Low Complexity Kernel Affine Projection-type Algorithms with a Coherence Criterion

Felix Albu
Faculty of Electronics and Information Technology
Valahia University of Targoviste
Targoviste, Romania
felix.albu@valahia.ro

Kiyoshi Nishikawa
Faculty of System Design
Tokyo Metropolitan University
Tokyo, Japan
knishikawa@m.ieice.org

Abstract— **In this paper, two new kernel adaptive algorithms are proposed. An approximation is used in order to derive the pseudo kernel affine projection algorithm and the pseudo kernel proportionate affine projection algorithm, respectively. The computational efficiency and performance of the proposed algorithms is verified for a nonlinear system identification application.**

Keywords— kernel affine projection algorithm; proportionate algorithms; nonlinear system identification;

I. INTRODUCTION

Linear adaptive filters have been used to identify an unknown system [1]. Several applications have been envisaged e.g., echo cancellation [2], active noise control [3] etc. Among the most promising adaptive algorithms are the affine projection (AP) algorithm [4] and proportionate affine projection (PAP) algorithm [5]. The kernel adaptive filters have been presented in [6] as an extension of the known families of the linear counterparts and their suitability for non-linear system identification has been investigated. The kernel methods [6] has been applied to the linear adaptive filters and several versions has been proposed (e.g., kernel affine projection (KAP) [7], the dichotomous coordinate descent KAP [8], the kernel proportionate affine projection (KPAP) algorithm [9], the kernel recursive least squares (KRLS) [10] and its fixed-budget version [11] etc.).

The paper proposes to apply the "pseudo" approximation used for the PAP algorithm [12] and adapt the idea to the kernel affine projection with a coherence criterion and the kernel proportionate affine projection respectively. The resemblance between KAP algorithm and the evolutionary affine projection algorithm firstly mentioned in [12] and efficiently implemented in [2] is exploited. To the best of our knowledge, this approximation hasn't been applied yet to the KAP based algorithms. The new algorithms are called pseudo kernel affine projection (PKAP) algorithm and pseudo kernel proportionate affine projection (PKPAP) algorithm respectively.

The paper is organized as follows. Section II presents the proposed algorithms and their numerical complexities is investigated. The simulation results for nonlinear system identification applications are presented in Section III. Finally, the conclusions are presented in Section IV.

II. THE PROPOSED ALGORITHMS

A. The KAP algorithm with a coherence criterion

The kernel methods are based on a non-linear transformation $\varphi(\cdot)$ of the input data $\mathbf{u}_i$ into a high-dimensional feature space [7]. In this space, the linear adaptive algorithms are applied to the transformed input signal $\varphi(\mathbf{u}_i)$ [7]. The kernel satisfies Mercer's conditions [6] and nonlinear versions of the linear adaptive algorithms are obtained using inner products [13].

$$k(\mathbf{u}_i,\mathbf{u}_j) = \langle\varphi(\mathbf{u}_i),\varphi(\mathbf{u}_j)\rangle \tag{1}$$

The function $\varphi(\cdot)$ has not an explicit formula and the most used kernels are the Gaussian kernel $k(\mathbf{u}_i,\mathbf{u}_j)=\exp(-\|\mathbf{u}_i-\mathbf{u}_j\|^2/2\beta_0^2)$ and the Laplacian kernel $k(\mathbf{u}_i,\mathbf{u}_j)=\exp(-\|\mathbf{u}_i-\mathbf{u}_j\|/\beta_0)$ where β_0 is the kernel bandwidth [6], [7]. In the feature space the unknown system is modeled as follows

$$\psi_n(\cdot)=\sum_{j=1}^{m}\alpha_j k(\cdot,\mathbf{u}_{\omega_j}) \tag{2}$$

at the time n, where ω_j's form an m-element subset of I_n of $\{1,\ldots,n\}$. $\{k(\cdot,\mathbf{u}_{\omega_j})\}_{j=1}^{m}$ is called the dictionary and m is the order of the kernel expansion [7], [9]. An adaptive algorithm is used to estimate α_j in (2). However, the kernel algorithms have some additional processes. The insertion of $k(\cdot,\mathbf{u}_n)$ into the dictionary is made if

$$\max_{\omega_j\in I_{n-1}}|k(\mathbf{u}_n,\mathbf{u}_{\omega_j})|\le\mu_0 \tag{3}$$

where $0 \le \mu_0 \le 1$ is a parameter [7]. The order of the kernel filters will increase in time and in some implementation a restriction is imposed [11].
The KAP algorithm with coherence criterion was proposed in [7]. The kernel output error vector is given by

