Affine Projection Adaptive Filters for the Identification of Sparse Systems

Cyro S. Hemsi

***Abstract*— In this paper, we focus on the identification of sparse systems, as is often the case in telecommunications and acoustics applications, using sparse affine projection (AP) algorithms, expected to perform better than sparse versions of the LMS and NLMS for a highly correlated input signal. Initially, we offer a concise review on the AP adaptive filter theory, followed by the analysis of sparse AP zero attractors from a geometric point-of-view. Then, we propose a sparse AP based on the SparseStep approximation of the ℓ_0-pseudo-norm. Finally, the proposed algorithm is numerically validated by comparing it with well-known sparse AP filters.**

***Keywords*— Affine Projection, Adaptive Filters, System Identification, Sparse, LMS Algorithm.**

I. Introduction

Often, in telecommunications and acoustics, the systems to be identified are known *a priori* to be sparse, i.e., few of the impulse response coefficients are large, while most of them are close to zero. In applications such as broadband channel estimation and echo cancellation, the conventional algorithms employed in systems identification are not capable of exploiting prior knowledge about the system sparsity, both to accelerate convergence and improve performance. This way, several sparse adaptive filters based on the Least Mean Square (LMS) algorithm [1], [2], [3] have been proposed in the literature [4], [5], [6], [7]. Norm-based sparse adaptive algorithms require choosing a sparsity-inducing penalty in the cost function [5]. As directly penalizing the ℓ_0-pseudo-norm, i.e., the number of non-zero elements, is an NP-hard problem, more practical solutions replace it by the ℓ_1-norm [4], ℓ_0 approximations [6], [8], [5] or heuristic approaches [4], [9].

The LMS family is often chosen in practice due to its reduced computational complexity [1], [2], [3], however, the conventional LMS filter may become unstable due to the variations in the input signal level. The Normalized LMS (NLMS) algorithm uses a normalized step-size parameter and achieves better stability and convergence speed. However, the performance of both filters is severely degraded in practical scenarios where the input signal is highly correlated (colored), leading to slow convergence and high steady-state error/misadjustment due to estimation noise. In order to overcome this, the affine projection (AP) algorithm has been developed in [10]. The AP algorithm belongs to the so-called data-reusing family, where, at each time instant, past data are reused. In this paper we focus on sparse AP algorithms, as a continuation of our work [11]. This paper proposes a sparse AP adaptive filter whose cost function is penalized by the SparseStep approximation of the ℓ_0-pseudo-norm [12], which is a simple but precise continuous function.

The contributions of this paper are twofold: (a) We offer a brief discussion on the geometric interpretation of sparse AP algorithms; (b) Furthermore, we extend the sparse adaptive filter SS-LMS proposed in [11] to the AP context. Finally, the performance of the so-called SS-AP is numerically validated by comparison with other sparse AP adaptive filters.

This paper is organized as follows. Section 2 provides background on the AP algorithm as an extension of gradient-descent adaptive filters; in Section 3, the proposed SS-AP update equation is derived. Section 4 provides numerical simulation and results, followed by Section 5, where final conclusions are drawn.

II. Problem Formulation

Let us consider an M-tap transversal finite-impulse response adaptive filter and real-valued signals. We denote vectors/matrices by boldface letters. Given the $M \times 1$ input vector $\boldsymbol{u}(n)$ and the tap-weight vector $\boldsymbol{w}(n)$, at any sample n, the filter output is given by $y(n) = \boldsymbol{u}^T(n)\boldsymbol{w}(n)$. In a system identification problem, adaptive filters are used to estimate the unknown coefficients of the system's impulse response. At each iteration, the adaptive filter output $\hat{y}(n)$, obtained with estimated coefficients $\hat{\boldsymbol{w}}(n)$, is subtracted from a (noisy) observation $d(n)$ of the reference (desired) signal, resulting in the estimation error $\hat{e}(n) \triangleq d(n) - \hat{y}(n)$. The estimated $\hat{\boldsymbol{w}}(n)$ are iteratively adjusted until the error $\hat{e}(n)$ has been minimized in the mean-square sense.

The LMS update equation is given by:

$$\hat{\boldsymbol{w}}(n+1) = \hat{\boldsymbol{w}}(n) + \mu \boldsymbol{u}(n)\hat{e}(n), \tag{1}$$

where μ is a positive step-size. The update vector $\Delta\hat{\boldsymbol{w}}(n)$ from the current $\hat{\boldsymbol{w}}(n)$ to $\hat{\boldsymbol{w}}(n+1)$ is:

$$\Delta\hat{\boldsymbol{w}}(n) = \mu[d(n) - \boldsymbol{u}^T(n)\hat{\boldsymbol{w}}(n)]\boldsymbol{u}(n), \tag{2}$$

being an affine function of $\hat{\boldsymbol{w}}(n)$. The filter will converge provided $\boldsymbol{u}(n)$ and $d(n)$ are joint stationary and μ is chosen adequately. However, in eq. (2), $\Delta\hat{\boldsymbol{w}}(n)$ is directly proportional to the time-varying level of the input signal $\boldsymbol{u}(n)$, so when the input signal is large, the amount of update is large and the algorithm becomes unstable, so it is difficult to adjust the value of μ.

