Yazdanpanah *et al. EURASIP Journal on Advances in Signal Processing* (2017) 2017:72
DOI 10.1186/s13634-017-0507-7

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

On the robustness of set-membership adaptive filtering algorithms

Hamed Yazdanpanah*, Markus V. S. Lima and Paulo S. R. Diniz

Abstract

In this paper, we address the robustness, in the sense of l_2-stability, of the set-membership normalized least-mean-square (SM-NLMS) and the set-membership affine projection (SM-AP) algorithms. For the SM-NLMS algorithm, we demonstrate that it is robust regardless of the choice of its parameters and that the SM-NLMS enhances the parameter estimation in most of the iterations in which an update occurs, two advantages over the classical NLMS algorithm. Moreover, we also prove that if the noise bound is known, then we can set the SM-NLMS so that it never degrades the estimate. As for the SM-AP algorithm, we demonstrate that its robustness depends on a judicious choice of one of its parameters: the constraint vector (CV). We prove the existence of CVs satisfying the robustness condition, but practical choices remain unknown. We also demonstrate that both the SM-AP and SM-NLMS algorithms do not diverge, even when their parameters are selected naively, provided the additional noise is bounded. Numerical results that corroborate our analyses are presented.

Keywords: Adaptive filtering, Robustness, Set-theoretic estimation, Set-membership filtering, SM-NLMS, SM-AP

1 Introduction

The classical adaptive filtering algorithms are iterative estimation methods based on the *point estimation theory* [1]. This theory focuses on searching for a unique solution that minimizes (or maximizes) some objective function. Two widely used classical algorithms are the normalized least-mean-square (NLMS) and the affine projection (AP) algorithms. These algorithms present a trade-off between convergence rate and steady-state misadjustment, and their properties have been extensively studied [2, 3].

On the other hand, there are few adaptive filtering algorithms employing the *set estimation theory* [4]. This is the case of the algorithms following the *set-membership filtering* (SMF) paradigm. In set estimation theory, a set Θ of feasible solutions is defined and any solution within Θ is equally acceptable. As real-world problems face many different kinds of uncertainties (due to noise, quantization, interference, and modeling errors, for example), it makes more sense to search for an acceptable solution rather than trying to find a single/unique solution, as in point estimation theory.

The SMF combines the set estimation theory with data selection strategy to introduce the set-membership (SM) adaptive filtering algorithms [5]. The data selection is responsible for reducing the computational complexity of the SM algorithms, as their filter coefficients are updated only when the estimation error is larger than a prescribed upper bound, i.e., SM algorithms evaluate the innovation on the input data before incorporating them in the learning process [2, 5–7]. Two important SM algorithms are the set-membership NLMS (SM-NLMS) and the set-membership AP (SM-AP) algorithms, proposed in [8, 9], respectively. These algorithms keep the advantages of their classical counterparts, but they are more accurate, more robust against noise, and also reduce the computational complexities due to the data selection strategy previously explained [2, 10–12]. Various applications of SM algorithms and their advantages over the classical algorithms have been discussed in the literature [13–21].

Despite the recognized advantages of the SM algorithms, they are not broadly used, probably due to the limited analysis of the properties of these algorithms. The steady-state mean-squared error (MSE) analysis of the SM-NLMS algorithm has been discussed in [22, 23]. Also, the steady-state MSE performance of the SM-AP algorithm has been analyzed in [10, 24, 25]. In addition,

*Correspondence: hamed.yazdanpanah@smt.ufrj.br
DEE–DEL/Poli & PEE/COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

the authors of the current paper have presented a few properties of the SM-NLMS algorithm in [26].

In this paper, the robustness of the SM-NLMS and the SM-AP algorithms are discussed in the sense of l_2 stability [3, 27]. Section 2 describes the robustness criterion. The robustness of the SM-NLMS algorithm is studied in Section 3, where we also discuss the cases in which the noise bound is assumed known and unknown. Section 4 presents the local and global robustness properties of the SM-AP algorithm with the details of the mathematical manipulations left to appendices A and B. Section 5 contains the simulations and numerical results. Finally, concluding remarks are drawn in Section 6.

Notation: Scalars are denoted by lower-case letters. Column vectors (matrices) are denoted by lowercase (uppercase) boldface letters. For a given iteration k, the optimum solution, the adaptive filter coefficient vector, the difference between the optimal solution and the adaptive filter coefficient vector, and the input vector are denoted by $\mathbf{w}_o$, $\mathbf{w}(k)$, $\tilde{\mathbf{w}}(k)$, $\mathbf{x}(k) \in \mathbb{R}^{N+1}$, respectively, where N stands for the filter order. The desired signal, output signal, error signal, and noise signal are denoted by $d(k)$, $y(k)$, $e(k)$, $n(k) \in \mathbb{R}$, respectively. The output signal and the error signal are defined by $y(k) \triangleq \mathbf{x}^T(k)\mathbf{w}(k) = \mathbf{w}^T(k)\mathbf{x}(k)$ and $e(k) \triangleq d(k) - y(k)$, respectively, where the superscript $(\cdot)^T$ stands for vector or matrix transposition. The l_2-norm of a vector $\mathbf{w} \in \mathbb{R}^{N+1}$ is denoted as $\|\mathbf{w}\| \triangleq \sqrt{\sum_{k=0}^{N} |w(k)|^2}$.

2 Robustness criterion

At every iteration k, assume that the desired signal $d(k)$ is related to the unknown system $\mathbf{w}_o$ by

$$d(k) \triangleq \underbrace{\mathbf{w}_o^T \mathbf{x}(k)}_{\triangleq y_o(k)} + n(k), \tag{1}$$

where $n(k)$ denotes the unknown noise and accounts for both measurement noise and modeling uncertainties or errors. Also, we assume that the unknown noise sequence $\{n(k)\}$ has finite energy [3], i.e.,

$$\sum_{k=0}^{j} |n(k)|^2 < \infty, \qquad \text{for all } j. \tag{2}$$

Suppose that we have a sequence of desired signals $\{d(k)\}$ and we intend to estimate $y_o(k) = \mathbf{w}_o^T \mathbf{x}(k)$. For this purpose, assume that $\hat{y}_{k|k}$ is an estimate of $y_o(k)$ and it is only dependent on $d(j)$ for $j = 0, \cdots, k$. For a given positive number η, we aim at calculating the following estimates $\hat{y}_{k|k} \in \{\hat{y}_{0|0}, \hat{y}_{1|1}, \cdots, \hat{y}_{N|N}\}$, such that for any $n(k)$ satisfying (2) and any $\mathbf{w}_o$, the following criterion is satisfied:

$$\frac{\sum\limits_{k=0}^{j} \|\hat{y}_{k|k} - y_o(k)\|^2}{\tilde{\mathbf{w}}^T(0)\tilde{\mathbf{w}}(0) + \sum_{k=0}^{j} |n(k)|^2} < \eta^2, \quad \text{for all } j = 0, \cdots, N \tag{3}$$

where $\tilde{\mathbf{w}}(0) \triangleq \mathbf{w}_o - \mathbf{w}(0)$ and $\mathbf{w}(0)$ is our initial guess about $\mathbf{w}_o$. Note that the numerator is a measure of estimation-error energy up to iteration j and the denominator includes the energy of disturbance up to iteration j and the energy of the error $\tilde{\mathbf{w}}(0)$ that is due to the initial guess.

So, the criterion given in (3) requires that we adjust estimates $\{\hat{y}_{k|k}\}$ such that the ratio of the estimation-error energy (numerator) to the energy of the uncertainties (denominator) does not exceed η^2. When this criterion is satisfied, we say that bounded disturbance energies induce bounded estimation-error energies, and therefore, the obtained estimates are robust. The interested reader can refer to [3], pages 719 and 720, for more details about this robustness criterion.

3 Robustness of the SM-NLMS algorithm

In this section, we discuss the robustness of the set-membership NLMS (SM-NLMS) algorithm. In subsections 3.1 and 3.2, we briefly introduce the algorithm and present some robustness properties, respectively. We address the robustness of the SM-NLMS algorithm for the cases of unknown noise bound and known noise bound in subsections 3.3 and 3.4, respectively. Then, in subsection 3.5, we introduce a time-varying error bound aiming at achieving simultaneously fast convergence, low computational burden, and efficient use of the input data.

3.1 The SM-NLMS algorithm

The SM-NLMS algorithm is given by the following recursion [2]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu(k)}{\|\mathbf{x}(k)\|^2 + \delta} e(k)\mathbf{x}(k), \tag{4}$$

where

$$\mu(k) \triangleq \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

and $\bar{\gamma} \in \mathbb{R}_+$ is the upper bound for the magnitude of the error signal that is acceptable and it is usually chosen as a multiple of the noise standard deviation σ_n [2, 10]. The parameter $\delta \in \mathbb{R}_+$ is a regularization factor, generally adopted as a small constant, used to avoid divisions by 0.

3.2 Robustness of the SM-NLMS algorithm

Let us consider the problem of identifying the unknown system $\mathbf{w}_o \in \mathbb{R}^{N+1}$, such that

$$d(k) = \mathbf{w}_o^T \mathbf{x}(k) + n(k), \tag{6}$$

where $d(k), n(k) \in \mathbb{R}$ denote the desired (reference) signal and the additive measurement noise, respectively.

