LUIZ FERNANDO DE OLIVEIRA CHAMON

COMBINATIONS OF ADAPTIVE FILTERS

São Paulo
2015

LUIZ FERNANDO DE OLIVEIRA CHAMON

COMBINATIONS OF ADAPTIVE FILTERS

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

São Paulo
2015

LUIZ FERNANDO DE OLIVEIRA CHAMON

COMBINATIONS OF ADAPTIVE FILTERS

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

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Cássio G. Lopes

São Paulo
2015

To my parents, Edna and Marco.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Cássio G. Lopes, for his guidance and indelible advice, as well as for the constant challenges and opportunities that kept me motivated.

I would also like to thank my parents, Marco and Edna, for all their support over these years, and my brother Paulo for the conversations and debates that keep me on my toes.

It is difficult to express my gratitude to my friends at the University of São Paulo, Wilder, Fernando, Amanda, Renato, David, Matheus, Yannick, and Humberto, for their help in both shaping this work and, at times, forgetting about it.

I am also indebted to all my professors, at the University of São Paulo and elsewhere, for having shared their knowledge, opinions, and experience. In particular, I would like to thank Prof. Vítor Nascimento and Prof. Magno Silva for their contributions to this work and to my formation.

Finally, I cannot thank enough my beloved fiancée Bárbara B. Lucena for her presence, support, and understanding in good and bad times.

This work was supported by a fellowship from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

ABSTRACT

Adaptive filtering has grown to become a fundamental topic in signal processing, increasingly attracting attention from the community. Important factors in this popularization were their low computational complexity and model-free nature, adapting even to nonstationary characteristics of the systems and/or signals under study. Nevertheless, many adaptive algorithms introduce trade-offs, for instance, between convergence rate, nonstationary signals tracking, and steady-state error, which can hinder their use in practical applications. Furthermore, some adaptive filters can become unstable when word length is reduced and/or the input data are highly correlated. Recently, combination of adaptive filters was put forward as a solution for such issues. This approach consists in combining a pool of filters by means of a supervisor that attempts to make the overall system at least as good (usually in the mean-square sense) as the best filter in the set. Examples of these structures have been shown to successfully solve this problem, although well-known limitations remain to be addressed. Moreover, due to the relative novelty of this topic, developments in combination of adaptive filters are difficult to accommodate into a common theoretical framework. This work studies combination of adaptive filters and addresses the aforementioned issue by (i) classifying the existing combinations and proposing a taxonomy that exposes the similarities and differences in their forms; (ii) proposing new combinations; (iii) devising a general framework for studying combinations of adaptive filters and using such framework in performance analyses.

RESUMO

Filtragem adaptativa vem ganhando destaque desde seu surgimento tornando-se um tópico de estudo fundamental em processamento de sinais. A versatilidade de dispensarem total conhecimento das propriedades estatísticas dos sinais, aliada à simplicidade computacional de seus métodos, foram importantes fatores em sua consagração. Apesar disto, muitos filtros adaptativos apresentam compromissos envolvendo, por exemplo, velocidade de convergência, rastreamento de sinais não-estacionários e erro em regime, que podem dificultar sua aplicação na prática. Ademais, alguns algoritmos adaptativos são instáveis quando suas entradas são altamente correlacionados e/ou a precisão dos cálculos é reduzida. Uma solução recente para estes problemas é o uso de combinações de filtros adaptativos. Esta abordagem baseia-se em combinar um conjunto de filtros por meio de um supervisor que procura fazer com que o sistema global seja pelo menos tão bom (em geral no sentido quadrático médio) quanto o melhor filtro do conjunto. Exemplos destas estruturas já mostraram a eficácia deste método, apesar de ainda existirem reconhecidas limitações. Além disso, em se tratando de um tópico relativamente recente, os desenvolvimentos na área de combinação de filtros adaptativos não possuem uma estrutura teórica unificada. Este trabalho propõe abordar estas questões (i) classificando as combinações existentes e criando uma taxonomia que explicita semelhanças e diferenças entre elas; (ii) introduzindo novas combinações; e (iii) desenvolvendo uma forma unificada de descrever combinações de filtros adaptativos e usando-a em análises de desempenho.

LIST OF FIGURES

1	The analogy: (a) schema of a member and (b) the workgroup	15
2	The system identification scenario	22
3	A supervised adaptive filter	24
4	Physical interpretation of the ECR: (a) general adaptive filter; (b) NLMS (total reflexion)	27
5	Combination of adaptive filters	30
6	$\mathcal{FOB}$ description	30
7	A digraph $\mathcal{D} = (V, A)$	32
8	Example digraphs of combination topologies: (a) stand-alone adaptive filter; (b) arbitrary topology.	33
9	Data distribution networks: (a) data sharing; (b) stand-alone data reusing adaptive filter; (c) circular buffer; (d) randomized.	38
10	$\mathcal{FOB}$ description of adaptive networks: (a) incremental and (b) diffusion. .	45
11	Timeline	49
12	Taxonomy of combinations of adaptive filters	51
13	Hierarchical parallel combination of four adaptive filters	58
14	The parallel-independent topology digraph.	62
15	Convergence stagnation issue of the parallel-independent topology	63
16	The parallel topology with coefficients leakage digraph.	64
17	The parallel topology with cyclic coefficients feedback digraph.	64
18	Comparison between coefficients leakage and coefficients feedback	65
19	Tracking of an LMS+RLS combination with and without coefficients feedback	67
20	$2 \cdot \text{LMS}$ combination with coefficients feedback ($L = 1$)	68
21	Designing the cycle period	69

22	Iterations-to-convergence for different L	69
23	The effect of the convexity constraint on the incremental supervisor	74
24	Comparison between different supervising rules for a combination of two LMS filters.	75
25	Incremental topology digraphs: (a) linear incremental topology and (b) ring incremental topology.	75
26	Comparison between the $\{\text{LMS}\}^N$ and the APA.	77
27	Cyclically incremental, parallel combination of LMS filters	79
28	Graph representation of parallel-incremental topologies: (a) parallel com- bination of incremental components and (b) parallel combination of incre- mental combinations.	80
29	Parallel-incremental combination of LMS filters	81
30	Variable step size algorithm based on a 2-LMS combination with coefficients feedback (nonstationary SNR)	85
31	Variable step size algorithm based on a 2-LMS combination with coefficients feedback (nonstationary $\mathbf{w}^o$)	86
32	Graphical representation of the complexity of different algorithms with similar performances	88
33	Diagram of relations between the APA and combinations of adaptive filters	88
34	APA and $f\{K \cdot \text{LMS}\}$	91
35	APA and DR- $\{\text{LMS}\}^N$ (stationary scenario)	92
36	APA and DR- $\{\text{LMS}\}^N$ (nonstationary scenario)	93
37	Parallel-incremental combination (white input)	94
38	Parallel-incremental combination (correlated input)	95
39	Performance of the $\{\text{sign-error LMS}\}^N$ in fixed point precision (stationary scenario)	96
40	Performance of the $\{\text{sign-error LMS}\}^N$ in fixed point precision (nonstation- ary scenario)	97
41	APA and DR- $\{\{\text{sign-error LMS}\}^N\}^L$	98

42	Component filters behavior under cyclic coefficients feedback ($2 \cdot \text{LMS}$) . . .	104
43	Steady-state of parallel combinations with cyclic coefficients feedback for different cycle periods	105
44	Steady-state analysis of a $2 \cdot \text{LMS}$ combination	113
45	Tracking analysis of a $2 \cdot \text{LMS}$ combination	113
46	Steady-state analysis of an $\text{LMS} + \text{LMF}$ combination	115
47	Tracking analysis of the LMMN using an $\text{LMS} + \text{LMF}$ combination.	116
48	Steady-state analysis of a $2 \cdot \text{RLS}$ combination	119
49	Tracking analysis of an $\text{LMS} + \text{RLS}$ combination	121
50	Supervising parameter variance under different topologies (normalized con- vex supervisor	128
51	Transient analysis of a $2 \cdot \text{LMS}$ combination with a convex supervisor. . . .	129
52	Transient analysis of the $2 \cdot \text{LMS}$ combination with an affine supervisor . .	130
53	Separation principle A.13	136
54	Steady-state analysis validation	138
55	Mean convergence analysis of an $\{\text{LMS}\}^N$	140
56	Incremental mean-square analysis of a $\{\text{LMS}\}^2$	144
57	Optimal supervisor for the incremental combination	146

CONTENTS

1	Motivation, contributions, and notation	14
1.1	An analogy: The workgroup	14
1.2	Motivation	15
1.3	Objectives	16
1.4	Contributions	17
1.5	Publications list	17
1.6	Notation	18
1.6.1	The system identification scenario	20
2	Stand-alone adaptive filters	23
2.1	Figures of merit	25
2.2	Energy conservation relation	26
2.3	Concluding remarks	28
3	Combinations of adaptive filters	29
3.1	A formal definition	29
3.2	The topology	31
3.2.1	Digraphs	31
3.2.2	Topologies and digraphs	32
3.2.3	The parallel/incremental dichotomy	34
3.3	The data reusing method	36
3.3.1	Data distribution networks	38
3.4	The supervisor	40
3.5	An <i>FOB</i> illustration	41

3.6	Relation to adaptive networks	44
3.7	Concluding remarks	46
4	Taxonomy and review	47
4.1	Timeline	47
4.2	Classifying combinations	50
4.2.1	Component filters	52
4.2.2	Topology	52
4.2.3	Data reusing method	52
4.2.4	Supervising rule	52
4.3	Concluding remarks	53
5	Parallel combinations	56
5.1	Supervising rules	56
5.1.1	Unsupervised (“static”)	58
5.1.2	The convex supervisor	59
5.1.3	The affine supervisor	60
5.2	Parallel topologies	61
5.2.1	The parallel-independent topology	61
5.2.2	The parallel topology with coefficients leakage	63
5.2.3	The parallel topology with coefficients feedback	65
5.2.3.1	Design of the cycle period	66
5.3	Concluding remarks	68
6	Incremental combinations	71
6.1	Supervising rules	72
6.1.1	Unsupervised (“static”)	73
6.1.2	The convex supervisor	73

6.1.3	The $\{\text{LMS}\}^2$ -specific supervisor	73
6.2	Incremental topologies	74
6.2.1	Linear incremental topology	75
6.2.2	Ring incremental topology	76
6.3	Concluding remarks	76
7	Combinations with arbitrary topologies	78
7.1	Cyclically incremental, parallel topology	78
7.2	Parallel-incremental combination	79
7.3	Concluding remarks	80
8	Stand-alone adaptive algorithms as combinations of adaptive filters	82
8.1	Mixed cost function algorithms	82
8.2	Data reusing algorithms	83
8.3	Variable step size algorithms	84
8.4	Concluding remarks	85
9	Combination as a complexity-reduction technique	87
9.1	$\text{DR-}\{\text{LMS}\}^N$	88
9.1.1	The APA and the $f\{K \cdot \text{LMS}\}$	88
9.1.2	The incremental counterpart of $f\{K \cdot \text{LMS}\}$	90
9.2	$\text{DR-}\{\text{LMS}\}^N + \text{LMS}$	92
9.3	$\{\text{sign-error LMS}\}^N$	93
9.4	$\text{DR-}\{\{\text{sign-error LMS}\}^N\}^L$	97
9.5	Concluding remarks	97
10	Performance analysis of parallel combinations	101
10.1	Steady-state and tracking performance	101
10.1.1	Global steady-state analysis	102

10.1.2	Cyclic feedback and steady-state	104
10.1.3	Variance and covariance relations for $L \rightarrow \infty$	105
10.1.4	Variance and covariance relations for $L = 1$	107
10.1.5	Supervisor analysis	109
10.2	Steady-state and tracking performance of specific combinations	111
10.2.1	2 · LMS	111
10.2.1.1	Without feedback	111
10.2.1.2	With feedback	112
10.2.2	LMS + LMF	113
10.2.2.1	Without feedback	114
10.2.2.2	With feedback	115
10.2.3	2 · RLS	116
10.2.3.1	Without feedback	117
10.2.3.2	With feedback	118
10.2.4	LMS + RLS	119
10.2.4.1	Without feedback	119
10.2.4.2	With feedback	120
10.3	Transient Performance	121
10.3.1	Component filters analysis	122
10.3.2	Supervisor analysis	125
10.3.3	Complete transient analysis	127
10.4	Concluding remarks	128
11	Performance analysis of incremental combinations	132
11.1	Overall recursion	133
11.2	Steady-state performance	134
11.2.1	Energy conversation relation	134

11.2.2	Mean-square analysis	135
11.3	Transient performance	139
11.3.1	Mean convergence analysis	139
11.3.2	Unbiasedness constraint	141
11.3.3	Mean-square convergence analysis	142
11.3.4	Optimal supervisor	144
11.4	Concluding remarks	146
12	Conclusion	148
13	Suggestions for future work	149
	References	151

1 MOTIVATION, CONTRIBUTIONS, AND NOTATION

Analogies prove nothing, that is quite true, but they can make one feel more at home.

-- Sigmund Freud, "New Introductory Lectures on Psychoanalysis"

1.1 An analogy: The workgroup

Analogies used in the context of local or distributed adaptive signal processing usually involve some sort of learning experience [1–6]. These, however, are limited inasmuch as they only deal with the adaptation capability of the algorithms without explicitly addressing the estimation/filtering step. What is more, distributed scenarios require an analogy for the exchange of information between nodes and the direct passing of “learned concepts” among peers is hard to relate to our own experience with knowledge transmission. An extension of the typical analogy is therefore proposed below.

Define a *workgroup* as a set of peers that collaborate to achieve a common goal. Each member relies on the information available to him to *learn* concepts and *apply* these concepts to produce work (Figure 1a). The latter step is the fundamental difference with previous analogies, as the members of the workgroup no longer only learn but also perform an action based on their learning. The workgroup has a leader (supervisor) responsible for generating a single piece of work based on the work produced by the group members¹. In a more concrete example, the information available can be thought of as books and the goal is to write an essay (e.g., a dissertation on combinations of adaptive filters). The assignment of the group leader is then to hand in a unique essay based on those written by the members, as well as the available books.

Individually, the group members may have different reading/learning/writing characteristics. Some may be more careful and take longer to deliver their version of the essay, while others work faster at the cost of making more mistakes. Also, the books available to the workgroup share a common topic and may therefore repeat themselves (their information is correlated). Some members may be better than others at identifying and

¹This analogy also applies to distributed settings in which there is no leader. In this case, however, there is no longer a unique essay to hand in and all group members must thrive to produce their own copy of the best possible essay in the group (i.e., to achieve *spatial universality* [7]).

Figure 1: The analogy: (a) schema of a member and (b) the workgroup

skipping these parts. Finally, it could be that the concept under study is still under debate (nonstationary) and people’s ability to update their essays to reflect changes can vary. The objective of working in a group is taking advantage of each member’s strength while counterbalancing their weaknesses.

But how to organize this cooperation? Should the members write their essays independently (parallel-independent topology, Chapter 5) or should they take turns writing the final essay (incremental topology, Chapter 6)? Should the group leader give feedback on the members work (coefficients feedback, Section 5.2.3)? Is it better for all workers to read the same book or should they each get a different one (data reusing method, Section 3.3)? The same questions can be asked for combinations and the result of their analyses in the remainder of this work can be interpreted in the context of the workgroup analogy.

1.2 Motivation

As is usually the case in engineering, estimation and detection problems come with trade-offs and limitations that may be hard to overcome (e.g., bias/variance or type I/type II errors compromises [8,9]). Combination is an established pattern to address this situation in several domains.

In the context of universal prediction, for example, algorithms with different predictive capabilities are weighted and aggregated to form a more robust predictor [10–12]. In fact, such an algorithm selection approach can be used to construct a “deterministic” theory of statistics relying on the concept of *individual sequences* [11, 12]. In machine learning (ML), *boosting* techniques combine weak learners (estimators) to produce high accuracy algorithms (e.g., AdaBoost and boosting trees) [13]. These techniques were inspired by the works of Kearns and Valiant [14, 15] and Schapire [16] that showed that

weak and strong learnability are actually equivalent. These algorithms are among the most effective ones in ML today [17].

Adaptive filters (AFs) are well-suited for combinations due to the compromises involved in their applications. Indeed, despite their versatility, AFs have inherent trade-offs that usually hinge on three aspects: transient performance, steady-state performance, and computational complexity. For instance, increasing the convergence rate of an adaptive algorithm typically involves degrading its steady-state error level [18–20]. Otherwise, another more complex AF must be used (such as the APA or RLS). Many of these algorithms, however, suffer from robustness issues that hinder their use in some scenarios (e.g., highly correlated signals and reduced precision [18, 19, 21]). Other solutions have been proposed, such as *data reusing* (DR) [22–25], *mixed-norm* updates [26–29], and *variable step size* (VSS) [30–32]. Combinations of AFs, however, have shown several advantages over these techniques in terms of performance and robustness [33–49]. In some cases, it even provides a more general framework under which these adaptive algorithms can be cast (as argued in Chapter 8).

1.3 Objectives

This work proposes to study combinations of AFs in a general and unified way. In doing so, it bridges the gap between different combinations, as well as combinations and stand-alone AFs, that were previously not properly related. The objectives of this research can be stated as:

I. Classify existing combinations

The development of a taxonomy for combinations of AFs helps clarify the similarities between existing combinations and better understand the elements that constitute these structures. This classification and the similarities found constitute an important step towards Objectives II and III.

II. Propose new combinations

Based on the classification devised for Objective I, unexplored algorithms/combinations can be investigated to formulate new structures that can then be analyzed and compared to the existing ones. These analyses can then be used to improve the general framework of Objective III.

III. Devise a general framework for combinations of adaptive filters

As a broader classification and novel combinations are elaborated, a more complete

framework for combinations can be created. This framework involves the formulation of a precise definition of combination of AFs, through which performance analyses can be derived. These analyses contribute to a better understanding of combinations and, thus, to Objective I.

1.4 Contributions

- (i) Extensive literature review and a taxonomy to classify combination of adaptive filters.
- (ii) A definition of combination and combination of adaptive filters, along with a specific notation to describe them.
- (iii) A formal study of combination topology using graph-theoretic arguments.
- (iv) Parallel topology with cyclic coefficients feedback: motivation, proposition, and analyses.
- (v) Data reusing combinations: motivation, data distribution networks, and performance improvements.
- (vi) Interpreting stand-alone adaptive algorithms as combinations of AFs and bridging the gap between combinations and adaptive networks.
- (vii) Combination as a complexity reduction technique.
- (viii) Unified performance analysis of parallel combinations: energy conservation relation, steady-state, tracking, and transient behavior for different component filters and supervising rules.
- (ix) Performance analysis of incremental combinations: steady-state, transient, unbiasedness constraint, and optimal supervisor.

1.5 Publications list

- [1] L.F.O. Chamon and C.G. Lopes. “Combinations of adaptive filters with coefficient feedback”. Work in progress (*Journal*), 2015.
- [2] H.F. Ferro, L.F.O. Chamon, and C.G. Lopes. “FIR-IIR adaptive filters hybrid combination”. *IET Electronics Letters*, 50[7]:501–503, 2014.

- [3] L.F.O. Chamon and C.G. Lopes. “There’s plenty of room at the bottom: Incremental combinations of sign-error LMS filters”. In: *International Conference in Acoustic, Speech, and Signal Processing (ICASSP)*, p. 7248–7252, 2014.
- [4] L.F.O. Chamon and C.G. Lopes. “Transient performance of an incremental combination of LMS filters”. In: *European Signal Processing Conference (EUSIPCO)*, 2013.
- [5] L.F.O. Chamon and C.G. Lopes. “On parallel-incremental combinations of LMS filters that outperform the Affine Projection Algorithm”. In: *Brazilian Telecommunication Symposium (SBrT)*, 2013.
- [6] 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*, p. 406–410, 2012.
- [7] L.F.O. Chamon, W.B. Lopes, and C.G. Lopes. “Combination of adaptive filters with coefficients feedback”. In: *International Conference in Acoustic, Speech, and Signal Processing (ICASSP)*, p. 3785–3788, 2012.
- [8] L.F.O. Chamon and C.G. Lopes. “Combination of adaptive filters for relative navigation”. In: *European Signal Processing Conference (EUSIPCO)*, p. 1771–1775, 2011.

1.6 Notation

The main conventions and symbols used in this work are collected in Table 1. In particular, note that **boldface** is used to denote vectors and matrices and that all vectors are column vectors except for the regressor vector $\mathbf{u}$, which is taken as a row vector for convenience. When iteration-dependent, scalars are indexed as in $x(i)$ and vectors as in $\mathbf{x}_i$. All variables in this text are *real-valued*². Deterministic and random variables are clear from the context.

In the context of combinations, component filters values are indexed by n . For instance, $e_n(i)$, $\mathbf{w}_{n,i}$, and $\{\mathbf{u}_{n,i}, d_n(i)\}$ refer to the n -th component output estimation error, coefficient vector, and data pair, all at iteration i . Buffered data, on the other hand, are indexed using k . Thus, the k -th data pair in the buffer (k -th row of $\mathbf{U}_i$ and $\mathbf{d}_i$) is referred to as $\{\mathbf{u}_{k,i}, d_k(i)\}$. Often times, $\{\mathbf{u}_{k,i}, d_k(i)\} = \{\mathbf{u}_{i-k}, d(i-k)\}$, in which case the

²Although the complex-valued case is not considered in this work, the extension of the results are straightforward under mild assumptions, such as circularity. Moreover, widely linear versions with similar computational complexity can be derived using multichannel counterparts of the adaptive algorithms and concatenating the real and imaginary parts of their input data [50].

latter form is preferred for explicitness. Notice that component filters and input data use different indices (n and k , respectively), a distinction that is important when discussing DR methods. Also relative to combinations, variables without the n index refers to a global quantity (e.g., the global coefficients $\mathbf{w}_i$ or the output of the combination $y(i)$).

When referring to combinations, the notation introduced in [45, 47] is adopted in this text to avoid the use of long cryptic acronyms. Inspired by transfer function composition in system theory [51], parallel combinations are represented by *addition*, whereas incremental combinations are represented by *multiplication*. Typical precedence rules and intuitive extensions can be used as shorthands for larger combinations. For instance, for two LMS filters, their parallel combination is written $\text{LMS} + \text{LMS}$ or $2 \cdot \text{LMS}$ and their incremental combination, $\text{LMS} \cdot \text{LMS}$ or $\{\text{LMS}\}^2$. When necessary, combinations with *coefficients feedback* or *data buffering* are distinguished using the prefixes f or DR respectively, as in $f\{\text{LMS} + \text{LMS}\}$ and $\text{DR-}\{\text{LMS}\}^N$.

In the sequel, the data model and assumptions used in the performance analyses and simulations are introduced.

