THE KERNEL PROPORTIONATE NLMS ALGORITHM

Felix Albu[1], Kiyoshi Nishikawa[2]

[1]Valahia University of Targoviste,
18-24 Bd. Unirii, Targoviste, Romania.
Email: felix.albu@valahia.ro
[2]Tokyo Metropolitan University,
Hino 191-0065, Tokyo, Japan.
Email: knishikawa@m.ieice.org

ABSTRACT

In this paper, the kernel proportionate normalized least mean square algorithm (KPNLMS) is proposed. The proportionate factors are used in order to increase the convergence speed and the tracking abilities of the kernel normalized least mean square (KNLMS) adaptive algorithm. We confirm the effectiveness of the proposed algorithm for nonlinear system identification and forward prediction using computer simulations.

Index Terms— Kernel normalized least mean square algorithm, proportionate-type algorithms, nonlinear system identification, forward prediction.

1. INTRODUCTION

Linear adaptive filters have been used in a variety of applications, e.g., echo or noise cancelation, equalization in wireless communication channels etc. In system identification applications, the main goal is to identify an unknown system using an adaptive filter [1]. For example, in case of network and acoustic echo cancellation scenarios, the echo path is sparse in nature, i.e., a small percentage of the impulse response components have a significant magnitude while the rest are zero or small. The sparseness character of the echo paths inspired the idea to "proportionate" the algorithm behavior, i.e., to update each coefficient of the filter independently of the others, by adjusting the adaptation step-size in proportion to the magnitude of the estimated filter coefficient [2]. The proportionate normalized least-mean-square (PNLMS) algorithm [3] was one of the first proportionate-type algorithms. The improved PNLMS algorithm proposed in [4] has superior convergence properties, especially for sparse echo paths. Other proportionate-type algorithms can be found in [5].

Recently, as an extension of the linear counterparts, kernel adaptive filters have been proposed to identify non-linear systems [6]. Kernel adaptive filters are derived by applying the kernel method to linear adaptive filters, and several algorithms were proposed, i.e., the kernel recursive least squares (KRLS) [7], the kernel least mean square (KLMS) [8], the kernel normalized LMS (KNLMS) [10], kernel affine projection (KAP) [9] and its efficient implementation [10], the kernel ERLS-DCD [11] algorithms etc.

The kernel algorithms require some settings (e.g. the kernel functions, the kernel parameters, or the parameter for sparseness [6], [9]). In this paper, we propose to apply the proportionate principle proposed for linear filters [4] to the non-linear filters.

The contribution of this paper is that the proportionate idea is adapted for the KNLMS algorithm. To the best of our knowledge, this work is the first that proposes the proportionate principle for kernel based algorithms.

The paper is organized as follows. Section 2 represents an overview of the kernel methods, of the KNLMS algorithm and the proportionate-type algorithms. In Section 3, the proposed KPNLMS algorithm is introduced and its numerical complexity is investigated. The simulation results presented in Section 4 compare the proposed algorithm with KNLMS in different scenarios. Finally, the conclusions are given and ideas for further improvements are proposed.

2. OVERVIEW OF THE KERNEL METHODS, KNLMS AND PROPORTIONATE ALGORITHMS

2.1. The kernel method

The conventional kernel filters were described in many works (e.g. [6], [9], [12]). The input signal $x(n)$ at moment n is transformed into a high dimensional feature space F by a transformation function $\Phi(x)$ and the output of the adaptive filter is given by

$$f\big(\mathbf{x}(n)\big) = \Phi^T\big(\mathbf{x}(n)\big)\mathbf{w}(n), \tag{1}$$

where $\mathbf{w}(n) = [w_0(n), w_1(n),...,w_{M-1}(n)]$, are filter coefficient vector of the adaptive filter, where $w_i(n)$, and M is the i-th coefficient of the filter at moment n and the length of the filter respectively. The tap-input vector is $\mathbf{x}(n) = [x(n), x(n-1),...,x(n-M+1)]$. We assume that the filter vector $\mathbf{w}(n)$ is expressed as

$$\mathbf{w}(n) = \sum_{j=1}^{m} \alpha_j \Phi(\mathbf{y}(j)). \tag{2}$$

The vectors $\mathbf{y}(j)$ are a subset of $\mathbf{x}(l), l = 0,1,...,n-1$ and α_j is the weight corresponding to $\mathbf{y}(j)$. Then, the output in (1) is expressed [6] as

$$f(\mathbf{x}(n)) = \sum_{j=1}^{m} \alpha_j \left(\Phi^T(\mathbf{x}(n)) \Phi(\mathbf{y}(j)) \right). \tag{3}$$