Cyro S. Hemsi received his Ph.D. (2021) from the Department of Telecommunication and Control (PTC) of the University of São Paulo (USP), São Paulo-Brazil. E-mail: cyro.hemsi@alumni.usp.br.

A. *Normalized LMS*

The NLMS algorithm normalizes the vector $\Delta\hat{\boldsymbol{w}}(n)$ by the squared Euclidean norm of the input vector $\boldsymbol{u}(n)$, leading to a more stable algorithm with faster convergence. The NLMS update equation can be derived from the following constrained optimization problem [1], [2], [3]:

$$\begin{aligned} \underset{\hat{\boldsymbol{w}}(n+1)\in\mathbb{R}^{M\times 1}}{\text{minimize}} \quad & \|\hat{\boldsymbol{w}}(n+1)-\hat{\boldsymbol{w}}(n)\|^2 \\ \text{subject to} \quad & d(n)-\boldsymbol{u}^T(n)\hat{\boldsymbol{w}}(n+1)=0 \end{aligned} \tag{3}$$

where $\|\cdot\|^2$ is the squared Euclidean norm and assuming measurement noise is negligible. The difference between the updated coefficient vector $\hat{\boldsymbol{w}}(n+1)$ and the current one is minimized (minimal disturbance principle), while forcing to zero the *a posteriori* error (i.e., computed using $\hat{\boldsymbol{w}}(n+1)$). The resulting NLMS update equation is given by:

$$\hat{\boldsymbol{w}}(n+1)=\hat{\boldsymbol{w}}(n)+\frac{\mu_N}{\epsilon+\|\boldsymbol{u}(n)\|^2}\boldsymbol{u}(n)\hat{e}(n), \tag{4}$$

where the normalized step-size μ_N is dimensionless and invariant under scaling of the input signal, in the range $0<\mu_N\leq 2$. Also $\epsilon>0$ is a regularization factor that prevents division by a value close to zero.

The optimization problem in eq. (3) can have the following geometric interpretation in the special case of $\mu_N=1$ and $\epsilon=0$. Let us consider the Hilbert space $\mathcal{H}:=\mathbb{R}^N$, as in [13]. At each iteration n, the coefficient vector $\hat{\boldsymbol{w}}(n)$ is represented by a point in $\mathcal{H}$ and the constraint from eq. (3) leads to an affine subspace (hyperplane) of $\mathcal{H}$, namely:

$$\Pi_n \triangleq \{\boldsymbol{w}\in\mathbb{R}^N \mid d(n)-\langle\boldsymbol{u}(n),\boldsymbol{w}\rangle=0\}, \tag{5}$$

where $\langle\cdot,\cdot\rangle$ is the inner product. Note that $d(n)$ is a translation, so Π_n is orthogonal to $\boldsymbol{u}(n)$ and passes through the point $\frac{d(n)}{\boldsymbol{u}(n)^T\boldsymbol{u}(n)}\boldsymbol{u}(n)$. Among infinitely many vectors $\boldsymbol{w}\in\Pi_n$ the algorithm selected the closest one to $\hat{\boldsymbol{w}}(n)$ in the Euclidean norm sense, i.e., the orthogonal projection P_{Π_n} of $\hat{\boldsymbol{w}}(n)$ onto Π_n:

$$P_{\Pi_n}(\hat{\boldsymbol{w}}(n))=\hat{\boldsymbol{w}}(n)-\boldsymbol{u}(n)\frac{\langle\boldsymbol{u}(n),\hat{\boldsymbol{w}}(n)\rangle}{\boldsymbol{u}^T(n)\boldsymbol{u}(n)}. \tag{6}$$

By inspection, eq. (6) is the NLMS update equation in eq. (4) assuming $\mu_N=1$ and $\epsilon=0$, that is, $\hat{\boldsymbol{w}}(n+1)=P_{\Pi_n}(\hat{\boldsymbol{w}}(n))$.