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{H}_n \hat{\alpha}_{n-1} \tag{4}$$

where $\mathbf{d}_n = \left[d_n, \ldots, d_{n-p+1}\right]^T$ is the observations vector, $\mathbf{e}_n = \left[e_n, \ldots, e_{n-p+1}\right]^T$ is the output error, p is the order of the algorithm and $\mathbf{H}_n = \left[k(\mathbf{u}_{n-i+1}, \mathbf{u}_{\omega_j})\right]_{i=1,\ldots,p;\, j=1,\ldots,m}$ [7].

The update equations for $\hat{\alpha}_n$ are obtained from the following problem at time step n [7]

$$\min_{\alpha} \| \alpha - \hat{\alpha}_{n-1} \|^2 \quad \text{subject to } \mathbf{d}_n = \mathbf{H}_n \alpha \tag{5}$$

The solution to (5) is found by minimizing the Lagrangian function [7]

$$J(\alpha, \lambda) = \| \alpha - \hat{\alpha}_{n-1} \|^2 + \lambda^t (\mathbf{d}_n - \mathbf{H}_n \alpha) \tag{6}$$

where λ is the vector of Langrange multipliers. $k(\bullet, \mathbf{u}_n)$ is represented by the kernel functions of the dictionary if $\max_{j=1,\ldots,m} |k(\mathbf{u}_n, \mathbf{u}_{\omega_j})| > \mu_0$ [7] and the recursive update equation for $\hat{\alpha}_n$ is

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \eta \mathbf{H}_n^t \left(\varepsilon \mathbf{I} + \mathbf{H}_n \mathbf{H}_n^t\right)^{-1} \mathbf{e}_n \tag{7}$$

It can be easily seen the resemblance between the affine projection equations [1] and Equation (7). The size and time evolution is different although the order of the filter and the dictionary are the same.
If $\max_{j=1,\ldots,m} |k(\mathbf{u}_n, \mathbf{u}_{\omega_j})| \le \mu_0$ $k(\bullet, \mathbf{u}_n)$ is inserted into the dictionary where it is denoted by $k(\bullet, \mathbf{u}_{\omega_{m+1}})$ [7]. The Equation (5) is modified as

$$\min_{\alpha} \| \alpha_{1,\ldots,m} - \hat{\alpha}_{n-1} \|^2 + \alpha_{m+1}^2 \text{ subject to } \mathbf{d}_n = \mathbf{H}_n \alpha \tag{8}$$

As shown in [7], one entry is added to the vector $\hat{\alpha}_n$ and $\mathbf{H}_n$ is increased by appending the column $[k(\mathbf{u}_n, \mathbf{u}_{\omega_{m+1}}) \ldots k(\mathbf{u}_{n-p+1}, \mathbf{u}_{\omega_{m+1}})]^T$. If $\mathbf{e}_n = \mathbf{d}_n - \mathbf{H}_n \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix}$ the update equation is the following [9]

$$\hat{\alpha}_n = \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix} + \eta \mathbf{H}_n^t \left(\varepsilon \mathbf{I} + \mathbf{H}_n \mathbf{H}_n^t\right)^{-1} \mathbf{e}_n \tag{9}$$

B. The KPAP algorithm with a coherence criterion

The kernel proportionate AP (KPAP) algorithm proposed in [9] uses the proportionate coefficients based on the $\hat{\alpha}_n$ coefficients as in [14]:

$$c_{n-1}^l = \frac{1-\beta}{2m} + \frac{\left|\hat{\alpha}_{n-1}^l\right|(1+\beta)}{2\sum_{i=0}^{m-1}\left|\hat{\alpha}_{n-1}^i\right| + \xi}, \tag{10}$$

where $\hat{\alpha}_{n-1}^l$ are the coefficients of $\hat{\alpha}_n$. We have $\mathbf{C}_{n-1} = diag\left\{c_{n-1}^0, \ldots, c_{n-1}^{m-1}\right\}$ and the updating of the equation is as follows [9]

