2019 27th European Signal Processing Conference (EUSIPCO)

A Simple Sparsity-aware Feature LMS Algorithm

Gabriel S. Chaves*, Markus V. S. Lima*, Hamed Yazdanpanah†, Paulo S. R. Diniz*, Tadeu N. Ferreira‡

*Poli & COPPE
Federal University of
Rio de Janeiro (UFRJ)
P.O. Box 68504
Rio de Janeiro, RJ, Brazil
Zip 21941-972

†Department of Computer Science,
Institute of Mathematics and Statistics
University of São Paulo (USP)
Rua do Matão, 1010
São Paulo, Brazil
Zip 05508-090

‡Engineering School
Fluminense Federal
University (UFF)
Rua Passo da Pátria 56 block E room 406
Niterói, RJ, Brazil
Zip 24210-240

***Abstract*—Many real systems have inherently some type of sparsity. Recently, the feature least-mean square (F-LMS) has been proposed to exploit hidden sparsity. Unlike the existing algorithms, the F-LMS algorithm performs a linear combination of the adaptive coefficients to reveal and then exploit the hidden sparsity. However, many systems have also plain besides hidden sparsity, and the F-LMS algorithm is not able to exploit the former. In this paper, we propose a new algorithm, named simple sparsity-aware F-LMS (SSF-LMS) algorithm, that is capable of exploiting both kinds of sparsity simultaneously. The hidden sparsity is exploited just like in the F-LMS algorithm, whereas the plain sparsity is exploited by means of the discard function applied to the filter coefficients. By doing so, the proposed SSF-LMS algorithm not only outperforms the F-LMS algorithm when plain sparsity is also observed, but also requires fewer arithmetic operations. Numerical results show that the proposed algorithm has faster speed of convergence and reaches lower steady-state mean-squared error (MSE) than the F-LMS and classical algorithms, when the system has plain and hidden sparsity.**

***Index Terms*—adaptive filtering, LMS algorithm, feature matrix, discard function, sparsity**

I. Introduction

Adaptive filtering algorithms have been utilized in several applications over the last decades. In particular, the least-mean square (LMS) is one of the most popular algorithms and since its development in 1960 [1], it has been considered the benchmark in the field of adaptive learning. The LMS algorithm has been employed in many real problems, such as active noise control [2], digital equalizers [3], continuous-time filter tuning [4], system identification [5], just to mention a few.

Recently, it has been verified that many systems have some type of sparsity, be it *plain* or *hidden*. The plain sparsity occurs when the system has most coefficients with low magnitude, i.e., sparsity is directly observed in the current representation of the system. Hidden sparsity, on the other hand, is observed when some mathematical manipulation is applied to reveal the system sparsity. Unfortunately, the LMS algorithm does not take advantage of any type of sparsity. The recently proposed feature LMS (F-LMS) algorithm [6], on the other hand, benefits from the hidden sparsity by exploiting some features inherent to the unknown system.[1] Indeed, the F-LMS algorithm improves steady-state mean-squared error (MSE) and convergence speed through linear combinations (responsible for revealing hidden sparsity) of the adaptive coefficients. The problem of plain sparsity was addressed for a while, and there exist many algorithms that exploit it [8]–[15]. Many works have verified that plain sparsity can be best represented by the ℓ_0-norm [16]–[18], and that is the main idea behind the ℓ_0-norm LMS algorithm [19]. By adding a ℓ_0-norm penalty on the filter coefficients to the cost function, the sparsest solution is acquired.

Many systems have both plain and hidden sparsity. However, the F-LMS and ℓ_0-norm LMS algorithms are not able to exploit both of them simultaneously. Therefore, by imposing plain sparsity to the cost function of the F-LMS algorithm, we can improve its performance so that the new algorithm can exploit both kinds of sparsity. With that in mind, we propose to include a penalty function on the adaptive coefficients in the cost function of the F-LMS algorithm. This penalty function relies on the so-called *discard function* [20] so that the simple sparsity-aware feature LMS (SSF-LMS) algorithm requires fewer arithmetic operations, thus saving computational resources, and outperforms the F-LMS algorithm when the system has plain and hidden sparsity[2].

This work is organized as follows. Section II introduces the F-LMS algorithm and presents some examples of feature matrices. Section III describes the proposed SSF-LMS algorithm. Section IV presents the simulation results of the experiments. Finally, the conclusions are drawn in Section V.