Figure 1, created using the Matlab drawLA toolbox [14], illustrates the geometric interpretation in $\mathbb{R}^3$ of the adaptive algorithms discussed in this paper, assuming $\mu_N=1$ and $\epsilon=0$. The affine subspaces Π_{n-1} and Π_n, respectively in red and blue in the figure, are translated linear subspaces (planes), where the estimated coefficient vectors should be located. Given the vector of coefficients $\hat{\boldsymbol{w}}(n)$ (point 1) in Π_{n-1} (red), the constraint in eq. (5) imposes that the NLMS updated vector $\hat{\boldsymbol{w}}(n+1)$ is located in the subspace Π_n (blue). Furthermore, from eq. (6), the updated vector is the orthogonal projection of $\hat{\boldsymbol{w}}(n)$ onto the subspace Π_n (point 2). Note that, in the absence of noise, the optimum coefficient vector $\boldsymbol{w}^o$ should be in the intersect subspace $\Pi_n\cap\Pi_{n-1}$ (point 3). The figure also shows, in green, the subspaces where sparse coefficient vector are located. Each of these subspaces is aligned with two coordinate axes of $\mathbb{R}^3$. Finally, note that when the angle between the input vectors $\boldsymbol{u}(n)$ and $\boldsymbol{u}(n-1)$ is close to zero or 180°, i.e., the input vectors are highly correlated, the rate of convergence towards $\boldsymbol{w}^o$ is slow.

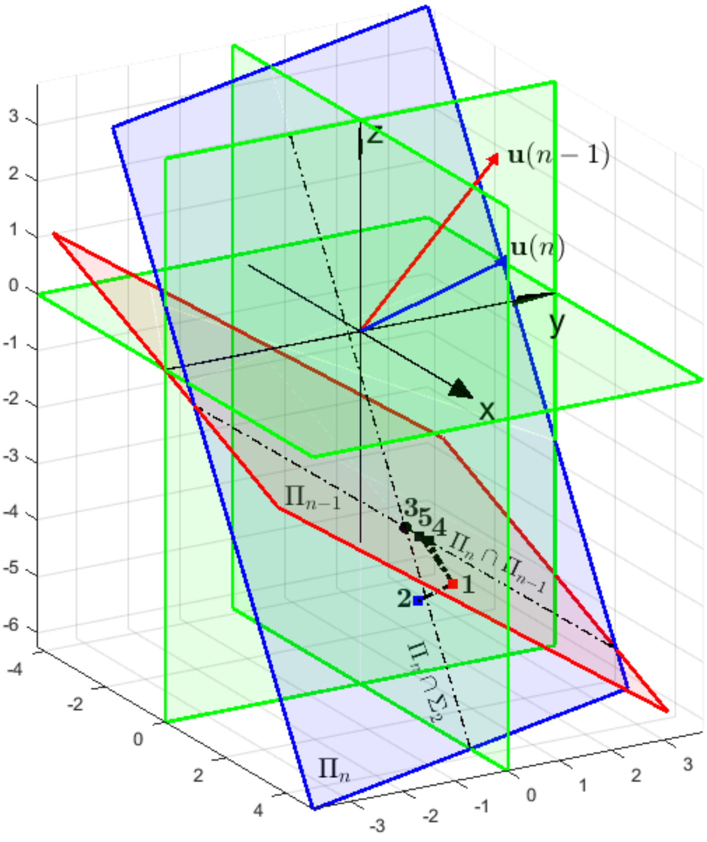


Fig. 1: Geometric interpretation of the algorithms.

B. *Affine Projection*

The AP algorithm is an extension of the NLMS, whose goal is to improve the NLMS convergence rate for highly correlated input data. Assuming that previous data is available, the AP algorithm uses at each iteration the L most recent $M\times 1$ input vectors $\boldsymbol{u}(n-i)$, for $i=0,\dots,L-1$ (where L is the projection order) and the corresponding desired responses $d(n-i)$. The algorithm assembles the $M\times L$ input matrix $\boldsymbol{U}(n)=\begin{bmatrix}\boldsymbol{u}(n) & \boldsymbol{u}(n-1) & \dots & \boldsymbol{u}(n-L+1)\end{bmatrix}$ and the response vector $\boldsymbol{d}(n)=\begin{bmatrix}d(n) & d(n-1) & \dots & d(n-L+1)\end{bmatrix}^T$. The filter output vector is given by:

$$\hat{\boldsymbol{y}}(n)=\boldsymbol{U}^T(n)\hat{\boldsymbol{w}}(n) \tag{7}$$

and the estimation error vector is $\hat{\boldsymbol{e}}(n)\triangleq\boldsymbol{d}(n)-\hat{\boldsymbol{y}}(n)$.