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \eta \mathbf{C}_{n-1} \mathbf{H}_n^t \left(\varepsilon \mathbf{I} + \mathbf{H}_n \mathbf{C}_{n-1} \mathbf{H}_n^t\right)^{-1} \mathbf{e}_n \tag{11}$$

where η is the normalized step-size parameter in the range $0 < \eta < 2$. For the order increase case the updating is given by

$$\hat{\alpha}_n = \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix} + \eta \mathbf{C}_{n-1} \mathbf{H}_n^t \left(\varepsilon \mathbf{I} + \mathbf{H}_n \mathbf{C}_{n-1} \mathbf{H}_n^t\right)^{-1} \mathbf{e}_n \tag{12}$$

The KPAP algorithm complexity is increased with $4m$ multiplications and additions. This complexity increase is small if compared with $O(mp^2)$ of KAP algorithm [7].

C. The proposed "pseudo" versions

The numerical complexity of KAP and KPAP algorithms can be reduced if similar approximations with those used to derive the pseudo affine projection algorithms [12], [15] are used. If we note by $\mathbf{R}_n = \varepsilon \mathbf{I} + \mathbf{H}_n \mathbf{H}_n^t$ then Equation (7) is written as

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \eta \mathbf{H}_n^t \mathbf{s}_n \tag{13}$$

where $\mathbf{s}_n = \mathbf{R}_n^{-1} \mathbf{e}_n$. Like in [12], instead of the very complex operation of inverting $\mathbf{R}_n$ the following update is made for the pseudo kernel affine projection (PKAP) algorithm

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \eta \mathbf{H}_n^t \mathbf{s}_n' \tag{14}$$

where $\mathbf{s}_n'$ is obtained by solving $\mathbf{R}_n \mathbf{s}_n' = \left[\mathbf{e}_{n,1}\ \mathbf{0}^t\right]^T$ and $\mathbf{e}_{n,1}$ is the first element of $\mathbf{e}_n$. If the order is increasing

$$\hat{\alpha}_n = \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix} + \eta \mathbf{H}_n^t s_n' \qquad (15)$$

This approximation reduce the complexity about p times [16]. The equations of PKAP algorithm are shown in Table I.

Algorithm 1 The PKAP Algorithm with coherence criterion

Initialization:

Fix the memory length p, the step-size η, the regularization parameter ϵ, $m = 1$, $\hat{\alpha}_p = 0$ Insert $k(\cdot, \mathbf{u}_p)$ into the dictionary, denote it by $\kappa(\cdot, \mathbf{u}_{\omega_j})$.
$\mathbf{H}_p = [\kappa(\mathbf{u}_p, \mathbf{u}_{\omega_1}) \; \ldots \; \kappa(\mathbf{u}_1, \mathbf{u}_{\omega_1})]^t$

Iteration:

for $n > p$ **do**
  Get $(\mathbf{u}_n, d_n)$
  if $\max_{j=1,\ldots,m} |\kappa(u_n, u_{\omega_j})| > \mu_0$ **then**
    compute $\mathbf{H}_n$, solve $\mathbf{R}_n \mathbf{s}_n' = [\mathbf{e}_{n,1} \; \mathbf{0}^t]^t$ and $\hat{\alpha}_n$ using equation (14)
  end if
  if $\max_{j=1,\ldots,m} |\kappa(u_n, u_{\omega_j})| \le \mu_0$ **then**
    $m = m + 1$, insert $\kappa(\cdot, \mathbf{u}_n)$ into the dictionary, denote it by $\kappa(\cdot, \mathbf{u}_{\omega_m})$, solve $\mathbf{R}_n \mathbf{s}_n' = [\mathbf{e}_{n,1} \; \mathbf{0}^t]^t$ and calculate $\hat{\alpha}_n$ using equation (15)
  end if
end for

