

# THERE'S PLENTY OF ROOM AT THE BOTTOM: INCREMENTAL COMBINATIONS OF SIGN-ERROR LMS FILTERS

*Luiz F. O. Chamon and Cássio G. Lopes*

Signal Processing Lab  
Dept. of Electronic Systems Engineering, University of São Paulo – Brazil  
chamon@usp.br, cassio@lps.usp.br

## ABSTRACT

The introduction of data reuse in the incremental topology made it possible for combinations of LMS filters to outperform algorithms such as the Affine Projection Algorithm (APA) with lower complexity. This work poses and extends the concept of combinations as a complexity reduction technique by proposing an incremental combination of sign-error LMS filters that matches and even outperforms stand-alone LMS filters with reduced complexity. This combination is then analyzed and used as a building block in a larger combination that is able to match the APA at a reduced computational cost. Numerical simulations illustrate the performance of this novel combination in different scenarios.

**Index Terms**— Adaptive filtering, Combination of adaptive filters, Data reuse incremental combinations, Affine Projection Algorithm, Sign-error LMS

## 1. INTRODUCTION

Combinations of adaptive filters (AFs) have become an established technique to improve the performance of adaptive algorithms [1–9]. It consists of mixing a pool of AFs—the *components*—so that the overall system is at least as good—usually in the mean-square error (MSE) sense—as the best filter in the set. It has been used to reduce transient/steady-state trade-offs [2, 4, 6, 7] and improve tracking [3, 9] and prediction [1, 5] in different scenarios. Though effective, combinations imply an increase in complexity, as several AFs must be evaluated at each iteration. A combination usually has the computational complexity of the sum of its components, although it can be reduced in some special cases [10–13].

Recently, a different view of combinations was put forward with the development of data reusing (DR) incremental structures. In this approach, combinations are used to create algorithms with the same performance as more powerful—and complex—AFs but with lower computational complexity. For instance, combinations of LMS filters have been shown to match and outperform the Affine Projection Algorithm (APA) with up to ten times less operations. Furthermore, it bridged the gap between DR adaptive algorithms [14–17] and combinations of AFs [8, 18].

This work explores and extends this novel concept of *combinations as a complexity-reduction technique* by (i) introducing an incremental combination of sign-error LMS filters that matches the performance of the LMS with less multiplications; (ii) showing that the resulting  $\{\text{sign-error LMS}\}^N$  asymptotically approaches the NLMS recursion<sup>1</sup>; (iii) designing the components so as to minimize

<sup>1</sup>In our notation,  $\text{LMS} \cdot \text{LMS} \cdots \text{LMS} = \{\text{LMS}\}^N$  represents an incremental combination of  $N$  LMS filters [18].

their number; and (iv) using the  $\{\text{sign-error LMS}\}^N$  as building blocks in combinations that may outperform the APA.

## 2. INCREMENTAL COMBINATIONS

### 2.1. Adaptive filtering

In a system identification scenario, an AF operates over the data pair  $\{u_i, d(i)\}$ , where  $u_i$  is a  $1 \times M$  regressor vector that captures samples  $u(i)$  of a real-valued zero mean input signal with variance  $\sigma_u^2$  and  $d(i) = u_i w^o + v(i)$  is a measurement of the output of an unknown system modeled by the  $M \times 1$  vector  $w^o$ . This measurement is corrupted by a noise  $v(i)$ , modeled as a zero mean i.i.d. real Gaussian random variable with variance  $\sigma_v^2$ . At iteration  $i$ , the adaptive algorithm updates a previous estimate  $w_{i-1}$  of  $w^o$  by

$$w_i = w_{i-1} + \mu p, \quad (1)$$

using a step size  $\mu$  and an update direction  $p = -B \nabla^T J(w_{i-1})$ , with  $B$  a positive-definite matrix,  $J(w_{i-1})$  the underlying cost function of the AF, and  $^T$  denoting the transpose operation. Usually, AFs attempt to minimize the MSE:  $J(w_{i-1}) = E e^2(i)$ , where  $e(i) = d(i) - u_i w_{i-1}$  is the output estimation error [19].

