WILDER BEZERRA LOPES

INCREMENTAL STRATEGIES IN COMBINATION OF ADAPTIVE FILTERS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2012

WILDER BEZERRA LOPES

INCREMENTAL STRATEGIES IN COMBINATION OF ADAPTIVE FILTERS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração:

Sistemas Eletrônicos

Orientador:

Prof. Dr. Cássio G. Lopes

São Paulo
2012

To my family.

ACKNOWLEDGMENTS

To the friend and advisor Prof. Cássio Lopes for his trust in my work, the full-time support, the insightful comments and the many hours spent in fruitful discussions (technical or not!) drinking coffee around the world.

To my parents Wilson and Derci, and my sister Petúnia, who even many kilometers far away are very present and supportive in many ways: my deepest gratitude.

To the friendships I made at the University of São Paulo: Fernando, Murilo, Matheus, Amanda, Manolis, David, Renato, Chamon, Yannick, Humberto. Thank you guys a lot for helping me to keep a piece of mind during the hard times.

To the professors at the University of São Paulo, my gratitude for the knowledge and help offered during the courses.

To the friends spread around the globe, thanks for the support.

To all the good people who have crossed my way since I became a citizen of São Paulo three years ago.

And at last but not least, I would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support to this work.

ABSTRACT

In this work a new strategy for combination of adaptive filters is introduced and studied. Inspired by incremental schemes and cooperative adaptive filtering, the standard convex combination of parallel-independent filters is rearranged into a series-cooperative configuration, while preserving computational complexity. Two new algorithms are derived employing recursive least-squares (RLS) and least-mean-squares (LMS) algorithms as the component filters. In order to assess the performance of the incremental structure, tracking and steady-state mean-square analysis is derived. The analysis is carried out assuming the combiners are fixed, so that the universality of the new structure may be studied decoupled from the supervisor's dynamics. The resulting analytical model shows good agreement with simulation results.

Keywords – Adaptive filtering, combination of adaptive filters, incremental strategies, convex combination.

RESUMO

Neste trabalho uma nova estratégia de combinação de filtros adaptativos é apresentada e estudada. Inspirada por esquemas incrementais e filtragem adaptativa cooperativa, a combinação convexa usual de filtros em paralelo e independentes é reestruturada como uma configuração série-cooperativa, sem aumento da complexidade computacional. Dois novos algoritmos são projetados utilizando *recursive least-squares* (RLS) e *least-mean-squares* (LMS) como subfiltros que compõem a combinação. Para avaliar a performance da estrutura incremental, uma análise de média quadrática é realizada. Esta é feita assumindo que os combinadores têm valores fixos, de forma a permitir o estudo da universalidade da estrutura desacoplada da dinâmica do supervisor. As simulações realizadas mostram uma boa concordância com o modelo teórico obtido.

Palavras-Chave – Filtragem adaptativa, combinação de filtros adaptativos, estratégias incrementais, combinação convexa.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Why and How to Aggregate Adaptive Filters

A common problem arising in the design of adaptive filters (AFs) is the compromise between convergence speed and steady-state error. In several scenarios it is desirable to obtain the best performance (fast converge and low steady-state error), which makes the accurate design of a single filter a difficult task. The issue can be circumvented by simultaneously employing different filters [1–3], which can be properly combined to obtain the desired behavior. In this sense, the combination of AFs has been explored in the literature in order to improve filtering performance.

The combination approach is an instance of universal estimation [1, 4], in which a pool of experts (estimators) is supervised so that the overall estimation performance is at least as good as that of the best expert in the pool.

In the context of adaptive filtering, the experts are AFs, and the supervisor can take different forms [1,5], but usually it is another adaptive system attempting to minimize the overall mean-square error in terms of mixing parameters, that aggregate the experts' estimates.

Usually, independent AFs are arranged in parallel, running simultaneously. Their weight vectors are aggregated directly via a set of combiners. The corre-

sponding global estimate implies a global error, which is minimized in some sense (mean-square) with respect to the combiners [1, 5].

For the parallel-independent case, several combining rules have been studied. The convex combination of two LMS filters with different step sizes had its mean-square analysis devised in [1], showing its universality. Similar study was developed for an affine combination of AFs in [5]. Combinations employing unidirectional conditional transfer of coefficients are studied in [3]. Cyclical feedback of the overall weights to all the component filters is introduced in [6]. Combinations of AFs with different orders and different adaptive algorithms are available in [2, 7–9]. They all address successfully the problem of combining AFs to improve performance.

1.2 Contributions of the Work

The parallel structure has limitations that can prevent the overall combination to appropriately explore what the sub-filters have to offer [10].

Inspired by the incremental-gradient algorithm [11–13], a new combination structure for AFs is proposed. Basically it is a cascade of AFs that cooperate directly with each other. The filters are also combined via a set of mixing parameters so that the overall performance is improved. The case of two filters is studied in detail via simulations and performance analysis.

To cope with the terminology adopted in the literature, we employ interchangeably convex and parallel-independent, as well as series and incremental-cooperative [1, 10].

The text is organized in the following manner. Chapter 2 presents background material that is employed throughout the work. In Chapter 3 a new combination structure is proposed and assessed in stationary scenarios. Simulations using LMS

and RLS filters illustrate how the new topology can circumvent the limitations of the parallel structure. In Chapter 4 the focus is on adverse scenarios (non-stationary). A study on the universality of the incremental structure is carried out. It is showed that the RLS–LMS combination is universal, and considerably outperforms its sub-filters. It is also quite superior to the parallel case under those circumstances. The LMS–LMS combination does not present advantages in the same scenario; further study is required. However, the incremental structure is more robust than the parallel, which also holds for the RLS–LMS case. Finally, Chapter 5 presents concluding remarks and points to future developments.

2 PRELIMINARIES

This chapter introduces background material to support the presentation of the chapters in the sequel.

2.1 Notation

The notation adopted in this text is that of [14], and it is summarized below:

- Boldface letters are used for random quantities and normal font letters for deterministic quantities. For instance, z is a realization (deterministic) of the random variable $\boldsymbol{z}$;

- Capital letters are used for matrices and small letters for *both* vectors and scalars. For instance, Z is a matrix and z is a scalar or vector. From the previous convention, $\boldsymbol{Z}$ is a random matrix and Z is a matrix realization (deterministic). In the same way, $\boldsymbol{z}$ is a vector with random entries and z is a vector realization;

- The time-dependency of a scalar quantity is denoted by parentheses, while subscripts are employed to denote the time-dependency of a vector or a matrix. For instance, $x(i)$ is a time-varying scalar and x_i is a time-varying

vector, while X_i is a time-varying matrix;

- The * symbol is used to represent the conjugate transposition. For real quantities it becomes the usual transposition. This distinction is made because the analysis framework presented in Section 4.2 is valid for both real and complex quantities.

2.2 The System Identification Problem

As it is typical in the adaptive filtering literature, a system identification scenario is used throughout this work to evaluate the performance of the algorithms.

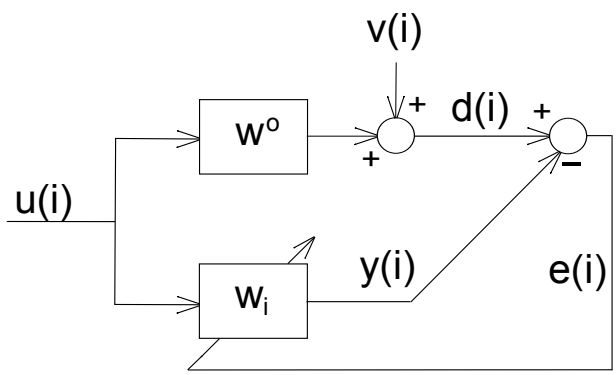


Figure 1: The system identification problem

Consider the scenario depicted in Fig. 1. The problem here is to estimate the weight coefficients of an unknown plant (or system) modelled by an $M \times 1$ vector w^o, which relates the system input-output via

$$d(i) = u_i w^o + v(i), \tag{2.1}$$

where u_i is a 1xM row vector and $v(i)$ represents measurement noise typically modelled as a white Gaussian process with variance σ_v^2 [14]. In the non-stationary case, the plant becomes time-varying, so that $w^o \longrightarrow w_i^o$.

The unknown plant and an adaptive system w_i are simultaneously fed with the same input signal $u(i)$. The output of the unknown system is contaminated by measurement noise $v(i)$ and the adaptive system output $y(i)$ is subtracted from the result. This generates the output estimation error $e(i)$ [14] that is fed back into the adaptive system in order to update its coefficients. The adaptation aims at the minimization of $e(i)$, usually in the mean-square sense, continuing until the adaptive system has converged (steady-state), when w_i is an estimate (or model) of the unknown plant w^o.

2.3 The Incremental Gradient-Descent Algorithm

Adaptive filters are essentially cost-functions minimizers, relying on iterative algorithms such as the gradient-descent

$$w_i = w_{i-1} - \mu[\nabla J(w_{i-1})]^*. \tag{2.2}$$

When the cost-function $J(w)$ has the following property

$$J = \sum_{k=1}^{K} J_k, \tag{2.3}$$

then two different algorithms arise from (2.2): the true-gradient and the incremental-gradient [11, 12]. The true-gradient is given by:

$$\begin{aligned} w_{0,i} &\leftarrow w_{i-1} \\ w_{k,i} &= w_{k-1,i} - \mu_k[\nabla J_k(w_{i-1})]^*, k = 1, ..., K \\ w_i &\leftarrow w_{K,i}, \end{aligned} \tag{2.4}$$

whereas the incremental-gradient is given by

$$\begin{aligned} w_{0,i} &\leftarrow w_{i-1} \\ w_{k,i} &= w_{k-1,i} - \mu_k[\nabla J_k(w_{k-1,i})]^*, k = 1, ..., K \\ w_i &\leftarrow w_{K,i}. \end{aligned} \tag{2.5}$$

Note that in both cases at each iteration i the new w_i is calculated by summing up the contributions from each sub-function J_k, or increments. The difference is *where* the "partial" gradient is evaluated at each sub-iteration k, and $w_{k,i}$ may be interpreted as the kth sub-filter at iteration i.

Each sub-filter operates upon the estimation delivered by the previous sub-filter in the cascade, naturally generating a cooperative structure [13].

The link between gradient-descent and AFs is made when the gradient is approximated from instantaneous data [14–16].

2.4 A Cascade of Cooperative Filters

A well established technique in the literature is to combine a set of independent AFs in parallel via a supervisor that generates a global estimate that should be universal with respect to the component filters. This approach can be posed in general terms as follows. Let $w_{k,i}$ be the kth sub-filter adapted according to

$$w_{k,i} = w_{k,i-1} + \mu_k p_{k,i}, \tag{2.6}$$

where $w_{k,i}$ is the $M \times 1$ weights estimate at iteration i, μ_k is its step size and p_k is selected as

$$p_{k,i} = -B_{k,i}\nabla^* J_k(w_{k,i-1}), \tag{2.7}$$

in which $B_{k,i}$ is any positive definite matrix, and $J_k(w_{k,i-1})$ is the underlying cost function that the sub-filter attempts to minimize.