Table 1: List of symbols and notation

$\mathbb{R}$	Field of real numbers
$\mathbb{Z}$	Ring of integer numbers
$\mathbb{N}$	Set of natural numbers
$\mathbf{I}$	Identity matrix
$\mathbf{1}$	Column vector of ones
$(\cdot)^T$	Matrix transposition
$\text{diag}\{\mathbf{x}\}$	Diagonal matrix formed from the entries of $\mathbf{x}$
$\ \mathbf{x}\ $	Euclidian norm of $\mathbf{x}$
$\ \mathbf{x}\ _{\mathbf{W}}$	Weighted norm of $\mathbf{x}$ ($\mathbf{x}^T \mathbf{W} \mathbf{x}$)
$\ \mathbf{x}\ _1$	ℓ_1 -norm of $\mathbf{x}$
$ x $	Absolute value of the scalar x . Applies element-wise to a vector
$\delta(i)$	Kronecker delta
$\delta_L(i)$	Impulse train of period L : $\delta_L(i) = \sum_{r \in \mathbb{N}} \delta(i - rL)$
$u(i)$	Input signal
$\mathbf{u}_i$	Regressor vector ($1 \times M$)
$\mathbf{R}_{\mathbf{u}}$	Covariance matrix of the regressor vector $\mathbf{u}$
σ_u^2	Variance of the input signal $u(i)$
γ	Parameter of the autoregressive model for correlated $u(i)$

$v(i)$	Noise
σ_v^2	Variance of the noise $v(i)$
$\mathbf{w}^o$	Unknown system coefficient vector ($M \times 1$)
$\mathbf{q}_i$	System disturbance vector ($M \times 1$)
$\mathbf{Q}$	System disturbance covariance matrix
$d(i)$	Scalar measurement
$\mathbf{U}_i$	Buffer of regressor vectors ($K \times M$ matrix)
$\mathbf{d}_i$	Buffer of measurements ($K \times 1$ vector)
$\mathbf{w}_i$	Overall (global) coefficient vector
$y(i)$	Output of the combination
$e(i)$	Overall (global) output estimation error
$e_a(i)$	Overall (global) <i>a priori</i> error
$e_p(i)$	Overall (global) <i>a posteriori</i> error
$\tilde{\mathbf{w}}_i$	Overall (global) coefficients error vector
$\eta_n(i)$	Supervising parameter for the n -th component filter
$\{\mathbf{u}_{n,i}, d_n(i)\}$	Data pair (input) of the n -th component filter
$\mathbf{w}_{n,i}$	Coefficient vector of the n -th component filter
$\mathbf{w}_{n,a}$	<i>A priori</i> coefficient vector of the n -th component filter
$y_n(i)$	Output of the n -th component filter
$e_n(i)$	Output estimation error of the n -th component filter
$e_{a,n}(i)$	<i>A priori</i> error of the n -th component filter
$e_{p,n}(i)$	<i>A posteriori</i> error of the n -th component filter
$\tilde{\mathbf{w}}_{n,i}$	coefficients error vector of the n -th component filter
μ_n	n -th component filter step size

1.6.1 The system identification scenario

AFs can be used in a myriad of scenarios and setups, such as equalization, inverse control, and prediction [18–20]. However, for the purpose of performance analysis and simulation, a system identification scenario (Figure 2) is usually adopted [18–20], as it is representative of many applications, such as echo cancellation [52, 53], time delay estimation [54–59], and adaptive control [60].

In this scenario, the output of an unknown system is sampled to yield the measure-

ments

$$d(i) = \mathbf{u}_i \mathbf{w}_i^o + v(i), \quad (1.1)$$

where $\mathbf{w}_i^o$ is an $M \times 1$ vector that represents the unknown system at iteration i , $\mathbf{u}_i$ is the $1 \times M$ regressor vector (with covariance matrix $\mathbf{R}_u = \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i$) that captures samples $u(i)$ of a zero mean input signal with variance σ_u^2 , and $v(i)$ is a zero mean i.i.d. sequence with variance σ_v^2 that represents the measurement noise. The samples $\{u(i), v(i)\}$ are assumed to be realization of a Gaussian real-valued random process and to follow A.1 below. This assumption is used to render the derivations more tractable, since in practice the noise is at most uncorrelated with the input [18].

A.1 (*Noise independence*) $\{v(i), \mathbf{u}_j\}$ are independent for all i, j .

To account for different types of input in simulations, a Gaussian i.i.d. RV $x(i)$ is used to generate the samples $u(i)$. For white input data, $u(i) = x(i)$. On the other hand, to simulate input data correlation, a first-order AR model [AR(1)] is adopted [18, 19]. Explicitly,

$$u(i) = \gamma u(i-1) + \sqrt{1 - \gamma^2} x(i), \quad (1.2)$$

for some parameter $0 < \gamma < 1$ that control the degree of correlation (the closer to one, the more correlated).

To model nonstationary systems, the following first order random walk is adopted [18, 19]:

$$\mathbf{w}_i^o = \mathbf{w}_{i-1}^o + \mathbf{q}_i, \quad (1.3)$$

where the initial state is $\mathbf{w}_{-1}^o = \mathbf{w}^o$ and $\mathbf{q}_i$ is the realization of a zero mean i.i.d. Gaussian RV with covariance matrix $\mathbf{Q} = \mathbb{E} \mathbf{q}_i \mathbf{q}_i^T$. When the system is stationary, $\mathbf{q}_i = \mathbf{0} \Rightarrow \mathbf{Q} = \mathbf{0}$ and $\mathbf{w}_i^o = \mathbf{w}^o$. For tractability of the analysis, assume that

A.2 (*Random walk independence assumption*) The RV $\mathbf{q}_i$ is statistically independent of the initial conditions $\{\mathbf{w}_{-1}^o, \mathbf{w}_{-1}\}$, of $\{\mathbf{u}_j, v(j)\}$ for all i, j , and of $\{d(j)\}$ for $i > j$.

When studying DR techniques, it is convenient to define the data buffers $\mathbf{U}_i = [\mathbf{u}_{i-k}]$, a $K \times M$ regressor matrix, and $\mathbf{d}_i = [d(i-k)]$, a $K \times 1$ measurement vector, for $k \in \mathcal{K}$, where $\mathcal{K} \subset \mathbb{Z}_+$ is an index set of size K . A typical choice would be [18, 19, 61]

$$\mathbf{U}_i = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_{i-D} \\ \vdots \\ \mathbf{u}_{i-(K-1)D} \end{bmatrix} \quad \text{and} \quad \mathbf{d}_i = \begin{bmatrix} d(i) \\ d(i-D) \\ \vdots \\ d[i - (K-1)D] \end{bmatrix}, \quad (1.4)$$

Figure 2: The system identification scenario

The delay D between regressors was introduced in [62] to reduce the correlation between rows in $\mathbf{U}_i$, which decreases the condition number of $\mathbf{U}_i \mathbf{U}_i^T$ whose inverse is used in the recursion of some adaptive algorithms [18, 19]. Notice that the data model (1.1) can be written in terms of the buffers in (1.4) as

$$\mathbf{d}_i = \mathbf{U}_i \mathbf{w}^o + \mathbf{v}_i, \quad (1.5)$$

where $\mathbf{v}_i = [v(i) \ v(i-D) \ \cdots \ v[i-(K-1)D]]^T$. Observe that for $K = 1$, (1.5) reduces to (1.1).

2 STAND-ALONE ADAPTIVE FILTERS

*O todo sem a parte não é todo,
A parte sem o todo não é parte,
Mas se a parte o faz todo, sendo parte,
Não se diga, que é parte, sendo todo.*¹

-- Gregório de Matos, “Ao braço do mesmo Menino Jesus quando apareceu”

Adaptive filtering originated with the introduction of the Least Mean Squares (LMS) filter in the beginning of the 1960s by Widrow and Hoff [1]. Since then, it has gained considerable attention from the signal processing community and the number of adaptive algorithms and their applications have grown quickly (see [18–20, 52, 60, 63] and references therein).

An AF can be defined as a pair *filtering structure–adaptation algorithm* (Figure 3) [20]. It is important to distinguish this configuration from the *statistics estimator–Wiener solver* pair, which estimates statistical properties from the data and plugs them into a non-recursive formula for computing the filter parameters (which Haykin [20] calls the estimate-and-plug procedure). This second approach does not take advantage of iterative solutions, which are able to provide partial results as they converge. Furthermore, it most certainly leads to more complex algorithms. Considering this definition, the present work studies the class of AFs that employs Finite Impulse Response (FIR) filters adapted by means of supervised algorithms [18–20]. Note, however, that there is no restriction as to the algorithms that can be combined and combinations involving Infinite Impulse Response (IIR) [48, 64] and unsupervised [34, 65, 66] AFs are also possible.

Explicitly, a supervised adaptive algorithm is a procedure to update, at iteration i , an *a priori* coefficient vector $\mathbf{w}_a$ using a data pair $\{\mathbf{u}_i, d(i)\}$ (or more generally, $\{\mathbf{U}_i, \mathbf{d}_i\}$) so as to minimize a scalar cost function $J(\mathbf{w}_a)$. Most often, this is done using a stochastic gradient method [18–20] of general form

$$\mathbf{w}_i = \mathbf{w}_a - \mu \mathbf{p}, \quad (2.1)$$

where μ is a step size, $\mathbf{p} = \mathbf{B} \nabla^T J(\mathbf{w}_a)$ is an update direction, and $\mathbf{B}$ is a symmetric

¹The whole without the part is not whole, / The part without the whole is not a part, / But if the part makes the whole by being a part, / Let it not be called the part as it is the whole.

Figure 3: A supervised adaptive filter

positive-definite matrix.

The introduction of the concept of *a priori* coefficients in (2.1) is motivated by the view of AFs as solvers of local optimization problems [18]. From this viewpoint, an AF is a general algorithm that updates any given $\mathbf{w}_a$ according to its cost function. Even though most of them are made recursive by choosing $\mathbf{w}_a = \mathbf{w}_{i-1}$, the advantage of adopting (2.1) is that it accounts for interruptions in the regular operation of the adaptive algorithm, such as reinitializations or rescue techniques [18]. Moreover, the *a priori* coefficient vector is fundamental in the derivation of a general definition of combinations of AFs in Chapter 3. For the sake of clarity, however, the remainder of this chapter assumes $\mathbf{w}_a = \mathbf{w}_{i-1}$.

Different choices of $\mathbf{p}$ lead to different adaptive algorithms [18]. In stand-alone adaptive filtering, the cost function is typically chosen as the mean-square error (MSE) $J(\mathbf{w}_{i-1}) = \mathbb{E}e^2(i)$, where $e(i) = d(i) - y(i)$ is the output estimation error and $y(i) = \mathbf{u}_i \mathbf{w}_{i-1}$ is the output of the AF [18–20]. Nevertheless, other functions have also been successfully used. The recursion of the AFs found in this work are collected below for reference¹:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e(i) \quad (\text{LMS [1, 18]}) \quad (2.2)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T \text{sign}[e(i)] \quad (\text{Sign-error LMS [18]}) \quad (2.3)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\mu}{\|\mathbf{u}_i\|^2 + \epsilon} \mathbf{u}_i^T e(i) \quad (\text{NLMS [18]}) \quad (2.4)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e^3(i) \quad (\text{LMF [18, 67]}) \quad (2.5)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mathbf{P}_i \mathbf{u}_i^T e(i) \quad (\text{RLS [18]}) \quad (2.6)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\mu}{\mathbf{u}_i \mathbf{G}_i \mathbf{u}_i^T + \epsilon} \mathbf{G}_i \mathbf{u}_i^T e(i) \quad (\text{PNLMS [68, 69]}) \quad (2.7)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e(i) [\delta + (1 - \delta) e^2(i)] \quad (\text{LMMN [18, 28]}) \quad (2.8)$$

¹The acronyms stand for Least Mean Squares (LMS), Normalized LMS (NLMS), Least Mean Fourth (LMF), Recursive Least Squares (RLS), Proportionate NLMS (PNLMS), Least Mean Mixed-Norm (LMMN), Robust Mixed-Norm (RMN), Data Reusing LMS (DR-LMS), True Gradient LMS (TRUE-LMS), and Affine Projection Algorithm (APA).

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T \left\{ \delta e(i) + (1 - \delta) \text{sign}[e(i)] \right\} \quad (\text{RMN [29]}) \quad (2.9)$$

$$\begin{cases} \mathbf{w}_{0,i} = \mathbf{w}_{i-1} \\ \mathbf{w}_{k,i} = \mathbf{w}_{k-1,i} + \mu \mathbf{u}_i^T [d(i) - \mathbf{u}_i \mathbf{w}_{k-1,i}] \\ \mathbf{w}_i = \mathbf{w}_{K,i} \end{cases} \quad (\text{DR-LMS [70]}) \quad (2.10)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{U}_i^T \mathbf{e}_i \quad (\text{TRUE-LMS}^2 \text{ [24, 25]}) \quad (2.11)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{U}_i^T (\mathbf{U}_i \mathbf{U}_i^T + \epsilon \mathbf{I})^{-1} \mathbf{e}_i \quad (\text{APA [18, 71]}) \quad (2.12)$$

where $0 < \epsilon \ll 1$ is a regularization factor, $0 \leq \delta \leq 1$ is a mixing constant, $\text{sign}[\cdot]$ is the *signum* operator, $\mathbf{e}_i = [e_1(i) \ \cdots \ e_K(i)]^T = \mathbf{d}_i - \mathbf{y}_i$, with $\mathbf{y}_i = \mathbf{U}_i \mathbf{w}_{i-1}$,

$$\mathbf{P}_i = \lambda^{-1} \left[\mathbf{P}_{i-1} - \frac{\lambda^{-1} \mathbf{P}_{i-1} \mathbf{u}_i^T \mathbf{u}_i \mathbf{P}_{i-1}}{1 + \lambda^{-1} \mathbf{u}_i \mathbf{P}_{i-1} \mathbf{u}_i^T} \right], \quad \text{and} \quad \mathbf{G}_i = \text{diag} \left\{ \frac{|\mathbf{w}_{i-1}|}{\|\mathbf{w}_{i-1}\|_1 + \epsilon} \right\},$$

with a forgetting factor λ , $\mathbf{P}_{-1} = \epsilon^{-1} \mathbf{I}$, and $\|\cdot\|_1$ denoting the ℓ_1 -norm of a vector.

2.1 Figures of merit

Usually, the performance of AFs are analyzed in three different phases/conditions:

- *Transient performance*: refers to the initial phase of the filter operation, specially as to convergence rate phenomena;
- *Steady-state performance*: the behavior of the AF after convergence, sometimes characterized by its *misadjustment* [see (2.13)]; and
- *Tracking performance*: relevant in nonstationary scenarios, it analyzes the adaptive algorithm capability to track changes in the unknown system. It is usually considered a steady-state phenomenon [18].

²This algorithm is also confusingly referred to as DR-LMS in the literature [25]. The denomination TRUE-LMS, inspired by true gradient approaches in distributed optimization [72], was introduced in [73].

The most common figures of merit used in the mean and mean-square analyses of these phases are [18]

$$\begin{aligned}
\tilde{\mathbf{w}}_i &= \mathbf{w}_i^o - \mathbf{w}_i && (\text{coefficients error}) \\
e_a(i) &= \mathbf{u}_i(\mathbf{w}_i^o - \mathbf{w}_{i-1}) && (a \text{ priori error}) \\
e_p(i) &= \mathbf{u}_i(\mathbf{w}_i^o - \mathbf{w}_i) && (a \text{ posteriori error}) \\
E|\tilde{\mathbf{w}}_i| &&& (\text{mean coefficients error—bias}) \\
\text{MSE}(i) &= E e^2(i) && (\text{Mean-Square Error—MSE}) \\
\text{EMSE}(i) &= \zeta(i) = E e_a^2(i) && (\text{Excess Mean-Square Error—EMSE}) \\
\text{MSD}(i) &= E \|\tilde{\mathbf{w}}_{i-1}\|^2 && (\text{Mean-Square Deviation—MSD}) \\
\mathfrak{M} &= \frac{\text{EMSE}}{\sigma_v^2} && (\text{Misadjustment})
\end{aligned} \tag{2.13}$$

where the steady-state value of these quantities is represented by omitting the i index as in $\zeta = \text{EMSE} = \lim_{i \rightarrow \infty} \text{EMSE}(i)$. Notice that the *a priori* error refers to the error at the beginning of an iteration, before the filter updates the coefficient vector, whereas the *a posteriori* error refers to the updated $\mathbf{w}$. Both, however, are relative to the current system coefficients, $\mathbf{w}_i^o$. When the system is stationary, $\mathbf{w}_i^o = \mathbf{w}^o$ for all i , so that the *a priori* and *a posteriori* errors expressions simplify to $e_a(i) = \mathbf{u}_i \tilde{\mathbf{w}}_{i-1}$ and $e_p(i) = \mathbf{u}_i \tilde{\mathbf{w}}_i$. Throughout the simulations in this work, these performance measures are estimated using the average of 300 independent runs.

2.2 Energy conservation relation

The energy conservation relation (ECR) associates errors values between consecutive iterations of an adaptive algorithm. It holds for most AFs, in particular those that can be represented using recursions with error nonlinearities or data nonlinearities as in

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T g[e(i)] \quad \text{and} \quad \mathbf{w}_i = \mathbf{w}_{i-1} + \mu g[\mathbf{u}_i^T] e(i),$$

where $g[\cdot]$ is an arbitrary scalar function. Notice that this is the case of almost all AFs presented in (2.2)–(2.12). In these cases, the ECR can be stated as

$$\|\mathbf{w}_i^o - \mathbf{w}_i\|^2 + \frac{\mu}{\|\mathbf{u}_i\|^2} e_a^2(i) = \|\mathbf{w}_i^o - \mathbf{w}_{i-1}\|^2 + \frac{\mu}{\|\mathbf{u}_i\|^2} e_p^2(i), \tag{2.14}$$

assuming $\mathbf{u}_i \neq \mathbf{0}^T$, in which case it reduces to the trivial $\|\mathbf{w}_i^o - \mathbf{w}_i\|^2 = \|\mathbf{w}_i^o - \mathbf{w}_{i-1}\|^2$ [18]. It should be noted that the ECR is *exact*, i.e., no approximations or assumptions are used

Figure 4: Physical interpretation of the ECR: (a) general adaptive filter; (b) NLMS (total reflexion)

in its derivation.

One of the most interesting aspects of the ECR is its physical interpretation, namely as an optical system. This interpretation comes from comparing a rearranged version of (2.14) to Snell's law. For clarity's sake, only the stationary case is considered. Explicitly,

$$\underbrace{\|\tilde{\mathbf{w}}_{i-1}\|^2}_{n_{i-1}^2} \underbrace{\left[1 - \frac{(\mathbf{u}_i \tilde{\mathbf{w}}_{i-1})^2}{\|\mathbf{u}_i\|^2 \cdot \|\tilde{\mathbf{w}}_{i-1}\|^2}\right]}_{\sin^2(\theta_{i-1})} = \underbrace{\|\tilde{\mathbf{w}}_i\|^2}_{n_i^2} \underbrace{\left[1 - \frac{(\mathbf{u}_i \tilde{\mathbf{w}}_i)^2}{\|\mathbf{u}_i\|^2 \cdot \|\tilde{\mathbf{w}}_i\|^2}\right]}_{\sin^2(\theta_i)}, \quad (2.15)$$

where $e_a(i)$ and $e_p(i)$ were expanded to facilitate the interpretation. The relation between (2.15) and Snell's law is indicated under the equation and illustrated in Figure 4a.

An AF can therefore be interpreted as an interface between media $i-1$ and i oriented according to a normal vector $\mathbf{u}_i^T$. Its updates are then akin to the process of optical refraction. Indeed, the coefficient error vector $\tilde{\mathbf{w}}_{i-1}$ is a photon impinging on the AF surface from the $i-1$ -side at an angle that depends on its orientation relative to $\mathbf{u}_i$. Its speed and, consequently, the value of the medium's refractive index n_{i-1} , is $\|\tilde{\mathbf{w}}_{i-1}\|$. This photon is then refracted on the i -side with trajectory and speed determined by $\tilde{\mathbf{w}}_i$ in a similar manner [18].

Optical systems like those shown in Figure 4a also give rise to the well-known phenomenon of *total reflexion*. Indeed, there is a critical angle (function of the media refractive indices) at which, instead of entering medium i , the incoming photon is trapped on the interface. For this to happen, $\theta_i = \pi/2 \Rightarrow \sin(\theta_i) = 1 \Rightarrow e_p(i) = \mathbf{u}_i \tilde{\mathbf{w}}_i = 0$. Now, assume that the AF operates in a noiseless environment, such that $d(i) = \mathbf{u}_i \mathbf{w}^o$. Then, if the *a posteriori* error vanishes, so does the residual estimation error $r(i) = d(i) - \mathbf{u}_i \mathbf{w}_i$. It is a well known fact that for $\mu = 1$ and $\epsilon = 0$, this condition is enforced by the NLMS filter

in (2.4) [18]. Therefore, the NLMS filter can be interpreted as imposing the trajectory of the $\tilde{\mathbf{w}}_{i-1}$ photon such that it reaches the media interface at the critical angle (Figure 4b).

2.3 Concluding remarks

Stand-alone AFs are used in many applications, such as communications (e.g., channel equalization, network echo cancellation, beamforming), audio (e.g., speech echo cancellation, acoustic feedback control, noise control), and radar (e.g., adaptive array processing) [52, 60, 63, 74–76]. In most scenarios, their behavior is well understood and steady-state, tracking, and transient analyses are widely available in the literature. Nevertheless, they are not a panacea and display well-known limitations, usually involving a trade-off between steady-state performance, convergence rate, and computational complexity. Thus, the requirement for fast lock and low tracking error in communication systems, for instance, requires the use of adaptive algorithms with higher complexity. On the other hand, increasing battery life may require a reduction of the computational burden of algorithms, thus forcing convergence or misadjustment constraints to be relaxed. Combinations of AFs can be used to address these common compromises and provide adaptive filtering solutions that can meet more stringent requirements.

3 COMBINATIONS OF ADAPTIVE FILTERS

*Tous pour un, un pour tous.*¹

-- Alexandre Dumas, “Les trois mousquetaires”

In the literature, combinations of AFs are usually defined in a vague and often circular manner. For example, some authors discuss “combination schemes, where the outputs of several filters are mixed together to get an overall output of improved quality” [33] or talk about “combining the outputs of several different independently-run adaptive algorithms to achieve better performance than that of a single filter” [34]. These are statements of the purpose of combinations, not definitions. Others, rely on the relation to the mathematical concept of *linear combination* as in “weighted combination (or mixture) over all possible predictors” [77]. These citations are biased towards parallel combinations because they predate the incremental topology. Its introduction, however, has not changed this issue: “a set of AFs is aggregated via a supervisor, which attempts to achieve universal behavior” [41] or “aggregating a pool of AFs through mixing parameters, adaptive or not, and attempting to achieve universality” [73].

Indeed, it is hard to explain what a combination of AFs is in any way that does not involve words such as “mixture” or “aggregate”, which are simply synonyms. This is not an issue when it comes to their informal treatment: the term *combination* itself carries enough meaning for people to informally understand what a combination of AFs is. In contrast, the lack of a precise definition can hinder the formal study of combinations. In the sequel, this issue is addressed by proposing a formal definition of combinations of AFs, showing how it relates to each element that composes a combinations, and giving examples of its application.

3.1 A formal definition

In what follows, the AFs that compose a combination are called *component filters* or *components*. Recall that components are indexed using $n = 1, \dots, N$, whereas the data

¹All for one, one for all

Figure 5: Combination of adaptive filters

Figure 6: $\mathcal{FOB}$ description

pairs $\{\mathbf{u}_{i-k}, d(i-k)\}$ are index with $k \in \mathcal{K}$, for an index set $\mathcal{K}$ of size K . This distinction is fundamental for the discussion of DR methods in Section 3.3.