Different choices for  $p$  lead to different algorithms, such as the ones presented below in increasing order of complexity:

$$w_i = w_{i-1} + \mu u_i^T \text{sign}[e(i)] \quad (\text{sign-error LMS}) \quad (2a)$$

$$w_i = w_{i-1} + \mu u_i^T e(i) \quad (\text{LMS}) \quad (2b)$$

$$w_i = w_{i-1} + \frac{\mu}{\|u_i\|^2 + \epsilon} u_i^T e(i) \quad (\text{NLMS}) \quad (2c)$$

$$w_i = w_{i-1} + \mu U_i^T (U_i U_i^T + \epsilon I)^{-1} e_i, \quad (\text{APA}) \quad (2d)$$

where  $\text{sign}[\cdot]$  is the signum operator—i.e.,  $\text{sign}[a] = 1$ , for  $a > 0$ ;  $\text{sign}[a] = -1$ , for  $a < 0$ ; and  $\text{sign}[a] = 0$ , for  $a = 0$ —and  $\epsilon \ll 1$  is a regularization factor [19, 20]. The APA operates over the larger data set  $U_i = [u_i^T \cdots u_{i-K+1}^T]^T$ , a  $K \times M$  regressors matrix, and  $d_i = [d(i) \cdots d(i-K+1)]^T$ , a  $K \times 1$  measurements vector, so that  $e_i = d_i - U_i w_{i-1}$ . It is presented in (2d) in its standard form, though less complex recursions exist [21].

### 2.2. Data reusing incremental combinations

Initially, incremental combinations were proposed as a solution to the convergence stagnation issue of the parallel-independent structure [7]. Since then, they have been shown to improve the overall performance of the resulting algorithms [9, 18, 22], even more so when the DR techniques introduced in [8] are used.

DR usually involves either using a same data pair  $\{u_i, d(i)\}$  several times or operating over a set of data pairs  $\{U_i, d_i\}$ . A DR incremental combination of  $N$  LMS filters is then defined as

$$\begin{aligned} w_{0,i} &= w_{i-1} \\ w_{n,i} &= w_{n-1,i} + \eta_n(i) \mu_n u_{n,i}^T e_n(i) \\ w_i &= w_{N,i}, \end{aligned} \quad (3)$$

where the component filters are indexed by  $n = 1, \dots, N$ . Hence, for the  $n$ -th component,  $\mu_n$  is the step size,  $e_n(i) = d_n(i) - u_{n,i} w_{n-1,i}$ ,  $\eta_n(i)$  represents the effect of the supervisor, and  $\{u_{n,i}, d_n(i)\}$  accounts for different DR strategies [8]. The combination outputs the *global* coefficients  $w_i$ , which yields the *global* output estimation error  $e(i) = d(i) - u_i w_{i-1}$  [7–9].

Most combinations [2–7, 9] employ *data sharing*, a scheme in which  $\{u_{n,i}, d_n(i)\} = \{u_i, d(i)\}$ . Alternatively, a *data buffering* method where  $\{u_{n,i}, d_n(i)\} = \{u_{i-k}, d(i-k)\}$ , with  $k = (n-1) \bmod K$ , can be used [8, 18], so that the components go over the data set  $\{U_i, d_i\}$  as many times as required. Notice that several DR AFs can be recovered as special cases of (3)—such as the DR-LMS filters from [14–16] for  $N = K$ ,  $\eta_n = 1$ , and  $\mu_n = \mu$ .

### 3. “THERE’S PLENTY OF ROOM AT THE BOTTOM”

Modern adaptive filtering applications [5, 23–25], such as telecommunication and speech echo cancellation, simultaneously require good performance and high throughput—i.e., low complexity algorithms. However, performance and complexity are usually competing objectives: using AFs like the APA generally yields better convergence and misadjustment than an LMS filter but also entails more computation [19, 20]. Moreover, developers have been migrating from DSPs to dedicated processor solutions—FPGAs—, where multiply-and-accumulate (MAC) operations are costly and may not be readily available [25]. In this context, several solutions have been proposed to reduce the complexity of high performance AFs and conform their recursions for FPGA implementation, such as dichotomous coordinate descent (DCD) iterations [26, 27].

More recently, a different approach was introduced in [8, 18] with the development of low complexity *data buffering* incremental combinations of LMS filters that match—and even outperform—the APA in (2d) [19, 20]. These results gave rise to the counter-intuitive concept of *combinations as a complexity-reduction technique*. Now, Mr. Feynman might ask whether there is still room at the bottom [28]. Are LMS filters the smallest *quanta* of adaptivity? Would it be possible to combine smaller *particles* and reduce complexity without compromising performance?