The following relation can be derived from (4):

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) - \frac{\bar{\mu}(k)}{\alpha(k)} e(k)\mathbf{x}(k) f(e(k), \bar{\gamma}), \tag{7}$$

where $\tilde{\mathbf{w}}(k) \triangleq \mathbf{w}_o - \mathbf{w}(k)$ represents the discrepancy between $\mathbf{w}(k)$ and the quantity we aim to estimate $\mathbf{w}_o$, and $\bar{\mu}(k)$, $\alpha(k)$, and the indicator function f are defined as

$$\bar{\mu}(k) \triangleq 1 - \frac{\bar{\gamma}}{|e(k)|}, \tag{8}$$

$$\alpha(k) \triangleq \|\mathbf{x}(k)\|^2 + \delta, \tag{9}$$

$$f(e(k), \bar{\gamma}) \triangleq \begin{cases} 1 & \text{if } |e(k)| > \bar{\gamma}, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

In addition, observe that the error signal can be written as

$$\begin{aligned} e(k) &= d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \\ &= \mathbf{w}_o^T \mathbf{x}(k) + n(k) - \mathbf{w}^T(k)\mathbf{x}(k) \\ &= \underbrace{\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)}_{\triangleq \tilde{e}(k)} + n(k), \end{aligned} \tag{11}$$

where $\tilde{e}(k)$ represents the *noiseless error*, i.e., the error due to a mismatch between $\mathbf{w}(k)$ and $\mathbf{w}_o$.

By computing the energy of (7) and using (11), the robustness property given in Theorem 1 can be derived after some mathematical manipulations (refer to [26] for the proof).

Theorem 1 (Local Robustness of SM-NLMS) *For the SM-NLMS algorithm, it always holds that*

$$\|\tilde{\mathbf{w}}(k+1)\|^2 = \|\tilde{\mathbf{w}}(k)\|^2, \text{ if } f(e(k), \bar{\gamma}) = 0 \tag{12}$$

or

$$\begin{aligned} \|\tilde{\mathbf{w}}(k+1)\|^2 + \frac{\bar{\mu}(k)}{\alpha(k)} \tilde{e}^2(k) \\ < \|\tilde{\mathbf{w}}(k)\|^2 + \frac{\bar{\mu}(k)}{\alpha(k)} n^2(k), \end{aligned} \tag{13}$$

if $f(e(k), \bar{\gamma}) = 1$.

Theorem 1 presents local bounds for the energy of the coefficient deviation when running from an iteration to the next one. Indeed, (12) states that the coefficient deviation does not change when no coefficient update is actually implemented, whereas (13) provides a bound for $\|\tilde{\mathbf{w}}(k+1)\|^2$ based on $\|\tilde{\mathbf{w}}(k)\|^2$, $\tilde{e}^2(k)$, and $n^2(k)$, when an update occurs. Using Theorem 1, Corollary 1 can be easily demonstrated (refer to [26] for the proof).

Corollary 1 (Global Robustness of SM-NLMS) *For the SM-NLMS algorithm running from iteration 0 (initialization) to a given iteration K, the following relation holds*

$$\frac{\|\tilde{\mathbf{w}}(K)\|^2 + \sum\limits_{k \in \mathcal{K}_{up}} \frac{\bar{\mu}(k)}{\alpha(k)} \tilde{e}^2(k)}{\|\tilde{\mathbf{w}}(0)\|^2 + \sum\limits_{k \in \mathcal{K}_{up}} \frac{\bar{\mu}(k)}{\alpha(k)} n^2(k)} < 1, \tag{14}$$

where $\mathcal{K}_{up} \neq \emptyset$ is the set containing the iteration indexes k in which $\mathbf{w}(k)$ is indeed updated. If $\mathcal{K}_{up} = \emptyset$, then $\|\tilde{\mathbf{w}}(K)\|^2 = \|\tilde{\mathbf{w}}(0)\|^2$ due to (12), but this case is not of practical interest since $\mathcal{K}_{up} = \emptyset$ means that no update is performed at all.

Corollary 1 states that, for the SM-NLMS algorithm, l_2-stability from its uncertainties $\{\tilde{\mathbf{w}}(0), \{n(k)\}_{0 \leq k \leq K}\}$ to its errors $\{\tilde{\mathbf{w}}(K), \{\tilde{e}(k)\}_{0 \leq k \leq K}\}$ is invariably guaranteed. Unlike the NLMS algorithm, in which the step-size parameter has to be selected properly to guarantee such l_2-stability, for the SM-NLMS algorithm it is taken for granted (i.e., no restriction is imposed on $\bar{\gamma}$).

3.3 Convergence of $\{\|\tilde{\mathbf{w}}(k)\|^2\}$ with unknown noise bound

The robustness results mentioned in subsection 3.2 provide bounds for the evolution of $\{\|\tilde{\mathbf{w}}(k)\|^2\}$ in terms of other variables. However, we have experimentally observed that the SM-NLMS algorithm presents a well-behaved convergence of the sequence $\{\|\tilde{\mathbf{w}}(k)\|^2\}$, i.e., for most iterations we have $\|\tilde{\mathbf{w}}(k+1)\|^2 \leq \|\tilde{\mathbf{w}}(k)\|^2$. Therefore, in this subsection, we investigate under which conditions the sequence $\{\|\tilde{\mathbf{w}}(k)\|^2\}$ is (and is not) decreasing.

Corollary 2 *When an update occurs (i.e., $f(e(k), \bar{\gamma}) = 1$), $\tilde{e}^2(k) \geq n^2(k)$ implies $\|\tilde{\mathbf{w}}(k+1)\|^2 < \|\tilde{\mathbf{w}}(k)\|^2$.*

Proof By rearranging the terms in (13) we obtain

$$\|\tilde{\mathbf{w}}(k+1)\|^2 + \frac{\bar{\mu}(k)}{\alpha(k)} \left(\tilde{e}^2(k) - n^2(k)\right) < \|\tilde{\mathbf{w}}(k)\|^2, \tag{15}$$

which is valid for $f(e(k), \bar{\gamma}) = 1$. Observe that $\frac{\bar{\mu}(k)}{\alpha(k)} > 0$ since $\alpha(k) \in \mathbb{R}_+$ and $\bar{\mu}(k) \in (0,1)$ when $f(e(k), \bar{\gamma}) = 1$. Thus $\frac{\bar{\mu}(k)}{\alpha(k)} \left(\tilde{e}^2(k) - n^2(k)\right) \geq 0$ when $f(e(k), \bar{\gamma}) = 1$ and $\tilde{e}^2(k) \geq n^2(k)$. Therefore, when an update occurs, $\tilde{e}^2(k) \geq n^2(k) \Rightarrow \|\tilde{\mathbf{w}}(k+1)\|^2 < \|\tilde{\mathbf{w}}(k)\|^2$. □

In words, Corollary 2 states that the SM-NLMS algorithm improves its estimate $\mathbf{w}(k+1)$ every time an update is required and the energy of the error signal $e^2(k)$ is dominated by $\tilde{e}^2(k)$, the component of the error which is due to the mismatch between $\mathbf{w}(k)$ and $\mathbf{w}_o$.

Corollary 2 also explains why the SM-NLMS algorithm usually presents a *monotonic decreasing sequence* $\{\|\tilde{\mathbf{w}}(k)\|^2\}$ during its transient period. Indeed, in the early iterations, the absolute value of the error is generally large, thus $|e(k)| > \bar{\gamma}$ and $\tilde{e}^2(k) > n^2(k)$, implying that $\|\tilde{\mathbf{w}}(k+1)\|^2 < \|\tilde{\mathbf{w}}(k)\|^2$. In addition, there are a few iterations during the transient period in which the input data do not bring enough innovation so that no update is performed, which means that $\|\tilde{\mathbf{w}}(k+1)\|^2 = \|\tilde{\mathbf{w}}(k)\|^2$ for these few iterations. As a conclusion, it is very likely to have $\|\tilde{\mathbf{w}}(k+1)\|^2 \leq \|\tilde{\mathbf{w}}(k)\|^2$ for all iterations k belonging to the transient period.

After the transient period, however, the SM-NLMS algorithm may yield $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$ in a few iterations. Although it is hard to compute how often such an event occurs, we can provide an upper bound for the probability of this event as follows:

$$\begin{aligned}&\mathbb{P}\left[\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2\right]\\ &\leq \mathbb{P}[\,\{|e(k)| > \bar{\gamma}\} \cap \{\tilde{e}^2(k) < n^2(k)\}]\\ &< \mathbb{P}[\,|e(k)| > \bar{\gamma}] = \operatorname{erfc}\left(\sqrt{\frac{\tau}{2}}\right),\end{aligned} \tag{16}$$

where $\mathbb{P}[\cdot]$ and $\operatorname{erfc}(\cdot)$ are the probability operator and the complementary error function [28], respectively. The first inequality follows from the fact that we do not know exactly what will happen with $\|\tilde{\mathbf{w}}(k+1)\|^2$ when an update occurs and $\tilde{e}^2(k) < n^2(k)$ at the same time[1], and therefore, it corresponds to a *pessimistic bound*. The second inequality is trivial and the subsequent equality follows from [29] by parameterizing $\bar{\gamma}$ as $\bar{\gamma} = \sqrt{\tau\sigma_n^2}$, where $\tau \in \mathbb{R}_+$ (typically $\tau = 5$) and by modeling the error $e(k)$ as a zero-mean Gaussian random variable with variance σ_n^2.

From (16), one can observe that the probability of obtaining $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$ is small. For instance, for $2 \leq \tau \leq 9$ we have $0.0027 \leq \operatorname{erfc}\left(\sqrt{\frac{\tau}{2}}\right) \leq 0.1579$, and for the usual choice $\tau = 5$, we have $\operatorname{erfc}\left(\sqrt{\frac{\tau}{2}}\right) = 0.0253$.