Definition 3.1 (Combination of adaptive filters). Given a pool of N AFs, their $M_n \times 1$ coefficient vectors $\mathbf{w}_{n,i}$, and a set of data pairs $\{\mathbf{u}_{i-k}, d(i-k)\}_{k \in \mathcal{K}}$, where $\mathcal{K} \subset \mathbb{Z}_+$ is an index set of size K (Figure 5), define the $M \times N$ coefficient matrix

$$\mathbf{\Omega}_i = \begin{bmatrix} \mathbf{w}_{1,i} & \mathbf{w}_{2,i} & \cdots & \mathbf{w}_{N,i} \end{bmatrix}, \quad (3.1)$$

where $M = \max_n M_n$ and the coefficients are zero-padded when necessary. Then, a combination of AFs is defined by three mappings (Figure 6):

- map $\mathcal{F}$ (*Forward*) updates the *a priori* coefficient matrix $\mathbf{\Omega}_a$ yielding $\mathbf{\Omega}_i$;
- map $\mathcal{O}$ (*Output*) evaluates the global coefficient vector $\mathbf{w}_i$ based on the current coefficients $\mathbf{\Omega}_i$;
- map $\mathcal{B}$ (*Backward*) determines the coefficients that will be updated by the components in the next iteration. In other words, it gives the “new” *a priori* coefficient matrix $\mathbf{\Omega}_a$.

The combination of AFs problem can then be stated as finding $\mathcal{F}$, $\mathcal{O}$, and $\mathcal{B}$ that meet the performance criteria of interest (faster convergence in a given scenario, increased robustness, etc.).

As it stands, Definition 3.1 does not specify the form of the elements that compose a combination. However, by using of abstract mappings, it accounts for the vast majority of combinations (if not all) without resorting to convoluted descriptions. Moreover, the generality of $\mathcal{FOB}$ descriptions (named after the eponymous mappings) does not hinder their validity and application, as the examples in Section 3.5 well illustrate. In fact, they can be used to represent structures from stand-alone AFs to adaptive networks (ANs,

see Section 3.6). It is the abstraction of the mechanisms involved in combinations that confers power to Definition 3.1.

Although their forms remain to be established, the maps described in Definition 3.1 establish all the parts one expects to find in a combination. Mapping $\mathcal{F}$ represents the coefficients updates of the *component filters*, thus determining both their adaptive algorithm and the *data distribution method* employed by the combination. The *supervisor*, responsible for determining the output of the combination, is identified with $\mathcal{O}$ and the *topology* that connects the AFs is effectively obtained from $\mathcal{B}$.

In what follows, each of these parts is analyzed in more details.

3.2 The topology

The topology of a combination plays an important role in its performance since it dictates how the component filters cooperate. Until recently, however, only parallel combinations of AFs were considered in the literature and little attention was given to the performance impact of the topology. The introduction of the incremental topology showed that combination design needs to consider factors beyond component filters and supervisors, although specific studies of topologies remain scarce (if there are any at all).

In this section, the topology of combinations is considered from a network perspective without taking the component filters into account. To do so, digraphs are introduced as tools to describe and analyze combination topologies (Sections 3.2.1 and 3.2.2). Then, the study of two important classes of topologies is motivated in the context of optimization algorithms (Section 3.2.3).

Before proceeding, however, the mappings $\mathcal{O}$ and $\mathcal{B}$ are constrained to be linear combinations of the components coefficients. Although the theory from the next sections could be extended to other types of mappings, using linear combinations is enough for our development and accounts for all combinations presented in this work.

3.2.1 Digraphs

A *directed graph* (or *digraph*) is an ordered pair $\mathcal{D} = (V, A)$ consisting of a non-empty finite set V of elements called *vertices* or *nodes* and a finite set A of ordered pairs of nodes called *arcs*¹. Notice that, in contrast to the definition of edges in a graph, arcs are

¹This definition of digraph allows parallel arcs (distinct arcs from and to the same nodes) and loops (arcs from and to a same node), which are usually not allowed in digraphs. Such a structure

$$V = \{r, s, t, u, v, w\}$$

$$A = \{(r, u), (s, r), (s, t), (u, s), (u, v), (v, s), (v, t), (v, w), (w, w)\}$$

Figure 7: A digraph $\mathcal{D} = (V, A)$

ordered pairs. The *degree* of $\mathcal{D}$, $|\mathcal{D}|$, is the cardinality of its vertex set V . An arc (u, v) is drawn as an arrow (Figure 7) that starts in u (the *tail*) and ends in v (the *head*). It is said that the arc (u, v) goes *from* u *to* v . The *in-degree* (*out-degree*) of a vertex is the number of arcs of which it is a tail (head), i.e., the number of arcs that start (end) at the vertex. A *weighted digraph* is a digraph $\mathcal{D}$ along with a mapping ϕ that attributes weights to the arcs in A . In this work, instead of the customary $\phi : A \rightarrow \mathbb{R}$, the weights are taken as polynomials in z^{-1} with real coefficients.

A *path* on $\mathcal{D}$ is defined as an alternating sequence $x_1 a_1 x_2 a_2 \dots a_{n-1} x_n$ of distinct nodes x_j and arcs a_j such that a_j goes from x_j to x_{j+1} , for $j = 1, \dots, n-1$. Node $u \in V$ is said to be *connected* to $v \in V$ if there is at least one path between them. Already, the concept of connectivity is not as straightforward in digraphs as it is in undirected graphs. For instance, the ordered pair (r, t) in Figure 7 is connected, whereas (t, r) is not. For this reason, two distinct forms of connectivity are typically use with digraphs. Namely, a digraph is said be *strongly connected* if it contains a path between all pair of nodes $V \times V$. A more relaxed definition states that a digraph is *weakly connected* if the graph obtained by replacing all of its arcs with undirected edges is connected. It is easy to see that all strongly connected digraphs are weakly connected, although the converse is not necessarily true [78]. Figure 7, for instance, is weakly connected but not strongly connect.

3.2.2 Topologies and digraphs

The topology of a combination determines how its component filters are connected to each other and to the output of the combination. These mechanisms are represented in Definition 3.1 by the maps $\mathcal{B}$ and $\mathcal{O}$, respectively. In other words, the topology is the network used to transfer coefficients in the combination. This view motivates the use of graph-like structures to describe the combination topology, as is usually done in the study of networks [79–81].

is actually called a *directed pseudograph* [78]. However, this distinction is not necessary in the current work and for the sake of conciseness, the shorter “digraph” will be used instead.

Figure 8: Example digraphs of combination topologies: (a) stand-alone adaptive filter; (b) arbitrary topology.

In the combination of AFs context, a topology is therefore a weighted digraph of degree $N + 1$ whose node set V includes a node for each component and one *output node* that represents the output of the combination. They are differentiated by drawing the former as circles and the latter as a triangle. Each arc in this digraph corresponds to a weighted transfer of the current coefficient vector. Explicitly, an arc with weight ω exiting node n carries the coefficients $\omega \mathbf{w}_{n,i}$. The *a priori* coefficients of a component is evaluated by combining all arcs entering its node. For instance, consider the component filters $\{m, n, p\}$ connected by the arcs (m, p) and (n, p) with weights ω_m and ω_n . Such an arrangement is taken to mean $\mathbf{w}_{p,a} = \omega_m \mathbf{w}_{m,i} + \omega_n \mathbf{w}_{n,i}$. The arcs entering the output node are combined in a similar fashion, but to form the current global coefficients $\mathbf{w}_i$.

These conventions are illustrated in Figure 8. Figure 8a shows a topology equivalent to a stand-alone AF for which $\mathbf{w}_{1,a} = \mathbf{w}_{i-1}$ and $\mathbf{w}_i = \mathbf{w}_{1,i}$. An arbitrary topology is shown in Figure 8b, where $\mathbf{w}_{1,a} = 0.8\mathbf{w}_{i-1} + 0.2\mathbf{w}_{2,i-1}$, $\mathbf{w}_{2,a} = \mathbf{w}_{2,i-1}$, and $\mathbf{w}_i = 0.5\mathbf{w}_{1,i} + 0.5\mathbf{w}_{2,i}$. Notice that when an arc is drawn unmarked its weight is considered to be one.

Clearly, not all digraphs represent topologies. Firstly, the digraph has to be weakly connected. If not, then at least one component filter cannot reach the output and can therefore be removed with no effect on the performance of combination. Repeating this procedure for all nodes disconnected from the output yield a weakly connected digraph that forms a topology indistinguishable from the original one. Secondly, the in-degree of any node must be at least one. Indeed, if node n has a null in-degree, then $w_{n,a} = 0$ always and the AF is restarted every iteration, which defies the recursive nature of these algorithms. Finally, when the topology is dynamic (changes with i), these requisites need only be met by the union of its possible digraphs.

3.2.3 The parallel/incremental dichotomy

From Section 3.2.2, it is straightforward to see that even for combinations with a handful of component filters, the number of possible topologies is overwhelming. Hence, a direct approach to the study of these structures rapidly becomes intractable. However, two classes of topology have gained prominence in the literature, namely, the parallel and incremental topologies [33, 35, 41, 73]. These categories can be distinguished in terms of the *synchronicity* of their components updates:

Definition 3.2 (Parallel topology). A topology is said to be *parallel* when the updates of its component filters can be evaluated simultaneously (synchronous components), i.e., when each component filter only depends on past information from other filters;

Definition 3.3 (Incremental topology). A topology is said to be *incremental* when the updates of its component filters depends on current data from previous filters (asynchronous components), i.e., component n depends on current information from components $m < n$. In other words, the components are *daisy-chained*, as it is commonly said in electronics [82].

Separating topologies in only two classes may seem too restrictive at first, specially considering their number. However, it is usually the case that more complex configurations can be reduced to mixtures of these two categories. What is more, this dichotomy appears in several fields, from electrical circuits to data streaming (although sometimes under different names: parallel/series or parallel/serial). Specifically, the parallel/incremental dichotomy is pervasive in numerical linear algebra and optimization algorithms used to solve problems similar to those found in adaptive filtering. These instances justify analyzing combinations only for purely parallel and purely incremental topologies.

Consider, for example, the problem of solving a linear system of equations, i.e., finding an $\mathbf{x} \in \mathbb{R}^M$ that satisfies

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.2}$$

with $\mathbf{A} \in \mathbb{R}^{N \times M}$ and $\mathbf{b} \in \mathbb{R}^N$. Recall that this is the problem an adaptive algorithm is ultimately solving [18, 19]. When $\mathbf{A}$ is a square matrix ($M = N$) and $\text{diag}(\mathbf{A}) \neq 0$, this system can be solved using the Jacobi or Gauss-Seidel methods [83], which are in fact parallel and incremental counterparts. This fact is clear from the algorithms iterations,

which for the i -th iteration read

$$x_{m,i} = \frac{1}{a_n} \left(b_n - \sum_{p=1}^{n-1} a_{np} x_{p,i-1} - \sum_{p=n+1}^N a_{np} x_{p,i-1} \right) \quad (\text{Jacobi}) \quad (3.3a)$$

$$x_{m,i} = \frac{1}{a_n} \left(b_n - \sum_{p=1}^{n-1} a_{np} x_{p,i} - \sum_{p=n+1}^N a_{np} x_{p,i-1} \right) \quad (\text{Gauss-Seidel}) \quad (3.3b)$$

for an initial guess $\mathbf{x}_0$ and with $\mathbf{A} = [a_{mn}]$, $\mathbf{b} = [b_m]$, and $\mathbf{x}_i = [x_{m,i}]$. Indeed, the Jacobi updates depend only on elements of $\mathbf{x}_{i-1}$ and therefore evaluates $\mathbf{x}_i$ in a parallel fashion. On the other hand, the Gauss-Seidel algorithm relies on the current value of some elements of $\mathbf{x}$ to update $x_{n,i}$, specifically $\{x_{p,i}\}$, $p < n$. Hence, it can only evaluate $\mathbf{x}_i$ incrementally.

Another important pair of algorithms for solving (3.2) that gives rise to a parallel/incremental dichotomy are known as row projection methods and are usually used to solve overdetermined systems ($N > M$). Their name comes from the fact that they operate on row slices of the original problem $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_N]^T$ and $\mathbf{b} = [b_n]$, projecting candidate solutions onto the affine sets $H_n = \{\mathbf{x} : \mathbf{a}_n \mathbf{x} = b_n\}$. Depending on whether the projections occur over all H_n simultaneously or sequentially, the technique is named Cimmino or Kaczmarz respectively [84]. Explicitly, defining $\mathcal{P}_n$ as the projection operator onto H_n ,

$$\mathbf{x}_i = \sum_{n=1}^N \eta_n [\mathbf{x}_{i-1} - 2\mathcal{P}_n(\mathbf{x}_{i-1})] \quad (\text{Cimmino}) \quad (3.4a)$$

$$\mathbf{x}_i = \mathcal{P}_N \left[\cdots \mathcal{P}_2 [\mathcal{P}_1(\mathbf{x}_{i-1})] \right] \quad (\text{Kaczmarz}) \quad (3.4b)$$

with $\sum \eta_n = 1$ and $\eta_n > 0$.

The concept of sequential projections in (3.4) is also at the heart of a well-known solution to convex feasibility problems, which are found in adaptive algorithms such as the APA or set-membership (SM) filters [19]. *Projection onto convex sets* (POCS) methods are similar to row projection algorithms, but use arbitrary closed convex set instead of simply the affine H_n [85]. POCS is also at the origin of *proximal algorithms*, which are heavily used in many areas of non-smooth convex optimization [86].

As a last example of parallel/incremental dichotomy, consider the case of a distributed optimization problem whose objective function can be decomposed as a sum of functions as in

$$h(\mathbf{x}) = \sum_{n=1}^N h_n(\mathbf{x}), \quad (3.5)$$

where $h, h_n : \mathbb{R}^M \rightarrow \mathbb{R}$. When solving (3.5) using gradient descent, two methods can be derived depending on how the gradient of the f_n is evaluated [72]

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \alpha_i \sum_{n=1}^N \nabla h_n(\mathbf{x}_{i-1}) \quad (\text{True gradient descent}) \quad (3.6a)$$

$$\begin{cases} \psi_0 = \mathbf{x}_{i-1} \\ \psi_n = \psi_{n-1} - \alpha_i \nabla h_n(\psi_{n-1}), \quad n = 1, \dots, N \\ \mathbf{x}_i = \psi_N \end{cases} \quad (\text{Incremental gradient descent}) \quad (3.6b)$$

Notice that (3.6) is at the origin of some nomenclature used in the combination of AFs and ANs literature.

These examples, among many others, support the classification and analysis of combinations in terms of parallel and incremental topologies.

3.3 The data reusing method

In the adaptive filtering literature, DR consists of either using $K > 1$ times a single data pair $\{\mathbf{u}_i, d(i)\}$ or operating over a set of past data pairs $\{\mathbf{U}_i, \mathbf{d}_i\}$. Both techniques have been successfully used to improve the performance of stand-alone AFs, albeit at the cost of increasing their complexity and memory requirements [87]. These algorithms are particularly appropriate in applications where data is intermittent, such as communication systems, in which the rate of signaling is limited by bandwidth restrictions, or speech processing [87, 88]. Instead of idling between samples, the adaptive algorithm continues to operate over the available data to improve its estimates. Moreover, these AFs are able to trade off complexity and performance by choosing K , which is invaluable in many scenarios [24].

Embedding DR in combinations of AFs is an important step towards formulating high performance algorithms. Moreover, they allow most stand-alone DR AFs to be cast as combinations, providing new forms of analysis and generalization. In the context of combinations, however, differentiating between forms of data reuse is paramount. In fact, according to the definition for stand-alone AFs all combinations reuse data, since several AFs operate over the data every iteration. The terms *data sharing* and *data buffering* are therefore used to unambiguously refer to the different methods of reusing data.

In data sharing combinations, all component filters *share* a single data pair $\{\mathbf{u}_i, d(i)\}$. Until recently, this was the only form of DR in the combination of AFs literature [73]

and it remains by far the most common [33–36, 41–44, 59, 77, 89–96]. Overall, there is little advantage in using any other DR method when the input signal is white. The additional implementation and memory cost is unwarranted given the negligible performance gains [44, 45]. Correlated inputs, however, significantly degrade the convergence rate of combinations, specially when the component filters do not have input data decorrelating properties, such as the NLMS or the APA [18, 23, 25, 62, 71].

Data buffering combinations can be used to improve the transient behavior in that case. As in stand-alone adaptive filtering, this technique *buffers* past data pairs $\{\mathbf{u}_{i-k}, d(i-k)\}$ into the rows of $\{\mathbf{U}_i, \mathbf{d}_i\}$ and distributes them between the component filters. The input buffer usually contains the last K input samples, although this is not necessary. For instance, a buffer can be defined using $k = \{k'D : k' = 0, \dots, K-1\}$ for some constant $D \in \mathbb{N}$, similar to [62]. More generally, k can be chosen from any size- K subset of $\mathbb{Z}_+$.

An input buffer alone, however, does not fully specify a data buffering method, which also needs to allocate a data pair to each component filter. Decoupling *data storage* from *data usage* allows for more complex DR solutions than those usually found in stand-alone DR AFs [22, 87]. Indeed, the choice of protocols goes from the simple use of the n -th row of $\{\mathbf{U}_i, \mathbf{d}_i\}$ by the n -th filter when $N = K$ to randomized allocation of data pairs. Moreover, this framework separates memory usage from algorithm complexity, something not found in stand-alone DR adaptive algorithms.

A last DR strategy worth mentioning is *refiltering*. Here, output values from previous components are used as the input data of the current filter. These could be the actual output of the AFs $y_n(j)$ but also their output estimation errors $e_n(j)$ or their coefficients $\mathbf{w}_{m,j}$. For example, AFs in tandem use past error from other AFs as their desired signals, i.e., $d_n(i) = e_{n-1}(i - \Delta)$, for some delay $\Delta > 0$ [97]. Notice that, even if coefficient vectors were used as inputs it would not incur in a topology change. The combination topology determines a connection between coefficients and has no relation to the data over which the filters operate. Due to its lack of use, the remainder of this work considers only the data sharing and data buffering methods.

In the sequel, DR is cast as a data distribution protocol and embedded in a digraph description compatible with the one from Section 3.2. Example strategies are provided to illustrate this concept.

Figure 9: Data distribution networks: (a) data sharing; (b) stand-alone data reusing adaptive filter; (c) circular buffer; (d) randomized.

3.3.1 Data distribution networks

Data sharing and data buffering were described as two steps processes: first, data pairs are stored; then, data pairs are allocated to the component filters. Hence, one way of describing them is defining a set of indices $\mathcal{K}$ that determines a buffer from the data pairs $\{\mathbf{u}_{i-k}, d(i-k)\}_{k \in \mathcal{K}}$ and a map between this buffer and the AFs. Naturally, the data sharing case is trivial as $K = 1$ and $\mathcal{K} = \{0\}$ for the buffer and all components get the same data pair. Notice, however, that what these steps effectively do is connect a data source to each filter in the combination. They can therefore be collapsed into a network that specifies how information reaches the component filters, i.e., a *data distribution network*.

Such network can be described under the same formalism as combination topologies (Section 3.2), i.e., using weighted digraphs. To do so, define a node that represents the data source (denoted by a square). Then, using arcs as delay lines, connect this node to each component filter, effectively determining the data pair over which they operate and thus the DR strategy. Figure 9 illustrates the digraphs for data sharing (a) and the data distribution method found in stand-alone DR AFs (b). Note that arcs only exit the data source node, which must always have an in-degree of zero. Three data buffering methods are presented below to illustrate the application of the proposed formalism.

Example 3.1 (Circular buffer or barrel shifter). Using the data pair $\{\mathbf{u}_{i-n-1}, d(i-n-1)\}$ for the n -th component (i.e., $k = n$) as in stand-alone DR AFs (Figure 9b) restricts the

number of components in the combination to $N = K$. This limits the performance achievable by combinations, especially when less complex adaptive algorithms are used. In [73], this issue was addressed by using a *circular buffer* (also known as a *barrel shifter* in the digital hardware milieu [82, 98]). Explicitly, the n -th component filter operates over the row $k = (n - 1) \bmod K$ of $\{\mathbf{U}_i, \mathbf{d}_i\}$. The data distribution network induced by this method is illustrated in Figure 9c considering the buffer holds the K last data pair ($\mathcal{K} = \{0, \dots, K - 1\}$). This method has been shown to improve the performance of incremental combinations to the point where they outperform the APA with considerably lower complexity [45, 47, 73].

Example 3.2 (Data selective distribution). Stand-alone *data selective* AFs (also known as *set-membership*, SM) are motivated by assuming that the additive noise $v(i)$ is bounded and determining or estimating a feasibility set for the coefficient vector that meets the corresponding error bound specifications [19]. These adaptive algorithms are more flexible in managing computational resources and have been successfully used in a myriad of applications [19, 99–101]. SM combinations can be obtained by using a data distribution network that only selects data pairs that yield errors larger than some threshold. In a parallel combination where $N < K$, for example, the strategy could allocate only the data pairs relative to the N largest errors.

Example 3.3 (Randomized data distribution). The row projection methods presented in Section 3.2.3 are intimately related to data buffering combinations of AFs. Indeed, parallel and incremental combinations operating over the rows of $\{\mathbf{U}_i, \mathbf{d}_i\}$ are similar to using Cimmino and Kaczmarz algorithms, respectively, on a system of equations such as $\mathbf{U}_i \mathbf{w}_i = \mathbf{d}_i$. The Kaczmarz algorithm, in particular, operates sequentially over the rows of this system, such that its performance may be affected by the order of the equations. To eliminate this undesirable effect, a randomized version of the Kaczmarz algorithm was proposed and analyzed in [102–104]. However, instead of selecting rows of $\mathbf{U}_i$ uniformly at random, they are drawn with probabilities proportional to their Euclidian norms. One way to motivate this strategy is in terms of SNR improvements: in a system identification scenario, when the power of the regressor $\mathbf{u}_i$ fluctuates up, there is good chance that the instantaneous SNR of the measurements $d(i) = \mathbf{u}_i \mathbf{w}^o + v(i)$ also increases.

A similar version of incremental combinations can be obtained by appropriately choosing the distribution f_Δ in Figure 9d. More complex strategies to select f_Δ can also be used, such as assigning data pair probabilities using upper confidence bound (UCB) algorithms from multi-armed bandit problems [105]. Randomized SM strategies can also be devised.

3.4 The supervisor

The supervisor or supervising rule is the element that evaluates the output of the combination (i.e., the mapping $\mathcal{O}$ in Definition 3.1). In other words, it is the part responsible for actually *combining* the AFs. It has therefore a significant effect on the overall performance of mixtures. Not surprisingly then, supervisors have attracted considerable attention from the combinations of AFs community, lending their names to several structures [33, 35, 91, 92, 106–108].

Supervising rules, however, are not a panacea. Other elements of the combination, such as the component filters and the topology, have as big an impact on its performance and can impose limitations to the effectiveness of the entire structure that the supervisor cannot overcome. The convergence stagnation of parallel-independent combinations is a good example of such an issues and is investigated in details in Section 5.2.1.