In case of KPAP algorithm, if we note by $\mathbf{Q}_n = \varepsilon \mathbf{I} + \mathbf{H}_n \mathbf{C}_{n-1} \mathbf{H}_n^t$ then Equation (11) is written as

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \eta \mathbf{C}_{n-1} \mathbf{H}_n^t s_n \qquad (16)$$

where $s_n = \mathbf{Q}_n^{-1} \mathbf{e}_n$. Like above, the following update is made for the pseudo kernel proportionate affine projection (PKPAP) algorithm

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \eta \mathbf{C}_{n-1} \mathbf{H}_n^t s_n'' \qquad (17)$$

where s_n'' is obtained by solving $\mathbf{Q}_n s_n'' = \left[\mathbf{e}_{n,1} \; \mathbf{0}^t \right]^t$. If the order is increasing

$$\hat{\alpha}_n = \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix} + \eta \mathbf{C}_{n-1} \mathbf{H}_n^t s_n'' \qquad (18)$$

The equations of PKPAP algorithm are shown in Table II. The computation of the matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$ can be efficiently implemented as in made as in an efficient way as in [2] and [8]. Table III shows the computational cost per iteration of KAP, KPAP, PKAP and PKPAP algorithms.

Algorithm 2 The PKPAP Algorithm with coherence criterion

Initialization:

Fix the memory length p, the step-size η, the regularization parameter ϵ, $m = 1$, $\hat{\alpha}_p = 0$ Insert $k(\cdot, \mathbf{u}_p)$ into the dictionary, denote it by $\kappa(\cdot, \mathbf{u}_{\omega_1})$.
$\mathbf{H}_p = [\kappa(\mathbf{u}_p, \mathbf{u}_{\omega_1}), \; \ldots \; \kappa(\mathbf{u}_1, \mathbf{u}_{\omega_1})]^t$

Iteration:

for $n > p$ **do**
  Get $(\mathbf{u}_n, d_n)$
  Compute c_{n-1}^l using equation (10) and form the matrix $\mathbf{C}_{n-1} = \text{diag}\{c_{n-1}^0, \ldots, c_{n-1}^{m-1}\}$
  if $\max_{j=1,\ldots,m} |\kappa(u_n, u_{\omega_j})| > \mu_0$ **then**
    compute $\mathbf{H}_n$, solve $\mathbf{Q}_n \mathbf{s}_n'' = [\mathbf{e}_{n,1} \; \mathbf{0}^t]^t$ and $\hat{\alpha}_n$ using equation (17)
  end if
  if $\max_{j=1,\ldots,m} |\kappa(u_n, u_{\omega_j})| \le \mu_0$ **then**
    $m = m + 1$, insert $\kappa(\cdot, \mathbf{u}_n)$ into the dictionary, denote it by $\kappa(\cdot, \mathbf{u}_{\omega_m})$, solve $\mathbf{Q}_n \mathbf{s}_n'' = [\mathbf{e}_{n,1} \; \mathbf{0}^t]^t$ and calculate $\hat{\alpha}_n$ using equation (18)
  end if
end for

Figure 1 shows the ratios of numerical complexities in terms of additions and multiplications of PKAP/KAP and PKPAP/KPAP, respectively, for variable m and $p = 5$. It can be noticed that important computational savings are obtained regardless if there is or not an order increase, i.e. at least 55%. It is easily to see that the computational savings are higher for higher values of p. Also, the complexity ratio without order increase is much smaller than that with order increase for small m values. On the other hand, when m has high values there is a small difference between the complexities ratio with order increase and without order increase.

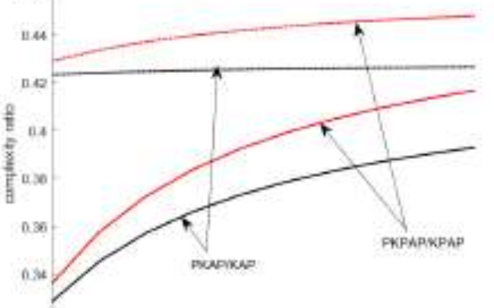

Fig. 1. The complexity ratios of PKAP/KAP and PKPAP/KPAP respectively for different m values and $p = 5$; without order increase (solid line), with order increase (dotted line).