The results in this subsection show that $\|\tilde{\mathbf{w}}(k+1)\|^2 \leq \|\tilde{\mathbf{w}}(k)\|^2$ for most iterations of the SM-NLMS algorithm, meaning that the SM-NLMS algorithm uses the input data efficiently. Indeed, having $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$ means that the input data was used to obtain an estimate $\mathbf{w}(k+1)$ which is further away from the quantity we aim to estimate $\mathbf{w}_o$, which is a waste of computational resources (it would be better not to update at all). Here, we showed that this rarely happens for the SM-NLMS algorithm, a property not shared by the classical algorithms, as it will be verified experimentally in Section 5.

3.4 Convergence of $\{\|\tilde{\mathbf{w}}(k)\|^2\}$ with known noise bound

In this subsection, we demonstrate that if the noise bound is known, then it is possible to set the threshold parameter $\bar{\gamma}$ of the SM-NLMS algorithm so that $\{\|\tilde{\mathbf{w}}(k)\|^2\}$ is a monotonic decreasing sequence. Theorem 2 and Corollary 3 address this issue.

Theorem 2 (Strong Local Robustness of SM-NLMS) *Assume the noise is bounded by a known constant $B \in \mathbb{R}_+$, i.e., $|n(k)| \leq B, \forall k$. If one chooses $\bar{\gamma} \geq 2B$, then $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ is a monotonic decreasing sequence, i.e., $\|\tilde{\mathbf{w}}(k+1)\|^2 \leq \|\tilde{\mathbf{w}}(k)\|^2, \forall k$.*

Proof If $f(e(k),\bar{\gamma}) = 1$, then $|e(k)| = |\tilde{e}(k) + n(k)| > \bar{\gamma}$, which means that: (i) $\tilde{e}(k) > \bar{\gamma} - n(k)$ for the positive values of $\tilde{e}(k)$ or (ii) $\tilde{e}(k) < -\bar{\gamma} - n(k)$ for the negative values of $\tilde{e}(k)$. Recalling that $n(k) \in [-B, B]$ and $\bar{\gamma} \in [2B, \infty)$, now we can find the bound for $\tilde{e}(k)$ by finding the minimum of (i) and the maximum of (ii) as follows:
(i) $\tilde{e}(k) > \bar{\gamma} - n(k) \Rightarrow \tilde{e}_{\min} > \bar{\gamma} - B \geq B$;
(ii) $\tilde{e}(k) < -\bar{\gamma} - n(k) \Rightarrow \tilde{e}_{\max} < -\bar{\gamma} + B \leq -B$.
Results (i) and (ii) above state that if $\bar{\gamma} \geq 2B$, then $|\tilde{e}(k)| > B$, which means that $|\tilde{e}(k)| > |n(k)|, \forall k$. Consequently, by using Corollary 2 it follows that $\|\tilde{\mathbf{w}}(k+1)\|^2 < \|\tilde{\mathbf{w}}(k)\|^2, \forall k$ in which $f(e(k),\bar{\gamma}) = 1$. In addition, if $f(e(k),\bar{\gamma}) = 0$ we have $\|\tilde{\mathbf{w}}(k+1)\|^2 = \|\tilde{\mathbf{w}}(k)\|^2$. Therefore, we can conclude that $\bar{\gamma} \geq 2B \Rightarrow \|\tilde{\mathbf{w}}(k+1)\|^2 \leq \|\tilde{\mathbf{w}}(k)\|^2, \forall k$. □

Corollary 3 (Strong Global Robustness of SM-NLMS) *Consider the SM-NLMS algorithm running from iteration 0 (initialization) to a given iteration K. If $\bar{\gamma} \geq 2B$, then $\|\tilde{\mathbf{w}}(K)\|^2 \leq \|\tilde{\mathbf{w}}(0)\|^2$, in which the equality holds only when no update is performed along all the iterations.*

The proof of Corollary 3 is omitted because it is a straightforward consequence of Theorem 2.

3.5 Time-varying $\bar{\gamma}(k)$

After reading subsections 3.3 and 3.4, one might be tempted to set $\bar{\gamma}$ as a high value since it reduces the number of updates, thus saving computational resources and also leading to a well-behaved sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ that has high probability of being monotonously decreasing. However, a high value of $\bar{\gamma}$ leads to slow convergence, because the updates during the learning stage (transient period) are less frequent and the step-size $\mu(k)$ is reduced as well. Hence, $\bar{\gamma}$ represents a compromise between convergence speed and efficiency and therefore should be chosen carefully according to the specific characteristics of the application.

An alternative approach is to allow a time-varying error bound $\bar{\gamma}(k)$ generally defined as $\bar{\gamma}(k) \triangleq \sqrt{\tau(k)\sigma_n^2}$, where

$$\tau(k) \triangleq \begin{cases} \text{Low value (e.g., } \tau(k) \in [1,5]) \\ \qquad\qquad \text{if } k \in \text{transient period,} \\ \text{High value (e.g., } \tau(k) \in [5,9]) \\ \qquad\qquad \text{if } k \in \text{steady-state.} \end{cases} \tag{17}$$

By using such a $\bar{\gamma}(k)$, one obtains the best features of the high and low values of $\bar{\gamma}$ discussed in the first paragraph of this subsection. In addition, if the noise bound B is known, then one should set $\bar{\gamma}(k) \geq 2B$ for all k during the steady-state, as explained in subsection 3.4. It is worth mentioning that (17) provides a general expression for $\tau(k)$ that allows it to vary smoothly along the iterations even within a single period (i.e., transient period or steady-state).

In order to apply the $\bar{\gamma}(k)$ defined above, the algorithm should be able to monitor the environment to determine when there is a transition between transient and steady-state periods. An intuitive way to do this is to monitor the values of $|e(k)|$. In this case, one should form a window with the $E \in \mathbb{N}$ most recent values of the error, compute the average of these $|e(k)|$ within the window, and compare it against a threshold parameter to make the decision. An even more intuitive and efficient way to monitor the iterations relies on how often the algorithm is updating. In this case, one should form a window of length E containing Boolean variables (flags, i.e., 1-bit information) indicating the iterations in which an update was performed considering the E most recent iterations. Clearly, if many updates were performed within the window, then the algorithm must be in the transient period; otherwise, the algorithm is likely to be in steady-state.

4 Robustness of the SM-AP algorithm

In this section, we address the robustness of the set-membership affine projection (SM-AP) algorithm. First, we introduce the SM-AP algorithm in subsection 4.1 and then we study its robustness properties in subsection 4.2. In subsection 4.3, we demonstrate that the SM-AP algorithm does not diverge.

4.1 The SM-AP algorithm

It is widely known that data-reusing algorithms can increase convergence speed significantly for correlated-input signals [2, 30, 31]. For this purpose, let us define the input matrix $\mathbf{X}(k)$, the error vector $\mathbf{e}(k)$, the desired vector $\mathbf{d}(k)$, the additive noise vector $\mathbf{n}(k)$, and the constraint vector (CV) $\boldsymbol{\gamma}(k)$ as follows:

$$\begin{aligned}\mathbf{X}(k) &= [\,\mathbf{x}(k)\;\mathbf{x}(k-1)\;\cdots\;\mathbf{x}(k-L)] \in \mathbb{R}^{(N+1)\times(L+1)},\\ \mathbf{e}(k) &= [\,e(k)\;\epsilon(k-1)\;...\;\epsilon(k-L)]^T \in \mathbb{R}^{L+1},\\ \mathbf{d}(k) &= [\,d(k)\;d(k-1)\;\cdots\;d(k-L)]^T \in \mathbb{R}^{L+1},\\ \mathbf{n}(k) &= [\,n(k)\;n(k-1)\;...\;n(k-L)]^T \in \mathbb{R}^{L+1},\\ \boldsymbol{\gamma}(k) &= [\,\gamma_0(k)\;\gamma_1(k)\;\cdots\;\gamma_L(k)]^T \in \mathbb{R}^{L+1},\end{aligned} \tag{18}$$

where N is the order of the adaptive filter and L is the data-reusing factor, i.e., L previous data are used together with the data from the current iteration k. The error signal is given by $\mathbf{e}(k) \triangleq \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k)$, and the entries of the constraint vector should satisfy $|\gamma_i(k)| \leq \bar{\gamma}$, for $i = 0, \ldots, L$, where $\bar{\gamma} \in \mathbb{R}_+$ is the upper bound for the magnitude of the error signal $e(k)$.

The SM-AP algorithm is described by the following recursion [9]:

$$\mathbf{w}(k+1) = \begin{cases}\mathbf{w}(k) + \mathbf{X}(k)\mathbf{A}(k)(\mathbf{e}(k) - \boldsymbol{\gamma}(k)) & \text{if } |e(k)| > \bar{\gamma},\\ \mathbf{w}(k) & \text{otherwise,}\end{cases} \tag{19}$$

where we assume that $\mathbf{A}(k) \triangleq \left(\mathbf{X}^T(k)\mathbf{X}(k)\right)^{-1} \in \mathbb{R}^{L+1\times L+1}$ exists, i.e., $\mathbf{X}^T(k)\mathbf{X}(k)$ is a full-rank matrix. Otherwise, we could add a regularization parameter as explained in [2].