The action of the supervisor over the n -th component is represented by the supervising parameter $\eta_n(i)$. Originally, the mapping $\mathcal{O}$ in Definition 3.1 does not define how these parameter are used to transform the AFs to the output of the combination. Section 3.2, however, restricted supervising rules to the class of linear combinations of coefficients to make the description of topologies more concrete. Thus, this work considers the $\{\eta_n(i)\}$ to be linear combiners and the output of the combination to be evaluated as,

$$\mathbf{w}_i = \sum_{n=1}^N \eta_n(i) \mathbf{w}_{n,i}.$$

As far as supervisors go, this is by far the most common case, specially for parallel combinations [33, 35, 91, 92, 106–108]. A notable exception is the incremental supervisor used in [41, 43, 44].

Supervising rules, their characteristics, and constraints are intrinsically connected to the topology of the combination. They are therefore discussed in more details for the parallel (Chapter 5) and incremental (Chapter 6) cases separately. Nevertheless, their adaptation methods can be classified in five broad categories:

- *Unsupervised* (“static”): unsupervised combinations use static supervising parameters. Hence, the $\{\eta_n(i)\}$ are designed *a priori* and do not change during the operation of the combination. These supervisors are typically found in incremental combinations [43, 45, 47, 73] or when casting adaptive algorithms as combinations (see Chapter 8).

- *Statistical*: some rules update the supervising parameters using a statistical method, such as Bayesian filtering or maximum *a posteriori* probability (MAP) [8]. They were more common in the earlier days of combinations of AFs [77, 93–95].
- *Error statistics*: this set of techniques relies on error statistics to update the supervising parameters. Usually, they are based on estimates of the global MSE, but instantaneous values and higher-order moments have also been used (e.g., [26, 27]). The $\{\eta_n(i)\}$ are sometimes evaluated using an activation function [35, 41, 44, 106].
- *Stochastic gradient descent*: inspired by individual AFs, stochastic gradient descent algorithms can be used to update the supervising parameters to minimize some cost function, usually the global MSE. This technique has been shown to be very effective and is ubiquitous in parallel combinations [33, 35, 39, 109, 110].
- *Newton’s method* (“normalized”): the normalized version of stochastic gradient descent rules were proposed to improve the robustness and reduce the variance of the supervising parameters. They are usually considered to be more effective than their unnormalized counterparts [39, 91].

3.5 An $\mathcal{FOB}$ illustration

To illustrate the use of Definition 3.1, an example $\mathcal{FOB}$ description is provided in this section. Even though the mappings used are fairly straightforward, they account for a comprehensive set of combinations, with different components, DR methods, supervisors, and topologies. Notable exceptions are combinations that employ refiltering and incremental combinations supervised as in [41].

Explicitly, define the mappings

$$\mathcal{F} : \quad \mathbf{w}_{n,i} = \mathbf{w}_{n,a} + \mathbf{H}_{n,i} \mathbf{U}_i^T \mathbf{B}_{n,i} g_n(\mathbf{d}_i - \mathbf{U}_i \mathbf{w}_{n,a}) \quad (3.7a)$$

$$\mathcal{O} : \quad \mathbf{w}_i = \boldsymbol{\Omega}_i \boldsymbol{\eta}_i \quad (3.7b)$$

$$\mathcal{B} : \quad \boldsymbol{\Omega}_a = [\boldsymbol{\Omega}_i \quad \mathbf{w}_i] \mathbf{Z}_i^{-1}, \quad (3.7c)$$

where $\mathbf{w}_{n,a}$ is the n -th column of $\boldsymbol{\Omega}_a$, $\mathbf{B}_{n,i} = \mathbf{b}_k \mathbf{b}_k^T$, $\mathbf{b}_k$ is the k -th vector of the canonical base of $\mathbb{R}^K$, $\mathbf{H}_{n,i}$ is an $M \times M$ step size matrix, $\mathbf{Z}_i^{-1}$ is an $(N+1) \times N$ polynomial matrix in z^{-1} , the *unitary delay*, and $\boldsymbol{\eta}_i$ is the $N \times 1$ vector that captures the supervising parameters $\{\eta_n(i)\}$. The functions $g_n[\cdot]$ are possibly nonlinear and *diagonal*, i.e., they are applied element-wise to vectors or matrices [111].

Before proceeding, notice that the adjacency matrix of the topology digraph is readily available from $\mathcal{O}$ and $\mathcal{B}$ (more specifically, $\boldsymbol{\eta}_i$ and $\mathbf{Z}_i^{-1}$) in (3.7). Indeed, recall that the topology digraph is akin to a mapping $[\boldsymbol{\Omega}_i \ \mathbf{w}_i] \rightarrow [\boldsymbol{\Omega}_a \ \mathbf{w}_{i-1}]$, which can be derived from (3.7) as

$$\mathbf{A}_i = [\mathbf{I} \ \boldsymbol{\eta}_i] \left[\begin{array}{c|c} \mathbf{Z}_i^{-1} & \mathbf{0} \\ \hline & z^{-1} \end{array} \right]. \quad (3.8)$$

Note that $\mathbf{A}_i$ is iteration-dependent, which accounts for both dynamic topologies and adaptive supervising rules.

In the sequel, example uses of (3.7) are presented to illustrate the versatility of $\mathcal{FOB}$ descriptions and suggest that some stand-alone adaptive algorithms can be cast as combinations of AFs, an avenue further investigated in Chapter 8. Without loss of generality, assume $g_n[x] = x$ and $\mathbf{H}_n = \mu_n \mathbf{I}$, so that the components are all LMS filters.

Example 3.4 (LMS+LMS with convex supervisor [33]). This combination can be written as

$$\begin{aligned} a(i) &= a(i-1) + \mu_a e(i) [y_1(i) - y_2(i)] \eta(i) [1 - \eta(i)] \\ \eta(i) &= \frac{1}{1 + e^{-a(i)}} \\ \mathbf{w}_{n,i} &= \mathbf{w}_{n,i-1} + \mu_n \mathbf{u}_i^T [d(i) - \mathbf{u}_i \mathbf{w}_{n,i-1}], \quad n = 1, 2 \\ \mathbf{w}_i &= \eta(i) \mathbf{w}_{1,i} + [1 - \eta(i)] \mathbf{w}_{2,i}, \end{aligned} \quad (3.9)$$

with $e(i) = d(i) - \mathbf{u}_i \mathbf{w}_{i-1}$ and $a(i) \in [-a_{\max}, a_{\max}]$.

Its $\mathcal{FOB}$ description is obtained from (3.7) using $N = 2$, $K = 1$, $M_n = M$,

$$\mathbf{Z}^{-1} = \begin{bmatrix} z^{-1} & 0 \\ 0 & z^{-1} \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\eta}_i = \begin{bmatrix} \eta(i) \\ 1 - \eta(i) \end{bmatrix},$$

where $\eta(i)$ is evaluated as in (3.9). Note that despite its matrix form, $\mathcal{F}$ and $\mathcal{B}$ are equivalent to the recursion of two independent LMS filters.

Example 3.5 ($f\{\text{LMS} + \text{LMS}\}$ [42]). The LMS + LMS with cyclic coefficients feedback can be obtained by replacing $\mathbf{w}_{n,i-1}$ with $\mathbf{w}_{n,a}$ in (3.9) and defining

$$\mathbf{w}_{n,a} = \delta_L(i) \mathbf{w}_{i-1} + [1 - \delta_L(i)] \mathbf{w}_{n,i-1}, \quad (3.10)$$

where $\delta_L(i) = \sum_{r \in \mathbb{N}} \delta(i - rL)$ is an impulse train of period L and $\delta(i)$ is the Kronecker delta [42]. Its $\mathcal{FOB}$ description is identical to the one from Example 3.4, except that $\mathbf{Z}^{-1}$

is now a function of i . Explicitly,

$$\mathbf{Z}_i^{-1} = \begin{bmatrix} 1 - \delta_L(i) & 0 \\ 0 & 1 - \delta_L(i) \\ \delta_L(i) & \delta_L(i) \end{bmatrix} z^{-1}.$$

Indeed, $\mathbf{Z}_i^{-1}$ represents the dynamic topology of a parallel combination with cyclic coefficients feedback.

Example 3.6 (TRUE-LMS [24, 25]). The TRUE-LMS is a stand-alone DR adaptive algorithm and is not usually considered a combination. Using (3.7) to describe its recursion shows not only the versatility of $\mathcal{FOB}$ descriptions, but also that this AF can be cast as a combination of LMS filters [73]. Indeed, the recursion of the TRUE-LMS, given in (2.11) as

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{U}_i^T \mathbf{e}_i$$

can be described using (3.7) by choosing $K = N$, $\mathbf{B}_{n,i} = \mathbf{b}_n \mathbf{b}_n^T$, $M_n = M$,

$$\mathbf{Z}^{-1} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ z^{-1} & \cdots & z^{-1} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\eta}_i = \frac{1}{N} \cdot \mathbf{1},$$

where $\mathbf{1}$ is a column vector of ones.

Example 3.7 (DR- $\{\text{LMS}\}^N$ [73]). For $k = (n-1) \bmod K$, the data buffering incremental combination of N LMS filters is given by [73]

$$\begin{aligned} \mathbf{w}_{0,i} &= \mathbf{w}_{i-1} \\ \mathbf{w}_{n,i} &= \mathbf{w}_{n-1,i} + \mu_n \mathbf{u}_{i-k}^T [d(i-k) - \mathbf{u}_{i-k} \mathbf{w}_{n-1,i}], \quad n = 1, \dots, N \\ \mathbf{w}_i &= \mathbf{w}_{N,i}. \end{aligned}$$

It can be obtained from (3.7) using $M_n = M$ and $\mathbf{B}_{n,i} = \mathbf{b}_k \mathbf{b}_k^T$, with $k = (n-1) \bmod K$,

$$\mathbf{Z}^{-1} = \begin{bmatrix} 0 & & & \\ \vdots & \mathbf{I} & & \\ 0 & 0 & \cdots & 0 \\ z^{-1} & 0 & \cdots & 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\eta}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Notice that this choice of $\mathbf{Z}^{-1}$ couples the component filters recursions. Hence, contrary to the previous examples, the recursions in $\mathcal{F}$ cannot be evaluated simultaneously, thus yielding an incremental topology. Taking $N = 2$ and $K = 1$, the data sharing combination

from [41] can be recovered.

3.6 Relation to adaptive networks

ANs consist of a collection of agents (*nodes*) with adaptation and learning capabilities that are linked through a topology and interact to solve inference and optimization problems in a distributed and online manner. More specifically, each node estimates a set of parameters using an AF that operates on local data. Information from neighboring nodes is aggregated by means of combination weights and also exploited by the local AF [3, 4, 6, 112–115].

This description resembles the ones for combinations of AFs presented in the beginning of this chapter. They both involve a pool of AFs, each operating on individual data pairs, and some mixture of their estimates. ANs, however, are usually thought of as distributed entities, while combinations of AFs are considered local. Nevertheless, this need not be the case. An AN can be made of “virtual nodes” in a single processor, which is not uncommon with graph algorithms applied to image processing [116]. Much the same way, the component filters of combinations can be physically distributed, as would be the case if the combination were implemented in a highly parallel processing framework such as Apache Spark [117].

There is, however, one important distinction between combinations of AFs and ANs: their data source model. This is directly connected to their goal as signal processing algorithms. Combinations of AFs were proposed to improve the performance of adaptive algorithms and address issues such as transient/steady-state trade-off and robustness. Hence, just as stand-alone AFs, they operate over a single data source. On the other hand, the goal of ANs is to solve distributed estimation and optimization problems, such as those in wireless sensor networks [79]. Therefore, they rely on different sources observed by different nodes. In a way, combinations of AFs are replacements for adaptive algorithms, whereas ANs use these adaptive algorithms to solve larger problems.

These observations suggest that the $\mathcal{FOB}$ description from Definition 3.1 can be applied to ANs with very little modification. The first important distinction is that, in most applications, ANs do not have a single output. Indeed, it would defy the purpose of a fully distributed technique. Hence, the mapping $\mathcal{O}$ only has meaning in analyses, where the average network performance is typically used as a figure of merit [3, 4, 6, 112–115]. The second difference has to do with the data pairs over which each AF operates. Namely,

Figure 10: $\mathcal{FOB}$ description of adaptive networks: (a) incremental and (b) diffusion.

instead of delayed versions of a single input $\{\mathbf{u}_i, d(i)\}$, the buffers $\mathbf{U}_i$ and $\mathbf{d}_i$ in (3.7) are formed by observations from different sources. Explicitly, $\mathcal{FOB}$ descriptions of ANs can be derived from

$$\mathcal{F} : \mathbf{w}_{n,i} = \mathbf{w}_{n,a} + \mu_n \mathbf{u}_{n,i}^T (\mathbf{d}_n(i) - \mathbf{u}_{n,i} \mathbf{w}_{n,a}) \quad (3.11a)$$

$$\mathcal{B} : \Omega_a = \Omega_i \mathbf{Z}_i^{-1}, \quad (3.11b)$$

where $\{\mathbf{u}_{n,i}, d_n(i)\}$ is the data pair observed by node n . For clarity, only ANs with LMS nodes were considered. The use of (3.11) is illustrated in the examples below.

Example 3.8 (Incremental AN [3]). From a topology viewpoint, an incremental AN is similar to the incremental combinations from Example 3.7. Indeed, its $\mathcal{FOB}$ description is derived from (3.11) using

$$\mathbf{Z}^{-1} = \begin{bmatrix} 0 & & \\ \vdots & \mathbf{I} & \\ z^{-1} & \dots & 0 \end{bmatrix}.$$

The digraph representing this combination can be found in Figure 10a. Notice that the data distribution network now has N data sources.

Example 3.9 (Diffusion AN [4]). An LMS adaptive diffusion network is described as²

$$\boldsymbol{\psi}_{n,i-1} = c_{nn} \mathbf{w}_{n,i-1} + \sum_{\ell \in \mathcal{N}_n} c_{\ell n} \mathbf{w}_{\ell,i-1} \quad (3.12a)$$

$$\mathbf{w}_{n,i} = \boldsymbol{\psi}_{n,i-1} + \mu_n \mathbf{u}_{n,i}^* [d_n(i) - \mathbf{u}_{n,i} \boldsymbol{\psi}_{n,i-1}], \quad (3.12b)$$

²This formulation is sometimes referred to as Combine-Then-Adapt (CTA). An alternative formulation, the Adapt-Then-Combine (ATC), can be obtained by exchanging $\boldsymbol{\psi} \leftrightarrow \mathbf{w}$ and modifying the output variable. Given that they are equivalent [6], the original CTA from [4] is presented here.

where $\mathcal{N}_n$ denotes the neighborhood of node n , i.e., the set of nodes that are connected to n . Collecting the weights c_{mn} from (3.12a), with $c_{mn} = 0$ if nodes m and n are not connected, into a matrix $\mathbf{C}$, the $\mathcal{FOB}$ description (Figure 10b) of the diffusion AN (3.12) is obtained from (3.11) using

$$\mathbf{Z}^{-1} = \mathbf{C}z^{-1}.$$

Notice that the aggregated coefficients $\{\mathbf{p}\mathbf{s}\mathbf{i}_{n,i-1}\}$ in (3.12) are, in fact, the *a priori* coefficients $\mathbf{w}_{n,a}$ of the individual AFs.

3.7 Concluding remarks

This chapter presents a first attempt at formalizing combinations of AFs. In doing so, it provides a unified description of existing combinations which simultaneously bridges the gap between combinations of AFs and both ANs and stand-alone AFs (a topic that is explored in more details in Chapter 8). Moreover, the network viewpoint put forward in this has started the explicit study of topology and DR method, topics still scarcely explored in the literature. Nevertheless, these are still recent developments in the theory of combinations of AFs and, at this point, their application remains descriptive (i.e., the use of this formalism in analyses is still an open question). Hence, the analyses in this work forego the $\mathcal{FOB}$ formalism proposed in this chapter, relying mostly on the parallel-incremental dichotomy.

4 TAXONOMY AND REVIEW

*Le savant doit ordonner; on fait la science avec des faits comme une maison avec des pierres; mais une accumulation de faits n'est pas plus une science qu'un tas de pierres n'est une maison.*¹

-- Henri Poincaré, “La Science et l’Hypothèse”

4.1 Timeline

The use of combinations in the adaptive filtering literature dates back to the 1980s (Figure 11). It appears that the first adaptive algorithm that can be considered a combination was proposed by Ferrara and Widrow in 1981 [118]. In this paper, they developed the idea of a *time-sequenced adaptive filter* (TSAF), in which an LMS filter would sequentially update a set of coefficient vectors, one per iteration. TSAFs were motivated by scenarios in which the nonstationarity follows a periodic pattern, e.g., when the signals are cyclostationary. This combination was unsupervised in the sense that the coefficient vector are updated in a sequence that is independent of the input signal.

The idea of a supervising rule only appears in Andersson’s *Adaptive Forgetting through Multiple Models* (AFMM) in 1985 [93]. This combination was inspired by nonlinear filtering solutions based on Gaussian sums developed in the 1970s [119,120]. It consisted of combining independent RLS filters with different forgetting factors weighting their coefficients using an approximate Bayesian method. In the decade that followed, different combinations of RLS/Kalman filters were developed in control [94,95] and prediction [77] settings.

Combinations of AFs as they are known today, were introduced in the beginning of the years 2000s. Multistage AFs [92,121] evolved from the work by Singer on the universal linear predictor (ULP) [77] and were supervised by unconstrained Bayesian and Least-Squares (LS) methods. Almost simultaneously, the convex supervisor was introduced in [122] and became ubiquitous for parallel-independent combinations, specially after [33]. Its normalized version [91] was proposed in 2008, the same year as the affine supervisor [35] and its normalized counterpart [108].

¹The scientist must organize; science is made of facts just as a house is made of stones; but an accumulation of facts is no more a science than a pile of stones is a house.

All structures discussed so far use independent component filters whose coefficients are merged by a supervisor. Albeit effective, this combination suffers from a well-known convergence stagnation issue: due to the different learning rate of the AFs, the slower filter in the combination can take some time to reach the steady-state MSE of the faster filter; during this period, the output of the combination will stall at the error level of the faster filter. This issue was initially addressed using conditional unidirectional transfers of coefficients between the components [2, 33]. Although effective in many cases, this solution only addressed the problem partially [42, 123]. Recently, more effective solutions were found through changes in topology, introducing the concept of incremental combinations [41, 73] and coefficients feedback [42]. These contributions allowed combinations of AFs to be generalized in many ways, with the introduction of DR combinations [73], complex topologies [45, 47], hybrid combinations [48] etc. The considerable improvements offered by these extensions lead to the notion of *combinations as a complexity reduction technique*, that will be presented in Chapter 9.

Figure 11: Timeline

4.2 Classifying combinations

Definition 3.1 (see Chapter 3) leads to a unified approach to combinations of AFs that was not possible before. This unification allows interpretations and analyses to be generalized to a variety of combinations and facilitates the comparison of different structures. Such a general framework, however, also results in an overwhelming number of algorithms being considered combinations, which undermines the initial appeal of this approach. Hence the importance of separating combinations into manageable classes. The challenge, then, is to find a classification that is fine enough to capture the idiosyncrasies of the structures, but not as fine as to obscure their similarities.

The four elements extracted from the $\mathcal{FOB}$ mappings in Chapter 3 are natural criteria for this taxonomy. Indeed, they are motivated by and directly related to Definition 3.1, besides being supported by an extensive literature review. Thus, the proposed taxonomy classifies combinations of AFs as to their

1. component filters;
2. topology;
3. DR method; and
4. supervising rule.

Possible classes based on these features are illustrated in Figure 12 and used to classify a myriad of combinations in the sequel. Notice that several adaptive algorithms (such as DR and mixed cost function algorithms) that were not initially seen as combinations but fit Definition 3.1 are included in the following review¹.

This taxonomy and its categories are not intended to be definitive or exhaustive.

¹More details on casting stand-alone adaptive algorithms as combinations can be found in Chapter 8.

Figure 12: Taxonomy of combinations of adaptive filters

4.2.1 Component filters

Any set of AFs can be combined. Hence, characteristics of these component filters can be used to differentiate them. Filters with different lengths, for example, are investigated in [38, 89, 124] and combinations of different filtering structures (IIR/FIR), in [48, 64]. A particularly important aspect of AFs is their adaptive algorithm [18, 19] and a classification based on this feature is found in Table 2. Notable cases that did not fit in this table include IPNLMS combinations [36], the LMS + sign-error LMS (RMN) [29], APA combinations [37, 46, 49], and combinations involving nonlinear AFs [40, 125–127].

4.2.2 Topology

The number of topologies available to combine even a handful of component filters is overwhelmingly large. Indeed, it ultimately amounts to the number of weakly connected digraphs that can be drawn over a set of $N + 1$ nodes. That is, of course, without considering arc weights, dynamical topologies... However, Section 3.2 suggested that two large categories of topologies are worth noting, specially given that more complex structures can be built based on them. These categories are differentiated in terms of the their components updates: when they are synchronous the topology is said to be *parallel*; when they are not, the topology is named *incremental*. Table 3 shows this classification. Some noteworthy topologies can be described as hybrids of these classes, such as the cyclically incremental, parallel topology [96] and the parallel-incremental topology [45].

4.2.3 Data reusing method

The DR methods discussed in Section 3.3 were divided in three categories: data sharing, data buffering, and refiltering. Although different forms of data buffering techniques do exist, Tables 4 focuses on distinguishing between these three classes. Notice that data sharing is the most common type of DR strategy.

4.2.4 Supervising rule

Section 3.4 described the supervising rule as the part of the combination that evaluates its output. It therefore plays a definitive role in the combination performance. Most of these techniques fall into one of five categories (described in Section 3.4): *unsupervised* (or static parameters), *statistical* (Bayesian) techniques, adaptation based on *error statistics*,

Table 2: Classification as to the adaptive algorithm of the component filters

AF	LMS	LMF	RLS	NLMS	PNLMS
LMS	[2, 22, 25, 33–35, 39, 41, 42, 44, 45, 47, 70, 73, 88–92, 96, 97, 106–108, 110, 118, 121–123, 128–134]	[26–28, 90, 135, 136]	[34, 43, 110, 129]	-	-
LMF		[110]	-	-	[59]
RLS			[2, 34, 77, 92–95, 129]	-	-
NLMS				[22, 24, 25, 38, 91, 109, 124, 137]	[69, 138]
PNLMS					-

stochastic gradient descent, and *Newton's method*. In Table 5, these categories are crossed with the constraints imposed on the $\{\eta_n(i)\}$: *unconstrained*, *convex constraint* ($\sum \eta_n(i) = 1$ and $\eta_n(i) \geq 0$), and *affine constraint* ($\sum \eta_n(i) = 1$). Notice that these constraints are mostly warranted for parallel topologies (as discussed in Chapter 5), but have been used for incremental combinations as well [41, 43].