Table 1: Computational Cost per Iteration of KAP, KPAP, PKAP, PKPAP Algorithms

		without order increase	with order increase
KAP[7]	×	$(p^2+2p)m+p^3+p$	$(p^2+2p)m+p^3+2p^2+p$
	+	$(p^2+2p)m+p^3+p^2$	$(p^2+2p)m+p^3+p^2+p-1$
KPAP[9]	×	$(p^2+2p+2)m+p^3+p$	$(p^2+2p+2)m+2p^3+2p^2+p$
	+	$(p^2+2p+2)m+p^3+p^2$	$(p^2+2p+2)m+p^3+p^2+p-1$
PKAP	×	$3pm+p^3/6+p^2-p/6$	$3pm+p^3/6+2p^2+5p/6$
	+	$3pm+p^3/6+p^2-7p/6$	$3pm+p^3/6+2p^2-7p/6+1$
PKPAP	×	$(3p+2)m+p^3/6+p^2-p/6$	$(3p+2)m+p^3/6+2p^2+5p/6$
	+	$(3p+2)m+p^3/6+p^2-7p/6$	$(3p+2)m+p^3/6+2p^2-7p/6+1$

III. SIMULATION RESULTS

In the following simulations we consider the nonlinear system described by $d_n=\left(0.8-0.4\exp\left(-d_{n-1}^2\right)\right)d_{n-1}-\left(0.3+0.8\exp\left(-d_{n-1}^2\right)\right)d_{n-2}+0.05\sin\left(d_{n-1}\pi\right)$, where d_n is the desired signal [7]. The data was generated as in [7] starting from (0.1, 0.1) and d_n was corrupted by a zero-mean Gaussian distribution noise and 0.1 standard deviation. The Gaussian kernel $k\left(\mathbf{u}_i,\mathbf{u}_j\right)=\exp\left(-3.73\cdot\|\mathbf{u}_i-\mathbf{u}_j\|^2\right)$, $\xi=10^{-8}$ and the regularization parameter $\varepsilon=0.07$ were used. Figure 2a shows the MSE difference in dB between KAP and PKAP while figure 2b shows the difference between KPAP and PKPAP for p = 3, $\mu_0=0.6$ and $\beta=-0.9$. Fifty MSE curves were averaged for Fig. 2. The value of μ_0 was found as the best coherence value that provides the minimum mean MSE [9]. Also, it was shown in [9] that KPAP algorithm can obtain an improvement of about 1 dB average over that of KAP algorithm for most iterations of this application. It can be noticed that, apart from a difference in the initial converging phase of few dBs, the MSE difference is less than 0.5 dB after convergence. Also, it can be noticed that the amplitude of the MSE difference is higher for the proportionate algorithms in the initial convergence phase.

Therefore, the compromise in performance is worth considering if a much lower numerical complexity is needed. Similar conclusions were obtained for other parameters of the investigated algorithms.

Future work will be focused on investigating the multi-kernel approach as in [17], developing Gauss-Seidel or dichotomous coordinate descent versions as in [18] - [20], trying sparse versions as in [21] – [22], apply them to the active noise control as in [3], [18] and [23] or point spreads function estimation [24].

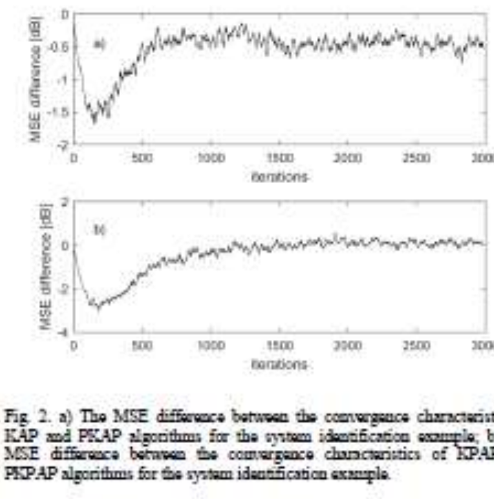

Fig. 2. a) The MSE difference between the convergence characteristics of KAP and PKAP algorithms for the system identification example; b) The MSE difference between the convergence characteristics of KPAP and PKPAP algorithms for the system identification example.