The global estimate, or the supervisor output, is typically obtained as

$$w_i = \sum_{k=1}^{K} \lambda_k(i) w_{k,i}, \tag{2.8}$$

where the set $\{\lambda_k(i)\}$ of mixing parameters is usually constrained by [1–3, 5]

$$\sum_{k=1}^{K} \lambda_k(i) = 1. \tag{2.9}$$

This gives rise to the affine combination strategy [5]. If the further constraint $\lambda_k(i) \in [0, 1]$ is imposed, then the convex combination strategy is obtained [1–3, 7–9].

In this work an alternative structure to combine filters is proposed, inspired by the incremental gradient-descent algorithm (see Section 2.3). A cascade of directly cooperating filters is rendered, in which each sub-filter's estimate is the initial condition to its successor in the chain, as follows

$$w_{k,i} = w_{k-1,i} + \lambda_k(i)\mu_k p_{k,i},\ k = 1, ..., K. \tag{2.10}$$

Note that in this arrangement any sub-filter that is underperforming can be annihilated by its corresponding combiner, while the cascade remains operational, i.e., the sub-filter simply relays the estimate to the next sub-filter. In this thesis, the $K = 2$ case is studied (see Eq. (2.10)) under the convex constraint. Hence, $\lambda_1(i) = \lambda(i)$ and $\lambda_2(i) = [1 - \lambda(i)]$. Moreover, the scope is limited to the LMS and RLS adaptive rules with $p_{k,i}$ chosen accordingly.

3 INCREMENTAL-COOPERATIVE STRATEGIES

In this chapter the standard parallel structure to combine adaptive filters is revisited, and a new one based on incremental-cooperative strategies is proposed. Simulations in stationary scenarios show the improvement originated from using the new topology to combine LMS and RLS sub-filters.

3.1 Parallel-Independent structure

The common ground for the convex structures currently available in the literature is that the component AFs operate independently from each other and in parallel. As depicted in Fig. 2, the outputs (or the weights) of a fast filter (LMS_1) and an accurate filter (LMS_2) are convexly aggregated via a combining parameter $\lambda(i)$ generating the overall weight vector

$$w_i = \lambda(i) w_{1,i} + [1 - \lambda(i)] w_{2,i} \tag{3.1}$$

where $w_{1,i}$ and $w_{2,i}$ are the individual LMS filters updated independently according to [14]

$$w_{k,i} = w_{k,i-1} + \mu_k u_i^* [d(i) - u_i w_{k,i-1}], \quad k = 1, 2, \tag{3.2}$$

in which u_i is a $1 \times M$ row regressor vector that captures samples of an input (white) signal $u(i)$ with variance σ_u^2 and μ_k is the filter step size. The combining factor $\lambda(i)$ is typically a function of another parameter $a(i)$ which is actually adapted in order to minimize the overall estimation error $e(i) = d(i) - u_i w_{i-1}$ in

the mean-square sense.

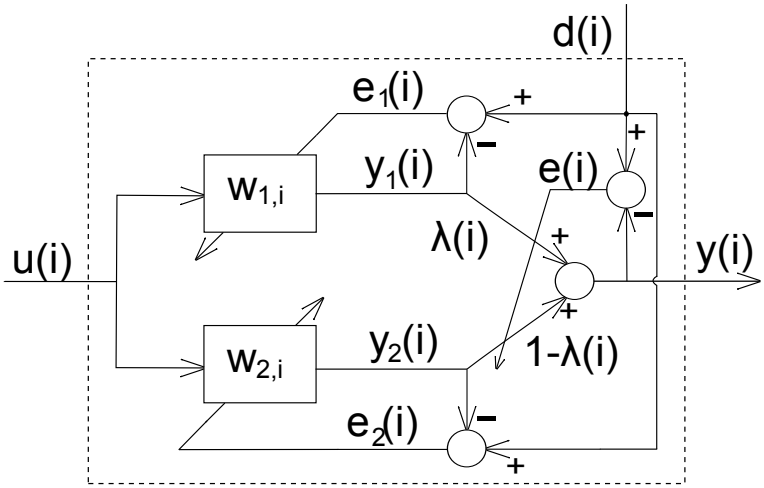


Figure 2: Convex combination of two adaptive transversal filters

$$\lambda(i) = \frac{1}{1 + e^{-a(i)}} \tag{3.3}$$

Function (3.3) is chosen to guarantee convexity, and can be seen as an activation function. Generally, a (stochastic) gradient-descent rule is adopted to tune $a(i)$ [1, 7–9]

$$a(i+1) = a(i) + \mu_a e(i)[y_1(i) - y_2(i)]\lambda(i)[1 - \lambda(i)], \tag{3.4}$$

where $y_k(i) = u_i w_{k,i-1}$, $k = 1, 2$, and μ_a is the step size.

The resulting algorithm is typically known as the convex LMS algorithm, or CLMS (convex) for short, and it is well known to present universal behavior. Fig. 3 depicts the excess mean-square error ($EMSE = E|u_i(w^o - w_{i-1})|^2$) curves for a typical example employing $\mu_1 = 0.07$, $\mu_2 = 0.007$, $\mu_a = 1000$, $\sigma_u^2 = 1$, and $\sigma_v^2 = 10^{-3}$. Note how CLMS is able to track the transient response of the faster filter (μ_1) and reach the steady-state performance of the more accurate (slower) filter (μ_2) [1].

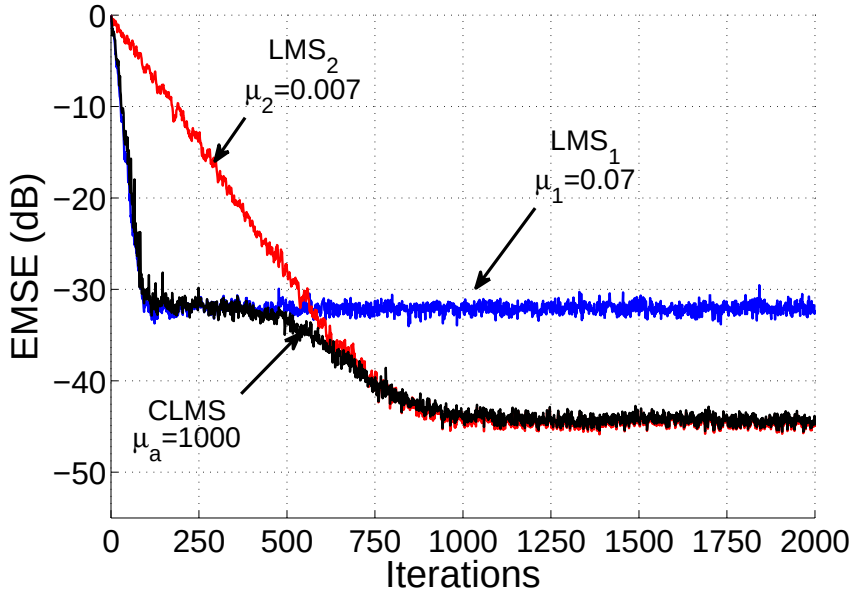


Figure 3: EMSE of the LMS filters and their convex combination averaged over 200 realizations.

3.2 Series-Cooperative Structure

Despite the clear CLMS advantage over the component filters, the transient of the accurate filter is not relevant and the steady-state of the fast filter is wasted: most of the time one filter is numerically annihilated by the combiner $\lambda(i)$. This is caused by the inherent parallel-independent structure, in which the overall system inexorably "awaits" the accurate filter to catch up in order to quickly commute. In stationary environments, $\lambda(i)$ can be interpreted as a switching mechanism. The great advantage offered by such structure is the simple design of the combiner.

In order to compensate for the aforementioned effect and anticipate the switching time, in the literature ad-hoc weight transfers ($w_1 \rightarrow w_2$) are conditionally performed [3], and further control mechanisms are required, since the accurate filter may be contaminated with the higher gradient noise arising from the fast filter. [1]

[1] Another possibility has been recently proposed: cyclic feedback of coefficients [6].

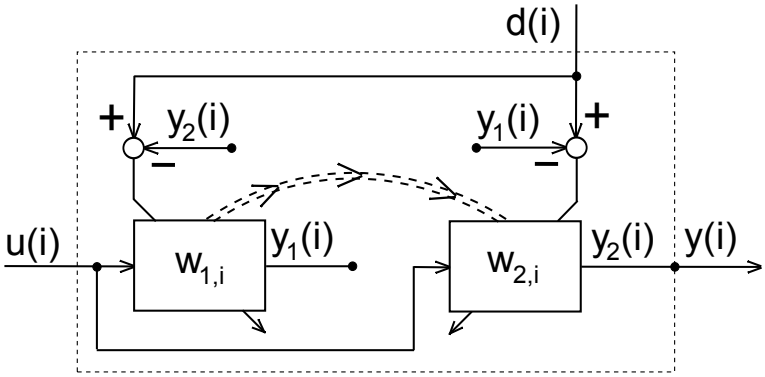


Figure 4: Proposed topology

3.2.1 A switching algorithm

The ad-hoc weight transfer procedure may be formally motivated and naturally implemented, without resorting to control mechanisms, inspired by incremental and cooperative structures described in Section 2.3 (INC-COOP) [17]. Topologically, the filters are rearranged in series, or incrementally – see Figs. 2 and 4, and $\lambda(i)$ now continuously and progressively transfer the weights. The resulting algorithm, INC-COOP$_1$, is quite simple and summarized as follows:

$$\begin{aligned} w_{1,i} &= w_{i-1} + \mu_1 \lambda(i) u_i^* [d(i) - u_i w_{i-1}] \\ w_{2,i} &= w_{1,i} + \mu_2 [1 - \lambda(i)] u_i^* [d(i) - u_i w_{1,i}] \\ w_i &\leftarrow w_{2,i} \end{aligned} \tag{3.5}$$

In Fig. 4, $\lambda(i)$ and $[1-\lambda(i)]$ are located inside the blocks $w_{1,i}$ and $w_{2,i}$, respectively. The incremental step is indicated by the dashed arrow. Note that the filters are no longer independent: they explicitly cooperate, balanced by $\lambda(i)$. Furthermore, the incremental arrangement allows $\lambda(i)$ to simultaneously play the role of a combiner while decreasing the net step size at the same time.

The way the recursion (3.5) is constructed, i.e., with $\lambda(i)$ and $[1 - \lambda(i)]$ multiplying only the adaptation term, has been motivated by the idea that, in case one filter is underperforming, it is possible to turn off its adaptation term by simply setting $\lambda(i) = 0$ or $\lambda(i) = 1$, turning it into a relay node. This procedure

avoids the necessity of implementing condition tests ("if-then") to turn on or off the entire component filter [3]. The serial-incremental cooperation naturally transfers the filter's coefficients along the cascade.