4.2 Robustness of the SM-AP algorithm

Suppose that in a system identification problem the unknown system is denoted by $\mathbf{w}_o \in \mathbb{R}^{N+1}$ and the desired (reference) vector is given by

$$\mathbf{d}(k) \triangleq \mathbf{X}^T(k)\mathbf{w}_o + \mathbf{n}(k). \tag{20}$$

By defining the coefficient mismatch $\tilde{\mathbf{w}}(k) \triangleq \mathbf{w}_o - \mathbf{w}(k)$, the error vector can be written as

$$\begin{aligned}\mathbf{e}(k) &= \mathbf{X}^T(k)\mathbf{w}_o + \mathbf{n}(k) - \mathbf{X}^T(k)\mathbf{w}(k)\\ &= \underbrace{\mathbf{X}^T(k)\tilde{\mathbf{w}}(k)}_{\triangleq\tilde{\mathbf{e}}(k)} + \mathbf{n}(k)\,,\end{aligned} \tag{21}$$

where $\tilde{\mathbf{e}}(k)$ denotes the noiseless error vector (i.e., the error due to a nonzero $\tilde{\mathbf{w}}(k)$). By defining the indicator function $f : \mathbb{R} \times \mathbb{R}_+ \to \{0, 1\}$ as in (8) and using it in (19), the update rule of the SM-AP algorithm can be written as follows:

$$\begin{aligned}\mathbf{w}(k+1) =& \mathbf{w}(k)\\ &+ \mathbf{X}(k)\mathbf{A}(k)(\mathbf{e}(k) - \boldsymbol{\gamma}(k))f(e(k), \bar{\gamma}).\end{aligned} \tag{22}$$

After subtracting $\mathbf{w}_o$ from both sides of (22), we obtain

$$\begin{aligned}\tilde{\mathbf{w}}(k+1) =& \tilde{\mathbf{w}}(k)\\ &- \mathbf{X}(k)\mathbf{A}(k)(\mathbf{e}(k) - \boldsymbol{\gamma}(k))f(e(k), \bar{\gamma}).\end{aligned} \tag{23}$$

Notice that $\mathbf{A}(k)$ is a symmetric positive definite matrix. To simplify our notation, we will omit the index k and the arguments of function f that appear on the right-hand side (RHS) of the previous equation, then by decomposing $\mathbf{e}(k)$ as in (21) we obtain

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}} - \mathbf{X}\mathbf{A}\tilde{\mathbf{e}}f - \mathbf{X}\mathbf{A}\mathbf{n}f + \mathbf{X}\mathbf{A}\,\boldsymbol{\gamma}f, \tag{24}$$

from which Theorem 3 can be derived.

Theorem 3 (Local Robustness of SM-AP) *For the SM-AP algorithm, at every iteration we have*

$$\|\tilde{\mathbf{w}}(k+1)\|^2 = \|\tilde{\mathbf{w}}(k)\|^2, \textit{ if } f(e(k), \bar{\gamma}) = 0 \tag{25}$$

otherwise

$$\begin{cases} \frac{\|\tilde{\mathbf{w}}(k+1)\|^2+\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(k)\|^2+\mathbf{n}^T\mathbf{A}\mathbf{n}} < 1, \; \textit{if } \boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} < 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} \\ \frac{\|\tilde{\mathbf{w}}(k+1)\|^2+\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(k)\|^2+\mathbf{n}^T\mathbf{A}\mathbf{n}} = 1, \; \textit{if } \boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} = 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} \\ \frac{\|\tilde{\mathbf{w}}(k+1)\|^2+\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(k)\|^2+\mathbf{n}^T\mathbf{A}\mathbf{n}} > 1, \; \textit{if } \boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} > 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} \end{cases}, \tag{26}$$

where the iteration index k has been dropped for the sake of clarity, and we assume that $\|\tilde{\mathbf{w}}(k)\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n} \neq 0$ *just to allow us to write the theorem in a compact form.*

Proof Proof is left to Appendix A. □

The combination of the first two inequalities in (26), which corresponds to the case $\boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} \leq 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$, has an interesting interpretation. It describes that for any constraint vector $\boldsymbol{\gamma}$ satisfying this condition we have

$$\|\tilde{\mathbf{w}}(k+1)\|^2 + \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} \leq \|\tilde{\mathbf{w}}(k)\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n}, \tag{27}$$

no matter what the noise vector $\mathbf{n}(k)$ is. In this way, we can derive the global robustness property of the SM-AP algorithm.

Corollary 4 (Global Robustness of SM-AP) *Suppose that the SM-AP algorithm running from 0 (initialization) to a given iteration K employs a constraint vector* $\boldsymbol{\gamma}$ *satisfying* $\boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} \leq 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ *at every iteration in which an update occurs. Then, it always holds that*

$$\frac{\|\tilde{\mathbf{w}}(K)\|^2 + \sum\limits_{k\in\mathcal{K}_{up}} \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(0)\|^2 + \sum\limits_{k\in\mathcal{K}_{up}} \mathbf{n}^T\mathbf{A}\mathbf{n}} \leq 1, \tag{28}$$

where $\mathcal{K}_{up} \neq \emptyset$ *is the set comprised of the iteration indexes k in which* $\mathbf{w}(k)$ *is indeed updated and the equality holds when* $\boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} = 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ *for every* $k \in \mathcal{K}_{up}$*. If* $\mathcal{K}_{up} = \emptyset$*, then* $\|\tilde{\mathbf{w}}(K)\|^2 = \|\tilde{\mathbf{w}}(0)\|^2$*, a case that has no practical interest since no update is performed.*

Proof Proof is left to Appendix B. □

Observe that, unlike the SM-NLMS algorithm, the SM-AP algorithm requires the condition $\boldsymbol{\gamma}^T\mathbf{A}\,\boldsymbol{\gamma} \leq 2\,\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ to be satisfied in order to guarantee l_2-stability from its uncertainties $\{\tilde{\mathbf{w}}(0), \{n(k)\}_{0\leq k\leq K}\}$ to its errors $\{\tilde{\mathbf{w}}(K), \{\tilde{e}(k)\}_{0\leq k\leq K}\}$. The next question is: are there constraint vectors $\boldsymbol{\gamma}$ satisfying such a condition? This is a very interesting point because the LHS of the condition is always positive, whereas the RHS is not. Corollary 5 answers this question and shows an example of such a constraint vector.

Corollary 5 *Suppose the CV is chosen as* $\boldsymbol{\gamma}(k) = c\mathbf{n}(k)$ *in the SM-AP algorithm, where* $\mathbf{n}(k)$ *is the noise vector defined in* (20)*. If* $0 \leq c \leq 2$*, then the condition* $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} \leq 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ *always holds, implying that the SM-AP algorithm is globally robust by Corollary 4.*

Proof Substituting $\boldsymbol{\gamma}(k) = c\mathbf{n}(k)$ in $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} \leq 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ leads to the following condition $(c^2-2c)\mathbf{n}^T(k)\mathbf{A}(k)\mathbf{n}(k) \leq 0$, which is satisfied for $c^2-2c \leq 0 \Rightarrow 0 \leq c \leq 2$ since $\mathbf{A}(k)$ is positive definite. Hence, due to Corollary 4 the proposed $\boldsymbol{\gamma}(k)$ leads to a globally robust SM-AP algorithm. □

It is worth mentioning that the constraint vector $\boldsymbol{\gamma}(k)$ in Corollary 5 is not practical because $\mathbf{n}(k)$ is not observable. Therefore, Corollary 5 is actually related to the existence of $\boldsymbol{\gamma}(k)$ satisfying $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} < 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$.

Unlike the SM-NLMS algorithm, the l_2-stability of the SM-AP algorithm is not guaranteed. Indeed, as demonstrated in Theorem 3 and Corollary 4, a judicious choice of the CV is required for the SM-AP algorithm to be l_2-stable. *It is worth mentioning that practical choices of* $\boldsymbol{\gamma}(k)$ *satisfying the robustness condition* $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} \leq 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ *for every iteration k are not known yet!* Even widely used CVs, like the simple-choice CV [32], sometimes violate this condition as will be shown in Section 5. However, this does not mean that the SM-AP algorithm diverges. In fact, it does not diverge regardless the choice of $\boldsymbol{\gamma}(k)$, as demonstrated in the next subsection.

4.3 The SM-AP algorithm does not diverge

When the SM-AP algorithm updates (i.e., when $|e(k)| > \bar{\gamma}$), it generates $\mathbf{w}(k+1)$ as the solution to the following optimization problem [2, 9]:

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ &\text{subject to } \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \boldsymbol{\gamma}(k). \end{aligned} \tag{29}$$

The constraint essentially states that the a posteriori errors $\epsilon(k-l) \triangleq d(k-l) - \mathbf{x}^T(k-l)\mathbf{w}(k+1)$ are equal to their respective $\gamma_l(k)$, which in turn are bounded by $\bar{\gamma}$, as explained in subsection 4.1. This leads to the following derivation:

$$\begin{aligned} |\epsilon(k-l)| = |d(k-l) - \mathbf{x}^T(k-l)\mathbf{w}(k+1)| &\leq \bar{\gamma}, \\ |\mathbf{x}^T(k-l)\tilde{\mathbf{w}}(k+1) + n(k-l)| &\leq \bar{\gamma}, \end{aligned} \tag{30}$$

which should be valid for all iterations and suitable values of the involved variables. Therefore, we have

$$\begin{aligned} -\bar{\gamma} - n(k-l) &\leq \mathbf{x}^T(k-l)\tilde{\mathbf{w}}(k+1) \\ &\leq \bar{\gamma} - n(k-l). \end{aligned} \tag{31}$$

Since the noise sequence is bounded and $\bar{\gamma} < \infty$, we have

$$-\infty < \sum_{i=0}^{N} x_i(k-l)\tilde{w}_i(k+1) < \infty, \tag{32}$$

where $x_i(k-l), \tilde{w}_i(k+1) \in \mathbb{R}$ denote the ith entry of vectors $\mathbf{x}(k-l), \tilde{\mathbf{w}}(k+1) \in \mathbb{R}^{N+1}$, respectively. As a result,

$|\tilde{w}_i(k+1)|$ is also bounded implying $\|\tilde{\mathbf{w}}(k+1)\|^2 < \infty$, which means that the SM-AP algorithm does not diverge even when its CV is not properly chosen. In section 5 we verify this fact experimentally by using a *general CV*, i.e., a CV whose entries are randomly chosen but satisfying $|\gamma_i(k)| \leq \bar{\gamma}$. Such general CV leads to poor performance, in comparison to the SM-AP algorithm using adequate CVs, but the algorithm does not diverge.