In the kernel adaptive filter, $\mathbf{\alpha}(n) = [\alpha_1, \alpha_2,...,\alpha_m]^T$ is regarded as the coefficient vector of the adaptive filter instead of $\mathbf{w}(n)$ [8], [9]. In these algorithms the inner product $\Phi^T(\mathbf{x}(n))\Phi(\mathbf{y}(j))$ in Eq. (3) is obtained via the kernel function. A kernel function $k(\cdot,\cdot)$ is given as:

$$\forall \mathbf{a}, \mathbf{b} \in X \qquad k(\mathbf{a}, \mathbf{b}) = \Phi^T(\mathbf{a})\Phi(\mathbf{b}) \tag{4}$$

and used to calculate the inner product in the space F [6]. The Gaussian kernel defined as below is widely used in kernel adaptive filtering:

$$k(\mathbf{a}, \mathbf{b}) = \exp\left(-\zeta \|\mathbf{a} - \mathbf{b}\|^2\right), \tag{5}$$

where $\|\cdot\|$ is the Euclidean norm and ζ is the kernel parameter.

2.2. The kernel normalized LMS algorithm

The kernel normalized LMS (KNLMS) algorithm was presented in [9]. First, Eq. (3) is re-written as $f(\mathbf{x}(n)) = \mathbf{h}(n)\mathbf{\alpha}(n)$ where

$$\mathbf{h}(n) = [k(\mathbf{x}(n), \mathbf{y}(1)),...,k(\mathbf{x}(n), \mathbf{y}(m))]^T \tag{6}$$
$$= [h_1,...,h_m]^T.$$

Then, the filter $\mathbf{\alpha}(n)$ is updated using a linear adaptive algorithm. We define the matrix D, called the dictionary, as $\mathbf{D} = [\mathbf{y}(1),...,\mathbf{y}(m)]$. The vectors stored in the dictionary $\mathbf{D}$ are m ($m \le n$) input vectors of the previous time, where m is a variable determined by the algorithm below. Then, $\mathbf{D}_n$ ($\mathbf{D}$ at time n) and $\mathbf{h}(n)$ are updated according to the pseudo algorithm from [12]. The value of the threshold γ_0 [9] is determined according to the sparseness of the signal, and

ranges between 0 and 1. Finally, the filter coefficients $\mathbf{\alpha}(n)$ are updated as follows:

$$\mathbf{\alpha}(n) = \mathbf{\alpha}(n-1) + \mu \frac{d(n) - \mathbf{h}^T(n)\mathbf{\alpha}(n-1)}{\delta + \|\mathbf{h}(n)\|^2} \mathbf{h}(n) \tag{7}$$

where μ is the normalized step-size parameter in the range $0 < \mu < 2$ and δ is a small regularization.

2.3. The improved PNLMS algorithm

In the context of echo cancellation the far-end signal is $x(n)$, the reference signal of the adaptive filter is $d(n)$ and an adaptive filter is used to model the echo path. Let us assume an adaptive filter defined by the real-valued coefficients vector $\hat{\mathbf{h}}(n) = [\hat{h}_0(n), \hat{h}_1(n),...,\hat{h}_{L-1}(n)]^T$, where L is the length of the adaptive filter. The error signal is defined as:

$$e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n), \tag{8}$$

where $\mathbf{x}(n) = [x(n), x(n-1),...,x(n-L+1)]^T$.