IV. CONCLUSIONS

The pseudo KAP and pseudo KPAP algorithms have been proposed. It was proved that the PKAP and PKPAP algorithms are more computationally efficient than the original algorithms having only a minor performance loss for the nonlinear system identification application.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific research and Innovation, CNCS/CCCDI-UEFISCDI project number PN-III-P2-2.1-PED-2016-0651.

REFERENCES

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

[2] F. Albu, C. Paleologu and J. Benesty, J. "A Variable Step Size Evolutionary Affine Projection Algorithm", Proceedings of ICASSP 2011, pp. 429-432, 2011.

[3] A. Gonzalez, F. Albu, M. Ferrer and M. Diego, "Evolutionary and variable step size affine projection algorithms for active noise control", IET Signal Processing, vol. 7, (6), pp. 471-476, 2013.

[4] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties", Electron. Commun. Japan., vol. 67A, (5), pp. 19–27, 1984.

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

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

[7] C. Richard, J. C. M. Bermudez and P. Honeine, "Online Prediction of Time Series Data with Kernels", IEEE Trans. on Signal Processing, 57(3), pp. 1058–1067, 2009.

[8] F. Albu, K. Nishikawa, M. Rotaru and D. Coltuc, "An efficient implementation of the kernel affine projection algorithm", Proceedings of IEEE ISPA, pp. 342-346, 2013.

[9] F. Albu and K. Nishikawa, "Nonlinear Adaptive Filtering with a Family of Kernel Affine Projection Algorithms", Handbook of Research on Advanced Intelligent Control Engineering and Automation, pp. 66-89, 2015.

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

[11] S. Van Vaerenbergh, I. Santamaria, W. Liu and J. Principe, "Fixed-Budget Kernel Recursive Least-Squares", Proceedings of ICASSP 2010, pp. 1882–1885, 2010.

[12] F. Albu and A. Fagan, "The Gauss-Seidel Pseudo Affine Projection Algorithm and its Application for Echo Cancellation", 37th Asilomar Conf. Sign., Syst., Comp., pp. 1303-1306, Nov. 2003.

[13] D. Rzepka, "Fixed-budget kernel least mean squares", Proceedings of IEEE 17th Conference on Emerging Technologies & Factory Automation, pp. 1-4, 2012.

[14] J. Benesty and S.L. Gay, "An improved PNLMS algorithm", Proceedings of ICASSP 2002, pp. 1881-1884, 2002.

[15] F. Bouteille, P. Scalart, M. Corazza, "Pseudo Affine Projection Algorithm New Solution for Adaptive Identification", Eurospeech, 1999

[16] G.H. Golub, C.F. Van Loan, Matrix Computation, Baltimore, MD, Johns Hopkins University Press, 1996.

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

[18] F. Albu, "An Efficient Multichannel Filtered-X Affine Projection Algorithm", IEE Electronics Letters, Vol. 42, Issue 7, pp. 59-60, 2006.

[19] F Albu, C Kotropoulos, "Modified Gauss-Seidel affine projection algorithm for acoustic echo cancellation", in Proc. of ICASSP 2005, pp. 121-124, 2005.

[20] F. Albu, H.K. Kwan, "Fast block exact Gauss-Seidel pseudo affine projection algorithm", IEE Electronics Letters, Vol. 40, Issue 22, pp. 1451-1453, 2004.

[21] Y. Li, Y. Wang, R. Yang and F. Albu, "A Soft Parameter Function Penalized Normalized Maximum Correntropy Criterion Algorithm for Sparse System Identification", Entropy 2017, 19(1), 45, 2017.

[22] Y. Li, Y. Wang, and F. Albu, "Sparse channel estimation based on a reweighted least-mean mixed-norm adaptive filter algorithm," in Proc. of EUSIPCO 2016, 2016.

[23] H. Bao, I. M. S. Panahi, "Active noise control based on kernel least-mean-square algorithm", Forty-Third Asilomar Conference, pp. 642-644, 2009.

[24] P. Corcoran, F. Albu, A. Drimbarean, A. Zamfir, "Adaptive PSF estimation technique using a sharp preview and a blurred image", US 8,204,330, June 2012.