The same reasoning could be applied to demonstrate that the SM-NLMS algorithm does not diverge as well. However, from Corollary 1, it is straightforward to verify that $\|\tilde{\mathbf{w}}(K)\|^2 < \infty$ for every K, as the denominator in (14) is finite.

5 Simulations

In this section, we provide simulation results for the SM-NLMS and SM-AP algorithms in order to verify their robustness properties addressed in the previous sections. These results are obtained by applying the aforementioned algorithms to a system identification problem. The unknown system $\mathbf{w}_o$ is comprised of 10 coefficients drawn from a standard Gaussian distribution. The noise $n(k)$ is a zero-mean white Gaussian noise with variance $\sigma_n^2 = 0.01$ yielding a signal-to-noise ratio (SNR) equal to 20 dB. The regularization factor and the initialization for the adaptive filtering coefficient vector are $\delta = 10^{-12}$ and $\mathbf{w}(0) = [0 \, \cdots \, 0]^T \in \mathbb{R}^{10}$, respectively. The error bound parameter is usually set as $\bar{\gamma} = \sqrt{5\sigma_n^2} = 0.2236$, unless otherwise stated.

5.1 Confirming the results for the SM-NLMS algorithm

Here, the input signal $x(k)$ is a zero-mean white Gaussian noise with variance equal to 1. Figure 1 aims at verifying Theorem 1. Thus, for the iterations k with coefficient update, let us denote the left-hand side (LHS) and the right-hand side (RHS) of (13) as $g_1(k)$ and $g_2(k)$, respectively. In addition, to simultaneously account for (12), we define $g_1(k) = \|\tilde{\mathbf{w}}(k+1)\|^2$ and $g_2(k) = \|\tilde{\mathbf{w}}(k)\|^2$ for the iterations without coefficient update. Figure 1 depicts $g_1(k)$ and $g_2(k)$ considering the system identification scenario described in the beginning of Section 5. In this figure, we can observe that $g_1(k) \leq g_2(k)$ for all k. Indeed, we verified that $g_1(k) = g_2(k)$ (i.e., curves are overlaid) only in the iterations without update, i.e., $\mathbf{w}(k+1) = \mathbf{w}(k)$. In the remaining iterations, we have $g_1(k) < g_2(k)$, corroborating Theorem 1.

Figure 2 depicts the sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ for the SM-NLMS algorithm and its classical counterpart, the NLMS algorithm. For the SM-NLMS algorithm, we consider three cases: fixed $\bar{\gamma}$ with unknown noise bound (blue solid line), fixed $\bar{\gamma}$ with known noise bound $B = 0.11$ (cyan solid line), and time-varying $\bar{\gamma}(k)$, defined as $\sqrt{5\sigma_n^2}$ during the transient period and $\sqrt{9\sigma_n^2}$ during the steady-state, with unknown noise bound (green solid line). For the results using the time-varying $\bar{\gamma}(k)$, the window length is $E = 20$ and when the number of updates in the window is less than 4, we assume the algorithm is in the steady-state period. For the NLMS algorithm, two different step-sizes are used: $\mu = 0.9$, which leads to fast convergence but high misadjustment, and $\mu = 0.05$, which leads to slow convergence but low misadjustment.

In Fig. 2, the blue curve confirms the discussion in Subsection 3.3. Indeed, we can observe that the sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ represented by this blue curve increases only 30 times along the 2500 iterations, meaning that the SM-NLMS algorithm did not improve its estimate $\mathbf{w}(k+1)$ only in 30 iterations. Thus, in this experiment we have $\mathbb{P}\left[\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2\right] = 0.012$, whose value is lower than its corresponding upper bound given by $\text{erfc}(\sqrt{2.5}) = 0.0253$, as explained in Subsection 3.3. Also,

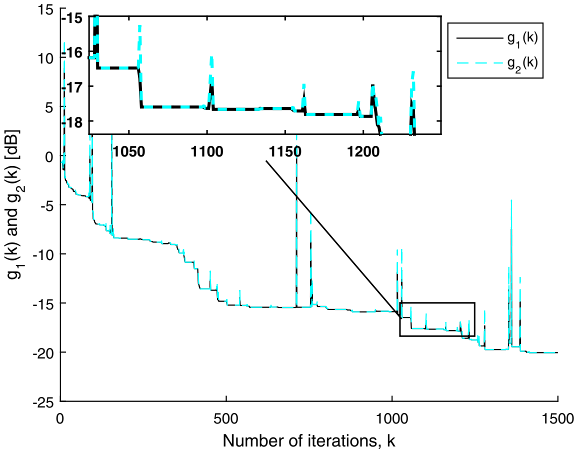

Fig. 1 Values of $g_1(k)$ and $g_2(k)$ over the iterations for the SM-NLMS algorithm corroborating Theorem 1

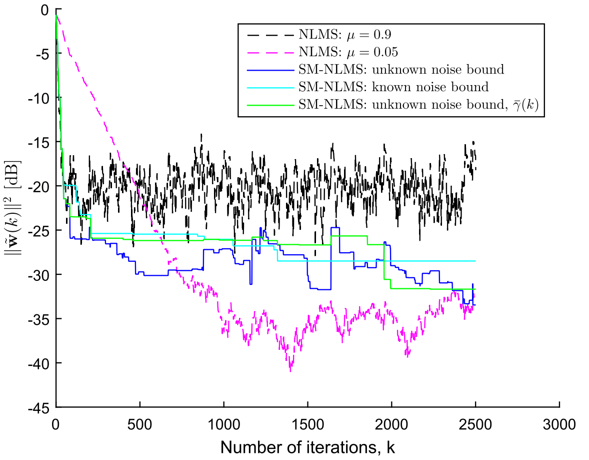

Fig. 2 $\|\tilde{\mathbf{w}}(k)\|^2 \triangleq \|\mathbf{w}_o - \mathbf{w}(k)\|^2$ for the NLMS and the SM-NLMS algorithms

we can observe that the event $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$ did not occur in the early iterations because in these iterations $\tilde{e}^2(k)$ is usually large due to a significant mismatch between $\mathbf{w}(k)$ and $\mathbf{w}_o$, i.e., the condition specified in Corollary 2 is frequently satisfied.

Also in Fig. 2, the cyan curve shows that when the noise bound is known we can obtain a monotonic decreasing sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ by selecting $\bar{\gamma} \geq 2B$, corroborating Theorem 2 and Corollary 3. The sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ represented by the green curve in Fig. 2 increases only 3 times, thus confirming the advantage of using a time-varying $\bar{\gamma}(k)$ when the noise bound is unknown, as explained in Subsection 3.5. As compared to the SM-NLMS algorithm, the behavior of the sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ for the NLMS algorithm is very irregular. Indeed, for the NLMS algorithm there are many iterations in which $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$, even when using a small step-size μ. Hence, the NLMS algorithm does not use the input data as efficiently as the SM-NLMS algorithm does, given that the NLMS performs many "useless updates".

In conclusion, an interesting advantage of the SM-NLMS algorithm over the NLMS algorithm is that the former can achieve fast convergence and has a well-behaved sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ (which rarely increases) at the same time. In addition, the SM-NLMS algorithm also saves computational resources by not updating the filter coefficients at every iteration. In Fig. 2, the update rates of the blue, cyan, and green curves are 4.6, 1.5, and 1.9%, respectively. They confirm that the computational cost of the SM-NLMS algorithm is significantly lower than that of the NLMS algorithm[2].

5.2 Confirming the results for the SM-AP algorithm

For the case of the SM-AP algorithm, the input is a first-order autoregressive signal generated as $x(k) = 0.95x(k-1) + n(k-1)$. We test the SM-AP algorithm employing $L = 2$ (i.e., reuse of two previous input data) and three different constraint vectors (CVs) $\boldsymbol{\gamma}(k)$: a general CV, the simple choice CV, and the noise vector CV. The general CV $\boldsymbol{\gamma}(k)$, in which the entries are set as $\gamma_l(k) = \bar{\gamma}$ for $0 \leq l \leq L$, illustrates a case where the CV is not properly chosen [5, 32]. The simple choice CV [5, 32] is defined as $\gamma_0(k) = \bar{\gamma}\frac{e(k)}{|e(k)|}$ and $\gamma_l(k) = \epsilon(k-l)$ for $1 \leq l \leq L$. The noise vector CV is given by $\boldsymbol{\gamma}(k) = \mathbf{n}(k)$.

The results depicted in Figs. 3, 4, 5, and 6 aim at verifying Theorem 3 and Corollary 5. We define $g_1(k)$ and $g_2(k)$ as the numerator and the denominator of (26) in Theorem 3, respectively, when an update occurs; otherwise, we define $g_1(k) = \|\tilde{\mathbf{w}}(k+1)\|^2$ and $g_2(k) = \|\tilde{\mathbf{w}}(k)\|^2$.

The results depicted in Fig. 3 illustrate that, for the general CV, there are many iterations in which $g_1(k) > g_2(k)$ (about 293 out of 1000 iterations). This is an expected behavior since the general CV does not take into account (directly or indirectly) the value of $n(k)$ and, therefore, it does not consider the robustness condition $\boldsymbol{\gamma}^T(k)\mathbf{A}(k)\boldsymbol{\gamma}(k) \leq 2\boldsymbol{\gamma}^T(k)\mathbf{A}(k)\mathbf{n}(k)$.

