, and J shows the number of entries in the dictionary. Only when this condition holds, $\mathbf{x}(n)$ will be added as a new entry to the dictionary.

2.3. Multi-kernel adaptive filter [5]

When we use the Gaussian kernel, the convergence characteristics are affected by the parameter ζ in (5). One of the practical problems of the kernel adaptive filters is to select an optimum value for this parameter. In order to relax this problem by using multiple kernels simultaneously, the multi-kernel adaptive filter was proposed [5, 6].

In this configuration, the filter output is expressed as

$$y_{\mathrm{k}}(n) = \sum_{m=1}^{M}\sum_{j=1}^{J_m} h_j^{(m)}(n)\kappa_m(\mathbf{x}(n), \mathbf{x}(j)) \quad (9)$$

where M shows the number of kernels used, and J_m the number of entries in the dictionary of m-th kernel. $h_j^{(m)}(n)$ shows the j-th coefficients of the adaptive filter of the m-th kernel at time n.

It is shown [5] that the configuration could relax the problem of selecting the optimum values for kernel parameters. Besides, it is shown in [7] that using a linear kernel as one of the multi-kernels can improve the convergence characteristics of the adaptive filter for acoustic echo canceler in which elimination of the linear component is important. Therefore, the two problems mentioned in Introduction seem to be solvable by using the multi-kernel structure with a linear kernel.

However, we note that there are some implementation problems in the structure of [7]. For example, the coefficient vector $\boldsymbol{w}_{\ell k}(n)$ of the adaptive filter corresponds to the linear kernel is equivalently updated by [8]

$$\boldsymbol{w}_{\ell k}(n+1) = \boldsymbol{w}_{\ell k}(n) + \frac{\eta e(n)\sum_{j=0}^{n-1}\mathbf{x}(j)\mathbf{x}^{\mathrm{T}}(j)\mathbf{x}(n)}{\sum_{m=1}^{M}\mathcal{X}_m^{\mathrm{T}}(n)\mathcal{X}_m(n)}. \quad (10)$$

Known from this equation, $\boldsymbol{w}_{\ell k}(n)$ depends on the selection of the other kernels including the kernel parameters because the denominator contains $\mathcal{X}_m(n)$ (see Eq. (3)).

The other point which may cause difficulty is that, even if the applied environment is linear dominant, the kernel adaptive filter still need to be updated, and this requires additional

computational complexity. In the following, we propose a structure that enables us to avoid these possible problems.

3. PROPOSED METHOD

Here, we describe the proposed structure which is based on the method in [8]. We show that, in the proposed structure, we can implement the kernel adaptive filter as an attachment to the linear adaptive filter.

3.1. Linear and multi-kernel filter configuration [8]

In [8], we considered a mixture configuration of a linear and a multi-kernel adaptive filters and derived an algorithm which update each filter independently in the structure. Although the configuration is similar to the one proposed in [7], they differ in the following point. Our method uses a linear adaptive filter instead of a linear kernel, and, we showed that the linear and the kernel filters can be updated independently using the algorithm summarized below.

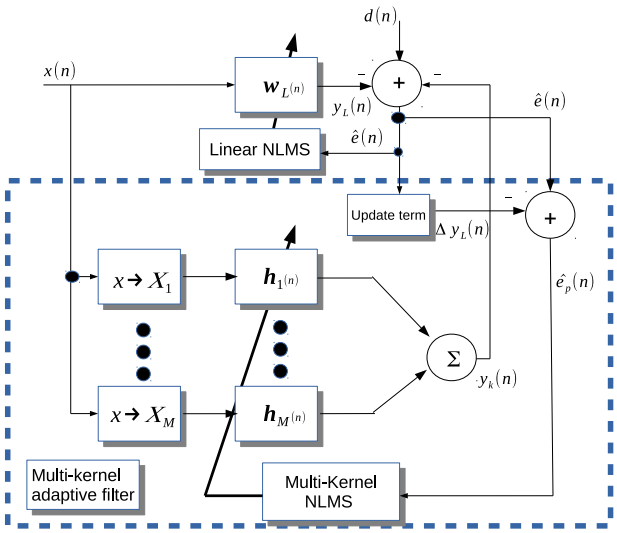

Fig. 1. Configuration of the method of [8]. The region in the dashed line is the multi-kernel adaptive filter and 'Update term' shows the calculation of the equation (14)

In Fig. 1, we show the configuration of [8]. As shown in the figure, we employ a linear adaptive filter simultaneously with the multi-kernel adaptive filter. We denote the filter coefficients of the linear filter as $\boldsymbol{w}_L(n)$ and is expressed as

$$\boldsymbol{w}_L(n) = [w_0(n) \;\; w_1(n) \;\; \dots \;\; w_{S-1}(n)]^{\mathrm{T}} \qquad (11)$$

where S shows the number of coefficients. Its output $y_{\mathrm{L}}(n)$ is given as $y_{\mathrm{L}}(n) = \mathbf{x}^{\mathrm{T}}(n)\boldsymbol{w}_L(n)$. Also, a multi-kernel adaptive filter is used whose kernels are assumed to be Gaussian with different values of the kernel parameter as in [5].

