# Improved Set-Membership Partial-Update Pseudo Affine Projection Algorithm

Felix Albu
EEEIT Department
Valahia University of Targoviste
Targoviste, Romania
felix.albu@valahia.ro

Paulo S. R. Diniz
DEL/Poli & PEE/COPPE/UFRJ
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
diniz@smt.ufrj.br

***Abstract*— In this paper, an improved set-membership partial-update pseudo affine projection (I-SM-PUPAP) algorithm is presented. An approximation that leads to solving a linear system with a direct method is used. It is proved that I-SM-PUPAP algorithm has a much lower numerical complexity and memory requirements than recently proposed I-SM-PUAP algorithm. Simulation results identify an inherent compromise between the convergence rate, complexity reduction and the number of updates.**

***Keywords—adaptive filtering; set-membership filtering; partial-update; identification problem.***

## I. Introduction

The adaptive filtering [1] has found application in many areas such as noise cancellation, active noise control, signal prediction, echo cancellation, communications, radar, speech processing etc. Various NLMS, affine projection (AP), recursive least square (RLS) or lattice RLS based algorithms have been proposed [1]-[3]. In many applications where the number of filter coefficients is high (e.g. acoustic echo cancellation) there is a need for less numerically complex algorithms. It was shown in many papers (e.g. [4]-[10]) that the average computational complexity of the adaptive filters can be reduced if not all filter weights are changed at each algorithm iteration.

Another approach to the above called partial-update (PU) algorithms is to use the set-membership filtering (SMF) idea [1]. In this approach the filter weights are not updated when the estimation error is lower than a chosen constant [9]. Implementation of SMF algorithms involves two main steps: 1) error calculation and comparison with a prescribed threshold, 2) parameter update. The set-membership versions of the NLMS (SM-NLMS) and affine projection (AP) algorithms (SM-AP) are less numerically complex than NLMS and AP respectively [11–13]. It has been verified [13] that smaller estimation bound accelerates the convergence speed, but also increases the number of updates. Also, using a higher error estimation threshold reduces the number of updates, but decreases the convergence rate.

The SM-PUAP algorithm, introduced in [1], was shown that simultaneously reduce the number of updates and accelerate the convergence speed. Firstly, a hypersphere centered at the present weight vector whose radius equals the distance between the present weight vector and the weight vector that would be obtained with the SM-AP algorithm is constructed [12]. This radius is used in the improved SM-PUAP (I-SM-PUAP) algorithm [12].

However the recently proposed I-SM-PUAP algorithm would be more useful if its internal calculations are examined closely aiming at further reduction in the computational complexity. The resulting algorithm should achieve this goal without compromising the good performance of the I-SM-PUAP algorithm in terms of speed of convergence and misadjustment in stationary environment.

In this paper, we propose a less computationally complex algorithm by incorporating previously used technique to derive the pseudo affine projection algorithm [14] from the AP algorithm. This algorithm is called improved set-membership partial update pseudo affine projection (I-SM-PUPAP) algorithm.

In the following section the I-SM-PUPAP algorithm is presented. Section III shows simulations of the considered algorithms and the tradeoff between complexity reduction and the convergence speed is investigated whereas section IV presents the conclusions.

## II. The I-SM-PUPAP algorithm

The following vectors, matrices and parameters are used: $\mathbf{X}(n)=\left[\mathbf{x}(n),...,\mathbf{x}(n-L+1)\right]$, where $L$ is the data reuse factor, $\mathbf{x}(n)=\left[x(n),...,x(n-N+1)\right]^T$, where $N$ is the order of adaptive filter, $\mathbf{I}$ is an $L\times L$ identity matrix, $\delta$ is a small positive constant, $\mathbf{e}(n)=\left[e_0(n)...e_{L-1}(n)\right]^T$ with $i=0,...,L-1$, and $\mathbf{d}(n)=\left[d(n)...d(n-L+1)\right]^T$.

In the partial-update algorithm $M$ out of $N$ coefficients are changed [12]. The coefficients that are updated are specified by $I_M(n)=\{i_1(n),...,i_M(n)\}$ with $\{i_j(n)\}_{j=1}^{M}$ from the

$\{1,...,N\}$ set. A comprehensive description of the I-SM-PUAP algorithm can be found in [12]. Table I shows the equations of I-SM-PUAP algorithm [12].