3.2.2 Enhancing performance: simultaneous operation

The potential of the series-cooperative structure can be further explored if simultaneous operation is implemented. For that, $\lambda(i)$ can be more efficiently used as follows:

$$\begin{aligned} w_{1,i} &= w_{i-1} + \mu_1 \lambda(i) u_i^* [d(i) - u_i w_{i-1}] \\ w_{2,i} &= w_{1,i} + \gamma \big[\mu_1 \lambda(i) + [1 - \lambda(i)]\mu_2\big] u_i^* [d(i) - u_i w_{1,i}] \\ w_i &\leftarrow w_{2,i} \ . \end{aligned} \tag{3.6}$$

This is the INC-COOP$_2$ algorithm, where $\gamma \in (0, 1]$ is a step-size contracting factor introduced to improve steady-state performance while keeping the same transient behavior. This is only possible in the series-cooperative arrangement; nevertheless, $\lambda(i)$ has to be designed so that the AFs operate simultaneously.

Partial results have shown that INC-COOP$_1$ (see Eq. (3.5))employing $\mu_2 \longrightarrow \gamma\mu_2$, in some cases, may present a comparable performance to INC-COOP$_2$. Additional research is required.

3.3 Design of the mixing parameter

As a matter of fact, algorithms (3.5) and (3.6) may be regarded as a resource reallocation of the original CLMS: the same AFs are employed (same complexity), using the same combiner and the same signals. Therefore, a direct comparison is fair. The challenge is to design $\lambda(i)$ properly, since the filters are explicitly impacting each other via the incremental procedure.

Initially, a deterministic design for $\lambda(i)$ is introduced to test the new algo-

rithms; in the sequel, a simple though effective way to design the mixing parameter automatically is presented.

3.3.1 Deterministic design

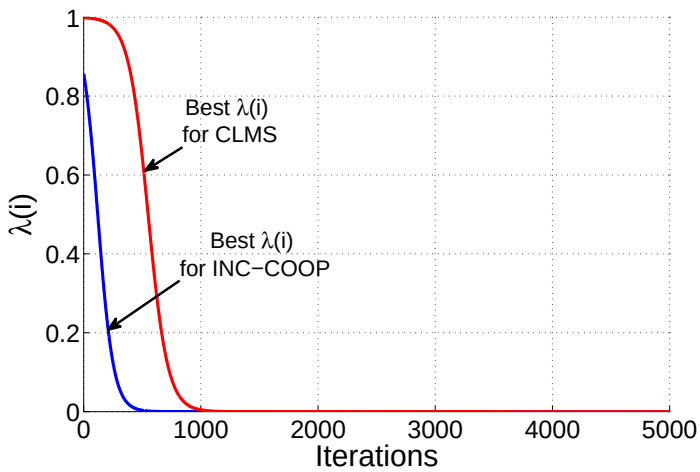
Due to the "switching nature" attributed to $\lambda(i)$, it can be chosen similarly to the parallel case

$$\lambda(i) = \frac{1}{1 + e^{s\cdot(i-n)}} \tag{3.7}$$

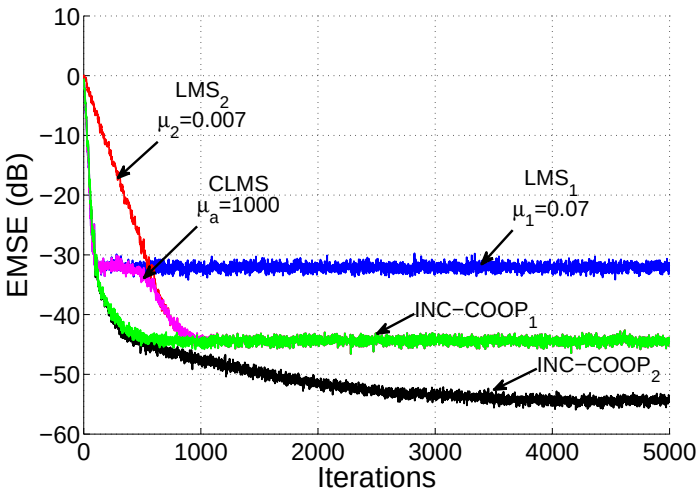
where now n is the *activation instant* and s controls the curve *smoothness*. The parameters $[s, n]$ have been tuned carefully to extract the best performance from the new INC-COOP algorithms and the CLMS algorithm, yielding a meaningful comparison.

Consider the system identification scenario and let $w^o = \frac{1}{\sqrt{10}}[1, 1, \cdots, 1]$, ($\|w^o\| = 1$), and $M = 10$, with signal variances $\sigma_u^2 = 1$ and $\sigma_v^2 = 10^{-3}$. All the LMS component filters used in the combinations (CLMS, INC-COOP$_1$ and INC-COOP$_2$) have step sizes $\mu_1 = 0.07$ and $\mu_2 = 0.007$. For the CLMS, $\mu_a = 1000$. Fig. 5 depicts $\lambda(i)$ (a) and EMSE (b) employed in the pilot experiment. For CLMS we have $[s = 0.012, n = 550]$ and for both INC-COOPs $[s = 0.015, n = 120]$. INC-COOP$_2$ uses $\gamma = 0.1$.

Note in Fig.5(b) the superior performance of the INC-COOP algorithms. INC-COOP$_1$ is able to promptly switch filters earlier, avoiding the stagnation experienced by the CLMS algorithm (which awaits the crossing point), and reproducing the steady-state performance of the accurate filter (LMS$_2$). Furthermore, the simultaneous operation imposed by INC-COOP$_2$ yields faster convergence and *smaller* error for the *same* $\lambda(i)$. Note that in *non-stationary* environments (currently under study), all algorithms are valid candidates.



(a)



(b)

Figure 5: (a) Time evolution of the deterministic $\lambda(i)$. (b) The correspondent EMSE averaged over 200 realizations.

3.3.2 A simple design for the mixing parameter

A simple rule for parameter adjustment in adaptive filtering is to low-pass filter a quantity $q(i)$ that captures the learning status and feed it back into the adaptive process, namely

$$a(i) = \alpha a(i-1) + \beta q(i) \tag{3.8}$$

in which $0 < \alpha < 1$. Such approach has been successfully adopted across several areas in adaptive filtering, such as step-size design [18], regularization control [19] and robust filtering [20]. Experience across the several fields in adaptive filtering aforementioned shows that $0.95 < \alpha < 0.99$ renders a good learning evolution for a wide Signal-to-Noise Ratio (SNR) range. For a quick design, one can assign $\beta = (1 - \alpha)$. Depending on the metric selected for $q(i)$, $\beta < (1 - \alpha)$ compensates for low SNR, say $\beta = 0.1 \cdot (1 - \alpha)$. Detailed analysis is required to show the impact of such parameters on system performance (future work).

Here $q(i)$ is selected equal to $e^2(i)$, where $e(i) = d(i) - u_i w_{2,i-1}$, to train the INC-COOP algorithms. Since $e^2(i)$ approaches zero, a bias is required in $\lambda(i)$ for the INC-COOP case

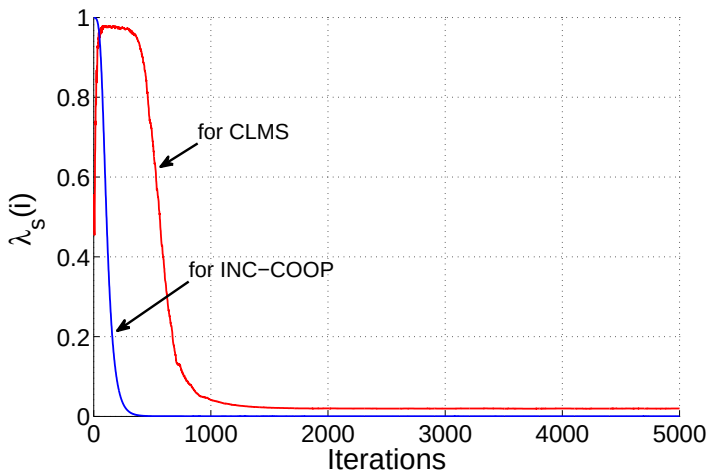
$$\lambda_s(i) = \frac{2}{1 + e^{-a(i)}} - 1, \tag{3.9}$$

in which the subscript 's' stands for "simple design", so that the full excursion $\lambda_s \in [0, 1]$ is guaranteed.

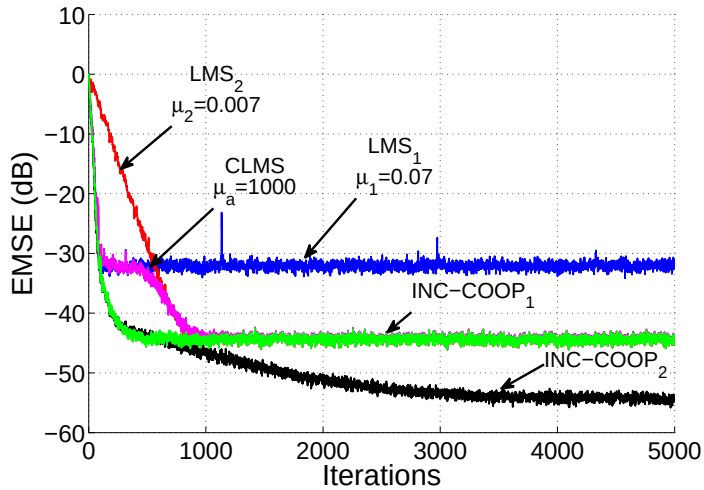
It is noteworthy to mention that a normalized gradient-descent rule has been designed and shows promising results. Nevertheless, as they are preliminary tests and need further study, their simulations are not covered here.

3.4 Simulations

The low-pass filter (3.8) and the $\lambda_s(i)$ (3.9) are implemented using $\alpha = 0.98$ and $a(-1) = 10$ in all the simulations to evaluate the performance of the INC-COOP algorithms as compared to the CLMS algorithm. Simulations are carried out in the system identification scenario previously described.



(a)



(b)

Figure 6: (a) Time evolution of $\lambda_s(i)$. (b) EMSEs of the combinations averaged over 200 realizations.

3.4.1 High SNR

The algorithms have been tested at $\sigma_v^2 = 10^{-3}$ (SNR = 30dB). Fig. 6 shows the EMSE curves of the INC-COOP algorithms and CLMS, averaged over 200 realizations. Both proposed algorithms present a transient response at least as fast as CLMS, allied with the *same* steady-state EMSE as that of the accurate LMS component filter (μ_2). In particular, INC-COOP$_2$ goes beyond and achieves a steady-state EMSE level approximately 10 dB *lower* than CLMS.

Due to limitations of the parallel-independent structure, the weight-transfer procedure employed by [3] compensates only partially the stagnation effect. Fig. 7 shows the superior performance of the INC-COOPs algorithms in comparison to CLMS ($\mu_a = 1000$) and transfer of coefficients ($\mu_a = 500$) for the same scenario.