After the calculation of the error signal $\hat{e}(n)$ by $\hat{e}(n) = d(n) - y_{\mathrm{L}}(n) - y_{\mathrm{k}}(n)$, the linear and kernel adaptive filters are updated.

We showed that the linear and the kernel filters can be updated independently by the following procedure. At first, we update only $\boldsymbol{w}_L(n)$ using the linear NLMS algorithm as

$$\boldsymbol{w}_L(n+1) = \boldsymbol{w}_L(n) + \alpha \frac{\hat{e}(n)}{\epsilon + \mathbf{x}^T(n)\mathbf{x}(n)}\mathbf{x}(n) \qquad (12)$$

where α and ϵ are the step-size and the stabilizing parameters respectively. By comparing Eq (12) with (10), we can see that, in the proposed method, $\boldsymbol{w}_L(n)$ can be updated independently of the kernel filter because (12) does not contain $\mathcal{X}_m(n)$. Then, using the updated $\boldsymbol{w}_L(n+1)$, we calculate the a posteriori error $\hat{e}_{\mathrm{p}}(n)$ of the linear filter as

$$\begin{aligned}\hat{e}_{\mathrm{p}}(n) &= d(n) - \mathbf{x}^{\mathrm{T}}(n)\boldsymbol{w}_L(n+1) - y_{\mathrm{k}}(n) \\ &= d(n) - (y_{\mathrm{L}}(n) + \Delta y_{\mathrm{L}}(n)) - y_{\mathrm{k}}(n) \\ &= \hat{e}(n) - \Delta y_{\mathrm{L}}(n) \end{aligned} \qquad (13)$$

where $\Delta y_{\mathrm{L}}(n)$ is defined as [8]

$$\Delta y_{\mathrm{L}}(n) = \alpha \frac{\hat{e}(n)}{\epsilon + \mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)}\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n). \qquad (14)$$

Then, the kernel filter is updated using $\hat{e}_{\mathrm{p}}(n)$ as

$$\boldsymbol{h}_m(n+1) = \boldsymbol{h}_m(n) + \eta \frac{\hat{e}_{\mathrm{p}}(n)\mathcal{X}_m(n)}{\sum_{k=1}^{M}\mathcal{X}_k^T(n)\mathcal{X}_k(n)} \qquad (15)$$

where $m = 1, 2, \dots, M$. Here, we can see that the kernel filter can be updated independently of the linear filter. Besides, the term $\Delta y_{\mathrm{L}}(n)$ in (14) can be calculated before updating $\boldsymbol{w}_L(n)$, and hence, we could update all the filters at once.

3.2. Proposed structure

Using the method just described, we propose a novel structure of the kernel filter. In the proposed structure, a kernel filter is an add-on component to a linear filter, and it can be attached or detached according to the characteristics of the target environment. This feature can be achieved by the independence of the linear and the kernel filters in the method. Besides, by using the multi-kernel adaptive filter, the problem of the selection of the kernel parameter could be relaxed.

In Fig.2, we show a conceptual structure of the proposed method. As shown in this figure, the kernel adaptive filter is an add-on component to the linear filter that is detachable when it is not required. The advantage of the proposed method is that the linear filter can be used without any modification while the kernel filter is attached or detached.

3.3. Algorithm

When the kernel filter is detached, only the linear filter will be updated

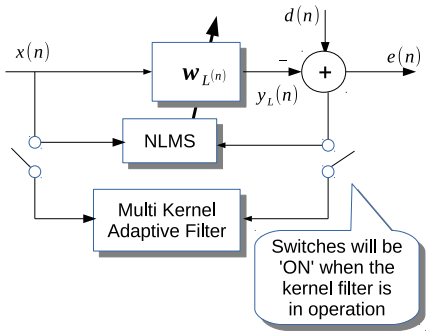

Fig. 2. Idea of the proposed structure. The kernel adaptive filter can be added to the linear filter according to characteristics of the applied environment. The modification of the linear adaptive filter is not required with or without the kernel filter.

1. The error signal $\hat{e}(n) = d(n) - y_{\mathrm{L}}(n)$
2. The linear adaptive filter $\boldsymbol{w}_L(n)$ is updated using Eq. (12)

where we assumed the NLMS algorithm is used.