#### 3.1. Incremental combinations of sign-error LMS filters

The sign-error LMS in (2a) was an early solution to the complexity problem: choosing  $\mu$  as a power of two replaces all multiplications—except for those involved in the filtering  $y(i) = u_i w_{i-1}$ —by simple bit shifts [19, 20, 29]. This AF, however, presents slow convergence speed—see Fig. 1—and bias issues: the mean coefficients error only converges to a ball around  $w^o$  with radius proportional to the step size as opposed to the LMS, which is unbiased [19, 20, 29].

Motivated by the convergence improvements enabled by the incremental topology [18, 22], an incremental—*data sharing*—combination of sign-error LMS filters is proposed. Explicitly,

$$\begin{aligned} w_{0,i} &= w_{i-1} \\ w_{n,i} &= w_{n-1,i} + \mu_n u_i^T \text{sign}[e_n(i)] \\ w_i &= w_{N,i}, \end{aligned} \quad (4)$$

#### Algorithm 1 The $\{\text{sign-error LMS}\}^N$

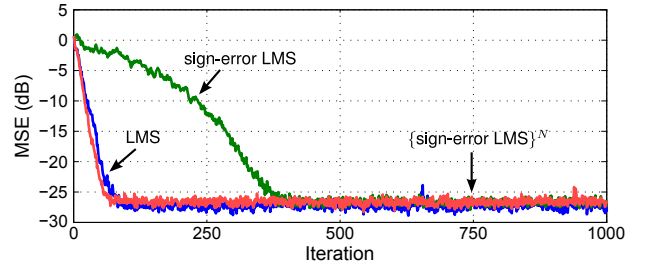
---

```

 $\|u_i\|^2 = \|u_{i-1}\|^2 - |u(i-M)|^2 + |u(i)|^2$  ▷ (1) ×
 $y(i) = u_i w_{i-1}$ ;  $e_1(i) = d(i) - y(i)$  ▷ (M) ×
 $w_{0,i} = w_{i-1}$ 
for  $n = 1, \dots, N$ 
   $w_{n,i} = w_{n-1,i} + \mu_n u_i^T \text{sign}[e_n(i)]$ 
   $e_{n+1}(i) = e_n(i) - \mu_n \|u_i\|^2 \text{sign}[e_n(i)]$ 
end
 $w_i = w_{N,i}$ 

```

---



**Fig. 1.** White stationary simulations with fixed point quantization (16 bits). **sign-error LMS**:  $\mu = 0.004$ ; **LMS**:  $\mu = 0.083$ ; and  **$\{\text{sign-error LMS}\}^N$** :  $N = 9$  and  $\mu_n = 2^{-3-n}$ .

for  $e_n(i) = d(i) - u_i w_{n-1,i}$  and  $\mu_n$  a power-of-two factor. Note that this combination is unsupervised, i.e.,  $\eta_n(i) = 1$ . In its current form, (4) requires  $NM$  multiplications. However, if  $u_i$  has shift structure—transversal filter—this combination can be implemented efficiently. Developing the  $(n+1)$ -th component error

$$\begin{aligned} e_{n+1}(i) &= d(i) - u_i w_{n,i} \\ &= d(i) - u_i \left\{ w_{n-1,i} + \mu_n u_i^T \text{sign}[e_n(i)] \right\} \\ &= e_n(i) - \mu_n \|u_i\|^2 \text{sign}[e_n(i)] \end{aligned} \quad (5)$$

where due to the shift structure of  $u_i$ , one has  $\|u_i\|^2 = \|u_{i-1}\|^2 - |u(i-M)|^2 + |u(i)|^2$ , which only requires 1 multiplication given that all  $|u(j)|^2$ , for  $j < i$ , are available from previous iterations.

This efficient implementation is summarized in Algorithm 1 and only requires  $M+1$  multiplications. Notice that this number does not depend on  $N$ , the number of component filters. Alternatively, the filtering process in  $y(i)$  could be implemented using distributed arithmetic (DA) [25, 30, 31], so that only one multiplication remains. Despite its lower complexity, this combination is indeed able to achieve the performance of an LMS filter, as shown in Fig. 1.

#### 3.2. $\{\text{sign-error LMS}\}^N$ as an efficient NLMS

A complete analysis of the combination in (4) is available, but lengthy. In the sequel, a simpler version is presented with enough to suggest that its improved performance comes from the fact that it approaches an NLMS recursion.