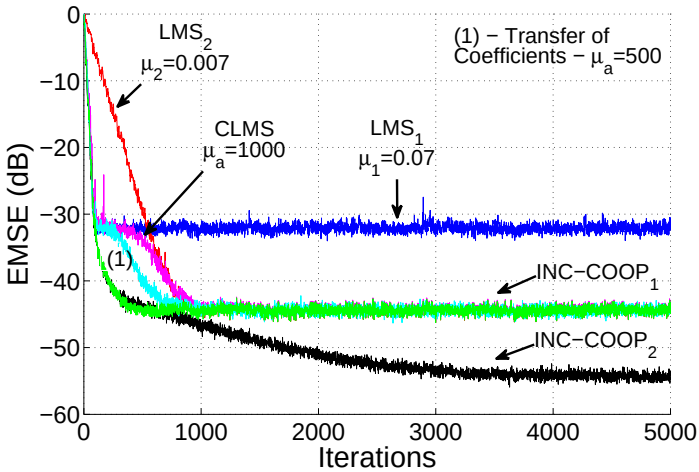


Figure 7: Performance comparison of INC-COOPs, CLMS and transfer of coefficients averaged over 200 realizations.

As $\lambda_s(i)$ is close to zero, the net step size of INC-COOP$_2$ is $\gamma\mu_2$ – see (3.6). Comparatively, the net step size of CLMS equals μ_2. Thus, CLMS was redesigned to have a net step size equal to $\gamma\mu_2$ in order to compare its performance to INC-COOP$_2$. As can be seen in Fig. 8, the redesigned CLMS reaches the same steady-state EMSE of INC-COOP$_2$. However, it takes much longer to converge.

3.4.2 Low SNR

Additional simulations have been performed at SNR $= \{10, 5, 3\}$ dB. With no change in the combinations parameters, it can be seen in Fig. 9 that the INC-COOP algorithms perform better than the CLMS. Moreover, INC-COOP$_2$ presents the best performance among the three combinations, significantly faster than the CLMS. Note also that INC-COOP algorithms are less susceptible to noise than CLMS and present lower variance. $\beta = (1 - \alpha)$ was used for SNR =

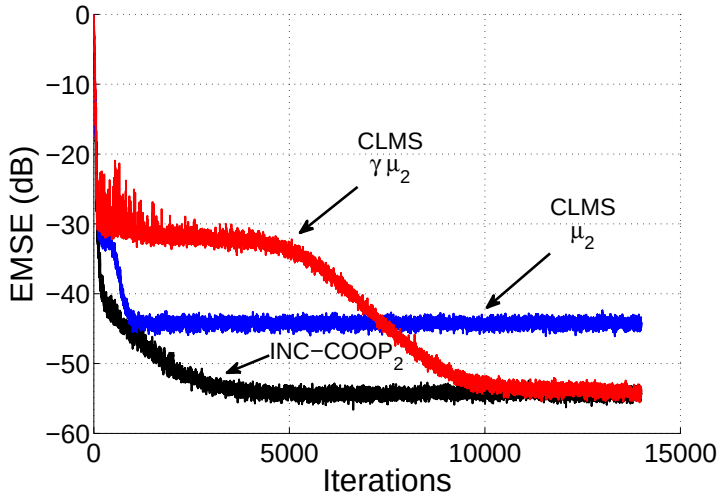


Figure 8: EMSE of INC-COOP$_2$, CLMS ($\mu = \mu_2$) and CLMS ($\mu = \gamma\mu_2$).

10 dB. For SNR $= \{5, 3\}$ dB, $\beta = 0.1 \cdot (1 - \alpha)$.

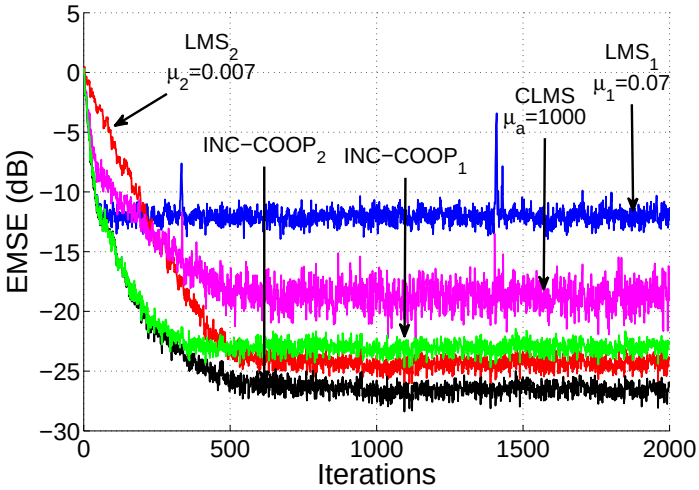
3.5 Combination with different types of filters - RLS–LMS

Consider the RLS rule

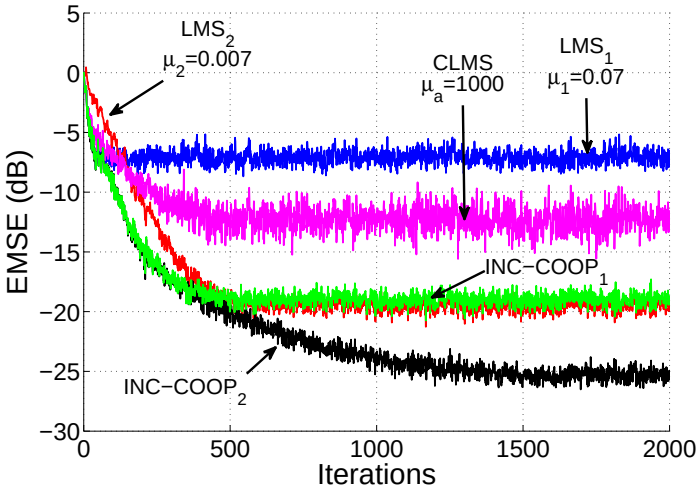
$$\begin{aligned} w_1 &= w_{i-1} + \lambda P_i u_i^* [d(i) - u_i w_{i-1}] \\ P_i &= \eta^{-1} \left[P_{i-1} - \frac{\eta^{-1} P_{i-1} u_i^* u_i P_{i-1}}{1 + \eta^{-1} u_i P_{i-1} u_i^*} \right], \; P_{-1} = \epsilon^{-1} I, \end{aligned} \tag{3.10}$$

where η is the forgetting factor and ϵ the regularization term.

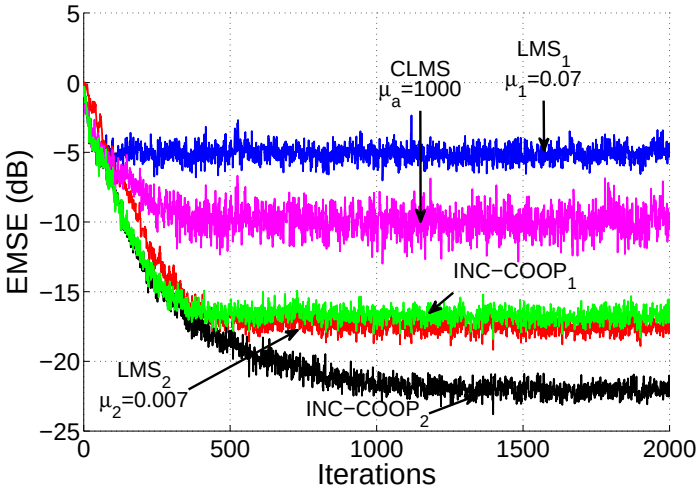
Back in the high SNR scenario and using INC-COOP$_1$ as a chain of a RLS followed by an LMS filter, it is showed in Fig. 10 that the adaptation halt is also circumvented. Both structures (convex and series) have the same RLS (forgetting factor η=0.95) and LMS ($\mu = 0.007$) as component filters [2]. The mixing parameter is designed according to Subsection 3.3.2.



(a)

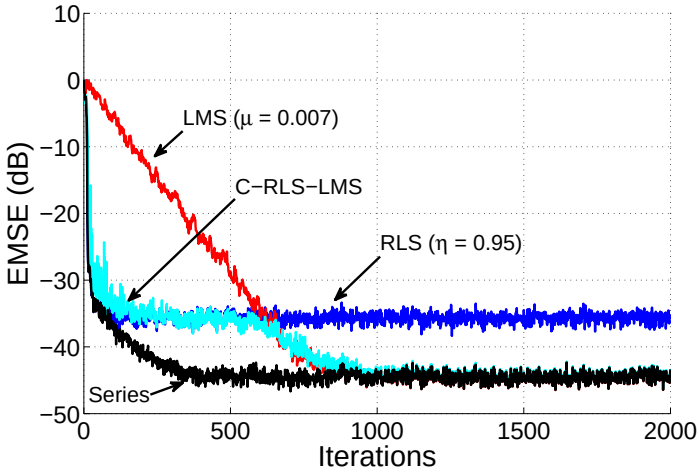


(b)

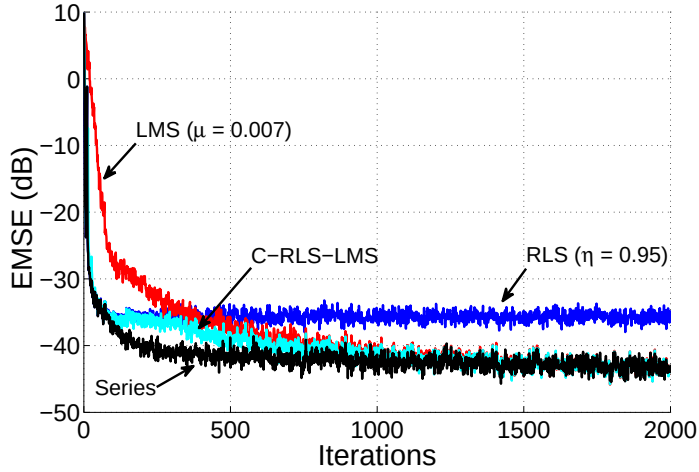


(c)

Figure 9: EMSE of the combinations for SNR = 10dB (a), SNR = 5dB (b), SNR = 3dB (c). The curves are averaged over 200 realizations.



(a)



(b)

Figure 10: Series and Convex combination of RLS and LMS filters in stationary scenario for (a) white - convex step size $\mu_a = 1700$, and (b) colored noise - convex step size $\mu_a = 1000$. The curves are averaged over 100 realizations.

4 INCREMENTAL COMBINATION IN NON-STATIONARY SCENARIOS

The design of the supervisor in the general case for the incremental structure can be rather challenging, since the component filters affect each other via the cooperative scheme throughout the adaptive process.

It is then crucial to assess whether the new structure may achieve universality. In other words, is it possible to design a λ so that the overall filter can outperform its component filters? To answer this, in this chapter a study is conducted on the mean-square performance of the new technique in terms of the (fixed) parameter λ.

The steady-state model places a performance benchmark for future adaptive designs for λ.

The derivations rely on Energy Conservation Relation (ECR) arguments [14], resulting in an analytical mean-square error model that describes the structure in steady-state for both stationary and non-stationary scenarios. The framework covers the $K = 2$ case across several adaptive rules. However, only the LMS–LMS and RLS–LMS arrangements were studied in detail.