The I-SM-PUAP algorithm performs an update whenever the adaptive filter coefficients are not at the intersection of the last $L$ constraint sets delimited by the threshold denoted by $\bar{\gamma}$. The coefficient update is determined by minimizing the following constrained cost function.

$$\min \frac{1}{2}\left\|\mathbf{w}(n+1)-\mathbf{w}(n)\right\|^2$$

subject to

$$\mathbf{d}(n)-\mathbf{X}^T(n)\mathbf{w}(n+1)=\gamma(n) \tag{1}$$

$$\mathbf{C}_{I_m(n)}\left[\mathbf{w}(n+1)-\mathbf{w}(n)\right]=\mathbf{0} \tag{2}$$

where in this paper we will consider only a single constraint set related to the error $e_0(n)$, meaning that the coefficients are updated whenever the error is such that $\left|e_0(n)\right|=\left|d(n)-\mathbf{w}^T(n)\mathbf{x}(n)\right|>\bar{\gamma}$. In this case the vector equation in the first constraint becomes a scalar equation. The solution to this optimization problem can be found in detail in [12].

TABLE I. THE I-SM-PUAP ALGORITHM

| Step | Initialization $\mathbf{w}(-1)=\mathbf{x}(-1)=[0...0]^T$; $\delta$, $\bar{\gamma}$ |
|---|---|
| | For $n \ge 0$ |
| 1 | $\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{X}^T(n)\mathbf{w}(n)$ |
| 2 | If $\left\|e_0(n)\right\|>\bar{\gamma}$ |
| 3 | $\mu(n)=\min\left(\frac{\left\|-e_0(n)\pm\bar{\gamma}\right\|}{\left\|\mathbf{x}(n)\right\|_2}\right)$ |
| 4 | $\mathbf{a}(n)=\mathbf{C}_{I_m(n)}\mathbf{X}(n)\left[\mathbf{X}^T(n)\mathbf{C}_{I_m(n)}\mathbf{X}(n)+\delta\mathbf{I}\right]^{-1}\mathbf{e}(n)$ |
| 5 | $\mathbf{w}(n+1)=\mathbf{w}(n)+\mu(n)\mathbf{a}(n)/\left\|\mathbf{a}(n)\right\|_2$ |
| 6 | Else |
| 7 | $\mathbf{w}(n+1)=\mathbf{w}(n)$ |
| 8 | End |
| 9 | End |

Using a similar approach to the pseudo affine projection algorithm [14], the first column of the inverse matrix, required in step 4 of Table II and denoted as $\hat{\mathbf{p}}(n)$ is computed, therefore the following linear system is solved by a direct technique:

$$\left(\mathbf{X}^T(n)\mathbf{C}_{I_m(n)}\mathbf{X}(n)+\delta\mathbf{I}\right)\hat{\mathbf{p}}(n)=[1,0...0]^T. \tag{3}$$

The weight update equation is modified as follows

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu(n)\hat{\mathbf{a}}(n)/\left\|\hat{\mathbf{a}}(n)\right\|_2, \tag{4}$$

where $\hat{\mathbf{a}}(n)$ is defined as follows

$$\hat{\mathbf{a}}(n)=\mathbf{C}_{I_m(n)}\mathbf{X}(n)\hat{\mathbf{p}}(n)e_0(n). \tag{5}$$

In this paper the $LDL^T$ method [15] is used due to its low complexity and stability. Table II shows the equations of the proposed I-SM-PUPAP algorithm. Important complexity savings in terms of multiplications are obtained by replacing steps 4 and 5 from Table I with Eqs. 3-5. Also, the step 1 of I-SM-PUPAP is ($L$ + 1) times less complex than step 1 of I-SM-PUAP. The numerical complexity of steps 3-5 of I-SM-PUAP algorithm in terms of multiplications for one iteration i.e., when $\left|e_0(n)\right|>\bar{\gamma}$ is

$$\begin{aligned} C_{I-SM-PUAP\_35} &= N^2\left(L^2+3L+2\right)+N\left(L+4\right)+ \\ & L\left(L^3+10L^2+23L+20\right)/6 \end{aligned} \tag{6}$$

The average numerical complexity of I-SM-PUAP algorithm is

$$C_{I-SM-PUAP}=N(L+1)+\beta_1 C_{I-SM-PUAP\_35} \tag{7}$$

where $\beta_1$ represents the percentage of performing steps 3-5 from all I-SM-PUAP algorithm iterations.