A proportionate-type NLMS algorithm [3] updates its coefficients according to:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mu \mathbf{G}(n-1)\mathbf{x}(n)e(n)}{\delta + \mathbf{x}^T(n)\mathbf{G}(n-1)\mathbf{x}(n)} \tag{9}$$

where μ is the step-size parameter, δ is the regularization constant, and $\mathbf{G}(n-1)$ is an L x L diagonal matrix which assigns an individual step-size to each filter coefficient. The NLMS algorithm is obtained as a particular case of Eq. (9) if $\mathbf{G}(n-1)$ is the unity matrix i.e. the update equation is:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mu \mathbf{x}(n)e(n)}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)}. \tag{10}$$

In the case of the IPNLMS algorithm [4], the diagonal elements of $\mathbf{G}(n-1)$, denoted in the following by $g_l(n-1)$, with $0 \le l \le L-1$, are evaluated as:

$$g_l(n-1) = \frac{1-\beta}{2L} + \frac{|\hat{h}_l(n-1)|(1+\beta)}{2\sum_{i=0}^{L-1}|\hat{h}_i(n-1)| + \xi}, \tag{11}$$

where $-1 \le \beta < 1$ and the small positive constant ξ avoids division by zero. Therefore we have

$$\mathbf{G}(n-1) = diag\{g_0(n-1),...,g_{L-1}(n-1)\}. \tag{12}$$

3. THE PROPOSED ALGORITHM

3.1. The KPNLMS algorithm

The proportionate factors idea can be adapted in order to "proportionate" the KNLMS algorithm behavior. If we note the output kernel error $e(n) = d(n) - \mathbf{h}^T(n)\boldsymbol{\alpha}(n-1)$ Eq. (7) can be re-written as:

$$\boldsymbol{\alpha}(n) = \boldsymbol{\alpha}(n-1) + \mu \frac{\mathbf{h}(n) \cdot e(n)}{\delta + \mathbf{h}^T(n)\mathbf{h}(n)}. \qquad (13)$$

The similarity of Eq. (13) with Eq. 10 is further exploited by computing the following proportionate coefficients:

$$p_l(n-1) = \frac{1-\beta}{2m} + \frac{|h_l(n-1)|(1+\beta)}{2\sum_{i=0}^{m-1}|h_i(n-1)| + \xi}, \qquad (14)$$

where $h_l(n)$ are the coefficients of $\mathbf{h}(n)$, with $0 \le l \le m-1$. The matrix $\mathbf{P}(n-1)$ is formed as an m x m diagonal matrix with null coefficients except the diagonal terms given by $p_l(n-1)$. Therefore we have:

$$\mathbf{P}(n-1) = diag\{p_0(n-1),...,p_{m-1}(n-1)\}. \qquad (15)$$

The kernel proportionate NLMS algorithm (KPNLMS) updates the $\boldsymbol{\alpha}_n$ coefficients according to:

$$\boldsymbol{\alpha}(n) = \boldsymbol{\alpha}(n-1) + \mu \frac{\mathbf{P}(n-1)\mathbf{h}(n) \cdot e(n)}{\delta + \mathbf{h}^T(n)\mathbf{P}(n-1)\mathbf{h}(n)}. \qquad (16)$$

3.2. Numerical complexity issues

The added complexity of using the proportionate factors is rather small if compared with the overall complexity of the KNLMS algorithm or other algorithms such us KAP or KRLS. The KNLMS algorithm has $3m+1$ multiplications and $3m$ additions (does not include the function kernel computation) [6]. The KPNLMS requires additional $2m$ multiplications, $2m$ additions and one division if compared with the KNLMS algorithm.

4. SIMULATION RESULTS

The performances of the proposed algorithm for system identification and forward prediction problems were investigated by computer simulations. For all the simulations a white Gaussian noise of SNR =40 dB with zero mean was added, $\delta = 0.001$, $\gamma_0 = 0.9$, and $\xi = 10^{-8}$ were used.

Figure 1 shows the comparison of KNLMS and KPNLMS in terms of the mean squared errors (MSEs) averaged over 100 independent trials. Two filter lengths were considered ($M = 2$ and $M = 10$). The following parameters of the algorithms were used: $\zeta = -0.13$, $\gamma_0 = 1$, and $\beta = 0.8$. The desired signal was generated with the equations:

$$v(n) = 1.1\exp(-|v(n-1)|) + u(n);\ d(n) = v^2(n). \qquad (17)$$

It can be seen from Figure 1 than KPNLMS has a faster convergence and simultaneously obtains a lower excess MSE than KNLMS. Also, fewer iterations are needed in order to get the same MSE value in case of filters with a smaller M value.

The influence of a smaller the step size μ is examined in Figure 2. It can be seen from Figures 1 and 2 that the convergence of the algorithms is slower and higher MSE values are obtained for a smaller μ value. Also, it can be noticed that KPNLMS convergence properties are superior to those of KNLMS.