Notation: Scalars are represented by lower-case letters. Vectors (matrices) are denoted by lowercase (uppercase) boldface letters. For a given iteration k, the weight vector and the input vector are denoted by $\mathbf{w}(k), \mathbf{x}(k) \in \mathbb{R}^{N+1}$, respectively, where N is the adaptive filter order. The optimum system coefficient

[1] While the F-LMS algorithm exploits features like lowpass or highpass spectrum through linear combinations of coefficients, there exist other features that could not be exploited in the same manner. For example, the tensor LMS algorithm is capable of exploiting impulse responses that can be decomposed as the Kronecker product of two lower-dimensional impulse responses [7].

[2] Recently, an alternative approach based on the ℓ_0 norm has been proposed in [21], but in such work the number of arithmetic operations required by the proposed algorithm is much larger in comparison with the SSF-LMS algorithm.

978-9-0827-9703-9/19/$31.00

is denoted by $\mathbf{w}_o$. The error signal at the k-th iteration is defined as $e(k) \triangleq d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$, where $d(k) \in \mathbb{R}$ is the desired signal. The l_1-norm of a vector $\mathbf{w} \in \mathbb{R}^{N+1}$ is given by $\|\mathbf{w}\|_1 = \sum_{i=0}^{N} |w_i|$.

II. The F-LMS Algorithm using ℓ_1-norm

The F-LMS algorithm proposed in [6] minimizes the following objective function

$$\xi_{\text{F-LMS}}(k) = \frac{1}{2}|e(k)|^2 + \alpha \mathcal{P}\left(\mathbf{F}(k)\mathbf{w}(k)\right), \tag{1}$$

where $\alpha \in \mathbb{R}_+$ represents the weight given to the penalty function $\mathcal{P}$, which promotes sparsity on the system, and $\mathbf{F}(k)$ is the *feature matrix* capable of exploiting the features inherent to the unknown system. This matrix is responsible for revealing its hidden sparsity, i.e., by applying $\mathbf{F}(k)$ to $\mathbf{w}(k)$ we perform a linear combination that intends to create a sparse vector. In practice, this matrix should be chosen based on some prior knowledge about the system to be identified. For example, due to the use of high sampling rates many analog systems exhibit lowpass feature. In this paper, we assume $\mathbf{F}(k)$ to be time-invariant $\mathbf{F}$ [6].

The penalty function $\mathcal{P}$ in (1) can be any almost everywhere differentiable sparsity-promoting function to allow for gradient-based methods [6], [16], [18], [20], [22], [23]. Like in [6], we choose $\mathcal{P}$ to be the ℓ_1-norm so that the complexity of the F-LMS algorithm is only slightly superior to the LMS complexity. Thus, the resulting objective function is

$$\xi_{\text{F-LMS}}(k) = \frac{1}{2}|e(k)|^2 + \alpha \|\mathbf{F}\mathbf{w}(k)\|_1, \tag{2}$$

and the general update equation is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) - \mu\alpha\mathbf{p}(k), \tag{3}$$

where $\mu \in \mathbb{R}_+$ is the step size, which should be small enough to ensure convergence [24], and $\mathbf{p}(k) \in \mathbb{R}^{N+1}$ is the gradient of function $\|\mathbf{F}\mathbf{w}(k)\|_1$ with respect to $\mathbf{w}(k)$.

The complete description of the general F-LMS algorithm is given in Algorithm 1.

Algorithm 1 The F-LMS using ℓ_1-norm

Initialization:
$\mathbf{x}(0) = \mathbf{w}(0) = [0\ 0\ \ldots\ 0]^T$
choose μ in the range $0 < \mu \ll 1$
choose α in the range $0 < \alpha < 1$
Do for $k \geq 0$
$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$
Compute $\mathbf{p}(k)$, refer to (5) and (7) for example
$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) - \mu\alpha\mathbf{p}(k)$

In the following subsections, we describe two versions of the F-LMS algorithm exploiting the lowpass and highpass features of the unknown systems.