4.3 Concluding remarks

Combinations of AFs are not novel and can be traced back to the early years of adaptive filtering, especially once certain classes of stand-alone adaptive algorithms are viewed as combinations (see Chapter 8). Their study, however, has recently picked up pace and remains an active research topic. Nevertheless, it is clear from the review presented in this chapter that there are still gaps in their understanding, particularly as it refers to the impact of topology on performance, DR methods, unsupervised/IIR component filters, and incremental combinations. The remaining chapters attempt to

Table 3: Classification as to the topologies

Topology	Parallel	[2, 24–29, 33–40, 42, 45–49, 59, 69, 77, 89–97, 106–110, 118, 121–130, 133–138]
	Incremental	[22, 24, 41, 43–45, 47, 64, 70, 73, 88, 96, 131, 132]

Table 4: Classification as to the data reusing method

DR method	References
Data sharing	[2, 22, 26–29, 33–49, 59, 64, 69, 70, 77, 88–96, 106–110, 121–130, 133, 133, 134, 136–138]
Data buffering	[22, 24, 25, 45, 47, 73, 118, 135]
Refiltering	[97, 131, 132]

address some of these issues, by first describing aspects specific to parallel and incremental combinations (Chapters 5 and 6); then, by introducing new uses for them (Chapters 8 and 9); and finally, by providing analyses of their performances (Chapters 10 and 11).

Table 5: Classification as to the supervising rule

Rule	Unconstrained	Convex constraint	Affine constraint
Unsupervised	[22, 24, 25, 44, 45, 47, 70, 73, 88, 97, 118, 135]	[28, 29, 41, 43, 69]	-
Statistical	[92, 121]	[77, 93–95]	-
Error statistics	[26]	[26, 27, 41, 49, 64, 138]	[35, 106, 109, 130]
Stochastic gradient descent	[107, 121, 131, 132]	[2, 33, 34, 36–38, 40, 42, 48, 59, 89, 90, 122, 123, 125–129, 134, 136, 137]	[35, 39, 90, 107, 108, 110]
Newton’s method	-	[40, 45, 46, 91, 96, 124, 126, 129, 133]	[39, 108]

5 PARALLEL COMBINATIONS

Any AFs mixture that employs a purely parallel topology is referred to as a *parallel combination*. This was the first type of combination of AFs to be introduced and remains ubiquitous in the literature. It has been studied for different adaptive algorithms, step sizes, filter lengths, and supervisors [2, 33–37, 40, 42, 46, 49, 59, 65, 77, 89–93, 96, 122, 123, 136, 139].

The main advantage in using a parallel combination is that the synchronicity of its component filters allows for parallel computation. Such a feature is of foremost importance in modern signal processing systems that increasingly rely on multicore processors and FPGAs [98]. Thus, parallel topologies can achieve higher throughputs in practice than other topologies, even if they are more computationally demanding. In fact, as long as there are enough resources to run all components in parallel, the throughput of the combination is on the order of that of an individual AF.

On the other hand, parallel combinations spend most of their operation exploiting the output of a single filter, although N AFs are running [33–35, 42, 90]. This is evident in stationary scenarios. Although solutions have been proposed to address this issue [96, 133], in platforms where power or processor size is limited, these combinations may be considered wasteful.

In the sequel, parallel combinations are explored by first studying parallel-specific supervising rules (Section 5.1) and then investigating its different topologies (Section 5.2).

5.1 Supervising rules

In Section 3.4, the supervisor (more specifically, $\mathcal{O}$) was restricted to the class of linear combinations of the components coefficients. It was argued then that this constraint has minimal impact in the generality of the derivations because it still accounts for most existing combinations. Indeed, as far as parallel combinations go, all supervising rules

introduced in the literature are of the form [33, 35, 92, 106, 107].

$$\mathbf{w}_i = \sum_{n=1}^N \eta_n(i) \mathbf{w}_{n,i}. \quad (5.1)$$

The supervising parameter $\eta_n(i)$ represents the action of the supervisor on the n -th component, which in this case is akin to controlling its influence on the output of the combination. Typically, the supervising parameters in (5.1) are chosen to minimize the global MSE under some unbiasedness constraint [33, 35, 39, 91, 106, 107]. This constraint guarantees that if the component filters are asymptotically unbiased, so is the combination. Since most adaptive algorithms are [18], this is an important feature to preserve.

To derive an unbiasedness constraint for parallel combinations, start by subtracting (5.1) from $\mathbf{w}^o$ to get

$$\tilde{\mathbf{w}}_i = \mathbf{w}^o - \sum_{n=1}^N \eta_n(i) \mathbf{w}_{n,i} \Leftrightarrow \tilde{\mathbf{w}}_i = \sum_{n=1}^N \eta_n(i) \tilde{\mathbf{w}}_{n,i} + \left[1 - \sum_{n=1}^N \eta_n(i) \right] \mathbf{w}^o. \quad (5.2)$$

Assuming $\mathbf{w}_{n,i}$ is conditionally independent of $\eta_n(i)$ for all n , the conditional expected value of (5.2) yields

$$\mathbb{E}[\tilde{\mathbf{w}}_i | \eta_n(i)] = \sum_{n=1}^N \eta_n(i) \mathbb{E} \tilde{\mathbf{w}}_{n,i} + \left[1 - \sum_{n=1}^N \eta_n(i) \right] \mathbf{w}^o. \quad (5.3)$$

This assumption is commonly used in the performance analysis of combinations and is valid for a myriad of supervising rules [39, 123, 128]. Alternatively, one could consider supervising parameters to be deterministic.

The first term in (5.3) is a linear combination of the component filters biases and the second term represents the additional bias due to the combination. Hence, to avoid increasing the bias of the individual AFs, $\{\eta_n(i)\}$ must be chosen such that

$$\sum_{n=1}^N \eta_n(i) = 1. \quad (5.4)$$

In this case, if the components are asymptotically unbiased, (5.3) will vanish as $i \rightarrow \infty$.

Imposing (5.4) is not always straightforward, specially if the supervising rule is non-linear (e.g., convex supervisor [33]). Moreover, although unconstrained supervising rules do exist [92, 121], their use is scarce due to their worst performance. In order to simplify the supervisor recursion, it is therefore common to design them for combinations of $N = 2$

Figure 13: Hierarchical parallel combination of four adaptive filters

AFs. In this case, the expression for the global coefficients (5.1) simplifies to

$$\mathbf{w}_i = \eta(i)\mathbf{w}_{1,i} + [1 - \eta(i)]\mathbf{w}_{2,i}, \quad (5.5)$$

which already enforces (5.4). This is by no means restrictive as larger parallel structures can be built by combining two-AFs structures hierarchically [36, 40, 46, 137]. For example, a parallel combination of four AFs can be constructed using two combinations of two AFs combined as if they were component filters (Figure 13). This hierarchical combination is equivalent to (5.1) under the unbiasedness constraint (5.4) and has the same computational complexity ($N - 1$ supervising parameters are evaluated). Although this structure is more common when using nonlinear supervisors (e.g., convex supervisor), it can be used with any supervising rule (e.g., affine supervisor). In view of this fact, the following rules are introduced for parallel combinations of two AFs.

5.1.1 Unsupervised (“static”)

Unsupervised parallel combinations can be used to cast several stand-alone adaptive algorithms as mixtures, a concept further explored in Chapter 8. The performance of these AFs is evidence of how effective the static supervisor can be [24, 25, 28, 29, 69, 97]. However, it has not been further explored in the combination of AFs literature because the optimal parallel combiner is not static, which suggests unsupervised parallel combinations can be readily improved [35, 39, 90, 128].

5.1.2 The convex supervisor

The convex supervising rule was introduced in [122] and analyzed in [33, 90, 123, 128]. This scheme constrains the global coefficients to the $(N - 1)$ -simplex formed by the components coefficients, imposing $\eta_n(i) \in [0, 1]$ as well as the unbiasedness constraint (5.4). For $N = 2$, this restriction is usually met using an activation function [2, 33, 34, 36, 38, 42, 59, 89, 90, 122, 136, 137]. Explicitly, the supervising parameters are defined as $\eta(i) = f[a(i)]$ for some monotonically increasing function $f : \mathbb{R} \mapsto [0, 1]$ with derivative¹ $f' > 0$, where $a(i) \in \mathbb{R}$ is an auxiliary parameter that is actually adapted. Extending this rule to arbitrary N is not straightforward.

The most common technique to adapt a is using a stochastic gradient algorithm that updates $a(i - 1)$ to minimize the global MSE. Explicitly,

$$a(i) = a(i - 1) + \mu_a e(i) [y_1(i) - y_2(i)] f'[a(i - 1)]. \quad (5.6)$$

For the widely adopted sigmoidal function

$$f(a) = \frac{1}{1 + e^{-a}}, \quad (5.7)$$

the classical convex supervisor from [33, 122] is recovered

$$\begin{aligned} a(i) &= a(i - 1) + \mu_a e(i) [y_1(i) - y_2(i)] \eta(i - 1) [1 - \eta(i - 1)] \\ \eta(i) &= \frac{1}{1 + e^{-a(i)}}, \end{aligned} \quad (5.8)$$

where the value of $a(i)$ is constrained to $[-a_{\max}, a_{\max}]$ so as not to slow down the adaptation (usually $a_{\max} = 4$). Nevertheless, the supervising parameter must reach 0 and 1 to achieve best performance, so that $\eta(i)$ in (5.5) is defined to be 1 (0) when $a(i) =$

¹This constraint ensures that the adaptation algorithm does not stalls, i.e., that there is no a for which $f'(a)$ vanishes and the supervising parameter becomes independent of the component filters.

$a_{\max}$ ($a(i) = -a_{\max}$). A complete implementation of this supervisor would therefore read²

$$\begin{aligned}
 a(i) &= \left[a(i-1) + \mu_a e(i) [y_1(i) - y_2(i)] \eta(i-1) [1 - \eta(i-1)] \right]_{-a_{\max}}^{a_{\max}} \\
 \eta(i) &= \frac{1}{1 + e^{-a(i)}} \\
 \eta'(i) &= \begin{cases} 1, & a(i) = a_{\max} \\ \eta(i), & -a_{\max} < a(i) < a_{\max} \\ 0, & a(i) = -a_{\max} \end{cases} \\
 \mathbf{w}_i &= \eta'(i) \mathbf{w}_{1,i} + [1 - \eta'(i)] \mathbf{w}_{2,i}
 \end{aligned}$$

Choosing μ_a in (5.8) is a compromise between the supervisor adaptation capability and the variance of $\eta(i)$. This issue is less accentuated when the supervising parameter is close to either 0 or 1 because of the small value of the activation function derivative [i.e., $\eta(1 - \eta)$], thus allowing the use of larger μ_a without hindering the combination misadjustment. This, however, can deteriorate transient performance, specially when the supervisor is switching between filters ($0 < \eta(i) < 1$). To address this issue, a normalized version of (5.8) was developed in [91]:

$$\begin{aligned}
 p(i) &= \beta p(i-1) + (1 - \beta) [y_1(i) - y_2(i)]^2 \\
 a(i) &= a(i-1) + \frac{\mu_a}{p(i) + \epsilon} e(i) [y_1(i) - y_2(i)] \eta(i-1) [1 - \eta(i-1)] \\
 \eta(i) &= \frac{1}{1 + e^{-a(i)}},
 \end{aligned} \tag{5.9}$$

where $0 \ll \beta < 1$ and ϵ is a regularization factor.

5.1.3 The affine supervisor

The affine supervisor was proposed in [35] as an alternative to the convex one [33, 122] in which only the unbiasedness constraint (5.4) is imposed. It was initially formulated for combinations with $N = 2$ components [35, 39] and later extended to the arbitrary N case [110]. Contrary to the convex rule, this generalization is straightforward and natural.

The stochastic gradient version of the affine supervisor can be derived as a particular case of (5.6) by choosing $f(a) = a$:

$$\eta(i) = \eta(i-1) + \mu_\eta e(i) [y_1(i) - y_2(i)]. \tag{5.10}$$

²Instead of limiting the value of $a(i)$, it is possible to modify the activation function f to take the limits $[-a_{\max}, a_{\max}]$ into account [140]. However, hard limiting requires less operations and is, therefore, better from an implementation point of view.

Although effective, this method is very sensitive to the choice of μ_η . Relatively large step sizes are needed for the supervising parameters to adapt fast enough, which increases the variance of $\eta(i)$ and hinders the performance of this supervising rule. Hence, accurate estimates of the SNR and components convergence rates are required to adequately design this supervisor [35, 39, 90, 110]. These issues can be partially addressed by using a normalized adaptation version of (5.10) [39]:

$$\begin{aligned} p(i) &= \beta p(i-1) + (1-\beta)[y_1(i) - y_2(i)]^2 \\ \eta(i) &= \eta(i-1) + \frac{\mu_\eta}{p(i) + \epsilon} e(i)[y_1(i) - y_2(i)] \\ w_i &= \eta(i)w_{1,i} + [1 - \eta(i)]w_{2,i}, \end{aligned} \tag{5.11}$$

where $0 \ll \beta < 1$ and ϵ is a regularization factor. Notice that this algorithm is similar to the one in (5.9).

Another solution is to constrain the value of $\eta(i)$ to a range that includes the optimal supervising parameter, thus limiting the supervising parameter variance without impairing its optimality. The error power ratio scheme from [35, 106], for example, uses this approach. The supervising parameter is evaluated using the ratio of estimates of the components MSE smoothed by an activation function that imposes $\eta(i) \in [1 - \kappa, 1]$. The value of κ is selected so that this range includes the optimal supervisor value. A selection procedure based on the theoretical value of component filters steady-state EMSEs was proposed in [106].

5.2 Parallel topologies

Even for a number as small as $N = 2$ filters, there are several ways of combining them in parallel. Although they fit in the same class, these topologies are not equivalent and their performance can vary considerably. Studying these structures is paramount to identify the performance gains and limits of the parallel topology and understand the part they should play in larger combinations.

5.2.1 The parallel-independent topology

The first and still most common parallel topology in combinations of AFs is the *parallel-independent* (Figure 14) [33–37, 40, 46, 49, 59, 65, 77, 89–93, 122, 123, 139]. Its name comes from the fact that its component filters run independently (their recursions do not depend on coefficients from any other filters) while the supervisor aggregates their

estimates. In other words, the components behave as if they were stand-alone AFs or as per Figure 14: $\mathbf{w}_{n,a} = \mathbf{w}_{n,i-1}$.

Although the component filters are independent in this topology, it does not mean they do not cooperate. Their cooperation, however, must arise from the supervisor ability to track the best filter in the set and exploit the cross-correlation between the AFs errors, which enables the combination to outperform its components [33–35, 90, 110, 128]. This effect can be seen in Figure 15 when the components learning curves cross and the combination has an EMSE lower than both filters. This feature is enhanced when the cross-correlation of the errors is lower and there is more information to exploit, such as in LMS + LMF combinations [90].

Figure 15 also illustrates the well-known convergence stagnation issue of this topology [33–35, 42]: the overall combination (black curve) stalls as the faster filter reaches steady-state while the slower filter is still converging. Since the combination attempts to minimize the global MSE, as long as the error of the slow component is larger than that of the fast one, the combination will track the output of the latter.

The convergence stagnation issue is inherent to the parallel-independent topology and can be readily understood from its digraph (Figure 14). As argued in Section 3.2, the cooperation between component filters is related to the connectivity of the topology. The parallel-independent topology, in particular, is clearly not strongly connected. In fact, the largest subset of strongly connected nodes (also known as a *strong component* [78]) has one element (one component filter). This means that the filters cooperation is very limited, since none of them can be influenced by other components or the output node. Hence, the transient performance of the combination is bounded by that of its component filters and since the combination has no effect on the individual AFs, the convergence stagnation issue is unavoidable.

Therefore, overcoming this issue then requires a change in topology. Conditional

Figure 14: The parallel-independent topology digraph.

Figure 15: Convergence stagnation issue of the parallel-independent topology ($2 \cdot \text{LMS}$). **White input data:** $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\mu_1 = 0.07$, $\mu_2 = 0.007$, and $\mu_a = 1000$ (convex supervisor).

coefficients leakage from fast to slow component (Section 5.2.2) [33, 123] and cyclic coefficients feedback (Section 5.2.3) [42] are two solutions that retain the parallel topology. Initially, the incremental topology was also motivated as a solution to the stagnation effect [41, 96], although it has since been shown to be better employed to other ends (see Chapter 6) [43, 44, 47, 73].

5.2.2 The parallel topology with coefficients leakage

coefficients leakage was proposed in an attempt to improve the convergence rate of the slower component filter by injecting the faster AF coefficient vector into its recursion [33]. For illustration purposes, assume that the components are LMS filters and that filter 2 is the slow one. Then, filter 1 runs independently ($\mathbf{w}_{1,a} = \mathbf{w}_{1,i-1}$) and the second filter recursion becomes

$$\mathbf{w}_{2,i} = \alpha(i)[\mathbf{w}_{2,i-1} + \mu_2 \mathbf{u}_i^T e_2(i)] + [1 - \alpha(i)]\mathbf{w}_{1,i-1}, \quad (5.12)$$

for some the factor $\alpha(i) \in [0, 1]$. Since this perturbation would deteriorate the misadjustment of component 2 (and consequently that of the combination), it is only enabled while filter 1 is converging. To do so, the transfer is made conditional on the supervising parameter value using $\alpha(i) = \alpha_0$, for $\eta(i) \geq \bar{\alpha}$, and $\alpha(i) = 1$, otherwise, for some threshold $\bar{\alpha} \approx 1$ and $\alpha_0 \in [0, 1]$. Typical values are $\bar{\alpha} = 0.98$ and $\alpha_0 = 0.97$ [33, 123].

Determining the digraph of this topology is not as straightforward as in the parallel-independent case, because it alters the recursion of the second AF. However, $\mathbf{w}_{2,i}$ is

Figure 16: The parallel topology with coefficients leakage digraph.

Figure 17: The parallel topology with cyclic coefficients feedback digraph.

actually a linear combination of the component coefficients. Hence, expressions for $\mathbf{w}_{n,a}$ and $\mathbf{w}_i$ can be obtained from (5.12) and (5.1) as

$$\begin{aligned}\mathbf{w}_{1,a} &= \mathbf{w}_{1,i-1} \\ \mathbf{w}_{2,a} &= \alpha \mathbf{w}_{2,i-1} + (1-\alpha) \mathbf{w}_{1,i-2} \\ \mathbf{w}_i &= \eta \mathbf{w}_{1,i} + (1-\eta) [\alpha \mathbf{w}_{2,i} + (1-\alpha) \mathbf{w}_{1,i-1}],\end{aligned}$$

which are drawn in Figure 16. The dependence on i was omitted where unambiguous for conciseness. Once again, the topology digraph makes it clear that the first component (red) behaves as if the combination were parallel-independent, whereas component 2 (green) is disturbed by past coefficients of component 1. Moreover, unless $1-\alpha$ vanishes as the combination approaches steady-state, the arc connecting filter 1 to the output node remains connected, increasing the global error of the combination. Finally, notice that although the coefficients leakage has improved the connectivity of this topology, it is not yet strongly connected and still limits the component filters cooperation.

As illustrated in Figure 18, this method effectively addresses the stagnation issue, although it presents several downsides from an implementation point of view. Firstly, it requires a conditional test on every iteration. Secondly, coefficients leakage is a unidirectional interaction (from component 1 to component 2) and determining the faster filter is not necessarily straightforward, specially when different adaptive algorithms are used. Moreover, there is little guarantee in practice that any one component will always be faster than the other (e.g., in nonstationary scenarios). These issues are further accentuated when more than two AFs are combined [42].

Figure 18: Comparison between coefficients leakage and coefficients feedback ($2 \cdot \text{LMS}$). **White input data:** $M = 7$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.07$, and $\mu_2 = 0.007$. (a) **Nonstationary w^o** ($w^o \rightarrow -w^o$ at $i = 1500$). **coefficients leakage:** $\alpha_0 = 0.95$, $\bar{\alpha} = 0.98$, $\mu_a = 200$ (convex supervisor); **cyclic coefficients feedback:** $L = 50$, $\mu_a = 300$ (convex supervisor). (b) **Nonstationary SNR** ($\sigma_v^2 = 10^{-1}$ at $i = 1000$ and $\sigma_v^2 = 10^{-3}$ at $i = 2000$). **coefficients leakage:** $\alpha_0 = 0.95$, $\bar{\alpha} = 0.98$, $\mu_a = 1$, $\beta = 0.9$, $\epsilon = 10^{-3}$ (normalized convex supervisor); **cyclic coefficients feedback:** $L = 50$, $\mu_a = 1$, $\beta = 0.9$, $\epsilon = 10^{-4}$ (normalized convex supervisor).

5.2.3 The parallel topology with coefficients feedback

Global coefficients feedback was proposed as a solution to the convergence stagnation that would also address the concerns presented at the end of the last section. This approach consists of using the global coefficient vector $\mathbf{w}_i$ from (5.1) as the *a priori* coefficients of all component filters in the combination (i.e., $\mathbf{w}_{n,a} = \mathbf{w}_{i-1}$). This means that all filters in the combination operate over the best current estimate of $\mathbf{w}^o$ as evaluated by the supervisor [42].

Since all filters update the same *a priori* coefficient vector, the difference between their outputs $y_n(i)$ is reduced. This slows down the supervisor adaptation which usually depends on this difference to identify the better component (see Section 5.1). To address this issue, the global coefficients are fed back cyclically as in

$$\mathbf{w}_{n,a} = \delta_L(i)\mathbf{w}_{i-1} + [1 - \delta_L(i)]\mathbf{w}_{n,i-1}, \quad (5.13)$$

where L is a scalar constant that defines the *cycle period*. Notice that the overall combination has two limiting behaviors depending on the value of the cycle period: (i) for $L = 1$, (5.13) reduces to $\mathbf{w}_{n,a} = \mathbf{w}_{i-1}$ and the global coefficients are always fed back to the components; (ii) for $L \rightarrow \infty$, one gets $\mathbf{w}_{n,a} = \mathbf{w}_{n,i-1}$ and the parallel-independent

combination from Section 5.2.1 is recovered.

Examining the digraph of this topology in Figure 17, it is clear that it better promotes cooperation between the components. In particular, for $\delta_L = 1$, the topology becomes strongly connected. At this point, all nodes in the combination have access to the coefficients of any other node, although this access is controlled by the supervisor [depends on $\eta(i)$]. Thus, the combination constantly exploits the component filters, as in the parallel-independent topology, but it also cyclically cooperates to improve their individual performances.

Conceptually, this topology is more natural than those with coefficients leakage because it provides all components with the global coefficients, which is evaluated by the supervisor to explicitly minimize the overall MSE. Moreover, it is the only topology so far to be (cyclically) strongly connected, which allows this combination to inherit several important aspects of stand-alone AFs. In particular, it follows a global version of the ECR (Section 2.2) and bridges the gap between stand-alone adaptive algorithms and combinations of AFs (for more details, see Chapter 8). In terms of performance, besides completely avoiding convergence stagnation (Figure 18), it can lead to improvements in the global misadjustment, specially in nonstationary scenarios (e.g., Figure 19). The feedback also has the side-effect of stabilizing the combination by reducing the supervising parameters variance (see Section 10.3). Finally, in contrast to the coefficients leakage, cyclic coefficients feedback is neither unidirectional nor limited to any pair of filters and are fit for efficient interruption-based implementations. What is more, as long as L is adequately designed, there is no need for the feedback to be conditional.