A global coefficients recursion of the  $\{\text{sign-error LMS}\}^N$  can be derived from (4) by grouping the equations into

$$w_i = w_{i-1} + u_i^T \sum_{n=1}^N \mu_n \text{sign}[e_n(i)]. \quad (6)$$

This relation can be put in a more convenient form by finding a recursion for  $\text{sign}[e_n(i)]$ . Indeed, taking the sign of (5) and noticing that  $e_{n-1}(i) = |e_{n-1}(i)| \text{sign}[e_{n-1}(i)]$  yields

$$\text{sign}[e_n(i)] = \text{sign}[e_{n-1}(i)] \text{sign} \left[ |e_{n-1}(i)| - \mu_{n-1} \|u_i\|^2 \right],$$

which can be iterated to arrive at

$$\text{sign}[e_n(i)] = \text{sign}[e(i)] \prod_{k=1}^{n-1} \text{sign}[|e_k(i)| - \mu_k \|u_i\|^2]. \quad (7)$$

Equation (7) can be used in (6) to get

$$w_i = w_{i-1} + \bar{\mu}(i) u_i^T \text{sign}[e(i)] \quad (8a)$$

$$\bar{\mu}(i) = \mu_1 + \sum_{n=2}^N r_n(i) \mu_n, \quad (8b)$$

where  $r_n(i) = \prod_{k=1}^{n-1} \mathcal{S}_k(i)$  and  $\mathcal{S}_k(i) = \text{sign}[|e_k(i)| - \mu_k \|u_i\|^2]$ .

The overall recursion (8a) has the form of a sign-error LMS with a variable step size (VSS)  $\bar{\mu}(i)$ . Thus, it is subject to the same convergence condition as any sign-error LMS filters [19, 20, 29]

$$\|\tilde{w}_i\|^2 \leq \|\tilde{w}_{i-1}\|^2 \Leftrightarrow |e(i)| \geq \bar{\mu}(i) \|u_i\|^2, \quad (9)$$

where  $\tilde{w}_i = w^o - w_i$  is the coefficients error vector. This condition can be enforced choosing, for some  $\alpha \in (0, 1]$ ,

$$\bar{\mu}^o(i) = \alpha \frac{|e(i)|}{\|u_i\|^2}. \quad (10)$$

Substituting (10) into (8a) leads to

$$w_i = w_{i-1} + \alpha \frac{|e(i)|}{\|u_i\|^2} u_i^T \text{sign}[e(i)] = w_{i-1} + \frac{\alpha}{\|u_i\|^2} u_i^T e(i), \quad (11)$$

which is the recursion of an NLMS with  $\epsilon = 0$  and  $\mu = \alpha$ . Hence, the VSS sign-error LMS in (8a) becomes an NLMS for  $\bar{\mu}(i) = \bar{\mu}^o(i)$ , which is implicitly imposed by the combination.

### 3.3. Successive approximations of the overall step size

Since the {sign-error LMS}<sup>N</sup> is equivalent to the VSS sign-error LMS in (8), showing that the {sign-error LMS}<sup>N</sup> → NLMS requires only that one demonstrates that, for  $\bar{\mu}(i)$  as in (8b),

$$|\bar{\mu}(i) - \bar{\mu}^o(i)| \rightarrow 0, \quad N \rightarrow +\infty. \quad (12)$$

To do so, assume, without loss of generality, that  $\alpha = 1$ . Then, substituting (8b) and (10) in (12) yields

$$\frac{1}{\|u_i\|^2} \left| \mu_1 \|u_i\|^2 + \sum_{n=2}^N r_n(i) \mu_n \|u_i\|^2 - |e(i)| \right|. \quad (13)$$

For  $\|u_i\|^2 \neq 0$ , suffices to show that the absolute value in (13) goes to zero. To derive a recursion for it, define  $r_1(i) = 1$  and note that  $e(i) = e_1(i)$  so that for any  $k$  the absolute value in (13) has the form

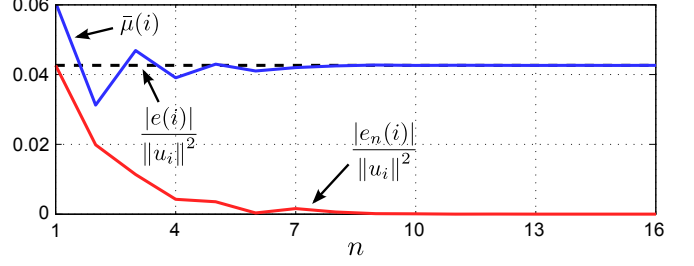
$$\left| \mu_k \|u_i\|^2 + r_k(i) \left[ \sum_{n=k+1}^N r_n(i) \mu_n \|u_i\|^2 \right] - |e_k(i)| \right|,$$

which is equivalent to

$$\left| r_k(i) \left[ \sum_{n=k+1}^N r_n(i) \mu_n \|u_i\|^2 \right] - [ |e_k(i)| - \mu_k \|u_i\|^2 ] \right|. \quad (14)$$