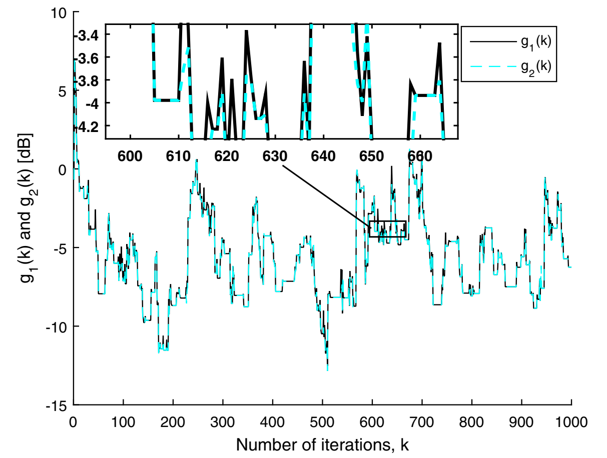

Fig. 3 Values of $g_1(k)$ and $g_2(k)$ over the iterations for the SM-AP algorithm with $\boldsymbol{\gamma}(k)$ as the general CV, where $g_1(k)$ and $g_2(k)$ are the numerator and denominator of (26) in Theorem 3, when an update occurs; otherwise, $g_1(k) = \|\tilde{\mathbf{w}}(k+1)\|^2$ and $g_2(k) = \|\tilde{\mathbf{w}}(k)\|^2$

For the SM-AP algorithm employing the simple choice CV, however, there are very few iterations in which $g_1(k) > g_2(k)$ (only 19 out of 1000 iterations), as shown in Fig. 4. This means that even the widely used simple choice CV does not lead to global robustness.

Figure 5 depicts the results for the SM-AP algorithm with $\boldsymbol{\gamma}(k) = \mathbf{n}(k)$. In this case, we can observe that $g_1(k) \leq g_2(k)$ for all k, corroborating Corollary 5. In other words, this CV guarantees the global robustness of the SM-AP algorithm.

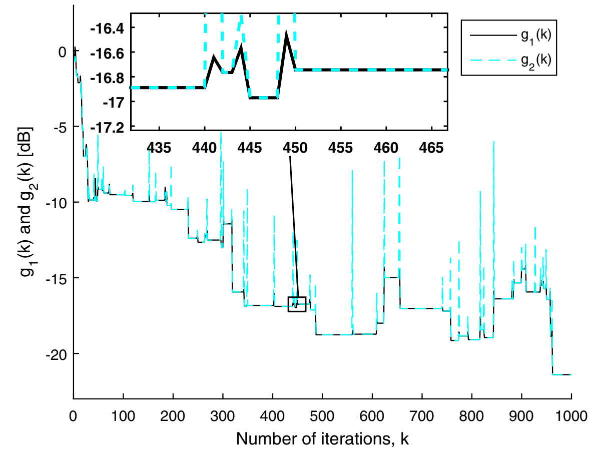

Fig. 4 Values of $g_1(k)$ and $g_2(k)$ over the iterations for the SM-AP algorithm with $\boldsymbol{\gamma}(k)$ as the SC-CV, where $g_1(k)$ and $g_2(k)$ are the numerator and denominator of (26) in Theorem 3, when an update occurs; otherwise, $g_1(k) = \|\tilde{\mathbf{w}}(k+1)\|^2$ and $g_2(k) = \|\tilde{\mathbf{w}}(k)\|^2$

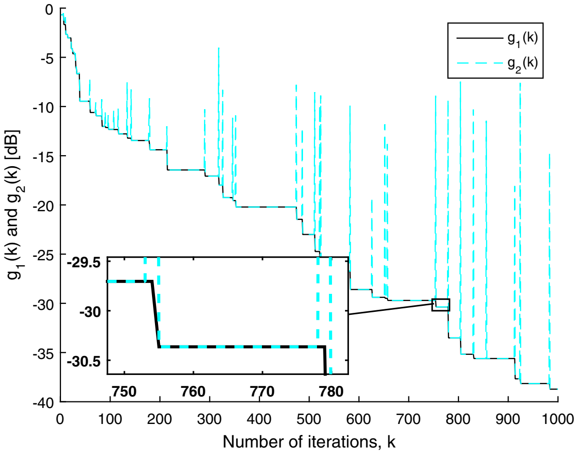

Fig. 5 Values of $g_1(k)$ and $g_2(k)$ over the iterations for the SM-AP algorithm with $\boldsymbol{\gamma}(k) = \mathbf{n}(k)$, where $g_1(k)$ and $g_2(k)$ are the numerator and denominator of (26) in Theorem 3, when an update occurs; otherwise, $g_1(k) = \|\tilde{\mathbf{w}}(k+1)\|^2$ and $g_2(k) = \|\tilde{\mathbf{w}}(k)\|^2$

Figure 6 illustrates $g_1(k)$ and $g_2(k)$ for the SM-AP algorithm with simple choice CV when the noise bound is known and 10 times smaller than $\bar{\gamma}$. In contrast with the SM-NLMS algorithm, for the SM-AP algorithm even when the noise bound is known and much smaller than $\bar{\gamma}$, we cannot guarantee that $g_1(k) \leq g_2(k)$ for all k. In Fig. 6, for example, we observe $g_1(k) > g_2(k)$ in 15 iterations.

Figure 7 depicts the sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ for the AP and the SM-AP algorithms. For the AP algorithm, the step-size μ is set as 0.9 and 0.05, whereas for the SM-AP algorithm the three previously defined CVs are tested. For the AP algorithm, we can observe an irregular behavior of $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$, i.e., this sequence increases and decreases very often. Even when a low value of μ is applied we still observe many iterations in which $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$ (425 out of 1000 iterations). The SM-AP algorithm using the general CV performs similar to the AP algorithm with high μ. But when the CV is properly chosen, like the simple choice CV for example, we observe that the number of iterations in which $\|\tilde{\mathbf{w}}(k+1)\|^2 > \|\tilde{\mathbf{w}}(k)\|^2$ is dramatically reduced (26 out of 1000 iterations), which means that the SM-AP with an adequate CV performs fewer "useless updates" than the AP algorithm. Another interesting, although not practical, choice of CV is $\boldsymbol{\gamma}(k) = \mathbf{n}(k)$, which leads to a monotonic decreasing sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$.

The MSE learning curves for the AP and the SM-AP algorithms are depicted in Fig. 8. These results were computed by averaging the squared error over 1000 trials for each curve. Observing the results of the AP algorithm, the

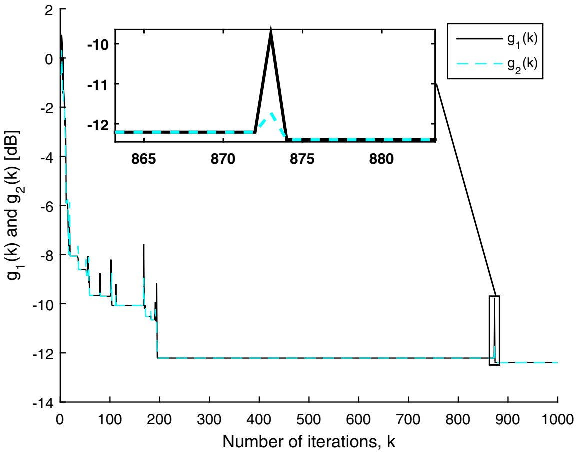

Fig. 6 Values of $g_1(k)$ and $g_2(k)$ over the iterations for the SM-AP algorithm with $\boldsymbol{\gamma}(k)$ as the SC-CV when the noise bound is known, where $g_1(k)$ and $g_2(k)$ are the numerator and denominator of (26) in Theorem 3, when an update occurs; otherwise, $g_1(k) = \|\tilde{\mathbf{w}}(k+1)\|^2$ and $g_2(k) = \|\tilde{\mathbf{w}}(k)\|^2$

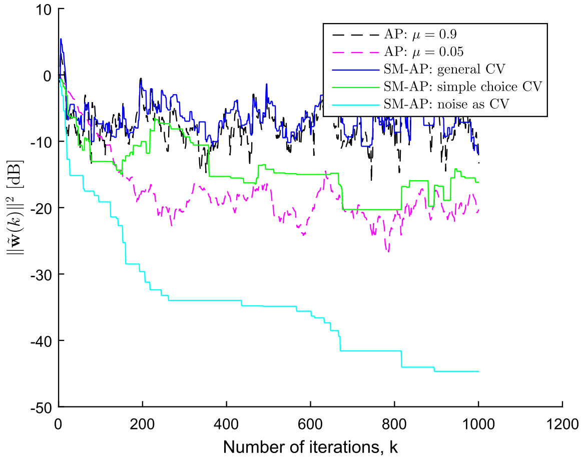

Fig. 7 $\|\tilde{\mathbf{w}}(k)\|^2 \triangleq \|\mathbf{w}(k) - \mathbf{w}_o\|^2$ for the AP and the SM-AP algorithms

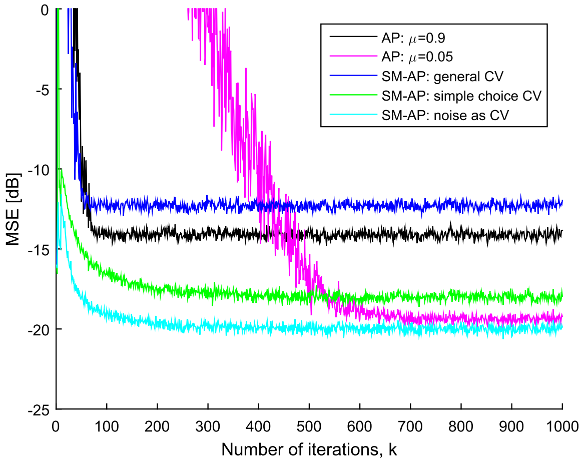

Fig. 8 Learning curves for the AP and SM-AP algorithm using different constraint vectors

trade-off between convergence rate and steady-state MSE is evident. Indeed, excluding the SM-AP with general CV (which is not an adequate choice for the CV), the AP algorithm could not achieve fast convergence and low MSE simultaneously, as the SM-AP algorithm did.