A. The F-LMS algorithm for lowpass systems

If the system has lowpass narrowband spectrum, then the difference between adjacent coefficients of $\mathbf{w}_o$ is small. By choosing the feature matrix properly we can minimize the sum of two adjacent coefficients. In this case, we set $\mathbf{F}$ as $\mathbf{F}_l$, where $\mathbf{F}_l$ is an $N \times N+1$ matrix defined as

$$\mathbf{F}_l = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix} \tag{4}$$

and $\|\mathbf{F}_l\mathbf{w}(k)\|_1 = \sum_{i=0}^{N-1} |w_i(k) - w_{i+1}(k)|$. Therefore, the F-LMS algorithm for lowpass systems is defined by the recursion given in (3), but replacing vector $\mathbf{p}(k)$ with $\mathbf{p}_l(k)$ whose entries are given by

$$p_{l,i}(k) = \begin{cases} \text{sgn}(w_0(k) - w_1(k)), & \text{if } i = 0 \\ -\text{sgn}(w_{i-1}(k) - w_i(k)) & \\ +\text{sgn}(w_i(k) - w_{i+1}(k)), & \text{if } i = 1, \cdots, N-1 \\ -\text{sgn}(w_{N-1}(k) - w_N(k)), & \text{if } i = N. \end{cases} \tag{5}$$

where $\text{sgn}(\cdot)$ denotes the sign function.

B. The F-LMS algorithm for highpass systems

Adjacent coefficients have similar absolute values with opposite signs if the system has highpass narrowband spectrum. Then, we seek to minimize the sum of adjacent adaptive coefficients $\mathbf{w}(k)$ since the sum of two consecutive coefficients is close to zero. This can be accomplished by selecting $\mathbf{F}$ as $\mathbf{F}_h$, where $\mathbf{F}_h$ is an $N \times N+1$ feature matrix defined as

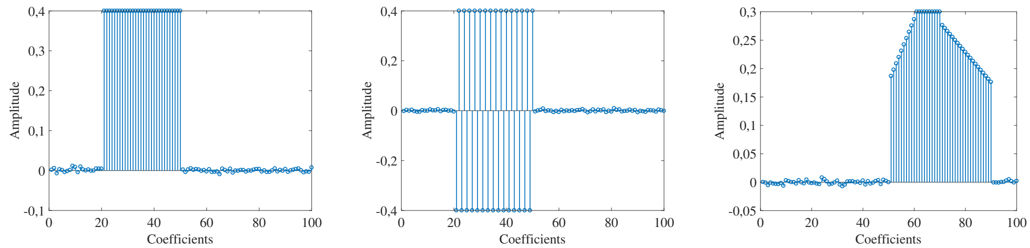
$$\mathbf{F}_h = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix}, \tag{6}$$

such that $\|\mathbf{F}_h\mathbf{w}(k)\|_1 = \sum_{i=0}^{N-1} |w_i(k) + w_{i+1}(k)|$. Like in the case of the lowpass filter, we can characterize the F-LMS algorithm for highpass systems by the recursion given in (3), but replacing $\mathbf{p}(k)$ with $\mathbf{p}_h(k)$, whose entries are

$$p_{h,i}(k) = \begin{cases} \text{sgn}(w_0(k) + w_1(k)), & \text{if } i = 0 \\ \text{sgn}(w_{i-1}(k) + w_i(k)) & \\ +\text{sgn}(w_i(k) + w_{i+1}(k)), & \text{if } i = 1, \cdots, N-1 \\ \text{sgn}(w_{N-1}(k) + w_N(k)), & \text{if } i = N. \end{cases} \tag{7}$$

III. The SSF-LMS algorithm

The F-LMS algorithms exploit the hidden sparsity in the parameters, i.e., the sparsity revealed through the application of $\mathbf{F}$ in $\mathbf{w}(k)$. However, there are many cases in which there exists plain sparsity in the parameters, i.e., $\mathbf{w}_o$ already represents a sparse vector. In this paper, we propose an algorithm that exploits both types of sparsity simultaneously.



(a) (b) (c)

Fig. 1. Impulse response of the unknown systems: (a) impulse response of the first simulation $\mathbf{w}_{o,l}$; (b) impulse response of the second simulation $\mathbf{w}_{o,h}$; (c) impulse response of the third simulation $\mathbf{w}'_{o,l}$ after 2000 iterations.