Noticing from (5) that  $|e_{k+1}(i)| = [ |e_k(i)| - \mu_k \|u_i\|^2 ] \cdot \mathcal{S}_k(i)$  and multiplying (14) by  $|\mathcal{S}_k(i)| = 1$ , it can be rearranged to read

$$\left| \mathcal{S}_k(i) r_k(i) \left[ \sum_{n=k+1}^N r_n(i) \mu_n \|u_i\|^2 \right] - |e_{k+1}(i)| \right| \quad (15)$$



**Fig. 2.** Evolution of a realization of  $\bar{\mu}(i)$  and  $|e_n(i)|$  during one iteration  $i$ :  $M = 10$ ,  $N = 16$ , and  $\mu_n = 2^{-4-n}$  (MATLAB precision).  $|e_n(i)|$  is normalized by  $\|u_i\|^2$  to remain within the range of the plot.

Given that  $r_{k+1}(i) = \mathcal{S}_k(i) r_k(i)$  (see (8b)), one gets

$$\begin{aligned} & \left| \mu_k \|u_i\|^2 + r_k(i) \left[ \sum_{n=k+1}^N r_n(i) \mu_n \|u_i\|^2 \right] - |e_k(i)| \right| = \\ & \left| \mu_{k+1} \|u_i\|^2 + r_{k+1}(i) \left[ \sum_{n=k+2}^N r_n(i) \mu_n \|u_i\|^2 \right] - |e_{k+1}(i)| \right| \end{aligned} \quad (16)$$

An inductive process can then be formed from (13) and (16). Notice that for a sufficiently small and decreasing sequence  $\{\mu_k\}$ , it holds from (5) that  $|e_{k+1}| < |e_k|$ , so that  $|e_k(i)| \rightarrow 0$ , along with  $\mu_k \|u_i\|^2$ , as one iterates over (16)—i.e., as  $k$  grows. Furthermore, the summation in (13) involves consecutively fewer terms with ever smaller  $\mu_k$ , thus also going to zero. Hence, (12) vanishes for large enough  $N$  and  $\bar{\mu}(i) \rightarrow \bar{\mu}^o(i)$ .

It is possible to formalize this demonstration and show that the {sign-error LMS}<sup>N</sup> asymptotically forces the *a posteriori* output estimation error  $e_{N+1}(i) = u_i w_i$  to vanish. Therefore, it can be stated as a well posed optimization problem similar to the *least perturbation principle* of the NLMS [19]. This phenomenon is illustrated in Fig. 2, where one can see the  $|e_n(i)|$  indeed tends to zero.

### 3.4. Minimizing the number of components

From an application viewpoint, using small  $\mu_n$  and large  $N$ —as in the previous section—to guarantee a good approximation for  $\bar{\mu}(i)$  is not practical. However, notice that (8b) can be seen as the truncated representation of a real number in terms of Rademacher functions [32]. Hence, it is in the space spanned by the component filters step sizes. Choosing  $\mu_n$  as a basis for the space in which  $\bar{\mu}(i)$  must lie then minimizes the number of components needed—as any basis is a minimal representation of the space it spans [33]. In the binary environment of embedded systems, an appropriate choice would be  $\mu_n = 2^{-P-n}$ , where  $P \in \mathbb{Z}$ , which yields  $\bar{\mu}(i) \in (0, 2^{-P+1})$ .

This design has several advantages: (i) it is a basis of the underlying number space, thus minimizing the number of components necessary; (ii) the  $\mu_n$  are power-of-two numbers, so that their multiplications can be replaced by bit shifts; and (iii) the common factor  $2^{-P}$  acts as the factor  $\alpha$  in (11) and constrains the maximum overall step size, increasing the robustness of the design against the truncated representation and noise. As illustrated in Fig. 2, this choice of  $\mu_n$  is indeed effective.