In addition, observe that $\boldsymbol{\gamma}(k) = \mathbf{n}(k)$ leads to the best results in terms of convergence rate and steady-state MSE, but the performance of the SM-AP with simple choice CV is quite close. The average number of updates required by the SM-AP algorithm using the general CV, the simple choice CV, and the noise CV are 35, 9.7, and 3.6%, respectively, implying that the last two CVs also have lower computational cost. It is worth noticing that even when using the general CV, the SM-AP algorithm still converges although it presents poor performance, as explained in subsection 4.3.

6 Conclusions

In this paper, we addressed the robustness (in the sense of l_2-stability) of the SM-NLMS and the SM-AP algorithms. In addition to the already known advantages of the SM-NLMS algorithm over the NLMS algorithm, regarding accuracy and computational cost, in this paper we demonstrated that: (i) the SM-NLMS algorithm is robust regardless the choice of its parameters and (ii) the SM-NLMS algorithm uses the input data very efficiently, i.e., it rarely produces a worse estimate $\mathbf{w}(k+1)$ during its update process. For the case where the noise bound is known, we explained how to set properly the parameter $\bar{\gamma}$ so that the SM-NLMS algorithm *never generates a worse estimate*, i.e., the sequence $\left\{\|\tilde{\mathbf{w}}(k)\|^2\right\}$ (the squared Euclidean norm of the parameters deviation) becomes monotonously decreasing. For the case where the noise bound is unknown, we designed a time-varying parameter $\bar{\gamma}(k)$ that achieves simultaneously fast convergence and efficient use of the input data.

Unlike the SM-NLMS algorithm, we demonstrated that there exists a condition to guarantee the l_2-stability of the SM-AP algorithm. This robustness condition depends on a parameter known as the constraint vector (CV) $\boldsymbol{\gamma}(k)$. We proved the existence of vectors $\boldsymbol{\gamma}(k)$ satisfying such a condition, but practical choices remain unknown. In addition, it was shown that the SM-AP with an adequate CV uses the input data more efficiently than the AP algorithm.

We also demonstrated that both the SM-AP and SM-NLMS algorithms do not diverge, even when their parameters are not properly selected, provided the noise is bounded. Finally, numerical results that corroborate our study were presented.

Endnotes

[1] This is because Corollary 2 provides a sufficient, but not necessary, condition for $\|\tilde{\mathbf{w}}(k+1)\|^2 < \|\tilde{\mathbf{w}}(k)\|^2$.

[2] In comparison to the NLMS algorithm, whenever the SM-NLMS algorithm updates it performs two additional operations: one division and one subtraction due to the computation of $\mu(k)$. However, for most of the iterations the SM-NLMS algorithm requires fewer operations because it does not update often.

Appendix A: Proof of Theorem 3

For convenience, let us start by rewriting Eq. (24):

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}} - \mathbf{X}\mathbf{A}\tilde{\mathbf{e}}f - \mathbf{X}\mathbf{A}\mathbf{n}f + \mathbf{X}\mathbf{A}\boldsymbol{\gamma}f. \qquad (33)$$

By computing the Euclidean norm of this equation and rearranging the terms, we get

$$\begin{aligned}
\|\tilde{\mathbf{w}}(k+1)\|^2& \\
=&\tilde{\mathbf{w}}^T\tilde{\mathbf{w}} - \tilde{\mathbf{w}}^T\mathbf{X}\mathbf{A}\tilde{\mathbf{e}}f - \tilde{\mathbf{w}}^T\mathbf{X}\mathbf{A}\mathbf{n}f \\
&+ \tilde{\mathbf{w}}^T\mathbf{X}\mathbf{A}\boldsymbol{\gamma}f - \tilde{\mathbf{e}}^T\mathbf{A}^T\mathbf{X}^T\tilde{\mathbf{w}}f \\
&+ \tilde{\mathbf{e}}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\tilde{\mathbf{e}}f^2 + \tilde{\mathbf{e}}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\mathbf{n}f^2 \\
&- \tilde{\mathbf{e}}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\boldsymbol{\gamma}f^2 - \mathbf{n}^T\mathbf{A}^T\mathbf{X}^T\tilde{\mathbf{w}}f \\
&+ \mathbf{n}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\tilde{\mathbf{e}}f^2 + \mathbf{n}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\mathbf{n}f^2 \\
&- \mathbf{n}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\boldsymbol{\gamma}f^2 + \boldsymbol{\gamma}^T\mathbf{A}^T\mathbf{X}^T\tilde{\mathbf{w}}f \\
&- \boldsymbol{\gamma}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\tilde{\mathbf{e}}f^2 - \boldsymbol{\gamma}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\mathbf{n}f^2 \\
&+ \boldsymbol{\gamma}^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}\boldsymbol{\gamma}f^2 \\
=&\|\tilde{\mathbf{w}}\|^2 - \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}f - \tilde{\mathbf{e}}^T\mathbf{A}\mathbf{n}f + \tilde{\mathbf{e}}^T\mathbf{A}\boldsymbol{\gamma}f \\
&- \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}f + \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}f^2 + \tilde{\mathbf{e}}^T\mathbf{A}\mathbf{n}f^2 \\
&- \tilde{\mathbf{e}}^T\mathbf{A}\boldsymbol{\gamma}f^2 - \mathbf{n}^T\mathbf{A}\tilde{\mathbf{e}}f + \mathbf{n}^T\mathbf{A}\tilde{\mathbf{e}}f^2 \\
&+ \mathbf{n}^T\mathbf{A}\mathbf{n}f^2 - \mathbf{n}^T\mathbf{A}\boldsymbol{\gamma}f^2 + \boldsymbol{\gamma}^T\mathbf{A}\tilde{\mathbf{e}}f \\
&- \boldsymbol{\gamma}^T\mathbf{A}\tilde{\mathbf{e}}f^2 - \boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}f^2 + \boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma}f^2\ , \qquad (34)
\end{aligned}$$

where it was used that $\mathbf{A}^{-1} = \mathbf{X}^T(k)\mathbf{X}(k)$ and $\tilde{\mathbf{e}}(k) = \mathbf{X}^T(k)\tilde{\mathbf{w}}(k)$. From the above equation, we observe that when $f = 0$ we have

$$\|\tilde{\mathbf{w}}(k+1)\|^2 = \|\tilde{\mathbf{w}}(k)\|^2 \qquad (35)$$

as expected, since $f = 0$ means that the algorithm does not update its coefficients. However, when $f = 1$ the following equality is achieved from (34):

$$\begin{aligned}
\|\tilde{\mathbf{w}}(k+1)\|^2 = \|\tilde{\mathbf{w}}\|^2 - \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} + \mathbf{n}^T\mathbf{A}\mathbf{n}& \\
- 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} + \boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma}.& \qquad (36)
\end{aligned}$$

After rearranging the terms of the previous equation, we obtain

$$\begin{aligned}
&\|\tilde{\mathbf{w}}(k+1)\|^2 + \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} = \\
&\|\tilde{\mathbf{w}}\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n} - 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} + \boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma}\ . \qquad (37)
\end{aligned}$$

Therefore, $\|\tilde{\mathbf{w}}(k+1)\|^2 + \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} < \|\tilde{\mathbf{w}}\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n}$ if $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} < 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$, $\|\tilde{\mathbf{w}}(k+1)\|^2 + \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} = \|\tilde{\mathbf{w}}\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n}$ if

$\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} = 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$, and $\|\tilde{\mathbf{w}}(k{+}1)\|^2{+}\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} > \|\tilde{\mathbf{w}}\|^2{+}\mathbf{n}^T\mathbf{A}\mathbf{n}$ if $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} > 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$.

Assuming $\|\tilde{\mathbf{w}}\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n} \neq 0$ we can summarize the discussion above in a compact form as follows:

$$\begin{cases} \frac{\|\tilde{\mathbf{w}}(k+1)\|^2+\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(k)\|^2+\mathbf{n}^T\mathbf{A}\mathbf{n}} < 1, \text{ if } \boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} < 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} \\ \frac{\|\tilde{\mathbf{w}}(k+1)\|^2+\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(k)\|^2+\mathbf{n}^T\mathbf{A}\mathbf{n}} = 1, \text{ if } \boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} = 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} \\ \frac{\|\tilde{\mathbf{w}}(k+1)\|^2+\tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(k)\|^2+\mathbf{n}^T\mathbf{A}\mathbf{n}} > 1, \text{ if } \boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} > 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n} \end{cases} . \tag{38}$$

Appendix B: Proof of Corollary 4

Denote by $\mathcal{K} \triangleq \{0, 1, 2, \ldots, K-1\}$ the set of all iterations. Let $\mathcal{K}_{\text{up}} \subseteq \mathcal{K}$ be the subset containing only the iterations in which an update occurs, whereas $\mathcal{K}_{\text{up}}^c \triangleq \mathcal{K} \setminus \mathcal{K}_{\text{up}}$ is comprised of the iterations in which the filter coefficients are not updated.