The SSF-LMS algorithm minimizes the following objective function

$$\xi(k) = \frac{1}{2}|e(k)|^2 + \alpha\|\mathbf{F}\left[\mathbf{f}_\epsilon\left(\mathbf{w}(k)\right)\right]\|_1, \tag{8}$$

where $\mathbf{f}_\epsilon(\mathbf{w}(k)) = [f_\epsilon(w_0(k))\ f_\epsilon(w_1(k)) \ldots f_\epsilon(w_N(k))]^T$ is the discard function, whose i^{th} element is defined as [20]

$$f_\epsilon(w_i(k)) = \begin{cases} w_i(k), & \text{if } |w_i(k)| \geq \epsilon \\ 0, & \text{if } |w_i(k)| < \epsilon, \end{cases} \tag{9}$$

the parameter $\epsilon \in \mathbb{R}_+$ is a threshold chosen by the user, generally close to the measurement noise [20]. In comparison to (2), the objective function in (8) generates the following update equation

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) - \mu\alpha\mathbf{p}(k)\mathbf{J}_{\mathbf{f}_\epsilon(\mathbf{w}(k))}, \tag{10}$$

where $\mathbf{J}_{\mathbf{f}_\epsilon(\mathbf{w}(k))}$ is a diagonal matrix whose diagonal elements are defined as

$$\mathbf{J}_{\mathbf{f}_\epsilon(\mathbf{w}(k))i,i} = \begin{cases} 1, & \text{if } |w_i(k)| \geq \epsilon \\ 0, & \text{if } |w_i(k)| < \epsilon. \end{cases} \tag{11}$$

Therefore, matrix $\mathbf{J}_{\mathbf{f}_\epsilon(\mathbf{w}(k))}$ selects the entries of vector $\mathbf{p}(k)$ which are relevant and, as a consequence, one can implement (10) efficiently by not computing the entries of $\mathbf{p}(k)$ related to the coefficients with small magnitude (plain sparsity) detected through (11).

The SSF-LMS algorithm is summarized in Algorithm 2. Vector $\mathbf{p}(k)$ in (10) is the same as those shown in Section II, for lowpass and highpass systems.

Algorithm 2 The SSF-LMS algorithm

Initialization:
  $\mathbf{x}(0) = \mathbf{w}(0) = [0\ 0\ \ldots\ 0]^T$
  choose μ in the range $0 < \mu \ll 1$
  choose α in the range $0 < \alpha < 1$
  choose ϵ small, close to measurement error
Do for $k \geq 0$
  $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$
  $\mathbf{p}(k) = [0\ 0\ \ldots\ 0]^T$
  Do for $i = 0$ to N
    if $|w_i(k)| > \epsilon$
      Compute $p_i(k)$
  $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) - \mu\alpha\mathbf{p}(k)$

IV. Simulations

In this section, we apply several LMS-based algorithms to identify some unknown sparse lowpass and sparse highpass systems aiming at verifying the potential benefits of exploiting both plain and hidden sparsity simultaneously. The competing algorithms are: (i) the LMS algorithm; (ii) the F-LMS algorithm, which exploits only hidden sparsity; and (iii) the ℓ_0-LMS algorithm, which exploits only plain sparsity and achieves better results than most sparsity-aware LMS-based algorithms, thus constituting a benchmark among them [18], [19].

The order of all unknown systems is 99, i.e., they have 100 coefficients, among which 30 to 40 are considered relevant (that is, their magnitudes are much greater than zero, as illustrated in Fig. 1). The adaptive filter order is $N = 99$. The input signal is a zero-mean white Gaussian noise with unit variance. The signal-to-noise ratio (SNR) is chosen as 20 dB. All algorithms are initialized with the vector $\mathbf{w}(0) = [0 \cdots 0]^T$, furthermore $\alpha = 0.05$, $\sigma_m^2 = 10^{-5}$ (to be explained in the next paragraph) and $\epsilon = 10^{-2}$. We used the Laplacian form with first order truncation for the ℓ_0-LMS, as in [19], and we set the weight given to the ℓ_0 approximation as $\kappa = 2 \times 10^{-3}$ and the parameter that controls the quality of the ℓ_0-norm approximation is $\beta = 5$ [18], [19]. The values of the step size μ are informed later for each simulation result. The MSE learning curves of the presented algorithms are computed averaging the outcomes of 500 independent trials.