## 4. MATCHING THE APA PERFORMANCE

The study of the {sign-error LMS}<sup>N</sup> presented in Section 3 showed that is able to at least match the performance of an LMS filter with

---

**Algorithm 2** The DR- $\{\text{sign-error LMS}\}^N$ <sup>L</sup>


---


$$\|u_i\|^2 = \|u_{i-1}\|^2 - |u(i-M+1)|^2 + |u(i)|^2 \quad \triangleright \text{(I)} \times$$

$$w_{0,i} = w_{i-1} \quad \triangleright \text{DR incremental combination}$$

**for**  $\ell = 1, \dots, L$

$$k = (\ell - 1) \bmod K$$

$$\varepsilon_1(i) = d(i-k) - u_{i-k} w_{\ell-1,i} \quad \triangleright \text{(M)} \times$$

$$w_{\ell,i}^{(0)} = w_{\ell-1,i}$$

**for**  $n = 1, \dots, N$   $\triangleright \{\text{sign-error LMS}\}^N$

$$w_{\ell,i}^{(n)} = w_{\ell,i}^{(n-1)} + \mu_n u_{i-k}^T \text{sign}[\varepsilon_n(i)]$$

$$\varepsilon_{n+1}(i) = \varepsilon_n(i) - \mu_n \|u_{i-k}\|^2 \text{sign}[\varepsilon_n(i)]$$

**end**

$$w_{\ell,i} = w_{\ell,i}^{(N)}$$

**end**

$$w_i = w_{L,i}$$


---

**Table 1.** Multiplications involved in the recursion of different AFs

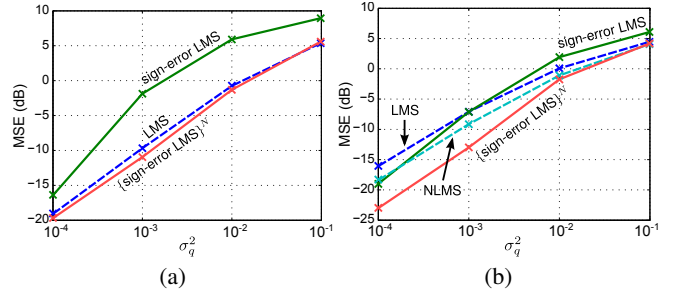
AF	$\times$ (e.g., Fig. 4b)
Standard APA	$(K^2 + 2K)M + K^3 + K = 13010$
DCD-APA [27]	$M + K^2 + 3K + 2 = 232$
DR- $\{\text{LMS}\}^L$ [8]	$(2M + 1)L, L \ll K^2 = 6030$
Algorithm 2	$LM + 1, L \ll K^2 = 3001$
Algorithm 2 (optimized)	$2M + K - 1 = 209$

reduced complexity. This suggests that *Algorithm 1* could be used as the building block of larger combinations, replacing the LMS filters in the DR- $\{\text{LMS}\}^L$  from [8] to further reduce its complexity. An overview of the resulting algorithm is presented in *Algorithm 2*. This implementation requires  $LM + 1$  multiplications. An optimized version can be found by further exploiting the shift structure of  $u_i$ , creating buffers for  $u(i-p-1)u(i-q-1)$  and  $u_{i-p-1}u_{i-q-1}^T$ ,  $p, q < K + M$ , and updating  $e_i$  at each iteration  $n$ . In this case, the complexity of the algorithm would reduce to  $2M + K - 1$ . Due to space constraints, this implementation will be presented in future works. A comparison to the complexity of other adaptive algorithms with similar performance can be found in Table 1.

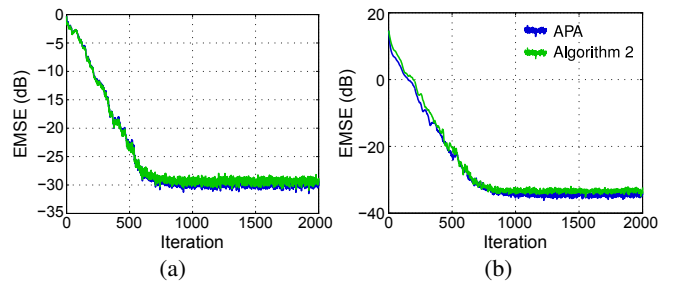
## 5. SIMULATIONS

Data for all simulations are taken from the zero mean Gaussian i.i.d. sequences  $\{x(i)\}$  and  $\{v(i)\}$ , with  $\sigma_x^2 = 1$  and  $\sigma_v^2 = 10^{-3}$ —SNR = 30 dB. White input experiments use  $u(i) = x(i)$ , whereas correlated inputs are generated using  $u(i) = \beta u(i-1) + \sqrt{1 - \beta^2} x(i)$  with  $\beta = 0.95$ , which results in a highly correlated signal. *Fixed point* simulations were performed quantizing both the data and the result of every operation to a 16 bits two's complements representation using a scaling factor of  $F = 13$  bits—i.e., 1 bit for the signal, 2 bits for the integer part, and 13 bits for the decimal part. Saturation is used to limit the values to  $[-2^2, 2^2 - 2^{-13}]$ . Finally, nonstationary scenarios were simulated using  $w^o(i) = \mathcal{Q}[w^o(i-1) + q_i]$ , where  $q_i$  is the realization of a zero mean Gaussian random vector with covariance matrix  $\sigma_q^2 I$  and  $\mathcal{Q}$  representing the quantization described earlier<sup>2</sup>. Curves for the MSE ( $E e(i)^2$ ) and EMSE ( $E[u_i \tilde{w}_{i-1}]^2$ ) are averaged over 200 and 500 independent realizations for stationary and nonstationary scenarios, respectively.