The AP update equation is obtained by solving the following constrained optimization problem [10], [15]:

$$\begin{aligned} \underset{\hat{\boldsymbol{w}}(n+1)\in\mathbb{R}^{M\times 1}}{\text{minimize}} \quad & \|\hat{\boldsymbol{w}}(n+1)-\hat{\boldsymbol{w}}(n)\|^2 \\ \text{subject to} \quad & \boldsymbol{d}(n)-\boldsymbol{U}^T(n)\hat{\boldsymbol{w}}(n+1)=\boldsymbol{0}_L \end{aligned} \tag{8}$$

Note that the multiple constraints in this problem reduce to the one in eq. (3) when $L=1$. Similarly to eq. (4), the AP adaptive filter update equation is given by:

$$\hat{\boldsymbol{w}}(n+1) = \hat{\boldsymbol{w}}(n) + \mu_N \boldsymbol{U}(n)\boldsymbol{S}(n)\hat{\boldsymbol{e}}(n), \tag{9}$$

where μ_N is the normalized step-size and $\boldsymbol{S}(n) \triangleq (\boldsymbol{U}^T(n)\boldsymbol{U}(n) + \epsilon \boldsymbol{I}_L)^{-1}$, where $\boldsymbol{I}_L$ is the $L \times L$ identity matrix. Alternatively, for $\epsilon = 0$ and assuming $\boldsymbol{U}(n)$ is full column rank, we can write the right-hand side as $\hat{\boldsymbol{w}}(n) + \mu_N \boldsymbol{U}^\dagger(n)\hat{\boldsymbol{e}}(n)$, where $\boldsymbol{U}^\dagger(n) = \boldsymbol{U}(n)(\boldsymbol{U}^T(n)\boldsymbol{U}(n))^{-1}$ is the Moore–Penrose pseudo-inverse of $\boldsymbol{U}(n)$. This way, $\hat{\boldsymbol{w}}(n+1)$ satisfies all the constraints in eq. (8) and is the orthogonal projection of $\hat{\boldsymbol{w}}(n)$ onto the (also affine) intersect subspace $\Pi^{(n)} \triangleq \Pi_n \cap \Pi_{n-1} \cap \cdots \cap \Pi_{n-L+1}$. In this case, the update vector $\Delta\hat{\boldsymbol{w}}(n)$ is orthogonal to $\Pi^{(n)}$, rather than to Π_n:

$$\Delta\hat{\boldsymbol{w}}(n) = \mu[\boldsymbol{d}(n) - \boldsymbol{U}^T(n)\hat{\boldsymbol{w}}(n)]\boldsymbol{U}^\dagger(n), \tag{10}$$

Going back to Figure 1, the AP updated coefficient vector (point 4) is in the intersect hyperplane $\Pi_n \cap \Pi_{n-1}$ ($L = 2$), thus satisfying both constraints $\boldsymbol{u}^T(n)\hat{\boldsymbol{w}}(n+1) = \boldsymbol{d}(n)$ and $\boldsymbol{u}^T(n-1)\hat{\boldsymbol{w}}(n+1) = \boldsymbol{d}(n-1)$ and is the orthogonal projection of point 1.

C. Sparse Affine Projection

Firstly, we define an N-length vector $\boldsymbol{w}$ as being k-sparse if it has at most $k \leq N$ non-zero entries, i.e., $\|\boldsymbol{w}\|_0 \leq k$. The set of all k-sparse vectors in $\mathbb{R}^N$ is defined as:

$$\Sigma_k \triangleq \{\boldsymbol{w} \in \mathbb{R}^N | \|\boldsymbol{w}\|_0 \leq k\} \tag{11}$$

and is not a subspace in $\mathbb{R}^N$, because the closure property for vector addition does not hold for two element vectors in the set with different supports. Instead, Σ_k consists of the union of $\binom{N}{k}$ subspaces of dimension k, each of which aligned with k out of N coordinate axes of $\mathbb{R}^N$. Figure 1 shows Σ_2, in green, as the union of three subspaces of 2-sparse vectors.

Let us now add a sparsity-inducing penalty to the standard AP optimization problem in eq. (8). The general solution for the sparse AP optimization problem is developed in [16], [17], leading to:

$$\begin{aligned}\hat{\boldsymbol{w}}(n+1) &= \hat{\boldsymbol{w}}(n) + \mu_N \boldsymbol{U}(n)\boldsymbol{S}(n)\hat{\boldsymbol{e}}(n) \\ &\quad + \mu_N\lambda[\boldsymbol{U}(n)\boldsymbol{S}(n)\boldsymbol{U}^T(n) - \boldsymbol{I}_M]\ \nabla\|\hat{\boldsymbol{w}}(n+1)\|_p,\end{aligned} \tag{12}$$

where $\|\cdot\|_p$, with $p = 0$ or 1, is a sparsity-inducing norm, $\nabla(\cdot)$ is the gradient vector and λ is a regularization parameter. The implementation of eq. (12) requires the choice of a sparsity-inducing penalty function, as discussed next. It also assumes that $\nabla\|\hat{\boldsymbol{w}}(n+1)\|_p \approx \nabla\|\hat{\boldsymbol{w}}(n)\|_p$ [16]. Several penalty choices have been proposed in the literature, among the most prominent are:

- ℓ_1-norm regularization. The first approach is the convex relaxation of the ℓ_0- by the ℓ_1-norm, given by $\|\boldsymbol{w}(n)\|_1 = \sum_{i=0}^{M-1}|w_i(n)|$, leading to the zero-attracting affine projection (ZA-APA) filter [16].
- ℓ_0-pseudo-norm approximations. The ℓ_0-pseudo-norm can be approximated in several ways, such as the AP-SSI filter in [17] and others [8], [7], [6].
- Heuristic approaches. In [16] the re-weighted zero attractor RZA-APA is derived, in which the zero attraction now depends on the tap-weight magnitudes. Other heuristic approaches are, for example, the proportionate NLMS (P-NLMS) [9].

III. Proposed Sparse Affine Projection

We propose a new AP adaptive filter, namely the SS-AP, derived from a smooth and continuous approximation of the ℓ_0-pseudo-norm, expressed as a weighted version of the ℓ_2-norm. This ℓ_0 approximation is the SparseStep proposed in [12]:

$$\|\boldsymbol{w}(n)\|_0 \approx \sum_{i=0}^{M-1} \frac{w_i^2(n)}{w_i^2(n) + \gamma^2}, \tag{13}$$

where $0 < \gamma^2 \ll 1$. Decreasing γ, the approximation for the counting norm becomes more and more accurate. The gradient of the ℓ_0-norm approximation in the previous equation can be expressed as $\nabla\|\boldsymbol{w}(n)\|_0 \triangleq \boldsymbol{\Theta}(n)\boldsymbol{w}(n)$, where $\boldsymbol{\Theta}(n)$ is a diagonal matrix, whose elements are the weights:

$$\theta_{i,i} = \frac{2\gamma^2}{(w_i^2 + \gamma^2)^2}. \tag{14}$$

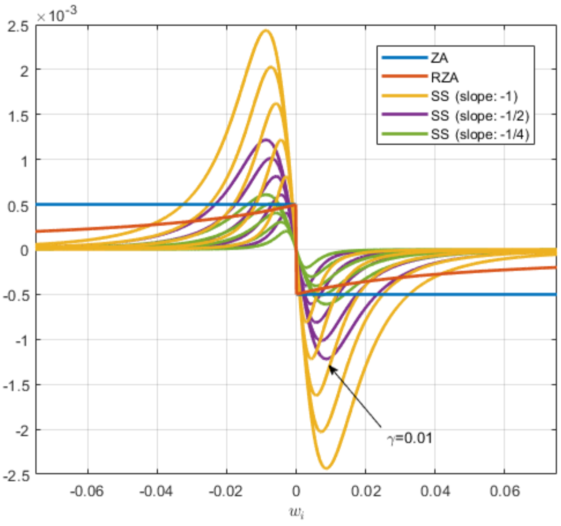


Fig. 2: Zero attractor functions.

Figure 2 compares the zero attractors of the ZA-APA, RZA-APA and SS-AP algorithms, respectively, $-\lambda sgn(\hat{\boldsymbol{w}}(n))$, $-\lambda sgn(\hat{\boldsymbol{w}}(n))/(1+\delta|\hat{\boldsymbol{w}}(n)|)$ and $-\lambda\boldsymbol{\Theta}(n)\hat{\boldsymbol{w}}(n)$, for a single coefficient w_i. The ZA-APA attractor is always equal to $\pm\lambda$, therefore, it applies an uniform attraction to all coefficient values. Meanwhile, the RZA-APA attractor selectively depends on the amplitude of each w_i. The figure shows three families of SS-AP attractors, with different slopes $\{-1/4, -1/2, -1\}$ in the linear region that can be observed when w_i is close to zero. Indeed, for $|w_i| \leq \gamma/2$, the SS-AP attractor is approximately linear with negative slope $-2\lambda/\gamma^2$. Note that each SS-AP curve is for a different value of γ, from 5e$-$3 to 1.5e$-$2, and the parameter γ defines the range (interval of w_i) for which the zero attractor is active. Finally, the SS attractor is larger than the RZA one when w_i is close to γ, while the RZA attractor is larger for $w_i \gg \gamma$. In summary:

- $|w_i| \leq \gamma/2 \Rightarrow$ zero attractor is $-2\lambda/\gamma^2 w_i$
- $|w_i| \gg \gamma \Rightarrow$ zero attractor is zero

and since the SS-AP attractor is not always active, it avoids estimation biases for large values of w_i. Next, the SS-AP update equation is obtained from eq. (12) as:

$$\begin{aligned}\hat{\boldsymbol{w}}(n+1) &= \hat{\boldsymbol{w}}(n) + \mu_N \boldsymbol{U}(n)\boldsymbol{S}(n)\hat{\boldsymbol{e}}(n) \\ &+ \mu_N\lambda[\boldsymbol{U}(n)\boldsymbol{S}(n)\boldsymbol{U}^T(n) - \boldsymbol{I}_M]\ \boldsymbol{\Theta}(n)\hat{\boldsymbol{w}}(n).\end{aligned} \tag{15}$$

Furthermore, in the SS-AP, the term $\lambda\boldsymbol{\Theta}(n)\hat{\boldsymbol{w}}(n)$ is projected orthogonally onto the intersect subspace $\Pi^{(n)}$, as discussed below. We can rewrite eq. (15), for $\mu_N = 1$ and $\epsilon = 0$, as follows:

$$\begin{aligned}\hat{\boldsymbol{w}}(n+1) \ = \ &\underbrace{\boldsymbol{U}^{\dagger}(n)\boldsymbol{d}(n)}_{\text{(affine) translation}} + \underbrace{[\boldsymbol{I}_M - \boldsymbol{U}^{\dagger}(n)\boldsymbol{U}^T(n)]\hat{\boldsymbol{w}}(n)}_{P^{\perp}_{\Pi_{\boldsymbol{U}(n)}}(\hat{\boldsymbol{w}}(n))} \\ - &\underbrace{[\boldsymbol{I}_M - \boldsymbol{U}^{\dagger}(n)\boldsymbol{U}^T(n)]\ \lambda\boldsymbol{\Theta}(n)\hat{\boldsymbol{w}}(n)}_{P^{\perp}_{\Pi_{\boldsymbol{U}(n)}}(\lambda\boldsymbol{\Theta}(n)\hat{\boldsymbol{w}}(n))},\end{aligned} \tag{16}$$

where $P^{\perp}_{\Pi_{\boldsymbol{U}(n)}}$ is the projection onto the orthogonal complement of $\Pi_{\boldsymbol{U}(n)} = span(\{\boldsymbol{u}(n-i)\}_{i=0}^{L-1})$, which is the subspace spanned by the columns of $\boldsymbol{U}(n)$, with dimension L. Furthermore, the projection $P_{\Pi^{(n)}}$ onto the intersection hyperplane defined in eq. (10) (of dimension $N-L$) is equal to $P^{\perp}_{\Pi_{\boldsymbol{U}(n)}}$, since each vector $\boldsymbol{u}(n-i)$ is perpendicular to the corresponding Π_{n-i}, for $i = 0, \ldots, L-1$. Therefore, the geometric interpretation of eq. (16) is that the vectors $\hat{\boldsymbol{w}}(n)$ and $-\lambda\boldsymbol{\Theta}(n)\hat{\boldsymbol{w}}(n)$ are projected onto the hyperplane $\Pi^{(n)}$ at each iteration of the algorithm. Going back to Figure 1, the projection of the resulting SS-AP vector (point 5) will be in the intersect hyperplane $\Pi_n \cap \Pi_{n-1}$ (for $L = 2$) and approaches the intersection between this hyperplane and the set Σ_2, closer to $\boldsymbol{w}^o$ at point 3.

IV. Numerical Results

This section validates the proposed SS-AP algorithm through numerical simulation under the following operating conditions: signal-to-noise ratios (SNR) of 10 and 25 dB, channel length $N = 16$, as in [17], and sparsity level [11] of 75%, i.e., $K = 4$ non-zero coefficients. Each simulation experiment consists of 100 independent Monte Carlo runs, each of which consisting of 3,000 iterations and the results are averaged. In each simulation run, the positions of the non-zero coefficients are randomly chosen and their values follow a Gaussian distribution. We compare the performance of the proposed SS-AP algorithm with the AP, ZA-APA, RZA-APA and AP-SSI Geman-McClure in [17], here named GM-AP, all with projection order $L = 3$, as well as the NLMS and SS-NLMS [11]. The input signal is a correlated Gaussian sequence derived through a first-order autoregressive process given by $\boldsymbol{u}(n) = 0.8\boldsymbol{u}(n-1) + \boldsymbol{v}(n)$, where $\boldsymbol{v}(n)$ is unit power Gaussian noise. The observation noise is also additive Gaussian with $\sigma_n^2 = 10^{-SNR/10}$. We evaluate the mean square error (MSE) between the reference signal and the estimated outputs, the mean square deviation (MSD) between the optimal and estimated coefficients and the number of iterations to convergence.