In adverse scenarios, the RLS–LMS structure is shown to be very promising, by simulations and theory. The LMS–LMS combination, on the other hand, does not perform similarly. Both arrangements, however, seem to be more robust than their convex parallel counterpart.

4.1 Motivating Examples

Extensive parametric simulations were conducted for both LMS–LMS and RLS–LMS arrangements covering the range $\lambda \in [0, 1]$. In the examples that follow, the plant becomes time-varying and evolves according to

$$w_i^o = w_{i-1}^o + q_i \ . \tag{4.1}$$

In the literature, q_i is generated as the realization of a zero-mean independent and identically distributed (i.i.d.) process with covariance matrix $E\boldsymbol{q}_i\boldsymbol{q}_i^* \triangleq Q = \sigma_q^2 I$.

It is important to emphasize that only adverse scenarios are considered, with highly correlated data and a considerably large degree of non-stationarity imposed on the adaptive system (which is captured by σ_q^2), when compared with the values in the literature [2, 3, 5].

The step sizes for the LMS–LMS are normalized via $\mu_o \in [0, 1]$ and $r_o \in [0, 1]$

$$\begin{aligned} \mu_1 &= \frac{\mu_o}{M\sigma_u^2} \\ \mu_2 &= \frac{r_o\mu_o}{M\sigma_u^2} \ , \text{ in which } r_o \triangleq \frac{\mu_2}{\mu_1} \in [0, 1], \end{aligned} \tag{4.2}$$

The RLS–LMS case employs $\mu = \frac{\mu_o}{M\sigma_u^2}$ only, and the forgetting factor is $\eta = 0.98$. In the extremes of the interval $\lambda \in [0, 1]$ (in which only one filter is selected) both parallel and series structures perform similarly. However, along the interval, there is a range where the incremental structure may considerably outperform the convex parallel structure.

Fig. 11 depicts the learning curves for $\lambda = 0.4$. The simulation was generated with $\sigma_v^2 = 10^{-3}$, $\sigma_q^2 = 10^{-4}$, and with regressors formed by a first-order auto-regressive process with transfer function $\sqrt{1-b^2}/(1-bz^{-1})$, $b = 0.98$, and $\sigma_u^2 = 1$. Note how the incremental arrangement achieves universality when the input data is highly-correlated and the plant is varying rapidly. Moreover, the convex

parallel combination is clearly outperformed.

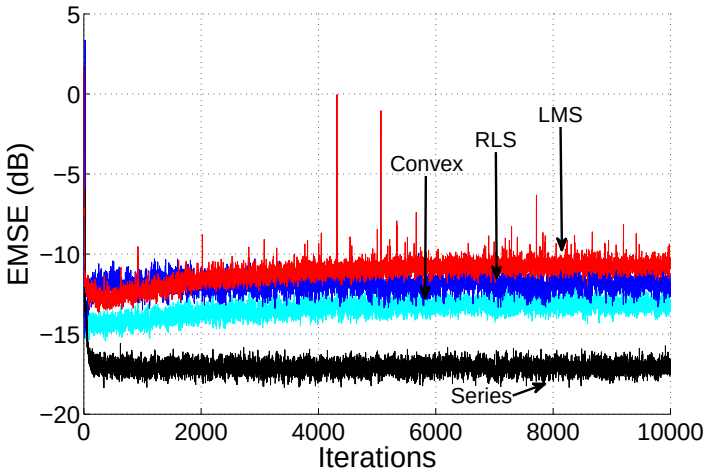


Figure 11: Universality achieved by the series combination of RLS and LMS filters for $\mu_o = 0.3$ and $\lambda = 0.4$ (500 realizations).

In the LMS–LMS case, on the other hand, there is no evidence that the series combination can perform better than its individual filters in the adverse scenario considered. This is shown more thoroughly in Section 4.4.

4.2 Tracking Analysis

The analysis derived in this section encompasses both stationary and non-stationary scenarios.

For the sake of generality, we consider the following incremental combination

$$\begin{aligned} w_1 &= w_{i-1} + \lambda H_{1,i} u_i^* [d(i) - u_i w_{i-1}] \\ w_i &= w_1 + (1-\lambda) H_{2,i} u_i^* [d(i) - u_i w_1]. \end{aligned} \tag{4.3}$$

where $H_{k,i}$, $k = 1, 2$, are chosen according to the desired adaptive rules.

Note that Eq. (4.3) is comprised by deterministic quantities, and this is what is implemented in practice. However, the theoretical study of the structure mean-square error assumes that the signals involved are random in nature, following known statistical distributions. Therefore, Eq. (4.3) becomes a stochastic equa-

tion

$$\begin{aligned} \boldsymbol{w}_1 &= \boldsymbol{w}_{i-1} + \lambda \boldsymbol{H}_{1,i} \boldsymbol{u}_i^* [\boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1}] \\ \boldsymbol{w}_i &= \boldsymbol{w}_1 + (1-\lambda) \boldsymbol{H}_{2,i} \boldsymbol{u}_i^* [\boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_1], \end{aligned} \tag{4.4}$$

The goal is to derive an expression for the mean-square error (MSE) in steady-state of any filter that can be written in form (4.4), via energy conservation relations [14]. The MSE is defined as

$$MSE = \xi \triangleq \lim_{i \to \infty} E|\boldsymbol{e}(i)|^2. \tag{4.5}$$

where

$$\boldsymbol{e}(i) = \boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1} \tag{4.6}$$

For that matter, since adaptive filters are non-linear, time-varying, and stochastic, it is necessary to adopt a set of simplifying assumptions collected into an extended data model [14]:

(1) There exits a vector $\boldsymbol{w}_i^o$ such that $\boldsymbol{d}(i) = \boldsymbol{u}_i \boldsymbol{w}_i^o + \boldsymbol{v}(i)$;

(2) The weight vector varies according to $\boldsymbol{w}_i^o = \boldsymbol{w}_{i-1}^o + \boldsymbol{q}_i$ (random-walk model);

(3) The noise sequence $\{\boldsymbol{v}(i)\}$ is i.i.d. with constant variance $\sigma_v^2 = E|\boldsymbol{v}(i)|^2$;

(4) The noise sequence $\{\boldsymbol{v}(i)\}$ is independent of $\boldsymbol{u}_i$ for all i, j ;

(5) The sequence $\{\boldsymbol{q}_i\}$ has covariance matrix $Q \triangleq E\boldsymbol{q}_i \boldsymbol{q}_i^*$ and is independent of $\{\boldsymbol{v}(i), \boldsymbol{u}_j\}$ for all i, j ;

(6) The initial conditions $\{\boldsymbol{w}_{-1}, \boldsymbol{w}_{-1}^o\}$ are independent of all $\{\boldsymbol{d}(j), \boldsymbol{u}_j, \boldsymbol{v}(j), \boldsymbol{q}_j\}$;

(7) The regressor covariance matrix is denoted by $R_u = E\boldsymbol{u}_i^* \boldsymbol{u}_i = Tr(R_u) > 0$;

(8) The random variables $\{\boldsymbol{d}(i), \boldsymbol{v}(i), \boldsymbol{u}_i, \boldsymbol{q}_i\}$ are zero mean;

(9) The weight vector $\boldsymbol{w}_i^o$ has constant mean w^o .

(4.7)

The ECR technique is an energy balance obtained from the following error quantities

$$\begin{aligned}
\tilde{\boldsymbol{w}}_{i-1} &\triangleq (\boldsymbol{w}_{i-1}^o - \boldsymbol{w}_{i-1}) \text{ weight-error vector} \\
\boldsymbol{e}_a(i) &= \boldsymbol{u}_i \tilde{\boldsymbol{w}}_{i-1} \text{ } a \text{ } priori \text{ estimation error} \\
\boldsymbol{e}_p(i) &= \boldsymbol{u}_i \tilde{\boldsymbol{w}}_i \text{ } a \text{ } posteriori \text{ estimation error}
\end{aligned} \tag{4.8}$$

together with the adaptive filter's recursion. The resulting energy equation leads to a variance relation from which the MSE and the EMSE can be derived; for details see [14].

Merging the two equations in (4.4) results

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \boldsymbol{H}_i \boldsymbol{u}_i^* \boldsymbol{e}(i), \tag{4.9}$$

where

$$\boldsymbol{H}_i \quad = [\lambda \boldsymbol{H}_{1,i} + (1-\lambda)\boldsymbol{H}_{2,i}(1 - \lambda \|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_{1,i}})] \tag{4.10}$$

and $\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_{1,i}} \triangleq \boldsymbol{u}_i \boldsymbol{H}_i \boldsymbol{u}_i^*$ is the weighted norm of $\boldsymbol{u}_i$. In the general case, $\|x\|^2_\Sigma = x\Sigma x^*$.

Subtracting (4.9) from $\boldsymbol{w}_i^o$ gives

$$(\boldsymbol{w}_i^o - \boldsymbol{w}_i) = (\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}) - \boldsymbol{H}_i \boldsymbol{u}_i^* \boldsymbol{e}(i). \tag{4.11}$$

Multiplying (4.11) from the left by $\boldsymbol{u}_i$ results in

$$\boldsymbol{e}_p(i) = \boldsymbol{e}_a(i) - \|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i} \boldsymbol{e}(i). \tag{4.12}$$

Substituting (4.12) in (4.11) gives

$$(\boldsymbol{w}_i^o - \boldsymbol{w}_i) + \frac{\boldsymbol{H}_i \boldsymbol{u}_i^*}{\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}} \boldsymbol{e}_a(i) = (\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}) + \frac{\boldsymbol{H}_i \boldsymbol{u}_i^*}{\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}} \boldsymbol{e}_p(i). \tag{4.13}$$

Using $\boldsymbol{H}_i^{-1}$ as weighting matrix and equating the squared weighted norms of

(4.13) results in

$$\|\boldsymbol{w}_i^o - \boldsymbol{w}_i\|^2_{\boldsymbol{H}_i^{-1}} + \overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = \|\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} + \overline{\mu}(i)|\boldsymbol{e}_p(i)|^2, \tag{4.14}$$

where $\overline{\mu}(i) \triangleq (\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i})^\dagger = \dfrac{1}{\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}}$, if $\boldsymbol{u}_i \neq 0$ or equals zero otherwise, with † representing the pseudoinverse operator [21].