5.2.3.1 Design of the cycle period

The choice of the cycle period is a compromise between the component filters cooperation and the supervisor adaptive capability. Choosing a small L allows the components to fully benefit from the combination and improves the overall steady-state behavior (Figure 19). However, small cycle periods tend to stall the supervisor, thus hindering the transient performance of the combination (Figure 20) [42].

If the goal is to improve steady-state or tracking performance regardless of convergence behavior, using values as small as $L = 1$ (typically, $L < 20$) is appropriate. Otherwise, the design procedure from [42] can be used to obtain a good period value. It is based on the rationale that, in a stationary scenario, the overall coefficients should be fed back as soon as the faster filter has converged, so that the slower filter can carry on improving

Figure 19: Tracking of an LMS+RLS combination with and without coefficients feedback. **Correlated input data:** $M = 5$, $\sigma_u^2 = 1/M$, $\sigma_v^2 = 10^{-2}$, $\gamma = 0.8$, $c_1 = 2 \cdot 10^{-6}$, $c_2 = 2 \cdot 10^{-5}$; $\mu_{\text{LMS}} = 0.04$, $\lambda_{\text{RLS}} = 0.92$, and $\epsilon = 10^{-6}$. **Parallel-independent:** $\mu_a = 100$ (convex supervisor); **cyclic coefficients feedback:** $\mu_a = 100$ and $L = 25$ (convex supervisor).

its estimate. The iteration at which this occurs can be approximated by determining the crossing point of transient and steady-state linearized models. This value is then used as the cycle period. An example design using an LMS filter is presented in the sequel. The data model from Section 1.6.1 is adopted throughout these derivations.

First, a simplified transient model for the fast filter is derived. To do so, assume $\mathbf{u}_i$ is white and Gaussian and use the weighted variance relation with independence for LMS filters [18, Eq. (23.19)] to get, by induction,

$$\mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 = \nu^i \mathbb{E} \|\mathbf{w}^o\|^2 + \mu \sigma_v^2 \sigma_u^2 M \sum_{k=0}^{i-1} \nu^k, \quad (5.14)$$

where $\nu = 1 - 2\mu\sigma_u^2 + \mu^2\sigma_u^4(M+2)$. A coarse approximation for (5.14) is obtained by assuming small step size and high SNR, leading to³

$$\mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 = \nu^i \mathbb{E} \|\mathbf{w}^o\|^2. \quad (5.15)$$

Finally, using the fact that the input is i.i.d. and that the noise is negligible, the MSE from (2.13) can be rewritten as $\text{MSE}(i) = \sigma_u^2 \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \sigma_v^2 \approx \sigma_u^2 \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2$. Expressing the MSE in dB yields the desired linearized model for the transient phase

$$\text{MSE}_{\text{dB}}(i) = 10i \log(\nu) + 10 \log(\sigma_d^2 - \sigma_v^2), \quad (5.16)$$

³For stable LMS filters, $0 \ll \nu < 1$.

Figure 20: 2 · LMS combination with coefficients feedback ($L = 1$). **White input data:** $M = 7$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.05$, $\mu_2 = 0.005$, and $\mu_a = 200$ (convex supervisor).

where the relation $\sigma_u^2 \|\mathbf{w}^o\|^2 = \sigma_d^2 - \sigma_v^2$ was used. Notice that (5.16) indeed takes on the form $r(i) = a \cdot i + b$, with $a = 10 \log(\nu)$ and $b = 10 \log(\sigma_d^2 - \sigma_v^2)$.

Finding L then reduces to solving $r(L) = 10 \log(\text{MSE}^{\{\text{LMS}\}})$ using (5.16) and the steady-state MSE expression for LMS filters [18, Lemma 16.1]

$$\text{MSE}^{\{\text{LMS}\}} = \frac{2\sigma_v^2(1 - \mu\sigma_u^2)}{2 - \mu(M + 2)\sigma_u^2}. \quad (5.17)$$

Several approximations were made in order to obtain such a simple expression for the cycle period. Besides the typical analyses approximations (data separation principle, Gaussian i.i.d. input, and noise independence [18]), (5.16) also relies on small step sizes and high SNR assumptions. Nevertheless, the results prove adequate for several values of SNR/step size, as shown in Figure 21a. Moreover, the overall performance of combinations with coefficients feedback is fairly robust to the choice of L . In fact, the number of iterations-to-convergence remains almost unchanged for a considerable range of L (Figure 22). The cycle period found for $\mu_1 = 0.05$ and $\text{SNR} = 20\text{dB}$ is used in the combination of Figure 21b. Notice that, albeit simple, the proposed design method works very well. This figure also illustrates the behavior of the component filters under coefficients feedback, which produces a cyclic “steady-state” behavior for filter 1.

5.3 Concluding remarks

Until recently, the parallel topology was the only one considered for combinations of AFs. Nevertheless, it has seen a clear evolution since the first parallel-independent com-

Figure 21: Designing the cycle period. **White input data:** $M = 7$ and $\sigma_u^2 = 1$. (a) **Design procedure.** (b) **Combination:** $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.05$, $\mu_2 = 0.005$, $L = 60$, and $\mu_a = 200$ (convex supervisor).

Figure 22: Iterations-to-convergence for different L (number of iteration to first attain the average steady-state EMSE value). **White input data:** $M = 7$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.05$, $\mu_2 = 0.005$, and $\mu_a = 500$ (convex supervisor).

binations, culminating with the introduction of the cyclic coefficients feedback. Indeed, this is the first strongly connected parallel topology, which accounts for its improved performance and robustness. Moreover, it has allowed several DR, mixed-norm, and VSS adaptive algorithms to be interpreted as combinations, allowing for a better understanding and improvements of their performance (for more details, see Chapter 8). In terms of supervising rules, parallel combinations have a myriad of possibilities and although stochastic gradient supervisors are the most common, other rules have been successfully applied, such as [29, 77, 93, 106, 107]. An important thing to notice about parallel supervising rules is that the unbiasedness constraint (5.4) is fundamental to achieve good performance. It is, however, specific to the parallel topology: its use in other context, for

instance, in incremental combinations, is unwarranted and can even deteriorate performance. This effect is explored in more details in the next chapter.

6 INCREMENTAL COMBINATIONS

Incremental combinations of AFs are those that operate over purely incremental topologies. In other words, the filters are daisy-chained such that the *a priori* coefficients of filter n is the updated coefficients of filter $n - 1$ ($\mathbf{w}_{n,a} = \mathbf{w}_{n-1,i}$). They were recently introduced by [41] as a solution to the convergence stagnation issue of the parallel-independent topology (Section 5.2.1). Since then, incremental combinations have been shown to display improved convergence rates and tracking capability [43–45, 73].

Contrary to their parallel counterpart, incremental combinations efficiently exploit the available resources, since every operation in the chain has an effect on the output. With many AFs operating over a coefficient vector at each iteration, it makes sense that these combinations display faster convergence [44, 45, 73] and better tracking [43]. In fact, incremental chains can usually achieve a given transient performance using less AFs than its parallel equivalent [44, 45, 73]. Therefore, these combinations have typically lower complexity.

Nevertheless, this improved convergence behavior comes at the cost of a larger misadjustment, at least for unsupervised combinations [44, 45]. This effect is amplified in longer chains, in which the steady-state error can become much larger than it would be if the filters operated individually (see Chapter 11 for an analysis of this behavior). Another issue with incremental topologies is that they are unable to take advantage of modern parallel computing platforms, such as FPGAs [98]. In contrast, they usually have smaller footprints (occupy less area in FPGAs or less memory in DSPs).

This chapter investigates different supervising rules (Section 6.1) and topologies (Section 6.2) for incremental combinations. Although incremental combinations involving different component filters are possible and have shown good results [43], combinations of LMS filters are considered in what follows for ease of presentation.

6.1 Supervising rules

In incremental combinations, the concept of supervisor is not as well defined as in the parallel case. Indeed, the output of an incremental combination is typically taken to be the last filter in the chain ($\mathbf{w}_{N,i}$) [41, 43, 45, 47, 73]. Under the restriction from Section 3.2 that the supervising rule is a linear combination of the component filters (i.e., $\mathbf{w}_i = \sum \eta_n(i) \mathbf{w}_{n,i}$), this is equivalent to choosing $\eta_n(i) = 0$ for all $n < N$ and $\eta_N(i) = 1$. Notice that, once again, the restriction on $\mathcal{O}$ has not impaired the generality of the $\mathcal{FOB}$ description. In [41], however, a different supervising rule was proposed. Instead of acting on the output of the combination, the supervising parameters affect the AFs updates directly. A supervised incremental combination of LMS filters, $\{\text{LMS}\}^N$, would then be

$$\begin{aligned} \mathbf{w}_{0,i} &= \mathbf{w}_{i-1} \\ \mathbf{w}_{n,i} &= \mathbf{w}_{n-1,i} + \eta_n(i) \mu_n \mathbf{u}_{n,i}^T [d_n(i) - u_{n,i} w_{n-1,i}] \\ \mathbf{w}_i &= \mathbf{w}_{N,i} \end{aligned} \tag{6.1}$$

for $n = 1, \dots, N$. Hence, the $\{\eta_n(i)\}$ play a dual role of supervisor and VSS [44].

Nevertheless, it could be argued that neither techniques are actually incremental supervising rules. Indeed, the first rule (linear combination of component filters) is equivalent to having a parallel combination of the AFs in the incremental chain. The second rule [(6.1)] impacts the update of the component filters in a VSS fashion. It could therefore be completely eliminated by replacing the components by VSS adaptive algorithms. In the absence of a consensus on the subject and since the first supervisor structure leads to a sort of parallel-incremental topology, supervisors of the form (6.1) are considered when discussing incremental supervising rules from now on.

Another issue with incremental supervising rules is that designing an effective method for adapting the $\{\eta_n(i)\}$ is challenging and remains an open issue [41, 44]. In particular, since the supervisor acts directly on the AFs updates, unbiasedness constraints and optimal supervising parameters depend on the component filters of the combination. Hence, general results as those obtained in Section 5.1 for the parallel supervisor are not available for incremental combinations. Nevertheless, different adaptation methods have been studied and are described in the sequel.

6.1.1 Unsupervised (“static”)

As for parallel combinations, “static” incremental supervisors can be used to model several DR stand-alone adaptive algorithms (see Chapter 8 for details). In contrast to the parallel topology, a strong case can also be made in favor of unsupervised incremental combinations of AFs. Indeed, the improved convergence and tracking behavior of these structures suggest that, as long as the component filters are adequately designed, there is no need for supervising parameters [43–45,73]. In fact, the optimal supervising parameters for an $\{\text{LMS}\}^2$ amounts to transforming the components into two identical VSS LMS filters [44]. Even though unsupervised incremental topologies degrade the misadjustment of the combination, the use of hybrid topologies is a far better solution than forcing the use of VSS components (see Chapter 7).

6.1.2 The convex supervisor

Together with the $\{\text{LMS}\}^2$, [41] put forward a convexly constrained adaptive supervisor for incremental combinations. Taking $N = 2$, $\eta_1(i) = \eta(i)$, and $\eta_2(i) = 1 - \eta(i)$ in (6.1), this supervising rule is written as

$$\begin{aligned} a(i) &= \alpha a(i-1) + \beta e^2(i) \\ \eta(i) &= \frac{2}{1 + e^{-a(i)}} - 1, \end{aligned} \tag{6.2}$$

with an initial value $a(-1)$, $0 \ll \alpha < 1$, and β chosen depending on the SNR [e.g., $\beta = 1 - \alpha$ for high SNR and $\beta = 0.1(1 - \alpha)$ for low SNR]. An incremental counterpart of the coefficients leakage from Section 5.2.2 was also used in [41] to improve the convergence of the slower filter. Nevertheless, this leakage becomes unnecessary when the convex constraint is lifted.

6.1.3 The $\{\text{LMS}\}^2$ -specific supervisor

Though effective, the convex constraint imposed in [41] was motivated by parallel combinations and is unwarranted for the incremental topology. Indeed, it has no relation to the unbiasedness of these structures and actually limits the performance of the combination. Figure 23 illustrates this effect by showing the performance of two supervisors with and without the convexity constraint. A small difference between the step sizes is used to make this effect more evident.

Based on the transient analysis of the $\{\text{LMS}\}^2$, [44] introduced a different supervising

Figure 23: The effect of the convexity constraint on the incremental supervisor ($\{\text{LMS}\}^2$). **White input data:** $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\mu_1 = 0.005$, and $\mu_2 = 0.003$. **Convexly constrained:** $\{1 + \exp[0.05(i - 1100)]\}^{-1}$ (deterministic supervisor); **no constraint:** $\{1 + \exp[0.05(i - 700)]\}^{-1}$ (deterministic supervisor)

rule based on global MSE estimates. To achieve maximum convergence speed, the supervising parameters in (6.1) are evaluated as $\eta_1(i) = \eta(i)/\mu_1$ and $\eta_2(i) = \eta(i)/\mu_2$ based on a single parameter $\eta(i)$ adapted as

$$\begin{aligned} a(i) &= \alpha a(i-1) + (1-\alpha)e^2(i) \\ \eta(i) &= f[a(i)], \end{aligned} \tag{6.3}$$

where $0 < \alpha < 1$ and $f(x)$ is a linear activation function, instead of the sigmoidal function employed in [41]. The coefficient of f depends on the value of the MSE [approximated by $a(i)$ in (6.3)] and, therefore, on the SNR and step sizes. To guarantee that the combination remains stable, $\eta(i)$ was limited by the maximum value of the optimal supervisor (see Section 11.3.4), explicitly

$$\eta(i) \leq \frac{1}{3(M+2)\sigma_u^2}. \tag{6.4}$$

Figure 24 shows that this approach is indeed an improvement over (6.2).

6.2 Incremental topologies

Purely incremental topologies are more limited than parallel ones, since the AFs have to form a chain. The structures presented below are different only in whether the chain is open (Section 6.2.1) or closed (Section 6.2.2).

Figure 24: Comparison between different supervising rules for a combination of two LMS filters. **White input data:** $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\mu_1 = 0.05$, and $\mu_2 = 0.005$. At iteration $i = 2000$, $\mathbf{w}^o \rightarrow -\mathbf{w}^o$. **Parallel-independent:** $\mu_a = 1.5$, $\beta = 0.9$, and $\epsilon = 10^{-4}$ (normalized convex supervisor); **cyclic coefficients feedback:** $L = 50$, $\mu_a = 1.1$, $\beta = 0.9$, and $\epsilon = 10^{-4}$ (normalized convex supervisor); **convexly constrained incremental:** $\alpha = 0.98$ and $\beta = 1 - \alpha$ [supervisor (6.2)]; **incremental-specific supervisor:** $\alpha = 0.97$ and $f(x) = 1.1x$ [supervisor (6.3)].

Figure 25: Incremental topology digraphs: (a) linear incremental topology and (b) ring incremental topology.

6.2.1 Linear incremental topology

The *linear incremental topology* is not very common in the adaptive filtering literature. Its name comes from the fact that it is described by a linear digraph (Figure 25a) [78]. They are therefore “open chains” of AFs: the first filter runs independently of the rest of the combination, which operates over its updated coefficients. In other words, no filter in the combination, except the first, is recursive since their *a priori* coefficients have no relation to the value of their coefficients in previous iterations.

Notice that the linear incremental topology is not strongly connected and thus cannot promote full cooperation between its component filters. This may, however, be the intended behavior, such as when an AF is used to initialize another. This is a classical case

of unilateral cooperation and is the setup illustrated in Figure 25a. As long as $\delta = 1$, the coefficients of filter 2 is initialized by filter 1, which updates its estimate independently of the rest of the combination. When $\delta = 0$, filter 1 is disconnected from the structure and filter 2 takes over the updates.

The literature has some examples of this initialization process. For instance, [67] proposes to use an LMS filter to initialize an LMF filter, since the stability and performance of the latter is influenced by its initial coefficients. Another example is the initialization of an IIR adaptive algorithm using an FIR AF. In [64], LMS and RLS initializations were used to improve the convergence and stability of IIR LMS and Steiglitz-McBride AFs.

6.2.2 Ring incremental topology

The ring incremental topology (or simply incremental topology) was used in the first incremental combinations of AFs [41] and it has remained ubiquitous since [43–45, 73]. Again, it is named after the form of its digraph (Figure 25b). This topology builds on the linear one by adding a global coefficients feedback, effectively forming a closed loop. In contrast to parallel topologies, there is typically no cycle in this feedback.

Examining Figure 25b, it is straightforward to see that it is a strongly connected topology. Once more, full cooperation between the component filters leads to a better performance: it is the ring topology that gives rise to the improved convergence and tracking behaviors credited to incremental combinations (Figure 26 compares it to the APA for illustration, see Chapter 9 for more details) [43–45, 73]. Besides the orientation of the arcs, an important distinction between the digraphs for the parallel topology with coefficients feedback (Figure 17) and the ring incremental topology (Figure 25b) is that the former has a delay (z^{-1}) in all the paths that connect its component filters. The presence of these delays is responsible for the difference in synchronicity between the components in these combinations.

6.3 Concluding remarks

The incremental topology has been introduced in combinations of AFs more recently and has, therefore, been less explored. Still, it has already shown fruitful results when it comes to improving convergence rate and tracking. In terms of supervising rules, a strong case can be made for unsupervised incremental combinations, contrary to parallel mixtures. Indeed, the supervisor proposed thus far can be likened to a VSS algorithm.

Figure 26: Comparison between the $\{\text{LMS}\}^N$ and the APA. **White input data:** $\sigma_u^2 = 1$ and $\sigma_v^2 = 10^{-3}$. **APA:** $\mu = 0.2$ and $K = 5$; **$\{\text{LMS}\}^N$:** $\mu_n = 0.03$ (unsupervised, $\eta_n(i) = 1$).

Moreover, outstanding performance has been repeatedly achieved using fixed supervising parameters, thus suggesting that they could be absorbed by the component filters recursions. Finally, the analyses conducted in Chapter 11 show that the optimal supervising parameter values are similar to the optimal instantaneous step size for individual AFs and that the combination's unbiasedness constraint has strong relations to mean-square stability results for stand-alone adaptive algorithms. On the other hand, unsupervised incremental combinations are known to degrade the misadjustment of its component filters. However, due to the intricacy of developing robust and effective adaptive supervisor for these combinations, the hybrid topology solutions explored in the next chapter appear to be a natural alternative.

7 COMBINATIONS WITH ARBITRARY TOPOLOGIES

You can't get to the moon by climbing successively taller trees.

-- Mo's Law of Evolutionary Development (Akin's Laws of Spacecraft Design)

The true power of combinations becomes evident when structures involving both parallel and incremental topologies are designed. The additional degrees-of-freedom allow these combinations to address more complex trade-offs and meet more stringent performance requirements. However, the literature still has few examples of such combinations, which is to be expected given that the study of the topology of combinations is a new topic. Two cases are illustrates in the sequel.

7.1 Cyclically incremental, parallel topology

In an attempt to improve the leakage solution to the convergence stagnation issue, a *coefficients handover* approach was explored in [96]. In this technique, the updated coefficient vector of the fast filter is cyclically and conditionally “copied” to the slow filter. Once again assuming component 1 is faster, this combination for two LMS filters is

$$\begin{aligned} \mathbf{w}_{1,i} &= \mathbf{w}_{1,i-1} + \mu_1 \mathbf{u}_i^T e_1(i) \\ \begin{cases} \mathbf{w}_{2,i} = \mathbf{w}_{1,i}, & \eta(i) \geq \tau \text{ and } \delta_L(i) = 1 \\ \mathbf{w}_{2,i} = \mathbf{w}_{2,i-1} + \mu_2 \mathbf{u}_i^T e_2(i), & \text{otherwise} \end{cases} \\ \mathbf{w}_i &= \eta(i) \mathbf{w}_{1,i} + [1 - \eta(i)] \mathbf{w}_{2,i}, \end{aligned} \tag{7.1}$$

where, as in Section 5.2.3, $\delta_L(i)$ is an impulse train of period L and $0 \ll \tau < 1$. Notice that while the combination is tracking filter 1 ($\eta(i) \approx 1$), its updated coefficients are cyclically transferred to filter 2, which violates synchronicity. Hence, according to our definitions this combination cannot be purely parallel. However, whenever $\eta(i) < \tau$ or $\delta_L(i) = 0$, (7.1) reduces to a parallel-independent combination. The structure in [96] has therefore a dynamical topology: it is a *cyclically incremental, parallel combination*, i.e., it is a parallel structure in which every L iterations, an incremental iteration is performed, if some condition is met.

Figure 27: Cyclically incremental, parallel combination of LMS filters. **White input data:** $M = 7$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.07$, and $\mu_2 = 0.007$. (a) **Nonstationary w^o** ($w^o \rightarrow -w^o$ at $i = 1500$): **cyclically incremental:** $L = 10$, $\tau = 0.98$, and $\mu_a = 250$ (convex supervisor); **cyclic coefficients feedback:** $L = 50$ and $\mu_a = 300$ (convex supervisor). (b) **Nonstationary SNR** ($\sigma_v^2 = 10^{-1}$ at $i = 1000$ and $\sigma_v^2 = 10^{-3}$ at $i = 2000$): **cyclically incremental:** $L = 2$, $\tau = 0.98$, $\mu_a = 1.9$, $\beta = 0.9$, and $\epsilon = 10^{-3}$ (normalized convex supervisor); **cyclic coefficients feedback:** $L = 50$, $\mu_a = 1$, $\beta = 0.9$, and $\epsilon = 10^{-4}$ (normalized convex supervisor).

As a solution to the convergence stagnation issue, this combination is indeed effective (Figure 27). However, it suffers from many of the problems of the coefficients leakage technique. Mainly, it is unidirectional, which can hinder its application in practical scenarios, and is hard to use in combinations with more than two AFs.

7.2 Parallel-incremental combination

Ring incremental combinations display improved convergence rates at the cost of degrading the misadjustment of the component filters. Increasing the length of an incremental chain therefore speeds up convergence, but also increases the steady-state error. A similar trade-off is found in stand-alone adaptive algorithms due to the step size [18]. A possible solution would be to dynamically adjust N so that the combination uses less filters as it approaches steady-state. This can be achieved using a structure as in Figure 28a, which shows a parallel combination operating over the AFs in an incremental chain. Notice that this is equivalent to adapting $\eta(i)$ in the $\mathcal{FOB}$ description (3.7). However, performing this adaptation in a way that is effective and robust is challenging due to the small difference between the outputs of AFs combined incrementally. This issue was addressed in parallel combinations by making the coefficients feedback cyclical, a solution

Figure 28: Graph representation of parallel-incremental topologies: (a) parallel combination of incremental components and (b) parallel combination of incremental combinations.

that is not suitable for this structure.

An alternative that leverages both incremental and parallel topologies was proposed in [45]. The *parallel-incremental* structure combines incremental branches in parallel, thus allowing a cyclic coefficients feedback to be embedded in the topology (Figure 28b). Although at a higher computational cost, the overall effect of this topology is the desired one: adjusting N dynamically. The number of branches determines how fine this adjustment is, but given the switching is smooth, few branches are usually enough. In [45], for example, only two branches were used with significant performance improvements. The results in this case are presented in Figure 29 for white and correlated inputs. Using additional incremental branches with less component filters would improve the convergence behavior (specially in the correlated scenario), although the steady-state performance would remain the same.