On the other hand, when the kernel filter is attached, the filters will be updated according to the following steps:

1. The error signal $\hat{e}(n) = d(n) - y_{\mathrm{L}}(n) - y_{\mathrm{k}}(n)$
2. The linear adaptive filter $\boldsymbol{w}_L(n)$ is updated using Eq. (12)
3. From equations (13) and (14), the a posteriori error $\hat{e}_{\mathrm{p}}(n)$ is calculated by

$$\hat{e}_{\mathrm{p}}(n) = \left(1 - \alpha \frac{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)}{(\epsilon + \mathbf{x}^{T}(n)\mathbf{x}(n))}\right)\hat{e}(n)$$

4. The kernel adaptive filter $\boldsymbol{h}(n)$ is updated using Eq. (15)

Note that we do not need add any modification to the update equations (12) or (15) regardless of the status of the kernel filter. Moreover, we can update the linear and kernel filter in parallel because $\hat{e}_{\mathrm{p}}(n)$ can be calculated without updating $\boldsymbol{w}_L(n)$ as mentioned in 3.1.

4. SIMULATION RESULTS

Here, we show the results of simulations to confirm the effectiveness of the proposed method. In the simulations, we compared three algorithms, namely, (i) the linear NLMS, (ii) the multi-kernel NLMS, and (iii) the multi-kernel NLMS with a linear kernel and (iv) the proposed. For (ii), (iii), and (iv), we used two Gaussian kernels with different kernel bandwidth which were selected to obtain near-optimal convergence. The step-size parameters of the linear and the kernel NLMS algorithms were $\alpha = 0.2$ and $\eta = 0.1$ respectively. The Gaussian noise was added to the desired signal whose mean was $\mu = 0$ and the variance was fixed as $\sigma_n^2 = 0.001$. The results shown in this section are ensemble averages of 300 independent simulations. For the proposed method, the kernel filter was detached at $n = 400$ and attached again at $n = 600$ to confirm the effect of the proposed structure.

4.1. Linear-dominant model

First, the filters are applied to an artificial mixture model of linear and non-linear systems whose input-output relation is given as

$$\begin{aligned} d(n) =& a_1 u(n) + a_2 u(n-1) + a_3(0.8 - 0.5\exp(-u^2(n)))u(n) \\ & - a_4(0.3 + 0.9\exp(-u^2(n))u(n-1) \\ & - 0.1 a_5 \sin(u(n)\pi) \end{aligned} \tag{16}$$

where the coefficients $a_1, \ldots, a_5$ were set as

$$[a_1 \;\; a_2 \;\; a_3 \;\; a_4 \;\; a_5] = \begin{cases} [0.5 \;\; 0.5 \;\; 0.2 \;\; 0.2 \;\; 0.2] & (n \le 700) \\ [0.3 \;\; 0.0 \;\; 0.0 \;\; 0.5 \;\; 0.5] & (n > 700) \end{cases} \tag{17}$$

We set $\mathcal{T}_0$ as 0.9, and ζ_m as -0.1 and -1.8 for the multi-kernel filters.

The results are shown in Fig. 3. From the figure, we can confirm the effectiveness of the linear filter used in the multi-kernel configuration. Besides, the MSE property of the proposed method is almost identical to that of the linear NLMS. Hence, for this configuration, we may detach the kernel filter for reducing the required computational complexity.

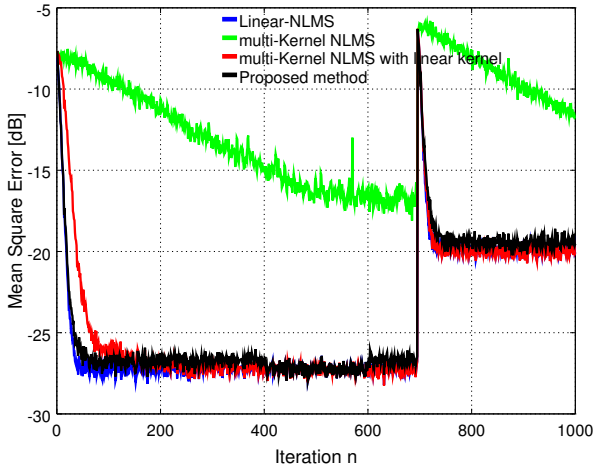

Fig. 3. Comparison of the algorithms in terms of MSE for the system defined by the equation (16).