TABLE II. THE I-SM-PUPAP ALGORITHM

| Step | Initialization $\mathbf{w}(-1)=\mathbf{x}(-1)=[0...0]^T$; $\delta$, $\bar{\gamma}$ |
|---|---|
| | For $n \ge 0$ |
| 1 | $e_0(n)=d(n)-\mathbf{x}^T(n)\mathbf{w}(n)$ |
| 2 | If $\left\|e_0(n)\right\|>\bar{\gamma}$ |
| 3 | $\mu(n)=\min\left(\frac{\left\|-e_0(n)\pm\bar{\gamma}\right\|}{\left\|\mathbf{x}(n)\right\|_2}\right)$ |
| 4 | Solve $\left(\mathbf{X}^T(n)\mathbf{C}_{I_m(n)}\mathbf{X}(n)+\delta\mathbf{I}\right)\hat{\mathbf{p}}(n)=[1,0...0]^T$ |
| 5 | $\hat{\mathbf{a}}(n)=\mathbf{C}_{I_m(n)}\mathbf{X}(n)\hat{\mathbf{p}}(n)e_0(n)$ |
| 6 | $\mathbf{w}(n+1)=\mathbf{w}(n)+\mu(n)\hat{\mathbf{a}}(n)/\left\|\hat{\mathbf{a}}(n)\right\|_2$ |
| 7 | Else |
| 8 | $\mathbf{w}(n+1)=\mathbf{w}(n)$ |
| 9 | End |
| 10 | End |

The number of multiplications of I-SM-PUPAP algorithm for one iteration of steps 3-6 having a similar $\bar{\gamma}$ is

$$C_{I-SM-PUPAP\_36} = N^2\left(L+1\right)+N\left(L^2+2L+5\right)+ L\left(L+1\right)\left(L+8\right)/6 \tag{8}$$

The average numerical complexity of I-SM-PUPAP algorithm is

$$C_{I-SM-PUPAP} = N + \beta_2 C_{I-SM-PUPAP\_36} \tag{9}$$

where $\beta_2$ represents the percentage of performing steps 3-6 for all I-SM-PUPAP algorithm iterations.

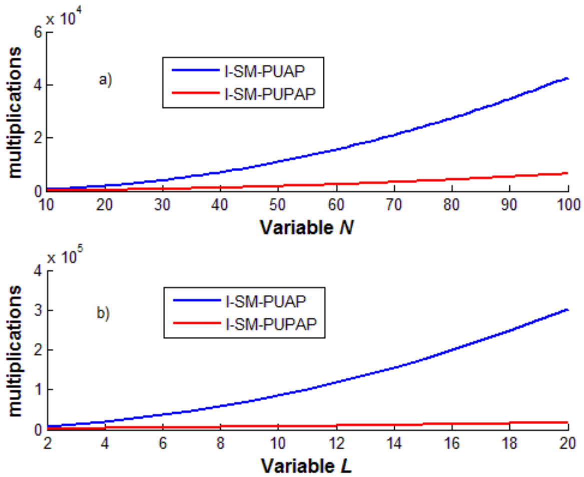


Figure 1. Numerical complexity of I-SM-PUAP and I-SM-PUPAP algorithms for two cases: a) $L = 5$ and variable $N$; b) $N = 80$ and variable $L$.

Figure 1 shows a comparison of the numerical complexity of the I-SM-PUAP and I-SM-PUPAP algorithms for two cases: $L = 5$, variable $N$ and $N = 80$, variable $L$. In both situations it was considered that $\beta_1 = \beta_2 = 10\%$ . It can be seen that I-SM-PUPAP algorithm is much less complex than I-SM-PUAP algorithm, especially for high values of $N$ or $L$. For example, if $N = 80$ and $L = 20$, the proposed I-SM-PUPAP algorithm is about 14 times less complex than I-SM-PUAP algorithm.

## III. Simulation Results

A system identification example is used in order to compare the I-SM-PUAP algorithm [12] and the proposed I-SM-PUPAP algorithm. Also the results of using NLMS with a step size of 0.1 and the lattice RLS (LRLS) [2] having a forgetting factor of 0.97 are shown. The order of the unknown system is $N$= 80 and its coefficients are random scalars with standard normal distribution [12]. The input signal is a Gaussian noise with zero mean and unitary variance, with SNR =20 dB ( $\sigma_n^2 = 0.01$ ). The threshold is $\bar{\gamma} = \sqrt{25\sigma_n^2}$ , $\delta = 10^{-12}$ , $\mathbf{w}(0) = [1,1,...,1]^T$ . The threshold bound vector $\bar{\gamma}(n)$ as $\bar{\gamma}_0(n) = \bar{\gamma}_{e0(n)} / \left|e_0(n)\right|$ is adopted and $\bar{\gamma}_i(n) = d(n-i) - \mathbf{w}^T(n)\mathbf{x}(n-i)$, for $i = 1,..,L-1$ [1], [12] and [16]. These are the same parameters used for simulations in [12] and [16] and the reasons for choosing them are explained in the previously mentioned references. The MSE curves are averaged over 50 trials.