7.3 Concluding remarks

The hybrid topologies explored in this chapter provide effective ways of improving further the performance of combinations, as well as exploiting the advantages of both parallel and incremental structures. Still, they remain simple topologies. Whether the use of more complex mixtures, actual networks of AFs, would allow to deal with more intricate performance requirements is still an open question. Designing such systems would require more general analysis frameworks, such as the one put forward in Chapter 3, that are still

Figure 29: Parallel-incremental combination of LMS filters: $M = 100$, $\sigma_u^2 = 1$, and $\sigma_v^2 = 10^{-3}$. (a) **White input data**: $N = 7$, $\mu_n = 0.006$, $\mu_{\text{LMS}} = 0.003$, $L = 500$, $\mu_a = 0.7$, $\beta = 0.95$, and $\epsilon = 10^{-4}$ (normalized convex supervisor). (b) **Correlated input data**: $\gamma = 0.95$, $N = 15$, $\mu_n = 0.002$, $\mu_{\text{LMS}} = 3 \cdot 10^{-4}$, $L = 500$, $\mu_a = 0.7$, $\beta = 0.9$, and $\epsilon = 10^{-4}$ (normalized convex supervisor).

being developed. Clearly, however, this is a fruitful avenue of research.

8 STAND-ALONE ADAPTIVE ALGORITHMS AS COMBINATIONS OF ADAPTIVE FILTERS

*What's in a name? that which we call a rose
By any other name would smell as sweet*

-- William Shakespeare, "Romeo and Juliet"

Section 3.2 showed that stand-alone AFs are special cases of combinations of AFs according to Definition 3.1, i.e., “combinations” of a single AF. Although this fact supports the generality of this definition, it is trivial and of limited conceptual novelty. On the other hand, some adaptive algorithms can be broken down into combinations of simpler AFs and cast under Definition 3.1. For instance, adaptation rules involving mixed cost functions and data reusing fit intuitively within the combination framework. They were, however, seen exclusively as indivisible units until the introduction of the incremental topology [41], DR combinations [73], and the concept of coefficients feedback [42]. Indeed, it was not possible to write stand-alone AFs as combinations without these structures, even though it made sense to do so. Hence, they are responsible for bridging a conceptual gap between stand-alone AFs and combinations.

Section 3.5 already provided an example of this concept by recasting a TRUE-LMS filter. The remainder of this chapter presents additional instances that illustrate how entire classes of adaptive algorithms can be written as combinations.

8.1 Mixed cost function algorithms

These algorithms were put forward as solutions to optimization problems involving composite cost functions, notably different moments of the output estimation error [26–29]. For instance, the Least Mean Mixed-Norm (LMMN) algorithm [28] minimizes the cost function

$$J(\mathbf{w}_a) = \text{E} \left[\delta e^2(i) + \frac{1}{2}(1 - \delta)e^4(i) \right],$$

where $0 \leq \delta \leq 1$ is a mixing constant, using the recursion in (2.8) reproduced here for ease of reference:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e(i) [\delta + (1 - \delta)e^2(i)] .$$

This algorithm can be posed as an LMS + LMF combination with coefficients feedback ($L = 1$). Indeed, assuming filter 1 is the LMS and filter 2 is the LMF, the global coefficients of a parallel combinations (5.5) can be expanded into

$$\begin{aligned} \mathbf{w}_i &= \eta(i)\mathbf{w}_{1,i} + [1 - \eta(i)]\mathbf{w}_{2,i} \\ &= \eta(i)[\mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e(i)] + [1 - \eta(i)][\mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e^3(i)] \\ &= \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T e(i) [\eta(i) + [1 - \eta(i)]e^2(i)] . \end{aligned}$$

For a static supervisor ($\eta(i) = \delta$), the LMMN recursion is indeed recovered. Extensions to other mixed-norm algorithms, such as the LMS+F (an adaptive LMS + LMF) [26, 27] and the Robust Mixed-Norm (RMN, an adaptive LMS + sign-error LMS) [29] are straightforward.

Mixed cost function AFs are not necessarily composed of different error norms. For instance, the Improved Proportionate NLMS (IPNLMS) [69] can be written as a parallel combination of an NLMS and a PNLMS with coefficients feedback. A similar idea was explored in [138] in which a heuristics was used to switch between the AFs. However, casting this algorithm as a combination motivate the use of any of the supervising rules presented in Section 3.4.

8.2 Data reusing algorithms

As introduced in Section 3.3, DR in adaptive filtering refers either to the simultaneous use of different data pairs or to the use of a single data pair several times. Both classes of algorithms can be cast as combinations, the former using a parallel topology and the latter, an incremental one.

For instance, take the TRUE-LMS and TRUE-NLMS¹. An $\mathcal{FOB}$ description of the former was already provided in Section 3.5, so it clearly fits in Definition 3.1. For illustration purposes, the equivalent combination of the TRUE-NLMS is derived directly from

¹Recall that these AFs are sometimes called DR-LMS and DR-NLMS [25], which are also the names used to denote algorithms that operate several times over the same data pair [22, 70].

its recursion, namely

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{U}_i^T \mathbf{N}_i \mathbf{e}_i, \quad (\text{TRUE-NLMS}) \quad (8.1)$$

where $\mathbf{e}_i = \mathbf{d}_i - \mathbf{U}_i \mathbf{w}_{i-1}$ and $\mathbf{N}_i = \text{diag} \{1/\|\mathbf{u}_i\|^2, \dots, 1/\|\mathbf{u}_{i-K+1}\|^2\}$. To do so, rewrite (8.1) as

$$\mathbf{w}_i = \sum_{k=0}^{K-1} \frac{1}{K} \underbrace{\left\{ \mathbf{w}_{i-1} + \frac{\mu'}{\|\mathbf{u}_{i-k}\|^2} \mathbf{u}_{i-k}^T [d(i-k) - \mathbf{u}_{i-k} \mathbf{w}_{i-1}] \right\}}_{\mathbf{w}_{n,i}}, \quad (8.2)$$

where $\mu' = \mu K$. Then, notice that (8.2) has the form of a parallel combination with $\eta_n(i) = 1/K$ and $\{\mathbf{w}_{n,i}\}$ as indicated. Hence, the TRUE-NLMS recursion (8.1) is equivalent to the data buffering parallel combination with coefficients feedback

$$\mathbf{w}_{n,i} = \mathbf{w}_{i-1} + \frac{\mu'}{\|\mathbf{u}_{i-k}\|^2} \mathbf{u}_{i-k}^T [d(i-k') - \mathbf{u}_{i-k'} \mathbf{w}_{i-1}] \quad (8.3a)$$

$$\mathbf{w}_i = \sum_{n=1}^N \eta_n \mathbf{w}_{n,i}, \quad (8.3b)$$

for $N = K$, $k' = n - 1$, $\eta_n = 1/K$, and $L = 1$ (permanent coefficients feedback).

The second type of DR algorithm (using a signal data pair several times) can be interpreted as an incremental combination. Indeed, take the DR-LMS from (2.10) as illustration [70, 88]:

$$\begin{cases} \mathbf{w}_{0,i} = \mathbf{w}_{i-1} \\ \mathbf{w}_{k,i} = \mathbf{w}_{k-1,i} + \mu \mathbf{u}_i^T [d(i) - \mathbf{u}_i \mathbf{w}_{k-1,i}] \\ \mathbf{w}_i = \mathbf{w}_{K,i} \end{cases}$$

Its recursion is identical to an unsupervised $\{\text{LMS}\}^N$ combination (Chapter 6). The data buffering version proposed in [22] can also be interpreted as a combination, specifically, a DR- $\{\text{LMS}\}^N$ with $N = K$. Notice that, although effective, this algorithm cannot match the performance of more complex DR AFs, such as the APA. In fact, only from a combination viewpoint was the use of $N > K$ component filters motivated, thus allowing this incremental structure to go as far as outperforming the APA [45, 73].

8.3 Variable step size algorithms

Whenever AFs with the same adaptive algorithm and different step sizes are combined using a parallel topology with coefficients feedback, the resulting overall algorithm reduces to a VSS version of the original filters [42]. Indeed, for $L = 1$ in the coefficients feedback

Figure 30: Variable step size algorithm based on a $2 \cdot \text{LMS}$ combination with coefficients feedback [nonstationary SNR: $\sigma_v^2 = 10^{-3}$ for $i \in [0, 1500)$ and $\sigma_v^2 = 10^{-1}$ for $i = [1500, 3000]$]. **White input data:** $M = 10$ and $\sigma_u^2 = 1$. **VSS-LMS** [31]: $\mu(i) = 0.99 \cdot \mu(i-1) + 10^{-3} \cdot e^2(i)$, $\mu_{\min} = 0.005$, and $\mu_{\max} = 0.05$; **coefficients feedback:** $\mu_1 = 0.05$, $\mu_2 = 0.005$, $L = 40$, $\mu_a = 1$, $\beta = 0.9$, and $\epsilon = 10^{-4}$ (normalized convex supervisor); **optimal step size (μ^o)**: evaluated as in [143, Eq. (4)].

relation (5.13) and denoting the adaptive algorithm update as $\mathbf{p}(\mathbf{w}_a)$, (5.1) yields

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \bar{\mu}(i)\mathbf{p}(\mathbf{w}_{i-1}), \quad (8.4)$$

where $\bar{\mu}(i) \triangleq \sum \eta_n(i)\mu_n$ is an iteration dependent linear combination of the components step sizes $\{\mu_n\}$. Given that $\bar{\mu}(i)$ is adjusted by the combination supervisor, this VSS algorithm inherits its complexity and robustness, making (8.4) a viable alternative to other step size adaptation techniques, such as [30, 31, 141, 142]. Indeed, Figure 30 shows that a $2 \cdot \text{LMS}$ combination with cyclic coefficients feedback is more robust to environment changes than the VSS-LMS from [31]. It is able to track the optimal step size in stringent nonstationary scenarios with low complexity (see Figure 31).

8.4 Concluding remarks

This chapter illustrated recent advances that allowed to present several stand-alone adaptive algorithms as combinations of AFs. This was made possible mainly by the introduction of coefficients feedback and the incremental topology. Though recent, this new viewpoint has lead to several developments in both stand-alone adaptive filtering and combinations of AFs. For instance, mean and mean-square performance analyses of the DR-LMS are now available using analyses of incremental combinations (Chapter 11). Conversely, this connection has also prompted the introduction of data reusing combinations. Another example is the development of VSS algorithms based on parallel

Figure 31: Variable step size algorithm based on a $2 \cdot \text{LMS}$ combination with coefficients feedback [nonstationary $\mathbf{w}^o$: $\sigma_q^2 = 10^{-4}$ for $i \in [1, 3000)$, $\sigma_q^2 = 10^{-5}$ for $i \in [3000, 6000)$, and $\sigma_q^2 = 10^{-6}$ for $i \in [6000, 10000)$]. **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.08$, and $\mu_2 = 0.005$. **VSS-LMS** [31]: $\mu(i) = 0.5 \cdot \mu(i-1) + 10^{-3} \cdot e^2(i)$, $\mu_{\max} = \mu_1$, and $\mu_{\min} = \mu_2$; **coefficients feedback:** $L = 10$, $\mu_a = 0.3$, $\beta = 0.9$, and $\epsilon = 10^{-2}$ (normalized convex supervisor); **optimal step size (μ^o):** evaluated as in [18, Lemma 21.1]. Dotted lines represent the one standard-deviation limits of the equivalent step size.

combinations with coefficients feedback, which inherit the robustness and simplicity of gradient supervising rules. Such similarities lead to the question of whether combinations can be designed to match or outperform established stand-alone adaptive algorithms, an interesting idea that is explored in the next chapter.

9 COMBINATION AS A COMPLEXITY-REDUCTION TECHNIQUE

It is better to solve the right problem approximately than the wrong problem exactly.

-- John Tukey (according to [144])

Combinations of AFs are usually viewed as a method that improves the performance of adaptive systems but increases their complexity. After all, combinations rely on the operation of more than one AF and therefore necessarily require more operations than any of their individual components. In fact, the established consensus is that a combination has the computational complexity of the sum of its components (given the low complexity of the typical supervisors) [33, 35, 40, 45, 47, 73, 92, 96, 133].

Recently, there have been attempts to reduce this overhead, specially in parallel combinations where only few components usually affect the output at any given moment. In [96], it was argued that there is no need to update the slow AF coefficients when they have just been transferred from the fast filter. Similar conditional updates could also be embedded in parallel combinations with coefficients feedback, although not necessarily without some performance cost. In contrast, the approach in [133] is more hardware-oriented: it aims at reducing the size of adders and multipliers by updating the difference between the component filters, which can be represented with a reduced word length.

The goal of these solutions can be summarized as follows: achieve the performance of a combination with the complexity of a single component filter. In contrast, this chapter is motivated by the broader objective of finding the lowest complexity combination with a given performance. Clearly, reducing the combinations computational overhead contributes considerably to this endeavor. However, these goals are not equivalent. In fact, the latter inspires comparisons between different forms of combinations as well as between combinations and stand-alone AFs. For instance, can a combination be designed to replace an APA or RLS filter in practical applications? Can it achieve the same performance? At what computational cost?

These questions are answered in the following sections by developing low complexity structures that can match and even outperform more complicated algorithms such as the APA. A graphical illustration of the complexity is shown in Figure 32, where the area

Figure 32: Graphical representation of the complexity of different algorithms with similar performances: APA, LMS, DR- $\{\text{LMS}\}^{20}$, $\{\text{sign-error LMS}\}^9$, and $\{\{\text{sign-error LMS}\}^{11}\}^{30}$ for $M = 100$ and $K = 10$. The area of the bubbles is proportional to the number of multiplications per iteration.

Figure 33: Diagram of relations between the APA and combinations of adaptive filters

of the bubbles are proportional to the number of multiplications required to get similar performances. These mixtures give rise to the counter-intuitive concept of *combination as a complexity-reduction technique*.

9.1 DR- $\{\text{LMS}\}^N$

The APA is one of the most celebrated stand-alone DR AFs [71, 145, 146]. It remains ubiquitous in applications such as speech echo cancellation, where its performance matches that of Fast RLS (FRLS) filters with three times less operations [52]. Hence, designing a combination that outperforms it with lower complexity is quite challenging. Still, this structure can be motivated in two steps: first, by deriving a relation between the APA and parallel combinations of LMS filters; then, by exploiting their incremental counterparts to take advantage of the improved convergence of this topology.

9.1.1 The APA and the $f\{K \cdot \text{LMS}\}$

This section derives a relation between the APA and parallel combinations of LMS filters. More specifically, the APA recursion is shown to converge to that of a TRUE-LMS¹ when it is over-regularized and/or operating under vanishing inputs. Notice that,

¹Recall from Section 3.5 that a TRUE-LMS with buffer size K is equivalent to an unsupervised parallel combination of K LMS filters with coefficients feedback.

although relations between the APA and the NLMS or between incremental combinations and the NLMS are well-established (Figure 33) [18, 19, 88], this result first appeared recently in [73].

To find a relation between the APA and the TRUE-LMS, compare their updates from (2.12) and (2.11). Explicitly,

$$\mu \mathbf{U}_i^T (\epsilon \mathbf{I} + \mathbf{U}_i \mathbf{U}_i^T)^{-1} \mathbf{e}_i \quad (\text{APA}) \quad (9.1a)$$

$$\mu \mathbf{U}_i^T \mathbf{e}_i \quad (\text{TRUE-LMS}) \quad (9.1b)$$

Thus, suffices to find a condition for which $(\epsilon \mathbf{I} + \mathbf{U}_i \mathbf{U}_i^T)^{-1} \approx \mathbf{I}$, in this case, $\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0$. Indeed, choosing the step size as $\mu = \mu_\epsilon \epsilon$, $\mu_\epsilon > 0$, so that the adaptation does not stall, the limit of (9.1a) as $\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0$ yields

$$\begin{aligned} & \lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0} (\mu_\epsilon \epsilon) \mathbf{U}_i^T (\epsilon \mathbf{I} + \mathbf{U}_i \mathbf{U}_i^T)^{-1} \mathbf{e}_i \approx \\ & \lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0} \mu_\epsilon \mathbf{U}_i^T \left[\lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0} \epsilon (\epsilon \mathbf{I} + \mathbf{U}_i \mathbf{U}_i^T)^{-1} \right] \mathbf{e}_i = \\ & \lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0} \mu_\epsilon \mathbf{U}_i^T \left[\mathbf{I} + \lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0} \epsilon^{-1} \mathbf{U}_i \mathbf{U}_i^T \right]^{-1} \mathbf{e}_i. \end{aligned} \quad (9.2)$$

Moving the limit in the second expression is an approximation justified by the fact that the function $h(\mathbf{A}) = \mathbf{U}^T \mathbf{A} \mathbf{e}(i)$ is continuous at $\mathbf{A} = \mathbf{I}$, in which case the TRUE-LMS update is recovered. The following theorem establishes the conditions under which the inner limit in (9.2) vanishes.

Theorem 9.1. *Let the $\{u(i)\}$ collected into the $1 \times M$ regressor vector $\mathbf{u}_i$ be samples of a zero mean real-valued wide-sense stationary random process $u(i)$ whose correlation function is $\rho(\ell) = \mathbb{E} u(i)u(i + \ell)$, $\rho(0) = \sigma_u^2$. With the covariance matrix $\mathbf{R}_u = \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i$,*

$$\lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow 0} \epsilon^{-1} \mathbf{U}_i \mathbf{U}_i^T = \mathbf{0} \quad (\text{a.s.}). \quad (9.3)$$

Proof. Multiplying the left-hand side of (9.3) by M/M evaluated as $M \rightarrow +\infty$ gives

$$\lim_{(M, \text{Tr}(\mathbf{R}_u)/\epsilon) \rightarrow (+\infty, 0)} \frac{\mathbf{U}_i \mathbf{U}_i^T}{M} M \epsilon^{-1}. \quad (9.4)$$

The strong law of large numbers [147] then guarantees that

$$\lim_{M \rightarrow +\infty} \frac{\mathbf{U}_i \mathbf{U}_i^T}{M} = \mathbf{R}_u = \begin{bmatrix} \sigma_u^2 & \cdots & \rho(M-1) \\ \vdots & \ddots & \vdots \\ \rho(M-1) & \cdots & \sigma_u^2 \end{bmatrix} \quad (\text{a.s.}),$$

so that (9.4) becomes equivalent to

$$\lim_{(M, \text{Tr}(\mathbf{R}_u)/\epsilon) \rightarrow (+\infty, \mathbf{0})} \begin{bmatrix} \sigma_u^2 & \cdots & \rho(M-1) \\ \vdots & \ddots & \vdots \\ \rho(M-1) & \cdots & \sigma_u^2 \end{bmatrix} M\epsilon^{-1} \quad (\text{a.s.}) \quad (9.5)$$

At this point, recall that a fundamental property of the correlation function is that $|\rho(\ell)| \leq \rho(0) = \sigma_u^2$, $\forall \ell \neq 0$. In other words, the off-diagonal elements in (9.5) are upper bounded by σ_u^2 [147]. Hence, suffices to show that the diagonal elements vanish to prove that the whole matrix converges to the null matrix. Indeed,

$$\lim_{(M, \text{Tr}(\mathbf{R}_u)/\epsilon) \rightarrow (+\infty, \mathbf{0})} \frac{M\sigma_u^2}{\epsilon} = \lim_{\text{Tr}(\mathbf{R}_u)/\epsilon \rightarrow \mathbf{0}} \frac{\text{Tr}(\mathbf{R}_u)}{\epsilon} = 0 \quad (\text{a.s.}) \quad \square$$

Applying Theorem 9.1 to the update expression (9.2) and rewriting the complete recursion leads to

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu_\epsilon \mathbf{U}_i^T \mathbf{e}_i,$$

which the same as an unsupervised parallel combination of LMS filters with coefficients feedback ($f\{K \cdot \text{LMS}\}$, Figure 33) or, equivalently, a TRUE-LMS.

Figure 34 illustrates the relation derived above. Figure 34a shows the effect of over-regularization and Figure 34b the effect of vanishing input. Note that μ_{APA} and σ_v^2 were chosen for clarity's sake, so as to emphasize the relation between APA and $f\{K \cdot \text{LMS}\}$. Also, in Figure 34b the step sizes were increased as the variance decreased to keep the all curves in the same iteration range. Nonetheless, it is clear that as $\sigma_u^2/\epsilon \rightarrow 0$, the APA's behavior approaches that of a parallel combination of LMS filters (TRUE-LMS) [73].

9.1.2 The incremental counterpart of $f\{K \cdot \text{LMS}\}$

Although related to the APA, the $f\{K \cdot \text{LMS}\}$ (TRUE-LMS) cannot match its transient behavior in most scenarios [25, 61]. However, it has been observed that incremental topologies are able to achieve faster convergence rates than parallel ones [44, 45, 73]. This suggests that an incremental counterpart of the TRUE-LMS filter is a good candidate to compete with the APA.

Indeed, the DR- $\{\text{LMS}\}^N$ was shown in [73] to be able to match (and sometimes outperform) the APA with a considerably lower complexity. To do so, it relies on a circular buffer data distribution strategy, so that more than K component filters can be

Figure 34: APA and $f\{K \cdot \text{LMS}\}$. **White input data:** $M = 16$ and $K = 4$. (a) $\epsilon \rightarrow \infty$: $\mu_\epsilon = 0.05$, $\mu_{K \cdot \text{LMS}} = K\mu_\epsilon$, $\mu_{\text{APA}} = \mu_\epsilon \epsilon$, $\sigma_u^2 = 1$, and $\sigma_v^2 = 10^{-3}$. (b) $\sigma_u^2 \rightarrow 0$: $\mu_{\text{APA}} = 0.3/\sigma_u^2$, $\mu_{K \cdot \text{LMS}} = K\mu_{\text{APA}}$, $\epsilon = 0.1$, and $\sigma_v^2 = 10^{-6}$.

used. The recursion of this combination reads

$$\begin{aligned}
 \mathbf{w}_{0,i} &= \mathbf{w}_{i-1} \\
 \mathbf{w}_{n,i} &= \mathbf{w}_{n-1,i} + \mu_n \mathbf{u}_{i-k}^T [d(i-k) - \mathbf{u}_{i-k} \mathbf{w}_{n-1,i}], \quad n = 1, \dots, N \\
 \mathbf{w}_i &= \mathbf{w}_{N,i},
 \end{aligned} \tag{9.6}$$

for $k = (n-1) \bmod K$. Notice that, when $N = K$, (9.6) does not make use of a circular buffer and reduces to a stand-alone DR AF introduced in [22].

The $\text{DR-}\{\text{LMS}\}^N$ presents several advantages over the APA. Its update only requires $O(NM)$ multiplications compared to $O(K^2M)$ for the classical APA [18] or $O(3K^2 + KM)$ for fast APA implementations [146]. The combination is therefore less computationally intensive for a wide range of N and, as simulations show (Figures 35 and 36), it can match the APA with $N \ll K^2$ (i.e., at a lower computational cost). Notice that the advantage of using data buffering becomes clear when the input data are correlated (see Figure 35b). For white input data (Figure 35a), data sharing and data buffering yield the same results (which is why curves for the latter were omitted).