<sup>2</sup>Note the high degree of nonstationarity compared to the literature  $\sigma_q^2 = [10^{-6}, 10^{-8}]$ .  $w^o$  is quantized so that its variance does not grow unbounded.



**Fig. 3.** Nonstationary scenario with  $M = 10$  and fixed point quantization (16 bits). (a) White input. **sign-error LMS**:  $\mu = 2^{-7}$ ; **LMS**:  $\mu = \mu^o = 0.0916$ ; and  **$\{\text{sign-error LMS}\}^N$** :  $N = 6$  and  $\mu_n = 2^{-4-n}$ . (b) Correlated input. **sign-error LMS**:  $\mu = 2^{-6}$ ; **LMS**:  $\mu = 0.02$ ; and **NLMS**:  $\mu = 0.08$  and  $\epsilon = 2^{-F+1}$ ; and  **$\{\text{sign-error LMS}\}^N$** :  $N = 5$  and  $\mu_n = 2^{-5-n}$ .



**Fig. 4.** APA and Algorithm 2 in a stationary scenario— $M = 100$  and  $K = 10$  (simulation performed in MATLAB precision due to the susceptibility of APA to reduced word length). (a) White input. **APA**:  $\mu = 0.3$  and  $\epsilon = 10^{-6}$ ; and **DR- $\{\text{sign-error LMS}\}^N$** <sup>L</sup>:  $L = 10, N = 20$ , and  $\mu_n = 2^{-n}$ . (b) Correlated input. **APA**:  $\mu = 0.06$  and  $\epsilon = 10^{-6}$ ; and **DR- $\{\text{sign-error LMS}\}^N$** <sup>L</sup>:  $L = 30, N = 10$ , and  $\mu_n = 2^{-15-n}$ .

The performance of  $\{\text{sign-error LMS}\}^N$  in a stationary environment was already illustrated in Fig. 1 and shown to match that of the LMS. For nonstationary systems, however, the combination largely outperforms other adaptive algorithms, with a gain of up to 7 dB over the LMS for colored inputs—Fig. 3. Notice that in Fig. 3a the LMS is using the optimal step size  $\mu^o$  for  $\sigma_q^2 = 10^{-4}$  [19, Lemma 21.1]. These results clearly justify the use of this combination as a direct replacement for LMS filters as well as a building block for larger combinations as in Section 4. Indeed, Fig. 4 show that Algorithm 2 is able to match the performance of APA with even lower complexity.

## 6. CONCLUSION

The concept of *combination as a complexity reduction technique* was explored and developed by proposing an incremental combination of sign-error LMS filters. This combination was shown to asymptotically approach the NLMS algorithm and simulations illustrated that it can match—and even outperform—LMS filters with lower complexity. The  $\{\text{sign-error LMS}\}^N$  was then used as building blocks for a larger combination that is able to match the performance of APA. Its behavior in different scenarios was simulated.