Figure 2 shows the averaged MSE performance for the considered algorithms for $L = 5$. It can be noticed that I-SM-PUP algorithm has a slightly slower initial convergence speed than I-SM-PUAP algorithm.

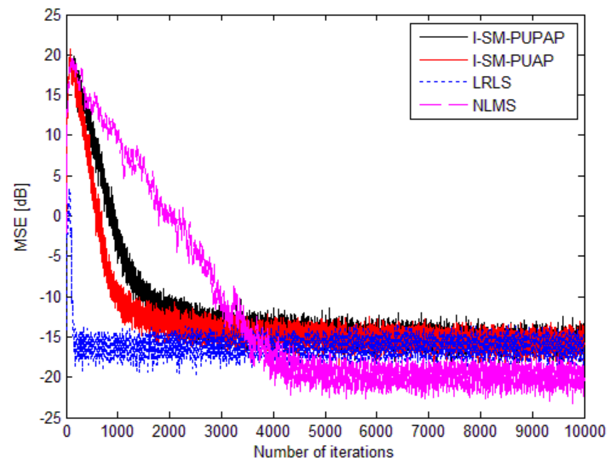


Figure 2. Averaged MSE curves of NLMS, LRLS, I-SM-PUAP and I-SM-PUPAP algorithms for a Gaussian input signal.

Figure 2 also shows that the performance of the AP variants is between that of NLMS and LRLS algorithms, which is an expected result [1]-[2]. The LRLS has the fastest convergence rate, while NLMS has the lowest convergence rate for the chosen parameters. Among the considered algorithms it is known that NLMS is the least complex while the LRLS is the most complex [1]-[2].

Figure 3 shows the averaged MSE curves for the considered algorithms for $L = 5$ having an AR(1) input signal (with a pole at 0.95).

The average number of updates for I-SM-PUAP for $L = 5$ is 6.4% while for I-SM-PUPAP is 9.8%. The performance/cost ratio defined as the average MSE on the last 1000 iterations divided by the number of multiplications per iteration [17] is 8.7802E-04 dB/multiplication for I-SM-PUAP while it is 3.72E-03 dB/multiplication for I-SM-PUPAP, i.e., the proposed algorithm is 4.2 times more efficient. This efficiency is increasing with $L$ (e.g. is 5.3 times more efficient for $L = 20$).

It can be noticed from Fig. 3 that the I-SM-PUPAP algorithm has a slower convergence speed and attains a higher MSE value than I-SM-PUAP. The average number of updates for I-SM-PUAP with $L = 5$ is 6.4% while for I-SM-PUPAP is 19%. The performance/cost ratio [17] is 7.9E-04 dB/multiplication for I-SM-PUAP while it is 1.6E-03 dB/multiplication for I-SM-PUPAP, i.e., the proposed algorithm is about two times more efficient.

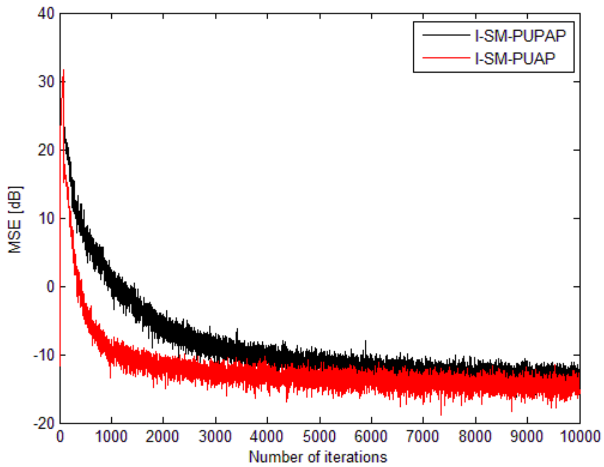


Figure 3 Averaged MSE performance of I-SM-PUAP and I-SM-PUPAP algorithms for a correlated input signal.