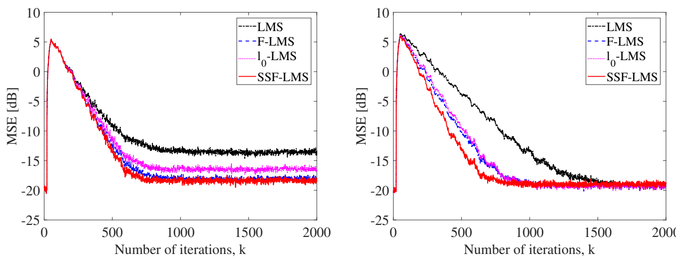
In the first experiment, we compare the performance of the aforementioned algorithms when the unknown systems do not change along the iterations. The first unknown system, $\mathbf{w}_{o,l}$, is a block sparse lowpass system such that the first 20 coefficients are zero-mean white Gaussian with variance σ_m^2, the next 30 coefficients are constant and equal to 0.4 and the last 50 coefficients are also zero-mean white Gaussian with variance σ_m^2. Fig. 1(a) depicts this system impulse response. The second unknown system, $\mathbf{w}_{o,h}$, is the same as the first one, but the non-white Gaussian coefficients with odd and even indexes are -0.4 and 0.4, respectively. We illustrate this system impulse response in Fig. 1(b).

In the second experiment, we test the tracking capability of the SSF-LMS algorithm in comparison to the others. The unknown system is the same block sparse lowpass system $\mathbf{w}_{o,l}$

used in the first experiment. However, after 2000 iterations, the system coefficients change to

$$\mathbf{w}'_{o,l}(n) = \begin{cases} \dfrac{0.1n}{9} - 0.38, & 51 \leq n < 61 \\ 0.3, & 61 \leq n < 71 \\ -\dfrac{0.1n}{19} + 0.65, & 71 \leq n < 91, \end{cases}$$

where n is the coefficient index. The other coefficients are zero-mean white Gaussian with variance σ_m^2. The other parameters are the same as in the first experiment. Fig. 1(c) depicts this impulse response.



(a) (b)

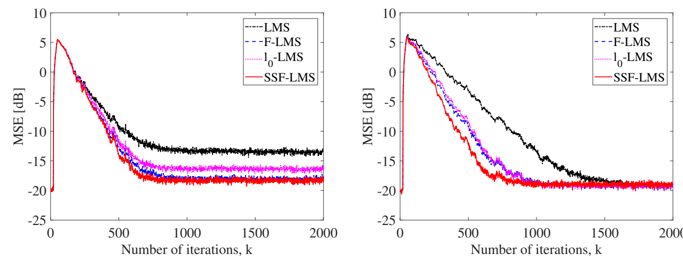
Fig. 2. MSE learning curves of the LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms considering $\mathbf{w}_{o,l}$: (a) all algorithms with the same step size: $\mu = 0.015$; (b) LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms with step sizes equal to 0.003, 0.0055, 0.005, and 0.007, respectively.

Fig. 2 illustrates the MSE learning curves of the LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms considering the block sparse lowpass system $\mathbf{w}_{o,l}$ in two different simulations. In Fig. 2(a), all algorithms use the same step size $\mu = 0.015$ so that they exhibit similar convergence speeds. We notice that the SSF-LMS algorithm achieves the lowest MSE, followed by the F-LMS, ℓ_0-LMS and LMS algorithms. Although the MSE results of the SSF-LMS algorithm are only slightly superior, in relation to the F-LMS algorithm, one must remind that the SSF-LMS algorithm performs fewer arithmetic operations due to the plain sparsity presented in $\mathbf{w}_{o,l}$. In Fig. 2(b), we compare the algorithms convergence rates fixing the steady-state MSE. Hence, we change the step sizes for the LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms to 0.003, 0.0055, 0.005, and 0.007, respectively. We observe that the SSF-LMS algorithm converges much faster than the others. It is worthy mentioning that the SSF-LMS algorithm reaches these results performing fewer arithmetic operations due to the existing plain sparsity. Therefore, the SSF-LMS algorithm outperforms the F-LMS, ℓ_0-LMS and LMS algorithms, for systems with plain and hidden sparsity.

Table I depicts the number of arithmetic operations for each algorithm during steady-state considering $\mathbf{w}_{o,l}$. One can notice that, in addition to achieving better performance, the SSF-LMS algorithm also requires fewer arithmetic operations in comparison to the F-LMS and ℓ_0-LMS algorithms. This reduction in computations occurs whenever there is plain sparsity in the unknown impulse response, since in this case the SSF-LMS algorithm does not compute every entry of vector $\mathbf{p}(k)$ (refer to Algorithm 2). Thus, in the limiting case where all coefficients are relevant (there is no plain sparsity in the system), the SSF-LMS algorithm would perform exactly the same number of operations required by the F-LMS algorithm.