As a consequence of Theorem 3, when an update occurs the inequality given in (27) is valid provided $\boldsymbol{\gamma}$ is chosen such that $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} \leq 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ is respected. In this way, by summing such inequality for all $k \in \mathcal{K}_{\text{up}}$ we obtain

$$\begin{aligned} &\sum_{k\in\mathcal{K}_{\text{up}}} \left(\|\tilde{\mathbf{w}}(k+1)\|^2 + \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}\right) \\ &\leq \sum_{k\in\mathcal{K}_{\text{up}}} \left(\|\tilde{\mathbf{w}}(k)\|^2 + \mathbf{n}^T\mathbf{A}\mathbf{n}\right). \end{aligned} \tag{39}$$

Observe that $\boldsymbol{\gamma}$, $\tilde{\mathbf{e}}$, $\mathbf{n}$, and $\mathbf{A}$ all depend on the independent variable k, which we have omitted for the sake of simplification. In addition, for the iterations without coefficient update, we have (25), which can be summed for all $k \in \mathcal{K}_{\text{up}}^c$ leading to

$$\sum_{k\in\mathcal{K}_{\text{up}}^c} \|\tilde{\mathbf{w}}(k+1)\|^2 = \sum_{k\in\mathcal{K}_{\text{up}}^c} \|\tilde{\mathbf{w}}(k)\|^2. \tag{40}$$

Summing (39) and (40) yields

$$\begin{aligned} &\sum_{k\in\mathcal{K}} \|\tilde{\mathbf{w}}(k+1)\|^2 + \sum_{k\in\mathcal{K}_{\text{up}}} \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} \\ &\leq \sum_{k\in\mathcal{K}} \|\tilde{\mathbf{w}}(k)\|^2 + \sum_{k\in\mathcal{K}_{\text{up}}} \mathbf{n}^T\mathbf{A}\mathbf{n}. \end{aligned} \tag{41}$$

Then, we can cancel several of the terms $\|\tilde{\mathbf{w}}(k)\|^2$ from both sides of the above inequality simplifying it as follows

$$\begin{aligned} &\|\tilde{\mathbf{w}}(K)\|^2 + \sum_{k\in\mathcal{K}_{\text{up}}} \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}} \\ &\qquad \leq \|\tilde{\mathbf{w}}(0)\|^2 + \sum_{k\in\mathcal{K}_{\text{up}}} \mathbf{n}^T\mathbf{A}\mathbf{n}. \end{aligned} \tag{42}$$

Assuming a nonzero denominator, we can write the previous inequality in a compact form

$$\frac{\|\tilde{\mathbf{w}}(K)\|^2 + \sum\limits_{k\in\mathcal{K}_{\text{up}}} \tilde{\mathbf{e}}^T\mathbf{A}\tilde{\mathbf{e}}}{\|\tilde{\mathbf{w}}(0)\|^2 + \sum\limits_{k\in\mathcal{K}_{\text{up}}} \mathbf{n}^T\mathbf{A}\mathbf{n}} \leq 1. \tag{43}$$

This relation holds for all K, provided $\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} \leq 2\boldsymbol{\gamma}^T\mathbf{A}\mathbf{n}$ is satisfied for every iteration in which an update occurs, i.e., for every $k \in \mathcal{K}_{\text{up}}$. The only assumption used in the derivation is that $\mathcal{K}_{\text{up}} \neq \emptyset$. Otherwise, we would have $\|\tilde{\mathbf{w}}(K)\|^2 = \|\tilde{\mathbf{w}}(0)\|^2$, which would occur only if $\mathbf{w}(k)$ is never updated, which is not of practical interest.

Acknowledgements

The authors would like to thank CAPES, CNPq, and FAPERJ agencies for funding this research work.

Authors' contributions

All authors have contributed equally. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 19 April 2017 Accepted: 8 October 2017
Published online: 19 October 2017

References

1. EL Lehmann, G Casella, *Theory of Point Estimation*, 2nd edn. (Springer, New York, 2003)
2. PSR Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 4th edn. (Springer, New York, 2013)
3. AH Sayed, *Adaptive Filters*. (Wiley-IEEE, New York, 2008)
4. PL Combettes, The foundations of set theoretic estimation. Proc. IEEE. **81**(2), 182–208 (1993). doi:10.1109/5.214546
5. MVS Lima, PSR Diniz, in *21st European Signal Processing Conference (EUSIPCO 2013)*. Fast learning set theoretic estimation (IEEE, Marrakech, 2013), pp. 1–5
6. E Fogel, Y-F Huang, On the value of information in system identification–bounded noise case. Automatica. **18**(2), 229–238 (1982). doi:10.1016/0005-1098(82)90110-8
7. JR Deller, Set-membership identification in digital signal processing. IEEE ASSP Magazine. **6**(4), 4–20 (1989). doi:10.1109/53.41661
8. S Gollamudi, S Nagaraj, S Kapoor, Y-F Huang, Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size. IEEE Signal Process. Lett. **5**(5), 111–114 (1998). doi:10.1109/97.668945
9. S Werner, PSR Diniz, Set-membership affine projection algorithm. IEEE Signal Process. Lett. **8**(8), 231–235 (2001). doi:10.1109/97.935739
10. MVS Lima, PSR Diniz, Steady-state MSE performance of the set-membership affine projection algorithm. Circ. Syst. Signal Process. **32**(4), 1811–1837 (2013). DOI:10.1007/s00034-012-9545-4
11. R Arablouei, K Dogancay, in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC 2012)*. Tracking performance analysis of the set-membership NLMS adaptive filtering algorithm (IEEE, Hollywood, 2012), pp. 1–6
12. A Carini, GL Sicuranza, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*. Analysis of a multichannel filtered-x set-membership affine projection algorithm, (2006). doi:10.1109/ICASSP.2006.1660623
13. S Gollamudi, S Kapoor, S Nagaraj, Y-F Huang, Set-membership adaptive equalization and updator-shared implementation for multiple channel

communications systems. IEEE Trans. Signal Process. **46**(9), 2372–2385 (1998). doi:10.1109/78.709523
14. S Nagaraj, S Gollamudi, S Kapoor, Y-F Huang, BEACON: An adaptive set-membership filtering technique with sparse updates. IEEE Trans. Signal Process. **47**(11), 2928–2941 (1999). doi:10.1109/78.796429
15. L Guo, Y-F Huang, Frequency-domain set-membership filtering and its applications. IEEE Trans. Signal Process. **55**(4), 1326–1338 (2007). doi:10.1109/TSP.2006.888890
16. S Werner, Jr. JAA, PSR Diniz, Set-membership proportionate affine projection algorithms. EURASIP J. Audio Speech Music. Process. **2007**(1), 1–10 (2007). doi:10.1155/2007/34242
17. WA Martins, MVS Lima, PSR Diniz, in *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2008)*. Semi-blind data-selective equalizers for QAM, (Recife, Brazil, 2008), pp. 501–505. doi:10.1109/SPAWC.2008.4641658
18. H Yazdanpanah, PSR Diniz, New trinion and quaternion set-membership affine projection algorithms. IEEE Trans. Circ. Syst. II Express Briefs. **64**(2), 216–220 (2017)
19. MZA Bhotto, A Antoniou, Robust set-membership affine-projection adaptive-filtering algorithm. IEEE Trans. Signal Process. **60**(1), 73–81 (2012). doi:10.1109/TSP.2011.2170980
20. S Zhang, J Zhang, Set-membership NLMS algorithm with robust error bound. IEEE Trans. Circ. Syst. II Express Briefs. **61**(7), 536–540 (2014). doi:10.1109/TCSII.2014.2327376
21. WL Mao, Robust set-membership filtering techniques on gps sensor jamming mitigation. IEEE Sensors J. **17**(6), 1810–1818 (2017). doi:10.1109/JSEN.2016.2558192
22. MVS Lima, PSR Diniz, in *7th International Symposium on Wireless Communication Systems (ISWCS 2010)*. On the steady-state MSE performance of the set-membership NLMS algorithm, (York, 2010), pp. 389–393. doi:10.1109/ISWCS.2010.5624323
23. N Takahashi, I Yamada, Steady-state mean-square performance analysis of a relaxed set-membership NLMS algorithm by the energy conservation argument. IEEE Trans. Signal Process. **57**(9), 3361–3372 (2009). doi:10.1109/TSP.2009.2020747
24. PSR Diniz, Convergence performance of the simplified set-membership affine projection algorithm. Circ. Syst. Signal Process. **30**(2), 439–462 (2011). doi:10.1007/s00034-010-9219-z
25. MVS Lima, PSR Diniz, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*. Steady-state analysis of the set-membership affine projection algorithm, (Dallas, 2010), pp. 3802–3805. doi:10.1109/ICASSP.2010.5495836
26. H Yazdanpanah, MVS Lima, PSR Diniz, in *9th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2016)*. On the robustness of the set-membership NLMS algorithm (IEEE, Rio de Janeiro, 2016)
27. M Rupp, Pseudo affine projection algorithms revisited: Robustness and stability analysis. IEEE Trans. Signal Process. **59**(5), 2017–2023 (2011). doi:10.1109/TSP.2011.2113346
28. JG Proakis, *Digital Communications*. (McGraw-Hill, USA, 1995)
29. JF Galdino, JA Apolinario, MLR de Campos, in *International Symposium on Circuits and Systems (ISCAS 2006)*. A set-membership NLMS algorithm with time-varying error bound, (2006), pp. 277–280
30. S Haykin, *Adaptive Filter Theory*, 4th edn. (Prentice Hall, Englewood Cliffs, 2002)
31. K Ozeki, T Umeda, An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. Electron. Commun. Jpn. **67-A**(5), 19–27 (1984)
32. WA Martins, MVS Lima, PSR Diniz, TN Ferreira, Optimal constraint vectors for set-membership affine projection algorithms. Signal Process. **134**, 285–294 (2017). doi:10.1016/j.sigpro.2016.11.025