Taking the expectations of (4.14) gives

$$E\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = E\|\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_p(i)|^2 \tag{4.15}$$

Using the random-walk model into the first term of the right-hand side of (4.15) yields

$$\begin{aligned} &E\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = \\ &E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} + E\|\boldsymbol{q}_i\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_p(i)|^2 \end{aligned} \tag{4.16}$$

In steady-state ($i \to \infty$), it is reasonable to assume

$$\tilde{\boldsymbol{w}}_{i-1} \text{ is independent of } \boldsymbol{u}_i, \tag{4.17}$$

so that

$$\begin{aligned} &E\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{H}_i^{-1}} = E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} \\ &E(\boldsymbol{H}_i^{-1}) \approx [E(\boldsymbol{H}_i)]^{-1} \triangleq H^{-1} \\ &E\boldsymbol{H}_i \triangleq H. \end{aligned} \tag{4.18}$$

Applying (4.18) in (4.16) results in the *variance relation*:

$$E\|\boldsymbol{u}_i\|^2_H|\boldsymbol{e}(i)|^2 + E\|\boldsymbol{q}_i\|^2_{H^{-1}} = 2R_e\{E\boldsymbol{e}_a^*(i)\boldsymbol{e}(i)\} \tag{4.19}$$

The separation principle states that in steady-state $\|\boldsymbol{u}_i\|^2_H$ is independent of $\boldsymbol{e}_a(i)$ [14]. From the data model one has

$$\begin{aligned} \boldsymbol{e}(i) &= \boldsymbol{d}(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1} \\ &= \boldsymbol{e}_a(i) + \boldsymbol{v}(i) \end{aligned} \tag{4.20}$$

which substituting into (4.19) leads to

$$2E|\boldsymbol{e}_a(i)|^2 = \sigma_v^2 E\|\boldsymbol{u}_i\|_H^2 + E\|\boldsymbol{u}_i\|_H^2 E|\boldsymbol{e}_a(i)|^2 + E\|\boldsymbol{q}_i\|_{H^{-1}}^2, \tag{4.21}$$

where $E|\boldsymbol{e}_a(i)|^2$ is the very definition of the Excess-Mean-Square Error (EMSE)

$$\text{EMSE} = \zeta \triangleq E|\boldsymbol{e}_a(i)|^2. \tag{4.22}$$

Plugging $E\|\boldsymbol{u}_i\|_H^2 = Tr(R_u H)$ and $E\|\boldsymbol{q}_i\|_{H^{-1}}^2 = Tr(QH^{-1})$ into (4.19) results in the steady-state EMSE, in non-stationary scenarios, for the incremental combination of two filters described by (4.4):

$$\zeta = \frac{\sigma_v^2 Tr(R_u H) + Tr(QH^{-1})}{2 - Tr(R_u H)}. \tag{4.23}$$

The MSE is obtained as follows (see Eq. (4.20))

$$\xi = \zeta + \sigma_v^2. \tag{4.24}$$

Eq. (4.23) holds under the data model (4.2), and requires the calculation of the data moments R_u and H (which is filter dependent).

4.3 RLS–LMS combination

For this combination,

$$\begin{aligned} \boldsymbol{H}_{1,i} &= \boldsymbol{P}_i \\ \boldsymbol{H}_{2,i} &= \mu I \end{aligned} \tag{4.25}$$

where $\boldsymbol{P}_i$ is obtained by the recursion [14],

$$\boldsymbol{P}_i = \eta^{-1}\left[\boldsymbol{P}_{i-1} - \frac{\eta^{-1}\boldsymbol{P}_{i-1}\boldsymbol{u}_i^*\boldsymbol{u}_i\boldsymbol{P}_{i-1}}{1+\eta^{-1}\boldsymbol{u}_i\boldsymbol{P}_{i-1}\boldsymbol{u}_i^*}\right], \boldsymbol{P}_{-1} = \epsilon^{-1}I, \tag{4.26}$$

with ϵ being the regularization parameter, I the MxM identity matrix, recalling that $\mu = \frac{\mu_o}{M\sigma_u^2}$, $\mu_o \in [0,1]$. Thus,

$$\boldsymbol{H}_i = [\lambda \boldsymbol{P}_i + (1-\lambda)\mu(1-\lambda\|\boldsymbol{u}_i\|^2_{\boldsymbol{P}_i})I]. \tag{4.27}$$

To calculate the EMSE and MSE of the structure, one needs to determine $E\boldsymbol{H}_i$ from (4.27). This is accomplished by using (4.18) and employing specific conditions for RLS [14],

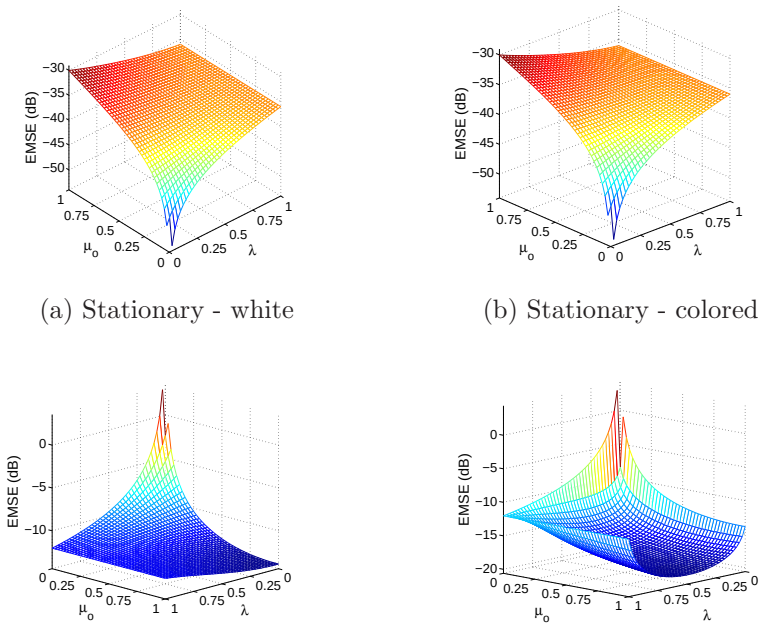
$$\begin{aligned}
&\lim_{i\to\infty} E(\boldsymbol{P}_i^{-1}) = \frac{R_u}{1-\eta} \triangleq P^{-1} \\
&E\boldsymbol{P}_i \approx [E(\boldsymbol{P}_i^{-1})]^{-1} = (1-\eta)R_u^{-1} = P \\
&\|\boldsymbol{u}_i\|^2_{\boldsymbol{P}i} \text{ is independent of } \boldsymbol{e}_a(i) \\
&E\|\boldsymbol{u}_i\|^2_{\boldsymbol{P}i} \approx E\|\boldsymbol{u}_i\|^2_P = Tr(R_uP) = (1-\eta)Tr(R_uR_u^{-1}) = (1-\eta)M,
\end{aligned} \tag{4.28}$$

resulting in

$$H = \{\lambda(1-\eta)R_u^{-1} + \mu(1-\lambda)[1-\lambda(1-\eta)M]I\}. \tag{4.29}$$

Substituting (4.29) into (4.23), together with (4.24) returns the EMSE (MSE) of the RLS–LMS incremental structure. It is a function of several parameters. Here η and M are fixed, and for a given R_u and Q the ζ behavior is explored in terms of μ and λ.

Fig. 12 depicts an example of the EMSE surface in four cases, varying μ_o and λ, in both stationary and non-stationary scenarios. Regressors are either white Gaussian or highly correlated, obtained from a first-order auto-regressive process with transfer function $\sqrt{1-b^2}/(1-bz^{-1})$, $b = 0.98$. The signals variances are $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, and $\sigma_q^2 = 10^{-4}$. The RLS forgetting factor is $\eta = 0.98$, regularization factor $\epsilon = 10^{-5}$ and the system order is $M = 20$. Figs. 12 a), b) and c) do not show improvement from the combination. On the other hand, it



(a) Stationary - white

(b) Stationary - colored

(c) Non-stationary - white

(d) Non-stationary - colored

Figure 12: RLS–LMS - EMSE theoretical surfaces for variable μ_o and λ in both stationary and non-stationary scenarios, with white and correlated input data.

is noticeable from the convex shape of the correlated data surface 12d that there are optimum pairs (μ_o, λ) that attain the minimum EMSE. This shows that the structure is universal and considerably outperforms the component filters in non-stationary and highly correlated scenario.

4.3.1 Simulations

In the next three examples, simulations are presented comparing the parallel with the series (incremental) structure. The EMSE curves are generated for $\mu_o = 0.2, 0.4, 0.9$ and for $\lambda \in [0, 1]$. The non-stationary plant is initialized with $\boldsymbol{w}^o_{-1}$ drawn from a unit variance white Gaussian process, and follows the random-walk model with $\sigma_q^2 = 10^{-4}$. Regressors are correlated and generated according to Section 4.1, namely $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\eta = 0.98$, $\epsilon = 10^{-5}$.

The first example is run with $\mu_o = 0.2$. Fig. 13 is an abacus comparing in steady-state the parallel and the series structure (theory and simulations).

The simulated curves were generated as the average of the last 300 points after convergence. Series combination clearly outperforms the convex as well as its component filters. This figure can be regarded as a slice of 12d. Fig. 14 presents the learning curves for $\lambda = 0.4$, with an improvement of nearly $4dB$ over the convex-parallel. Note how the theoretical and the simulated curves match.

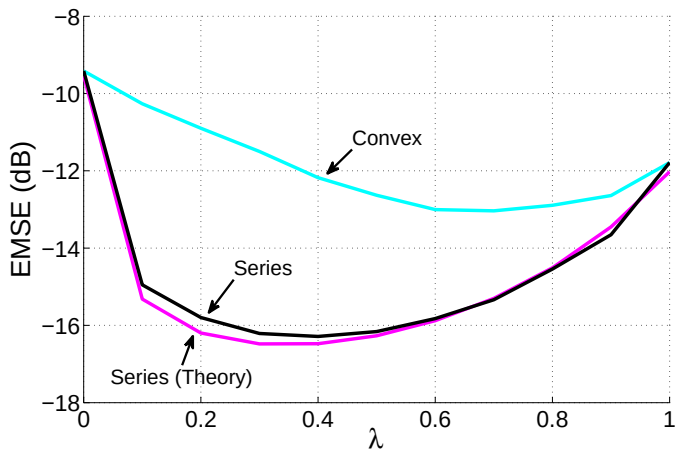


Figure 13: EMSE x λ for both combinations (RLS–LMS) with $\mu_o = 0.2$ and $\sigma_q^2 = 10^{-4}$.

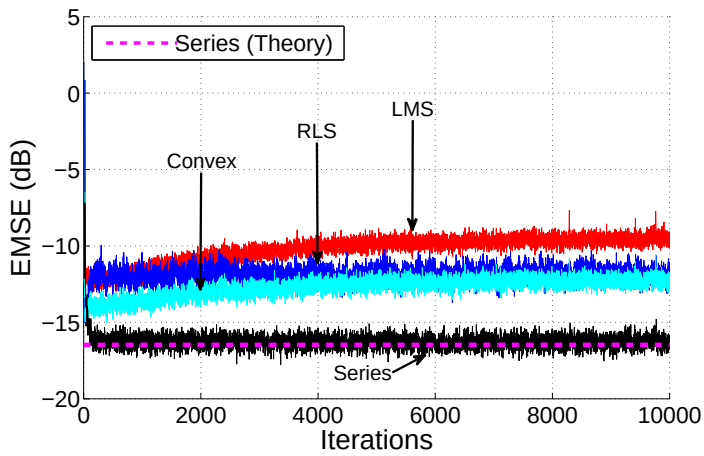


Figure 14: EMSE of the individual filters and the combinations for $\mu_o = 0.2$ and $\lambda = 0.4$ (500 realizations).