TABLE I
NUMBER OF ARITHMETIC OPERATIONS PER ITERATION DURING STEADY-STATE CONSIDERING $\mathbf{w}_{o,l}$.

Algorithm	# Multiplications	# Additions
SSF-LMS	232	321
F-LMS	301	497
LMS	201	200
ℓ_0-LMS	341	340



(a) (b)

Fig. 3. MSE learning curves of the LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms considering $\mathbf{w}_{o,h}$: (a) all algorithms with the same step size: $\mu = 0.015$; (b) LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms with step sizes equal to 0.003, 0.0055, 0.005, and 0.007, respectively.

In Fig. 3, we show the results for the block sparse highpass system $\mathbf{w}_{o,h}$. Fig. 3(a) depicts the performance of the algorithms when all of them have the same step size ($\mu = 0.015$). Once again the SSF-LMS algorithm reaches the lowest MSE but the difference between the F-LMS and the SSF-LMS algorithms is not remarkable. We can observe in Fig. 3(b) the speed of convergence of each algorithm, when they have different step size values. As in Fig. 2, the SSF-LMS algorithm achieves the best convergence rate.

Since the impulse response $\mathbf{w}_{o,h}$ has the same number of relevant coefficients as $\mathbf{w}_{o,l}$, the number of arithmetic operations of each algorithm during steady-state is exactly the same as those depicted in Table I.

According to Fig. 4 we observe that the SSF-LMS algorithm reaches the steady-state first, despite the sudden variation in the unknown impulse response.

Table II presents the number of arithmetic operations during steady-state considering $\mathbf{w}'_{o,l}$. Once again the SSF-LMS algorithm requires lower amount of operations (in total) than the F-LMS and the ℓ_0-LMS algorithms, but, in this simulation, it required more additions than the ℓ_0-LMS algorithm. However, this is not a major problem as the SSF-LMS algorithm performs much fewer multiplications, which are operations that demand more computational power than additions.

V. CONCLUSIONS

In this paper, we introduced a penalty function to the cost function of the F-LMS algorithm, so that both types of sparsity

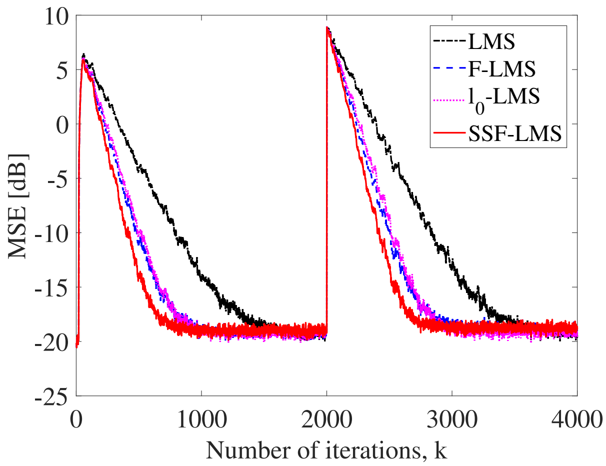


Fig. 4. MSE learning curves of the LMS, F-LMS, ℓ_0-LMS, and SSF-LMS algorithms considering that the unknown system is $\mathbf{w}_{o,l}$ in the first 2000 iterations, and suddenly changed to $\mathbf{w}'_{o,l}$. The step sizes for each algorithm are the same as those used in Fig 2(b).

TABLE II
NUMBER OF ARITHMETIC OPERATIONS PER ITERATION DURING STEADY-STATE CONSIDERING $\mathbf{w}'_{o,l}$.

Algorithm	# Multiplications	# Additions
SSF-LMS	242	361
F-LMS	301	497
LMS	201	200
ℓ_0-LMS	321	320

can be exploited. Indeed, the discard penalty function exploits the plain sparsity, whereas the feature matrix applied to the adaptive coefficients results in a sparse vector, i.e., reveals the hidden sparsity of the system. We evaluated the performance of the SSF-LMS algorithm, by elaborating some simple systems that present plain and hidden sparsity simultaneously. Finally, simulation results demonstrated that the SSF-LMS algorithm outperforms the other algorithms in terms of MSE, while also requiring fewer arithmetic operations than the F-LMS algorithm.