Typically, the number of updates of I-SM-PUPAP algorithm is higher than that of I-SM-PUAP, but it can be reduced by increasing the bound on the output estimation. The future work will be focused on finding the theoretical bound for the convergence rate of the proposed algorithm and investigate its use for other applications such as active feedback cancellation [18].

## IV. Conclusions

The I-SM-PUPAP algorithm aiming to obtain a good compromise between convergence rate and numerical complexity has been presented. Numerical simulations for system identification problem have confirmed that the I-SM-PUPAP algorithm has a lower convergence rate than recently proposed I-SM-PUAP algorithm, but also has a higher performance/cost ratio even if the number of updates is higher. In conclusion this work presents a new adaptive filtering algorithm that exploits possible resources to make the final implementation computationally efficient, while keeping the performance competitive in comparison to a related algorithm.

## Acknowledgment

The work of Felix Albu was supported by the CNCS-UEFISCDI PN-II-ID-PCE-2011-3-0097 grant of the Romanian National Authority for Scientific Research. The work of Paulo S. R. Diniz was sponsored by the CNPq and FAPERJ both Brazilian Research councils.

## References

[1] P. S. R. Diniz, “Data-selective adaptive filtering”, *Adaptive Filtering*, Springer, 2008.

[2] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Upper Saddle River, NJ, USA, 4th edition, 2002.

[3] F. Albu, J. Kadlec, C. Softley, R. Matousek,, A. Hermanek, A. Fagan, N. Coleman, "Implementation of (Normalized) RLS Lattice on VIRTEX", Field Programmable Logic and Applications, Gordon Brebner and Roger Woods Editors, 2001, pp. 91-100.

[4] K. Dogancay, *Partial-Update Adaptive Signal Processing: Design, Analysis and Implementation*, Academic Press, Elsevier, 2008.

[5] S. Werner, “Reduced complexity adaptive filtering algorithms with applications to communications systems,” D.Sc. dissertation, Helsinki University of Technology, 2002.

[6] S. Werner, M. L. R. de . Campos, and P. S. R. Diniz, “Partial-update nlms algorithms with data-selective updating,” *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 938–949, April 2004.

[7] F. Zhu, F. Gao , M. Yao, H. Zou, "Variable partial-update NLMS algorithms with data-selective updating", *Science China Information Sciences*, Vol. 57, Issue 4, pp. 1-11, April 2014.

[8] J. Wu and M. Doroslovacki, “Partial update NLMS algorithm for sparse system identification with switching between coefficient-based and input-based selection,” in *Proc. of CISS 2008,* pp. 237-240.

[9] P. S. R. Diniz, G. O. Pinto, and A. Hjorungnes, “Data selective partial-update affine projection algorithm,” in *Proc. of ICASSP* 2008, pp. 3833–3836.

[10] A. Tandon, M. N. S. Swamy, and M. O. Ahmad, “Partial-update l∞-norm based algorithms,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 2, pp. 411–419, 2007.

[11] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y. F. Huang, “Set-membership filtering and a set-membership normalized lms algorithm with an adaptive step size,” *IEEE Signal Processing Letters*, vol. 5, pp. 111–114, May 1998,.

[12] P. S. R. Diniz and H. Yazdanpanah, “Improved set-membership partial-update affine projection algorithm”, in *Proc. of ICASSP* 2016, pp. 4174-4178.

[13] S. Werner and P. S. R. Diniz, “Set-membership affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 8, no. 8, pp. 231–235, August 2001.

[14] F. Albu, and A. Fagan, “The Gauss-Seidel pseudo affine projection algorithm and its application for echo cancellation”, in *Proc. of ASILOMAR 2003*, Vol. 2, November 2003, pp. 1303 – 1306.

[15] G. H. Golub and C. F. Van Loan, *Matrix computation*, 3rd edition, The John Hopkins University Press, 1996.

[16] M. V. S. Lima and P. S. R. Diniz, “Fast learning set theoretic estimation,” in *Proc. of EUSIPCO 2013*, September 2013, pp. 1–5.

[17] F. Albu, M. Bouchard and Y. Zakharov, “Pseudo Affine Projection Algorithms for Multichannel Active Noise Control”, *IEEE Transaction on Audio, Speech and Language Processing,* Vol. 15, Issue 3, pp. 1044-1052, March 2007.

[18] M. Rotaru, F. Albu, H. Coanda, “A Variable Step Size Modified Decorrelated NLMS Algorithm for Adaptive Feedback Cancellation in Hearing Aids,” in *Proc. of ISETC 2012*, Nov. 2012, pp. 263-266.