Figures 15 and 16 repeat the previous simulation for $\mu_o = 0.4$. Due to the highly correlated input signal, the LMS component filter experiences spikes (outliers), and so does the convex as it can not fully combat this phenomenon. This

is clearly seen in the abacus (Fig. 15) and in the learning curves (Fig. 16) which have been generated for $\lambda = 0.6$.

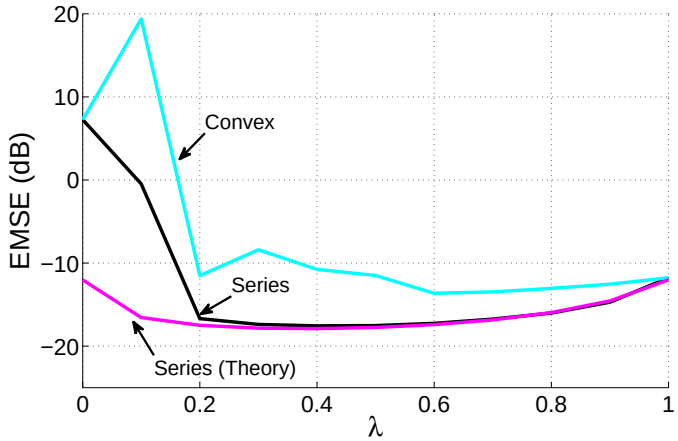


Figure 15: EMSE x λ for both combinations (RLS–LMS) with $\mu_o = 0.4$ and $\sigma_q^2 = 10^{-4}$.

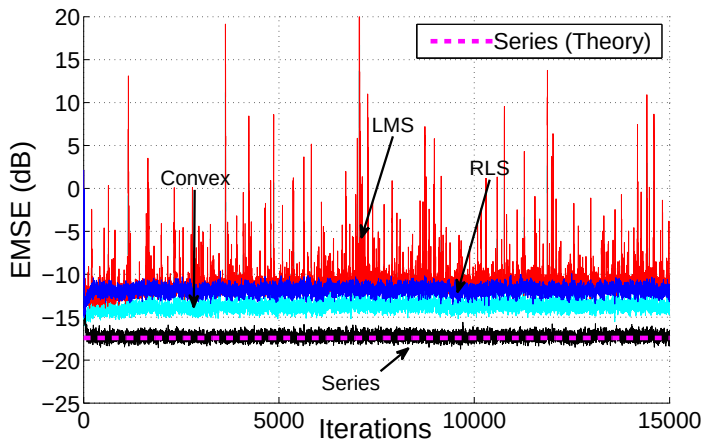
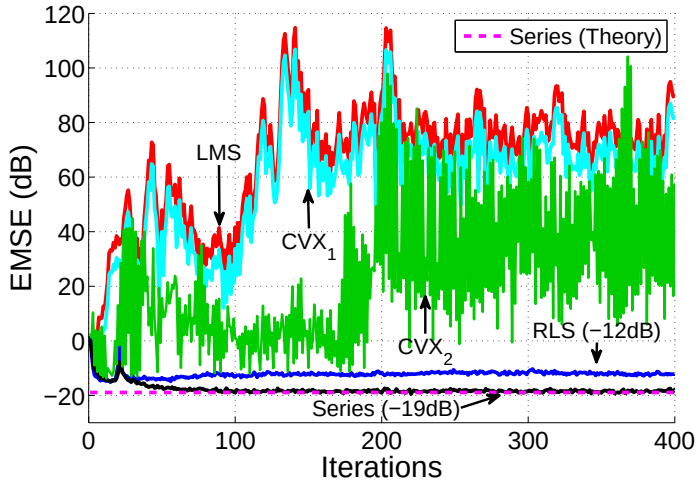
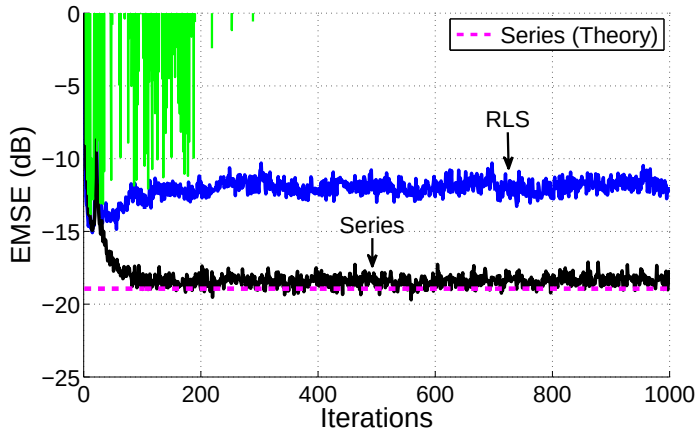


Figure 16: EMSE of the individual filters and the combinations for $\mu_o = 0.4$ and $\lambda = 0.6$ (500 realizations).

In the same vein, Fig. 17 presents the same scenario, only with the step size further increased to $\mu_o = 0.9$, while keeping $\lambda = 0.6$. The abacus has not been depicted as it is nearly meaningless: the convex-parallel diverges for the whole interval. The incremental case, on the other had, is stable for $\lambda \geq 0.5$. Steady-state performance for the series case has been improved almost $3dB$ compared to the previous examples. Although the RLS sub-filter can converge, the LMS is unstable, driving the convex combination with fixed λ (CVX_1) into divergence. The



(a)



(b)

Figure 17: (a)EMSE of the individual filters and the combinations for $\mu_o = 0.9$, $\lambda = 0.6$ and $\sigma_q^2 = 10^{-4}$ (500 realizations). CVX_1 is the convex combination with fixed λ. CVX_2 is the convex combination with adaptive λ. The stabilization effect results from the natural feedback in series structure. (b) A zoomed-in view of the convergent curves.

convex algorithm with adaptive λ (CVX_2: $\lambda \to \lambda(i)$) was included to illustrate that, even with an update rule, in this stringent scenario the parallel combination is not able to cope with LMS instability: the best the convex combination could do would be to track the RLS performance, setting $\lambda = 1$. Once again, the simulated curves corroborate the theory.

Fig. 17 illustrates the stabilization effect provided by the feedback nature of the structure. Even with a poorly designed step size for the LMS (for example,

$\mu_o = 0.9$), the series combination is able to circumvent this issue, making the final EMSE curve converge. In this situation, the only way the series combination can diverge is to set λ very close or equal to 0, turning the RLS component into a relay node, and consequently the combination becomes the unstable LMS filter.

Fig. 18 presents an abacus showing the universal behavior of the series-incremental combination for different degrees of non-stationarity (σ_q^2). Note that an improvement of about $5dB$ of the combination over the sub-filters remains over a wide range of σ_q^2.

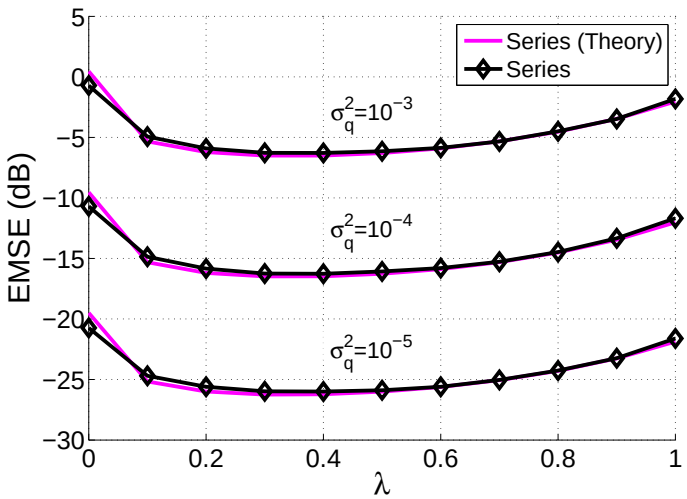


Figure 18: Abacus for RLS–LMS combination with $\mu_o = 0.2$ and variable σ_q^2

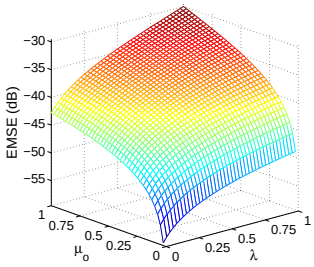
4.4 LMS–LMS combination

For this combination,

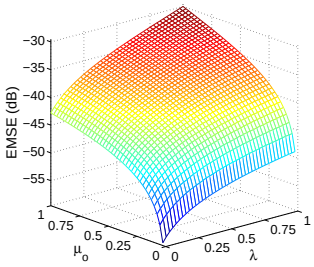
$$\begin{aligned} \boldsymbol{H}_{1,i} &= \mu_1 I \\ \boldsymbol{H}_{2,i} &= \mu_2 I \end{aligned} \tag{4.30}$$

where μ_1 and μ_2 are the filters step sizes defined by (4.2) with $r_o = 0.1$. Thus,

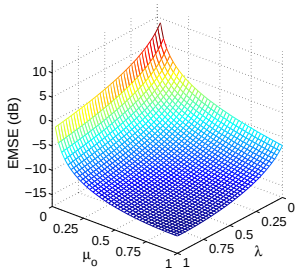
$$\boldsymbol{H}_i = [\lambda\mu_1 + (1-\lambda)\mu_2(1 - \lambda\mu_1\|\boldsymbol{u}_i\|^2)I], \tag{4.31}$$



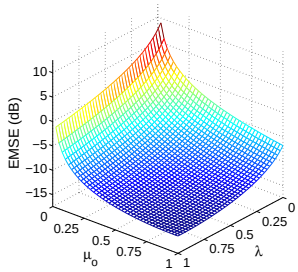
(a) Stationary - white



(b) Stationary - colored



(c) Non-stationary - white



(d) Non-stationary - colored

Figure 19: LMS–LMS - EMSE theoretical surfaces for variable μ_o and λ in both stationary and non-stationary scenarios, with white and correlated data.

and it readily follows that

$$H = \left[\lambda\mu_1 + (1-\lambda)\mu_2[1 - \lambda\mu_1 Tr(R_u)]\right] \cdot I \tag{4.32}$$

so that the EMSE (MSE) may be retrieved by (4.24).

Plots of the EMSE surface are depicted in Fig. 19 for four cases, varying μ_o and λ. The same scenarios and parameters (when applicable) from Section 4.3 are employed. None of the cases present a convex surface like in the RLS–LMS combination. Considering the ranges of μ_o and λ, the best the combination can do is to achieve the same EMSE level of the component filters. Nevertheless, as cited previously in RLS–LMS case, the incremental-cooperative nature of the series structure makes it more robust than the parallel case.

4.4.1 Simulations

As predicted by theory and corroborated by simulations (refer to Figs. 19, 20, and 21), the LMS–LMS combination does not improve performance with respect to the sub-filters. However, the series combination is more robust to the sub-filters as compared to the convex combination, as can be seen in Figs. 22 and 23.