In future works, we will explain strategies to learn the desired feature online, thus eliminating the need of prior information about the system.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. It was also partially financed by CNPq (CNPq Universal 431381/2016-0) and FAPERJ. The authors would like to thank the São Paulo Research Foundation (FAPESP) grants #2015/22308-2, #2019/06280-1, CNPq, FAPERJ, and CAPES Finance Code 001.

REFERENCES

[1] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *IRE WESCOM Conv. Rec.*, vol. 4, pp. 96–104, 1960.

[2] M. Rupp and F. Hausberg, "LMS algorithmic variants in active noise and vibration control," in *22th European Signal Processing Conference (EUSIPCO 2014)*, Lisbon, Portugal, Sept. 2014, pp. 691–695.

[3] S. Rebhi, R. Barrak, and M. Menif, "LMS-based digital pre-equalizer for cognitive RoF system," in *18th International Conference on Transparent Optical Networks (ICTON 2016)*, Trento, Italy, July 2016, pp. 1–4.

[4] D. T. Westwick, B. Maundy, and R. Salmeh, "Approximate LMS tuning of continuous time filters: convergence and sensitivity analysis," *IEE Proceedings - Circuits, Devices and Systems*, vol. 152, no. 1, pp. 1–6, Feb. 2005.

[5] S. Ciochină, C. Paleologu, J. Benesty, S. L. Grant, and A. Anghel, "A family of optimized LMS-based algorithms for system identification," in *24th European Signal Processing Conference (EUSIPCO 2016)*, Budapest, Hungary, Sept. 2016, pp. 1803–1807.

[6] P. S. R. Diniz, H. Yazdanpanah, and M. V. S. Lima, "Feature LMS algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4144–4148.

[7] M. Rupp and S. Schwarz, "A tensor LMS algorithm," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 3347–3351.

[8] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sept. 2000.

[9] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, vol. 2, Dallas, USA, May 2002, pp. 1881–1884.

[10] Y. Chen, Y. Gu, and A. O. Hero, "Regularized least-mean-square algorithms," *arXiv*, pp. 1–9, Dec. 2010.

[11] Y. Chen, Y. Gu, and A. Hero, "Sparse LMS for system identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, Taipei, Taiwan, April 2009, pp. 3125–3128.

[12] H. Deng and M. Doroslovacki, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 181–184, Mar. 2005.

[13] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted l_1 balls," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 936–952, March 2011.

[14] R. de Lamare and R. Sampaio-Neto, "Sparsity-aware adaptive algorithms based on alternating optimization and shrinkage," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 225–229, Feb. 2014.

[15] M. Petraglia and D. Haddad, "New adaptive algorithms for identification of sparse impulse responses – analysis and comparisons," in *7th International Symposium on Wireless Communication Systems (ISWCS 2010*, York, UK, Sept. 2010, pp. 384–388.

[16] M. V. S. Lima, W. A. Martins, and P. S. R. Diniz, "Affine projection algorithms for sparse system identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013, pp. 5666–5670.

[17] M. V. S. Lima, I. Sobron, W. A. Martins, and P. S. R. Diniz, "Stability and MSE analyses of affine projection algorithms for sparse system identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014, pp. 6399–6403.

[18] M. V. S. Lima, T. N. Ferreira, W. A. Martins, and P. S. R. Diniz, "Sparsity-aware data-selective adaptive filters," *IEEE Transactions on Signal Processing*, vol. 62, no. 17, pp. 4557–4572, Sept. 2014.

[19] Y. Gu, J. Jin, and S. Mei, "l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, Sept. 2009.

[20] H. Yazdanpanah, P. S. R. Diniz, and M. V. S. Lima, "A simple set-membership affine projection algorithm for sparse system modeling," in *24th European Signal Processing Conference (EUSIPCO 2016)*, Budapest, Hungary, Sept. 2016, pp. 1798–1802.

[21] H. Yazdanpanah, J. A. Apolinario Jr., P. S. R. Diniz, and M. V. S. Lima, "l0-norm feature LMS algorithms," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2018, pp. 311–315.

[22] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[23] R. Meng, R. de Lamare, and V. Nascimento, "Sparsity-aware affine projection adaptive algorithms for system identification," in *Sensor Signal Processing for Defence (SSPD 2011)*, London, U.K., Sept. 2011, pp. 1–5.

[24] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 4th ed. New York, USA: Springer, 2013.