4.2. Non-linear system model [13]

Next, we simulated the model used in [13], in which the signal $d(n)$ was generated by the equation below:

$$\phi(y(n)) = \begin{cases} \dfrac{y(n)}{3[0.1 + 0.9y^2(n)]^{1/2}} & \text{for } y(n) \le 0 \\ \dfrac{-y^2(n)[1 - \exp(0.7y(n))]}{3} & \text{for } y(n) > 0 \end{cases}$$

$$d(n) = \phi(y(n)) + \xi(n) \tag{18}$$

where $\xi(n)$ is the additive Gaussian noise, and $y(n)$ is given as

$$y(n) = \mathbf{a}^{\mathrm{T}}\mathbf{u}(n) - 0.2y(n-1) + 0.35y(n-2) \qquad (19)$$

where $\mathbf{a} = [1 \;\; 0.5]^{\mathrm{T}}$ and $\mathbf{u}(n)$ is $\mathbf{u}(n) = [u_1(n) \;\; u_2(n)]^{\mathrm{T}}$. The results are shown in Fig. 4. From the figure, for this model, we can see that the proposed method provides better convergence characteristics than other algorithms, especially the multi-kernel with a linear kernel configuration. The results demonstrate that the usage of the a posteriori error in the proposed method can eliminate the effect of the linear components in this model.

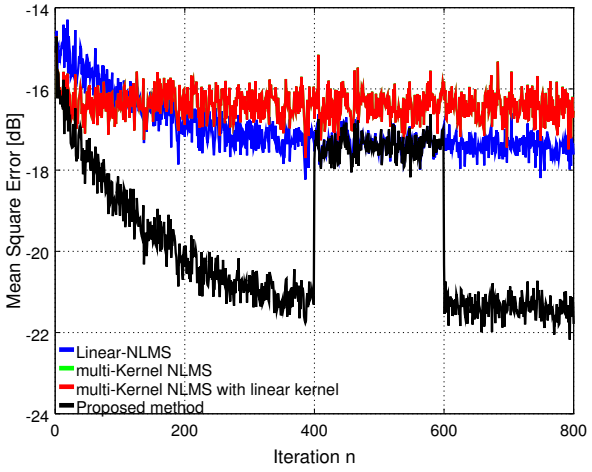

Fig. 4. Comparison of the algorithms in terms of MSE for the system defined by the equation (18).

5. CONCLUSIONS

In this paper, we proposed a novel structure of the kernel adaptive filter which can be attached to a linear adaptive filter as an add-on component. The proposed structure is based on the algorithm of [8] in which a linear and a kernel adaptive filters are independently updated. Featuring this characteristics, in the proposed structure, the kernel adaptive filter can be attached or detached according to the target environment without any modification to the linear filter. Through computer simulations, we confirmed the effectiveness of the proposed structure. As a future work, we will consider the switching algorithm for automatically attach or detach the kernel filter according to the environments.

REFERENCES

[1] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*, Wiley, 2010.

[2] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online Prediction of Time Series Data With Kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[3] F. Albu and K. Nishikawa, "The Kernel Proportionate NLMS Algorithm," in *Proc. EUSIPCO 2013*, Marrakech, Morocco, Sept. 2013.

[4] K. Nishikawa, Y. Ogawa, and F. Albu, "Fixed Order Implementation of Kernel RLS-DCD Adaptive Filters," in *Proc. APSIPA ASC 2013*, Oct. 2013.

[5] M. Yukawa, "Multikernel Adaptive Filtering," *Signal Processing, IEEE Transactions on*, vol. 60, no. 9, pp. 4672–4682, Sept 2012.

[6] O. Toda and M. Yukawa., "On Kernel Design for Online Model Selection by Gaussian Multikernel Adaptive Filtering," in *Proc. APSIPA ASC2014*, Dec. 2014.

[7] J. Gil-Cacho, M. Signoretto, T. van Waterschoot, M. Moonen, and S. Jensen, "Nonlinear Acoustic Echo Cancellation Based on a Sliding-Window Leaky Kernel Affine Projection Algorithm," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 9, pp. 1867–1878, Sept 2013.

[8] K. Nishikawa and F. Albu, "Consideration on the Performance of Kernel Adaptive Filters for the Mixture of Linear and Non-Linear Environments," in *Proc. APSIPA ASC 2014*, Dec. 2014.

[9] G. Sicuranza and A. Carini, "Nonlinear system identification by means of mixtures of linear-in-the-parameters nonlinear filters," in *Proceedings on ISPA2013*, Sept 2013, pp. 337–342.

[10] D. Comminiello, M. Scarpiniti, L. Azpicueta-Ruiz, J. Arenas-Garcia, and A. Uncini, "Functional link adaptive filters for nonlinear acoustic echo cancellation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 7, pp. 1502–1512, July 2013.

[11] D. Comminiello, M. Scarpiniti, L. Azpicueta-Ruiz, J. Arenas-Garcia, and A. Uncini, "Nonlinear acoustic echo cancellation based on sparse functional link representations," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 7, pp. 1172–1183, July 2014.

[12] W. Liu, I. Park, and J. C. Principe, "An information theoretic approach of designing sparse kernel adaptive filters.," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1950–61, Dec. 2009.

[13] W. Gao, J. Chen, C. Richard, and J. Huang, "Online Dictionary Learning for Kernel LMS," *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2765–2777, June 2014.