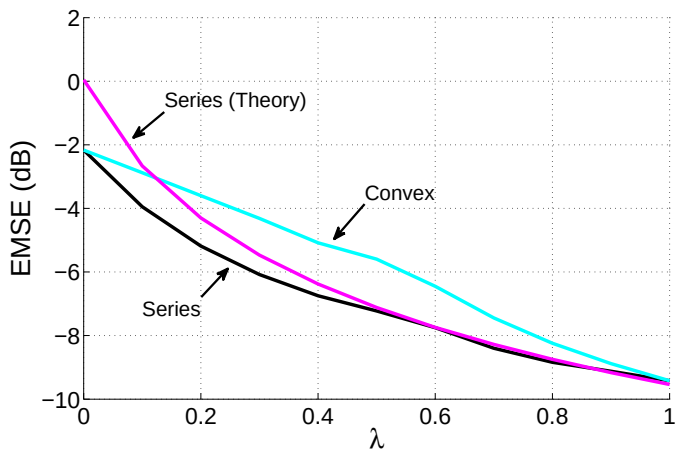


Figure 20: EMSE x λ for both combinations (LMS–LMS) with $\mu_o = 0.2$ and $r_o = 0.1$.

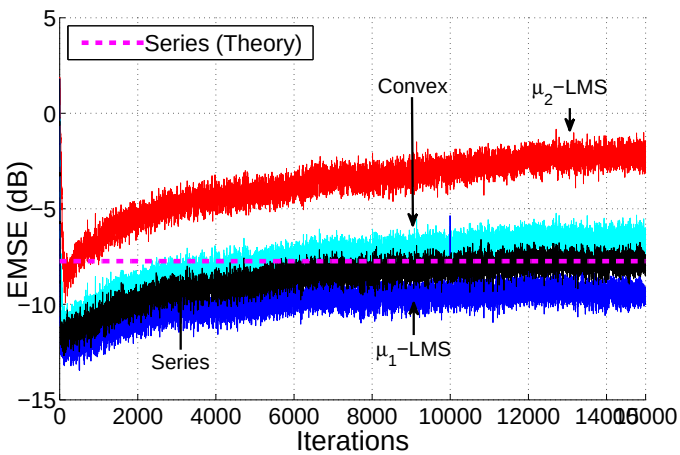


Figure 21: EMSE of the individual filters and the combinations for $\mu_o = 0.2$, $r_o = 0.1$ and $\lambda = 0.6$ averaged over 300 realizations.

Further study in the LMS–LMS case, for instance, in different degrees of non-stationarity, may show new perspectives and is the scope for future work.

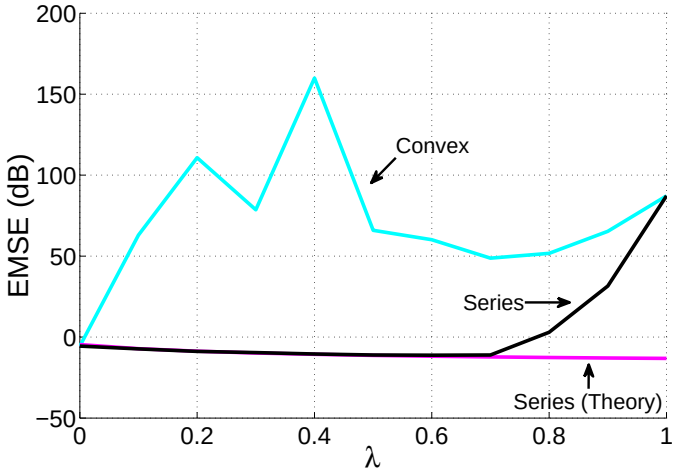
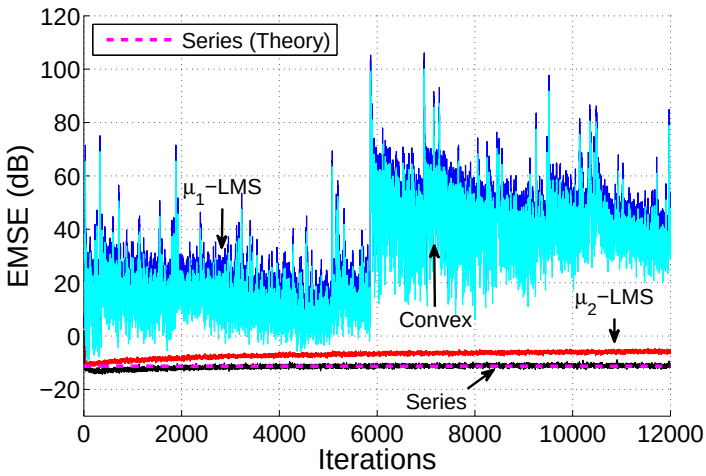
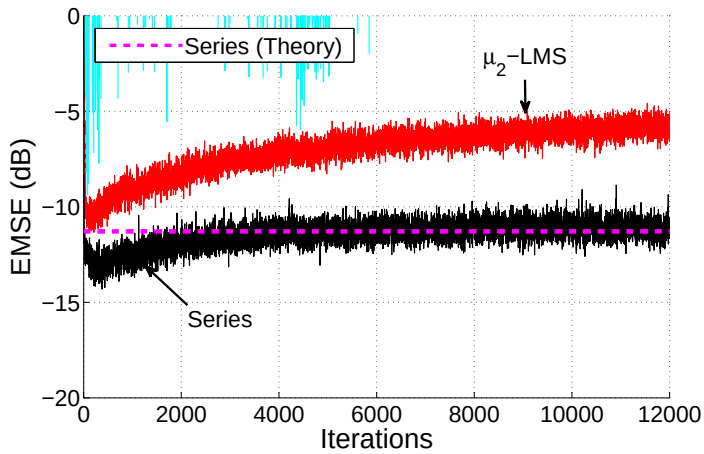


Figure 22: EMSE x λ for both combinations (LMS–LMS) with $\mu_o = 0.6$ and $r_o = 0.1$.



(a)



(b)

Figure 23: (a)EMSE of the individual filters and the combinations for $\mu_o = 0.6$, $r_o = 0.1$ and $\lambda = 0.5$ averaged over 300 realizations. (b) A zoomed-in view of the convergent curves.

5 CONCLUSION

This thesis proposed a new method to combine AFs inspired by incremental strategies. In stationary scenarios, the new technique is able to naturally circumvent the stagnation effect experienced by the convex-parallel topology, without sacrificing steady-state performance. This was validated for the LMS–LMS and RLS–LMS cases, employing a simple design for the mixing parameter ($\lambda(i)$). In the general case, the adaptive design may be challenging and new techniques have yet to be developed. Partial results were obtained (but not presented) for a normalized stochastic rule to design $a(i)$ ($\lambda(i)$), however further study is necessary.

In order to assess the incremental structure performance, mean-square analysis was derived for combination of filters from different families. Simulations and the analytical model show that the incremental RLS–LMS structure is universal for correlated data and rapidly varying plants. On the other hand, the LMS–LMS case did not present universality, but it outperforms the parallel structure as well. In any event, the series combination tends to be more robust than the parallel counterpart for the same component filters.

The analytical mean-square models matched well the simulations and have to be studied in more detail to explore the impact of other parameters, as the forgetting factor (η) or a wider range for σ_q^2. A more comprehensive model for the time-varying plant, as a non-diagonal Q, is also promising.

Since a new structure was introduced, several extensions are possible in terms

of different cost functions and filters or mixing strategies (affine combination). Some topics for future study are:

- New component filters, as NLMS–LMS or APA–LMS;

- Different cost functions, as blind algorithms and LMF–LMS (the series counterpart of the LMMN algorithm [14]);

- Affine combinations;

- Derive an effective adaptive rule to design $\lambda(i)$ and perform its mean-square analysis.

REFERENCES

[1] J. Arenas-Garcia, A.R. Figueiras-Vidal, and A.H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *Signal Processing, IEEE Transactions on*, vol. 54, no. 3, pp. 1078–1090, 2006.

[2] M.T.M. Silva and V.H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3137–3149, 2008.

[3] M.T.M. Silva, V.H. Nascimento, and J. Arenas-Garcia, "A transient analysis for the convex combination of two adaptive filters with transfer of coefficients," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3842–3845.

[4] N. Merhav and M. Feder, "Universal prediction," *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2124 –2147, oct 1998.

[5] N.J. Bershad, J.C.M. Bermudez, and J.Y. Tourneret, "An affine combination of two LMS adaptive filters – Transient mean-square analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1853–1864, 2008.

[6] L. F. O. Chamon, W.B. Lopes, and C.G. Lopes, "Combination of adaptive filters with coefficients feedback," in *Accepted for publication in ICASSP 2012*. IEEE.

[7] C.G. Lopes, E.H. Satorius, and A.H. Sayed, "Adaptive carrier tracking for direct-to-earth Mars communications," in *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on*. IEEE, 2007, pp. 1042–1046.

[8] C.G. Lopes, E.H. Satorius, P. Estabrook, and A.H. Sayed, "Efficient Adaptive Carrier Tracking for Mars to Earth Communications During Entry, Descent and Landing," in *Statistical Signal Processing, 2007. SSP'07. IEEE/SP 14th Workshop on*. IEEE, 2007, pp. 517–521.

[9] 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," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, no. 4, pp. 1865–1879, 2010.

[10] W.B. Lopes and C.G. Lopes, "Incremental-cooperative strategies in combination of adaptive filters," in *Int. Conf. in Acoust. Speech and Signal Process.* IEEE, 2011, pp. 4132 –4135.

[11] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim*, vol. 7, pp. 913–926, 1996.

[12] A. Geary and D.P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, 1999, vol. 1, pp. 907 –912 vol.1.

[13] C.G. Lopes and A.H. Sayed, "Incremental adaptive strategies over distributed networks," *Signal Processing, IEEE Transactions on*, vol. 55, no. 8, pp. 4064 –4077, aug. 2007.

[14] A.H. Sayed, *Adaptive filters*, Wiley-IEEE Press, 2008.

[15] S.S. Haykin, *Adaptive filter theory*, Prentice-hall Englewood Cliffs, NJ, 1986.

[16] Paulo Sergio Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Kluwer Academic Publishers, Norwell, MA, USA, 2 edition, 2002.

[17] C.G. Lopes and A.H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 3.

[18] C.G. Lopes and J.C.M. Bermudez, "Evaluation and design of variable step size adaptive algorithms," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. IEEE, 2002, vol. 6, pp. 3845–3848.

[19] M.C. Tsakiris, C.G. Lopes, and P.A. Naylor, "An Alternative Criterion for Regularization in Recursive Least-Squares Problems," *Image*, vol. 60, no. 40, pp. 20.

[20] M.C. Tsakiris and C.G. Lopes, "A robust affine projection algorithm with feedback compensation of the condition number," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 2057–2060.

[21] Alan J. Laub, *Matrix analysis - for scientists and engineers.*, SIAM, 2005.