## 7. REFERENCES

- [1] A.C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. Signal Process.*, vol. 47[10], pp. 2685–2699, 1999.
- [2] J. Arenas-García, A.R. Figueiras-Vidal, and A.H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54[3], pp. 1078–1090, 2006.
- [3] M.T.M. Silva and V.H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56[7], pp. 3137—3149, 2008.
- [4] N.J. Bershad, J.C.M. Bermudez, and J.-Y. Tournet, "An affine combination of two LMS adaptive filters—transient mean-square analysis," *IEEE Trans. Signal Process.*, vol. 56[5], pp. 1853—1864, 2008.
- [5] C.G. Lopes, E.H. Satorius, P. Estabrook, and A.H. Sayed, "Adaptive carrier tracking for mars to earth communications during entry, descent, and landing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46[4], pp. 1865–1879, 2010.
- [6] L.F.O. Chamon, W.B. Lopes, and C.G. Lopes, "Combination of adaptive filters with coefficients feedback," in *ICASSP*, 2012, pp. 3785–3788.
- [7] W.B. Lopes and C.G. Lopes, "Incremental-cooperative strategies in combination of adaptive filters," in *ICASSP*, 2011, pp. 4132–4135.
- [8] L.F.O. Chamon, H.F. Ferro, and C.G. Lopes, "A data reuse algorithm based on incremental combination of LMS filters," in *Asilomar Conference on Signals, Systems and Computers*, 2012.
- [9] W.B. Lopes and C.G. Lopes, "Incremental combination of RLS and LMS adaptive filters in nonstationary scenarios," in *ICASSP*, 2013, pp. 5676–5680.
- [10] V.H. Nascimento and R.C. de Lamare, "A low-complexity strategy for speeding up the convergence of convex combinations of adaptive filters," in *ICASSP*, 2012, pp. 3553–3556.
- [11] V.H. Nascimento, M.T.M. Silva, and J. Arenas-García, "A low cost implementation strategy for combination of adaptive filters," in *ICASSP*, 2013, pp. 5671—5675.
- [12] M. Ferrer, M. de Diego, A. Gonzalez, and G. Piñero, "Convex combination of affine projection algorithms," in *EUSIPCO*, 2009, pp. 431–435.
- [13] L. Arévalo, R. Sampaio-Neto, J.A. Apolinário, and M.L.R. Campos, "Convex combination of three affine projections adaptive filters," in *ISWCS*, 2013.
- [14] S. Shaffer and C.S. Williams, "Comparison of LMS, alpha LMS, and data reusing LMS algorithms," in *Asilomar Conference on Signals, Syst. and Comput.*, 1983, pp. 260–264.
- [15] B.A. Schnauffer and W.K. Jenkins, "New data-reusing LMS algorithms for improved convergence," in *Asilomar Conf. on Signals, Syst. and Comput.*, 1993, pp. 1584–1588.
- [16] J. Benesty and T. Gänsler, "On data-reuse adaptive algorithms," in *International Workshop on Acoustic Echo and Noise Control*, 2003, pp. 31–34.
- [17] H.-C. Shin, W.-J. Song, and A.H. Sayed, "Mean-square performance of data-reusing adaptive algorithms," *IEEE Signal Process. Lett.*, vol. 12[12], pp. 851–854, 2005.
- [18] L.F.O. Chamon and C.G. Lopes, "On parallel-incremental combinations of LMS filters that outperform the Affine Projection Algorithm," in *Symposium of the Brazilian Telecommunications Society*, 2013.
- [19] A.H. Sayed, *Adaptive filters*, Wiley-IEEE Press, 2008.
- [20] P.S.R. Diniz, *Adaptive filtering: Algorithms and practical implementation*, Springer, 4th edition, 2013.
- [21] M.C. Tsakiris and P.A. Naylor, "Fast exact affine projection algorithm using displacement structure theory," in *International Conference on Digital Signal Processing*, 2009.
- [22] L.F.O. Chamon and C.G. Lopes, "Transient performance of an incremental combination of LMS filters," in *EUSIPCO*, 2013.
- [23] Y.A. Huang and J. Benesty, Eds., *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Springer, 2004.
- [24] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [25] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2007.
- [26] Y.V. Zakharov, G.P. White, and Jie Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Signal Process. Lett.*, vol. 56[7], pp. 3150–3161, 2008.
- [27] Y.V. Zakharov, "Low-complexity implementation of the affine projection algorithm," *IEEE Signal Process. Lett.*, vol. 15, pp. 557–560, 2008.
- [28] R.P. Feynman, "There's plenty of room at the bottom," *Engineering and Science*, vol. 23[5], pp. 22–36, 1960.
- [29] W.A. Sethares and C.R. Johnson, "A comparison of two quantized state adaptive algorithms," vol. 37[1], pp. 138–143, 1989.
- [30] D.J. Allred, H. Yoo, V. Krishnan, W. Huang, and D.V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," *IEEE Trans. Circuits Syst. I*, vol. 52[1], pp. 1327–1337, 2005.
- [31] H.-J. Lo and D.V. Anderson, "A hardware-efficient implementation of the fast affine projection algorithm," in *ICASSP*, 2009, pp. 581–584.
- [32] M. Kac, *Statistical Independence in Probability, Analysis and Number Theory*, John Wiley and Sons, 1959.
- [33] O. Christensen, *An Introduction to Frames and Riesz Bases*, Birkhäuser, 2003.