The influence of γ_0 is examined in Figure 3 for the parameters used in Figure 1. The last 500 MSE values were averaged over 100 independent runs. The value of γ_0 was varied from 0.1 to 0.95 in increments of 0.05. It can be seen from Figure 3 that the minimum mean MSE value is obtained for $\gamma_0 = 0.9$ in case of both M values. A practical compromise between the model order and its performance could be obtained by $\gamma_0 = 0.6$ too.
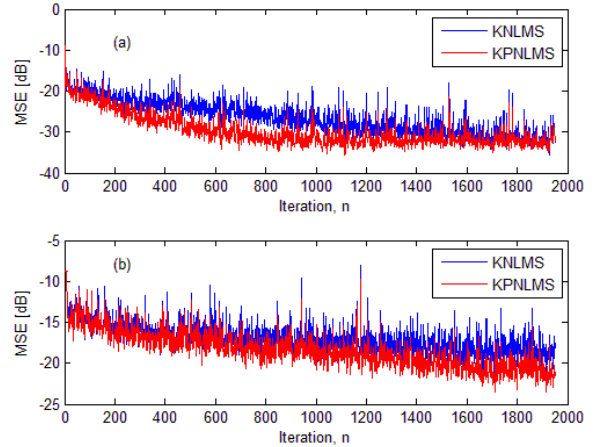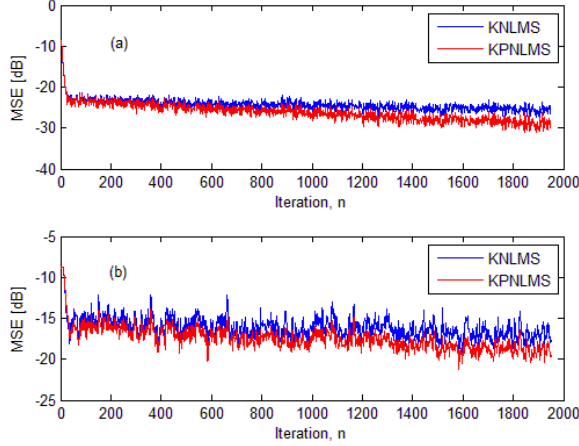

Figure 1: Comparison of convergence characteristics of KNLMS and KPNLMS applied to system identification for $\mu = 1$ in two cases: a) $M = 2$; b) $M = 10$.

Next, we show the results for the channel equalization of a multipath Rayleigh fading channel [6]. The length of the signal was 1000, $M = 5$, the maximum Doppler frequency was 100 Hz, and the sampling rate was set to 0.8s.
Figure 4 confirms that KPNLMS has better convergence properties than KNLMS. The evolution of the order of dictionary is also shown and emphasizes the necessity of using techniques for limiting its size without compromising the overall performances like those proposed in [6] or [13].

Next, we show the results of forward prediction. The following parameters were used: $\zeta = -3.73$, $\mu = 1$ and $\beta = 0.5$. For this case, the forward prediction equation is:

$$x(n) = \left(0.8 - 0.5\exp\left(-x(n-1)^{-2}\right)\right)x(n-1) - \tag{18}$$

$$\left(0.3 + 0.9\exp\left(-x(n-1)^2\right)\right)x(n-2)$$

$$+0.1\sin\left(x(n-1)\pi\right).$$

Figure 2: Comparison of convergence characteristics of KNLMS and KPNLMS applied to system identification for $\mu = 0.1$ in two cases: a) $M = 2$; b) $M = 10$.

It can be seen from Figure 5 that KPNLMS has a faster convergence and simultaneously maintain the lower excess MSE than KNLMS for both M values. Also, a smaller MSE value is obtained by using a higher M value.

Next, an investigation of the tracking ability is examined by simulating a non-stationary environment. The same parameters of Figure 5 were used. The non-stationary environment is simulated by a change of the equation starting at $n = 100$ for the forward prediction formula (Eq. 19).
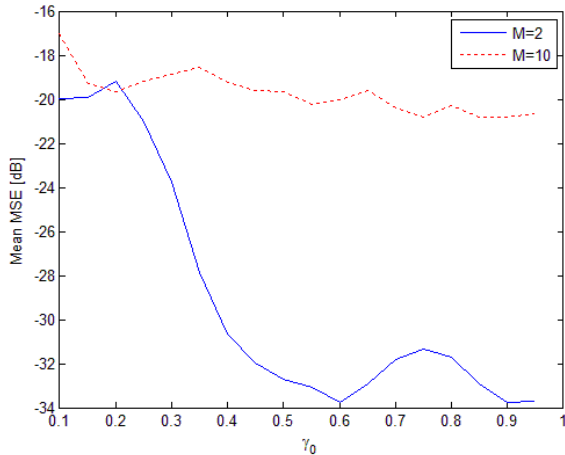