Firstly, for each operating condition, the parameters μ and λ are tuned by testing a grid of values, as in Figure 3-(a) and Figure 4-(a). In Figure 3-(a), the SS-AP algorithm is evaluated with SNR=25 dB on the ranges: μ from 0.05 to 0.25 and λ from zero to 1e-4 and we choose values close to the elbow, i.e., where the required iterations increase more rapidly, $\mu = 0.11$ and $\lambda = 1\text{e}{-4}$, as a compromise between MSE and iterations. Then we run the AP algorithms with these parameters. Figure 3-(b) compares the resulting MSD curves, showing that among the sparse AP algorithms, the SS-AP achieves the lowest MSD. Other parameters are $\epsilon = 1\text{e}{-5}$, $\delta = 20$ for RZA-APA, $\beta = 5$ for the GM-AP and $\gamma = \sqrt{2\lambda}$ for SS-AP, i.e., slope equal to -1. Likewise, in Figure 4-(a), the SS-AP algorithm is evaluated with SNR=10 dB on the ranges: μ from 0.02 to 0.12 and λ from zero to 4e-4 and we choose values close to the elbow, $\mu = 0.05$ and $\lambda = 3.5\text{e}{-4}$. Figure 4-(b) compares the resulting MSD curves, showing again that the SS-AP achieves the lowest MSD. Table I provides a summary of the numerical results discussed above. Finally, in Table II we present additional performance results for a new simulation experiment, in which sparsity level of 87.5%, or $K = 2$ non-zero coefficients, is considered, instead of $K = 4$.

TABLE I: Performance results with $K = 4$.

	SNR= 25dB			SNR= 10dB		
	MSE	MSD	Iter.	MSE	SMD	Iter.
NLMS	-23.83	-24.46	2888	-8.59	-7.00	2942
AP	-23.80	-24.70	281	-9.45	-13.32	434
ZA-APA	-23.96	-25.59	283	-9.52	-14.44	459
RZA-APA	-23.97	-25.52	280	-9.54	-14.23	438
SS-AP	-24.25	-29.62	276	-9.66	-18.22	465
GM-AP	-24.25	-27.88	276	-9.67	-17.00	441

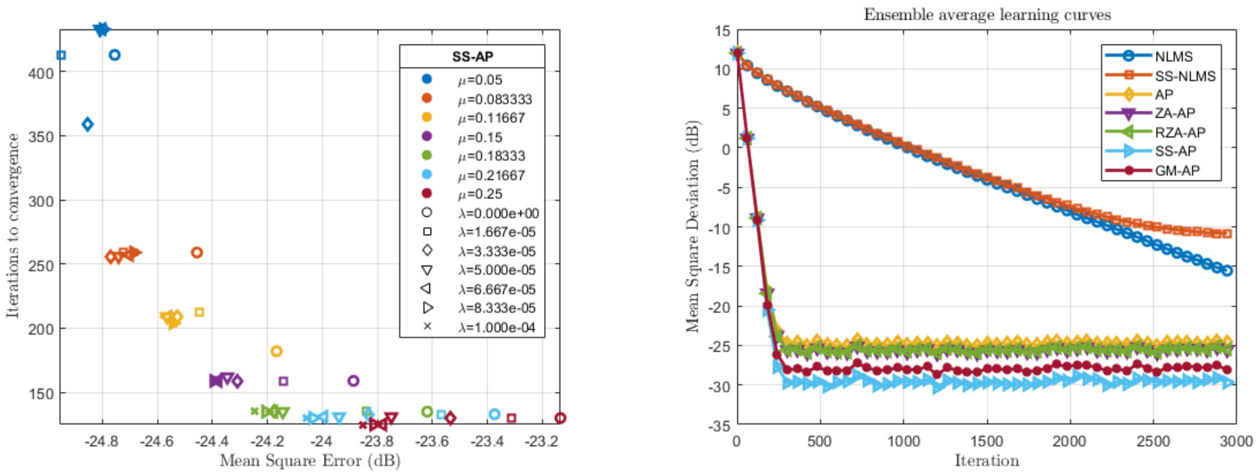
TABLE II: Performance results with $K = 2$.

	SNR= 25dB			SNR= 10dB		
	MSE	MSD	Iter.	MSE	SMD	Iter.
NLMS	-23.82	-24.65	2897	-8.72	-7.08	2945
AP	-23.79	-24.76	287	-9.44	-13.27	463
ZA-APA	-24.02	-25.93	288	-9.57	-15.05	491
RZA-APA	-23.99	-25.74	283	-9.54	-14.35	466
SS-AP	-24.53	-31.68	275	-9.76	-20.12	463
GM-AP	-24.39	-28.86	265	-9.74	-18.25	466

V. Conclusions

The following conclusions can be drawn from our study:

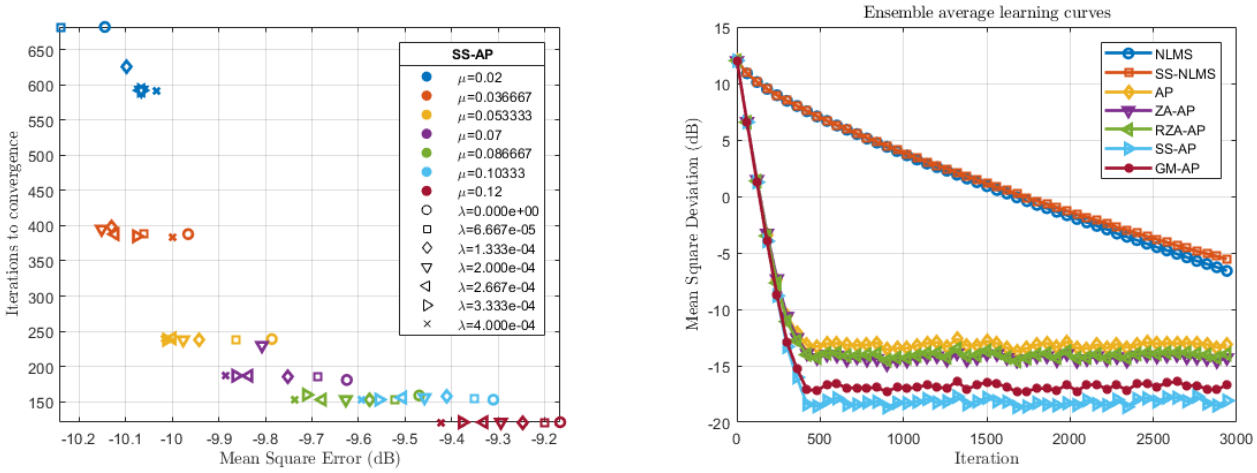
- From the numerical simulations, AP-based algorithms outperformed NLMS-based ones under correlated input signals. Sparse AP (zero attractor) algorithms outperformed standard AP in estimating zeroed coefficients.
- As the SNR decreases, the sparse AP parameters μ and λ need to be adjusted, respectively, decreased and increased. With the SS-AP, we can also adjust the attractor slope, by decreasing the parameter γ.
- The proposed SS-AP outperformed the other sparse AP algorithms in this study, specially under highly sparse settings, by further improving the estimation of zeroed coefficients. Finally, the numbers of iterations required by the sparse AP algorithms were equivalent.



(a) Tuning the parameters μ, λ.

(b) MSD (dB) curves.

Fig. 3: Performance evaluation ($K = 4$) with SNR=25 dB.



(a) Tuning the parameters μ, λ.

(b) MSD (dB) curves.

Fig. 4: Performance evaluation ($K = 4$) with SNR=10 dB.

References

[1] S. Haykin, *Adaptive Filter Theory*. Pearson, 2014.

[2] A. Sayed, *Adaptive Filters*. IEEE Press, Wiley, 2008.

[3] P. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Springer, 2008.

[4] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3125–3128, 2009.

[5] G. Gui and F. Adachi, "Improved adaptive sparse channel estimation using least mean square algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, pp. 1–18, Aug. 2013.

[6] 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, 2009.

[7] G. Gui and F. Adachi, "Improved least mean square algorithm with application to adaptive sparse channel estimation," *EURASIP Journal on Wireless Communications and Networking*, Aug. 2013.

[8] Y. Chen, Y. Gu, and A. Hero, "Regularized Least-Mean-Square algorithms," *ArXiv e-prints*, Dec. 2010.

[9] R. Seara, J. Zipf, L. Koehler, and F. Perez, "Algoritmo NLMS modificado para operação em ambientes esparsos," *Simpósio Brasileiro de Telecomunicações SBrT – SBrT 2016*, 2016.

[10] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan (Part I: Communications)*, vol. 67, no. 5, pp. 19–27, 1984.

[11] C. Hemsi, "Adaptive filter theory and application for the identification of sparse systems," *Simpósio Brasileiro de Telecomunicações SBrT – SBrT 2022*, 2022.

[12] G. J. van den Burg, P. J. Groenen, and A. Alfons, "SparseStep: Approximating the counting norm for sparse regularization," *Econ. Inst. Res. Pap.*, no. 1, pp. 1–15, 2017.

[13] M. Yukawa, "Adaptive filtering based on projection method - Lecture Notes," *Elite Master Study Course, University of Erlangen Nuremberg and Technical University of Munich, Germany*, 2010.

[14] V. Bondarenko, "DrawLA - Draw toolbox for linear algebra, MATLAB Central File Exchange," 2023.

[15] K. Ozeki, *Theory of Affine Projection Algorithms for Adaptive Filtering*. Springer Publishing Company, Incorporated, 1st ed., 2015.

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

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