Figure 3: Mean MSE of KPNLMS for the last 500 iterations for different γ_0 values in two cases: $M = 2$ and $M = 10$.
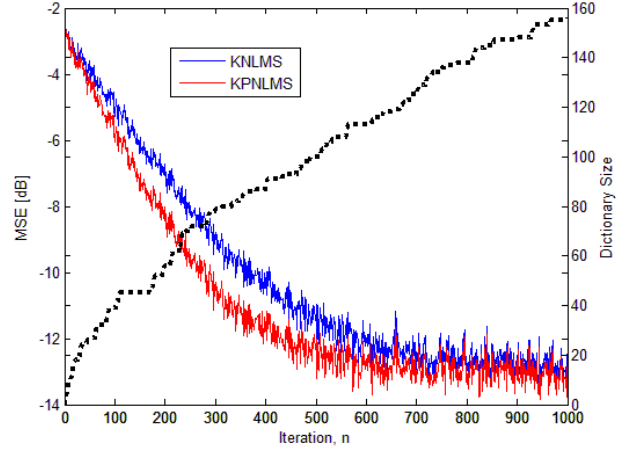

Figure 4: Comparison of convergence characteristics of KNLMS and KPNLMS for the channel equalization of a multipath Rayleigh fading channel
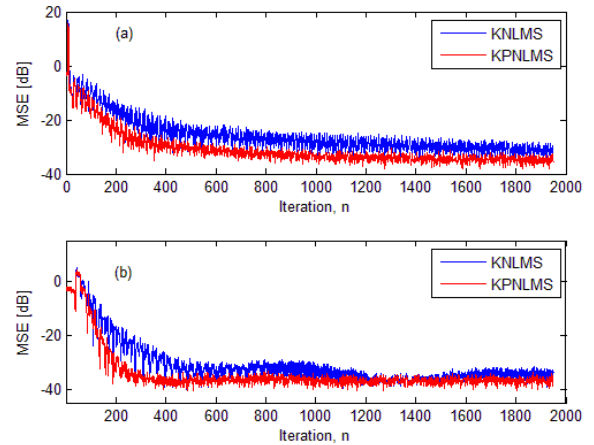

Figure 5: Comparison of convergence characteristics of KNLMS and KPNLMS applied to forward prediction problem for $\mu = 1$ in two cases: a) $M = 2$; b) $M = 10$.

For the first 100 iterations the following equation is used:

$$x(n) = \left(0.8\exp\left(-x(n-1)^{-2}\right)\right)x(n-1) - \tag{19}$$

$$\left(-0.1 - 0.9\exp\left(-x(n-1)^2\right)\right)x(n-2)$$

$$+0.5\sin\left(x(n-1)\pi\right).$$

It can be seen from Figure 6 that the tracking abilities of KPNLMS are superior to those of KNLMS. Also, from all previous simulations it can be seen that KPNLMS obtains better improvements over KNLMS especially for lower M values.

Like for the linear case, the value of β influences the convergence characteristics of KPNLMS. For the next

simulations $M = 5$, $\mu = 1$ and three values of β were used: -1, 0 and 1. In Figure 7a the system identification case from Figure 1 is used ($\zeta = -0.13$). In Figure 7b the non-stationary example of Figure 6 is used ($\zeta = -3.73$).
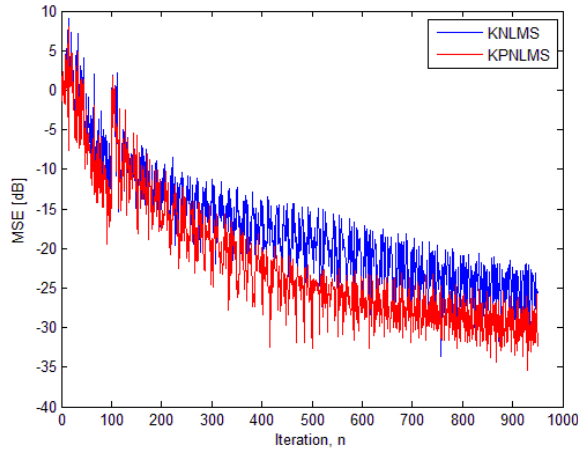

Figure 6: Comparison of convergence characteristics of KNLMS and KPNLMS applied to a non-stationary environment problem for $\mu = 1$ and $M = 2$.
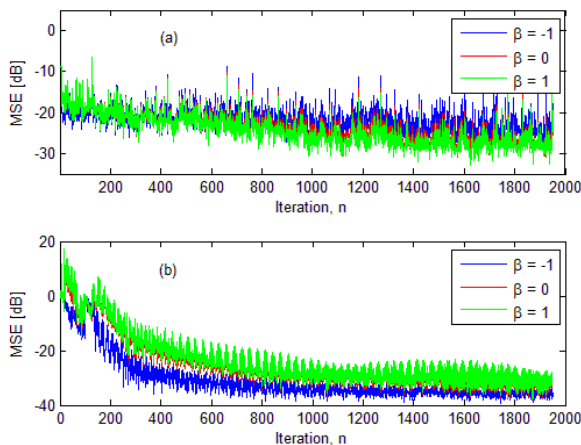

Figure 7: Comparison of convergence characteristics of KNLMS and KPNLMS in two cases for $M = 5$: a) system identification example; b) non-stationary environment example.

Figure 7a shows that for the system identification example the value of β should be closer to 1 in general, while for non-stationary environments (see Figure 7b) should be closer to -1. The value $\beta = 0$ leads to intermediate convergence performance in both situations.

5. CONCLUSIONS

In this paper, the superior convergence characteristics and tracking abilities of the KPNLMS algorithm over the KNLMS algorithm for system identification, forward prediction and non-stationary environment has been proved by computer simulations. The influence of different algorithm parameters has been investigated. It is also shown that the added complexity is rather small, therefore KPNLMS could prove a suitable choice for the investigated applications. Future work would be focused in adapting the proportionality idea to other kernel based algorithms and investigate their use in other applications.

REFERENCES

[1] A.H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, 2003.

[2] D. L. Duttweiler, Proportionate normalized least-mean-squares adaptation in echo cancellers, *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508–518, 2000.

[3] H. Deng, and M. Doroslovački, Proportionate adaptive algorithms for network echo cancellation, *IEEE Trans. Signal Processing*, vol. 54, no. 5, pp. 1794–1803, 2006.

[4] J. Benesty and S.L. Gay, An improved PNLMS algorithm, in *Proc. of IEEE ICASSP*, vol. 2, pp. 1881–1884, 2002.

[5] F. Albu, H.K.Kwan, New proportionate affine projection sign algorithms, in *Proc. of ISCAS 2013*, pp. 1789 – 1793, 2013.

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

[7] Y. Engel, S. Mannor, and R. Meir, The Kernel Recursive Least-Squares Algorithm, *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275–2285, 2004.

[8] W. Liu, P.P. Pokharel, and J. C. Principe, The Kernel Least-Mean-Square Algorithm, *IEEE Transactions on Signal Processing*, vol. 56, pp. 543–554, 2008.

[9] C. Richard, J. C. M. Bermudez, and P. Honeine, Online Prediction of Time Series Data With Kernels, *IEEE Transactions on Signal Processing*, vol. 57, pp. 1058–1067, 2009.

[10] F. Albu, K. Nishikawa, M. Rotaru, D. Coltuc, An efficient implementation of the kernel affine projection algorithm, accepted at *IEEE ISPA 2013*.

[11] Y. Ogawa and K. Nishikawa, A Kernel Adaptive Filter based on ERLS-DCD Algorithm, in *Proc. of Intl Tech. Conf. Circuits Systems, Computer, Communications 2011*, no.P4-13, pp.1228–1231, 2011.

[12] K. Nishikawa, and H. Nakazato, Mixture structure of kernel adaptive filters for improving the convergence characteristics, in *Proc. of APSIPA 2012,* 2012

[13] K. Nishikawa, and K. Makizaki, Fixed Order Implementation Method of Kernel Adaptive Filters with Lower Computational Complexity, in *Proc. of APSIPA ASC 2011*, 2011.