Besides its complexity, another important feature of this structure is that it inherits the well-known robustness and stability of the LMS filters it is based on [18, 148]. Given the recognized numerical issues of the APA in lower precision environments [18, 19, 21], the $\text{DR-}\{\text{LMS}\}^N$ is expected to yield even better results in embedded applications. Finally, note that in (9.6) the updates of combination output are made in blocks, i.e., after going through all components. This was done so that the algorithm is in the same time scale as the APA, although more frequent updates of the overall coefficients are possible.

Figure 35: APA and DR- $\{\text{LMS}\}^N$ (stationary scenario): $M = 100$, $K = 10$, $\sigma_u^2 = 1$, and $\sigma_v^2 = 10^{-3}$. (a) **White input data**: $\mu_{\text{APA}} = 0.05$ and $\mu_n = 5 \cdot 10^{-4}$. (b) **Correlated input data**: $\gamma = 0.95$, $\mu_{\text{APA}} = 1$, $\mu_{\{\text{LMS}\}^{30}} = 15 \cdot 10^{-4}$, and $\mu_{\text{DR-}\{\text{LMS}\}^{15}} = 0.002$.

9.2 DR- $\{\text{LMS}\}^N + \text{LMS}$

Although the convergence rate of (9.6) was fast enough to compete with the APA, this transient behavior comes at the cost of a larger misadjustment [45]. In fact, incremental combinations display a trade-off between transient and steady-state performance, as illustrated in Figure 35a. These observations motivate the use of the parallel-incremental topology with coefficients feedback from Section 5.2.3. In [45], two branches were enough to match the performance of even combinations of APA, leading to the DR- $\{\text{LMS}\}^N + \text{LMS}$. Its recursion is reproduced below for $n = 1, \dots, N$ and $k = (n - 1) \bmod K$ (circular buffer) [45]:

- i. *Cyclic coefficients feedback*

$$\mathbf{w}_{n,a} = \delta_L(i) \mathbf{w}_{i-1} + [1 - \delta_L(i)] \mathbf{w}_{n,i-1}$$

- ii. *Incremental combination with circular buffer branch (DR- $\{\text{LMS}\}^N$)*

$$\mathbf{w}'_{0,i} = \mathbf{w}_{1,a}$$

$$\mathbf{w}'_{n,i} = \mathbf{w}'_{n-1,i} + \mu_1 \mathbf{u}_{i-k}^T [d(i-k) - \mathbf{u}_{i-k} \mathbf{w}'_{n-1,i}]$$

$$\mathbf{w}_{1,i} = \mathbf{w}'_{N,i}$$

- iii. *LMS filter branch*

$$\mathbf{w}_{2,i} = \mathbf{w}_{2,a} + \mu_2 \mathbf{u}_i^T [d(i) - \mathbf{u}_i \mathbf{w}_{2,a}]$$

Figure 36: APA and $\text{DR-}\{\text{LMS}\}^N$ (nonstationary scenario): $M = 60$, $K = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, and $\mathbf{Q} = 10^{-6}\mathbf{I}$. **Correlated input data:** $\gamma = 0.95$, $\mu_{\text{APA}} = 0.1$, and $\mu_{\text{DR-}\{\text{LMS}\}^{20}} = 0.003$.

iv. *Parallel combination*

$$\mathbf{w}_i = \eta(i)\mathbf{w}_{1,i} + [1 - \eta(i)]\mathbf{w}_{2,i}$$

Figures 37 and 38 compare the performance of these combinations to an APA + NLMS without [37] and with [45] coefficients feedback. First, notice that the computational complexity of combinations of APAs with different ranks K is almost the same as that of a single APA. This is due to the fact that order-recursive methods can be used to solve the matrix inversion in its recursion. Hence, the inverse for all orders up to K is available [37]. The steady-state behavior of the algorithm can therefore be improved at almost no cost. To address the convergence stagnation issue in these combinations, [45] embedded a cyclic coefficients feedback in these structures. Still, the $\text{DR-}\{\text{LMS}\}^N + \text{LMS}$ achieves the same performance of these APA combinations with an almost ten-fold reduction in the number of operations.

9.3 $\{\text{sign-error LMS}\}^N$

Modern adaptive filtering applications, such as telecommunication and speech echo cancellation, simultaneously require good learning performance and high throughput (i.e., low complexity algorithms) [38, 74, 75, 98]. Typically, however, performance and complexity are competing objectives: AFs such as the APA or RLS generally achieve better

performance than LMS filters but also entail more computation [18, 19]. Moreover, developers have been migrating from DSPs to dedicated processing solutions (FPGAs), where multiply-and-accumulate (MAC) operations are costly and may not be readily available [98]. In this context, several solutions have been proposed to reduce the complexity of high performance AFs and adequate their recursions for FPGA implementation, such as the use of dichotomous coordinate descent (DCD) iterations [149, 150].

The sign-error LMS was also initially proposed as a reduced complexity alternative to the LMS filter [151]. Its recursion (2.3) is reproduced below for ease of reference

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i^T \text{sign}[e(i)].$$

Notice that if the step size μ takes on a power-of-two value, the whole algorithm can be implemented in hardware using only bit shifts (except, of course, for the multiplications involved in the filtering process $y(i) = \mathbf{u}_i \mathbf{w}_{i-1}$) [18, 19, 151]. Albeit effective, this AF displays slow convergence (illustrated in Figure 39) and bias issues: the mean coefficients only converge to a ball around $\mathbf{w}^o$ with radius proportional to μ [18, 19, 151]. It is therefore unable to match the performance of the LMS algorithm.

Motivated by the convergence improvements enabled by the incremental topology, the

Figure 37: Parallel-incremental combination. **White input data:** $M = 100$ and $K = 10$. **APA + NLMS:** $\mu_{\text{APA}} = 0.3$, $\mu_{\text{NLMS}} = 0.3$, $\mu_a = 0.8$, $\beta = 0.95$, and $\epsilon = 10^{-4}$ (normalized convex supervisor); **f{APA+NLMS}**: $L = 400$; **LMS+DR-{LMS}^N**: $\mu_{\text{LMS}} = 3 \cdot 10^{-3}$, $\mu_{\text{DR-LMS}} = 6 \cdot 10^{-3}$, $N = 7$, $L = 400$, $\mu_a = 0.7$, $\beta = 0.95$, and $\epsilon = 10^{-4}$ (normalized convex supervisor).

Figure 38: Parallel-incremental combination. **Correlated input data:** $M = 100$ and $K = 10$. **APA + NLMS:** $\mu_{\text{APA}} = 1$, $\mu_{\text{NLMS}} = 0.04$, $\mu_a = 0.8$, $\beta = 0.95$, and $\epsilon = 10^{-4}$ (normalized convex supervisor); **f{APA + NLMS}:** $L = 400$; **LMS+DR-{\textbf{LMS}}^N:** $\mu_{\text{LMS}} = 4 \cdot 10^{-4}$, $\mu_{\text{DR-LMS}} = 2 \cdot 10^{-3}$, $N = 15$, $L = 500$, $\mu_a = 0.7$, $\beta = 0.95$, and $\epsilon = 10^{-4}$ (normalized convex supervisor).

following incremental combination of sign-error LMS filters was proposed in [47]:

$$\begin{aligned}
 \mathbf{w}_{0,i} &= \mathbf{w}_{i-1} \\
 \mathbf{w}_{n,i} &= \mathbf{w}_{n-1,i} + \mu_n \mathbf{u}_i^T \text{sign}[e_n(i)] \\
 \mathbf{w}_i &= \mathbf{w}_{N,i},
 \end{aligned} \tag{9.7}$$

for $e_n(i) = d(i) - \mathbf{u}_i \mathbf{w}_{n-1,i}$ and $\mu_n = 2^{-P-n}$, where $P \in \mathbb{Z}$. Note that, once again, this combination is unsupervised ($\eta_n(i) = 1$). Moreover, the choice of step size in (9.7) is optimal in the sense that it minimizes the number of component filters in a finite precision binary environment [47].

In contrast to the structures from the previous sections, (9.7) is a data sharing combination. The reason for not exploiting more complex data distribution networks is to enable an efficient implementation of (9.7). This implementation is obtained by noticing that, for $\mathbf{u}_i$ with shift structure (transversal filter), the $(n+1)$ -th component error can be expanded to become

$$\begin{aligned}
 e_{n+1}(i) &= d(i) - \mathbf{u}_i \mathbf{w}_{n,i} \\
 &= d(i) - \mathbf{u}_i \{ \mathbf{w}_{n-1,i} + \mu_n \mathbf{u}_i^T \text{sign}[e_n(i)] \} \\
 &= e_n(i) - \mu_n \|\mathbf{u}_i\|^2 \text{sign}[e_n(i)].
 \end{aligned} \tag{9.8}$$

Due to the shift structure of $\mathbf{u}_i$, one has $\|\mathbf{u}_i\|^2 = \|\mathbf{u}_{i-1}\|^2 - u^2(i-M) + u^2(i)$, which only requires 1 multiplication given that all $u^2(j)$ for $j < i$ can be made available from previous

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

$$\|\mathbf{u}_i\|^2 = \|\mathbf{u}_{i-1}\|^2 - u^2(i - M) + u^2(i) \quad \triangleright (1) \times$$

$$y(i) = \mathbf{u}_i \mathbf{w}_{i-1} ; e_1(i) = d(i) - y(i) \quad \triangleright (M) \times$$

$$\mathbf{w}_{0,i} = \mathbf{w}_{i-1}$$

for $n = 1, \dots, N$

$$\mathbf{w}_{n,i} = \mathbf{w}_{n-1,i} + \mu_n u_i^T \text{sign}[e_n(i)]$$

$$e_{n+1}(i) = e_n(i) - \mu_n \|\mathbf{u}_i\|^2 \text{sign}[e_n(i)]$$

end

$$\mathbf{w}_i = \mathbf{w}_{N,i}$$

Figure 39: Performance of the $\{\text{sign-error LMS}\}^N$ in fixed point precision (stationary scenario). **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, signed 16 bits two's complement with 13 bits fractional part (Q2.13). **sign-error LMS:** $\mu = 0.004$; **LMS:** $\mu = 0.083$; **$\{\text{sign-error LMS}\}^9$:** $\mu_n = 2^{-3-n}$.

iterations. This efficient implementation is summarized in Algorithm 1 and only requires $M + 1$ multiplications. Notice that this value does not depend on N , the number of component filters. Alternatively, the filtering process in $y(i)$ could be implemented using distributed arithmetic (DA) [98, 152, 153], leaving a single multiplication by iteration. Despite its lower complexity, this combination is indeed able to achieve the performance of an LMS filter, as shown in Figure 39. In fact, it was shown in [47] that this incremental combination converges to an NLMS filter as $N \rightarrow \infty$ (in a W bits environment, it is enough to have $N > W$). This explains why it is able to compete with and outperform these AFs in nonstationary scenarios (Figure 40).

Figure 40: Performance of the $\{\text{sign-error LMS}\}^N$ in fixed point precision (nonstationary scenario). $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, signed 16 bits two's complement with 13 bits fractional part (Q2.13). (a) **sign-error LMS**: $\mu = 2^{-7}$; **LMS**: $\mu = \mu^o = 0.0916$ [18, Lemma 21.1]; **{sign-error LMS}**⁶: $\mu_n = 2^{-4-n}$. (b) **sign-error LMS**: $\mu = 2^{-6}$; **LMS**: $\mu = 0.02$; and **NLMS**: $\mu = 0.08$ and $\epsilon = 2^{-13+1}$; **{sign-error LMS}**⁵: $\mu_n = 2^{-5-n}$.

9.4 DR- $\{\{\text{sign-error LMS}\}^N\}^L$

The ability of the $\{\text{sign-error LMS}\}^N$ to match the performance of an LMS filter with lower complexity suggests that it could be used as the building block of larger combinations. More specifically, it could be used to replace the LMS filters in the DR- $\{\text{LMS}\}^N$ from Section 9.1 and further reduce its complexity. An overview of the resulting algorithm, the DR- $\{\{\text{sign-error LMS}\}^N\}^L$, is presented in Algorithm 2. This implementation requires $LM + 1$ multiplications. An optimized version (Algorithm 3) can be obtained by further exploiting the shift structure of $\mathbf{u}_i$, creating buffers for $u(i-p-1)u(i-q-1)$ and $\mathbf{u}_{i-p-1}\mathbf{u}_{i-q-1}^T$, $p, q < K + M$, and updating $\mathbf{e}_i$ after each component filter. Albeit more complicated, this implementation only uses $M + K$ multiplications. A comparison to the complexity of other adaptive algorithms with similar performance (Figure 41) can be found in Table 6. Notice that simulations in Figure 41 were performed in double precision due to the susceptibility of the APA to reduced word length [18, 19, 21].

9.5 Concluding remarks

This chapter explored a novel application of combinations of AFs, made possible by recent developments such as DR combinations and the incremental topology. Although it was presented as an answer to “Design a low complexity combination with the perfor-

Table 6: Number of multiplications for different algorithms with similar performance

AF	$\times$	e.g., Fig. 41b
Standard APA	$(K^2 + 2K)M + K^3 + K$	13010
DCD-APA [150]	$M + K^2 + 3K + 2$	232
DR- $\{\text{LMS}\}^L$ [73]	$(2M + 1)L, L \ll K^2$	6030
Algorithm 2	$LM + 1, L \ll K^2$	3001
Algorithm 3	$M + K$	110

Figure 41: APA and DR- $\{\{\text{sign-error LMS}\}^N\}^L$: $M = 100$ and $K = 10$. (a) **APA**: $\mu = 0.3$ and $\epsilon = 10^{-6}$; **{{sign-error LMS}}²¹**: $\mu_n = 2^{-n+1}$; **DR-{{sign-error LMS}}²¹**¹⁰: $\mu_n = 2^{1-n}$. (b) **APA**: $\mu = 0.06$ and $\epsilon = 10^{-6}$; and **{{sign-error LMS}}⁶**: $\mu_n = 2^{-6-n}$. **DR-{{sign-error LMS}}¹¹**³⁰: $\mu_n = 2^{-14-n}$.

mance of X", these combinations can also be interpreted as different (or approximate) implementation of classical algorithms. Take, for instance, the NLMS algorithm. It can be implemented either directly, using recursion (2.4), or as an infinite incremental combination of LMS filters. Interestingly, however, the performance of the NLMS filter can be matched almost exactly with a few dozens of LMS filters, thus avoiding the complexity of a division, as well as its regularization issues. The same effect can be produced using sign-error LMS filters, which are more hardware-friendly. Finally, even though incremental combinations are not parallelizable (contrary, for instance, to the APA), they can be accelerated when chip area is not an issue, by either pipelined or unrolled implementations [98].

Algorithm 2 The DR- $\{\{\text{sign-error LMS}\}^N\}^L$

$$\|\mathbf{u}_i\|^2 = \|\mathbf{u}_{i-1}\|^2 - u^2(i-M) + u^2(i) \quad \triangleright (\mathbf{1}) \times$$

$$\mathbf{w}_{0,i} = \mathbf{w}_{i-1}$$

for $\ell = 1, \dots, L$ $\triangleright$ DR incremental combination

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

$$\varepsilon_1(i) = d(i-k) - \mathbf{u}_{i-k} \mathbf{w}_{\ell-1,i} \quad \triangleright (\mathbf{M}) \times$$

$$\mathbf{w}_{\ell,i}^{(0)} = \mathbf{w}_{\ell-1,i}$$

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

$$\mathbf{w}_{\ell,i}^{(n)} = \mathbf{w}_{\ell,i}^{(n-1)} + \mu_n \mathbf{u}_{i-k}^T \text{sign}[\varepsilon_n(i)]$$

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

end

$$\mathbf{w}_{\ell,i} = \mathbf{w}_{\ell,i}^{(N)}$$

end

$$\mathbf{w}_i = \mathbf{w}_{L,i}$$

Algorithm 3 The DR- $\{\{\text{sign-error LMS}\}^N\}^L$ (optimized)

$$\mathbf{b}_{i-1} = [\mathbf{u}_{i-1}\mathbf{u}_{i-2}^T \cdots \mathbf{u}_{i-1}\mathbf{u}_{i-K}^T]^T$$

$$\mathbf{e}_{i-1} = [e(i-1) \cdots e(i-K)]^T$$

procedure UPDATE DATA

$$\|\mathbf{u}_i\|^2 = \|\mathbf{u}_{i-1}\|^2 - u^2(i-M) + u^2(i) \quad \triangleright (\mathbf{1}) \times$$

for $k = 1, \dots, K-1$

$$\mathbf{b}_i[k] = \mathbf{b}_{i-1}[k] - u(i-M)u(i-M-k) + u(i)u(i-k) \quad \triangleright (\mathbf{1}) \times$$

end

$$\mathcal{U}_i = \begin{bmatrix} \|\mathbf{u}_i\|^2 & \mathbf{b}^T \\ \mathbf{b} & \mathbf{B}_{i-1}[1:K-1] \end{bmatrix}$$

$$\mathbf{e}_i = [d(i) - \mathbf{u}_i\mathbf{w}_{i-1} \quad \mathbf{e}_{i-1}[1:K-1]]^T \quad \triangleright (M) \times$$

end

procedure COMBINATION

UPDATE DATA

$$\mathbf{w}_{0,i} = \mathbf{w}_{i-1}$$

for $n = 1, \dots, N$

$\triangleright$ DR incremental combination

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

$$\mathbf{e}_{i,1} = \mathbf{e}_i$$

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

$\triangleright \{\text{Sign-error LMS}\}^L$

$$\mathbf{w}_{\ell,i} = \mathbf{w}_{\ell-1,i} + \mu_\ell \mathbf{u}_{i-k}^T \text{sign}[\mathbf{e}_{i,\ell}[k+1]]$$

$$\mathbf{e}_{i,\ell+1} = \mathbf{e}_{i,\ell} - \mu_\ell \mathcal{U}_i[:, k+1] \text{sign}[\mathbf{e}_{i,\ell}[k+1]]$$

end

$$\mathbf{w}_{n,i} = \mathbf{w}_{L,i}$$

$$\mathbf{e}_i = \mathbf{e}_{i,L}$$

end

end

10 PERFORMANCE ANALYSIS OF PARALLEL COMBINATIONS

All models are wrong, but some are useful.

-- George Cox, "Empirical Model-Building and Response Surfaces"

Extensive performance analyses of parallel combinations have been carried out in the literature [33–35, 77, 90–93, 95, 96, 106, 108, 110, 123, 128]. These models, however, account only for parallel-independent and coefficients leakage topologies. Moreover, they were derived for specific component filters and supervising rules. In this chapter, a general steady-state and tracking analysis is obtained accounting for different topologies and supervisors (Section 10.1). This model is then used to analyze specific combinations (Section 10.2). The transient performance of the 2·LMS is also studied along with the analysis of general stochastic gradient supervising rules (Section 10.3).

10.1 Steady-state and tracking performance

In this section, the asymptotic behavior of parallel combinations of AFs with and without cyclic coefficients feedback is studied. First, the different steady-state regimes of the structure are introduced and a global analysis of the parallel combination output is developed (Sections 10.1.1 and 10.1.2). Then, general forms of the required second-order error statistics are simultaneously evaluated for stationary and nonstationary scenarios using an ECR argument similar to [18] (Section 10.1.3 and 10.1.4). Finally, a method to determine the steady-state values of the supervising parameters is presented (Section 10.1.5). Recall that even though all derivations are carried out for two component filters, the results are easily extendable to the case of N AFs by evaluating the additional statistics or using hierarchical combinations [36, 38].

Before proceeding, we introduce general forms of the component filters and supervising rule recursions to provide a flexible framework for analyses. Explicitly,

$$\mathbf{w}_{n,i} = \mathbf{w}_{n,a} + \mu_n \mathbf{u}_i^T g_n[d(i) - \mathbf{u}_i \mathbf{w}_{n,a}] \quad (10.1)$$

$$a(i) = a(i-1) + \mu_a e(i) [y_1(i) - y_2(i)] f'[a(i-1)]. \quad (10.2)$$

Notice that the recursion in (10.1) represents an AF with error nonlinearity for some possibly nonlinear positive-valued g_n . This function can be used to recover a myriad of adaptive algorithms, such as the LMS for $g_n[x] = x$; the NLMS for $g_n[x] = x/\|\mathbf{u}_i\|^2$; the LMF for $g_n[x] = x^3$; and the sign-error LMS for $g_n[x] = \text{sign}[x]$. For the sake of clarity, combinations involving RLS filters are studied separately in Sections 10.2.3 and 10.2.4 using $g_n[x] = x$ and replacing μ_n by $\mathbf{P}_{n,i}$ as in (2.6).

Also recall that (10.2) was used in Section 5.1 to describe stochastic gradient supervisors. It accounts for different supervising rules by appropriately choosing the activation functions f . For instance, the convex (5.8) and the affine (5.10) supervisors can be recovered by choosing $f(a)$ as the sigmoid from (5.7) with $f' = \eta(1 - \eta)$ and $f(a) = a$ with $f' = 1$, respectively.

10.1.1 Global steady-state analysis

In combinations of AFs, two types of asymptotic behavior can be distinguished:

Definition 10.1 (Local steady-state). A combination of AFs is said to operate at steady-state *locally* if $\mathbb{E} \tilde{\mathbf{w}}_{m,i}^T \tilde{\mathbf{w}}_{n,i} = \mathbb{E} \tilde{\mathbf{w}}_{m,i-1}^T \tilde{\mathbf{w}}_{n,i-1}$ for all m, n and as $i \rightarrow \infty$.

Definition 10.2 (Global steady-state). A combination of AFs is said to operate at steady-state *globally* if $\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 = \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2$, as $i \rightarrow \infty$.

Notice that Definition 10.1 is independent of the supervising parameter. As is usually the case, the present derivations are primarily interested in global steady-state performance measures. However, it is not always practical to analyze the overall combination directly, such as with parallel-independent structures ($L \rightarrow \infty$). Nevertheless, it is straightforward to show that, if the supervisor is at steady-state, Definition 10.1 implies Definition 10.2. The converse, in contrast, only holds in specific cases (as argued in Section 10.1.2).

Proceeding under either steady-state regimes, global figures of merit can be related to statistics of the supervising parameter and components errors. To do so, subtract the overall coefficients expression (5.5) at $i - 1$ from $\mathbf{w}_i^o$ to get

$$\mathbf{w}_i^o - \mathbf{w}_{i-1} = \eta(i-1)(\mathbf{w}_i^o - \mathbf{w}_{1,i-1}) + [1 - \eta(i-1)](\mathbf{w}_i^o - \mathbf{w}_{2,i-1}). \quad (10.3)$$

Then, multiplying (10.3) by $\mathbf{u}_i$ from the left, squaring, and taking the expected value

yields the global EMSE

$$\begin{aligned}\zeta(i) &= \mathbb{E} \eta^2(i-1) e_{a,1}^2(i) \\ &\quad + 2 \mathbb{E} \eta(i-1) [1 - \eta(i-1)] e_{a,1}(i) e_{a,2}(i) \\ &\quad + \mathbb{E} [1 - \eta(i-1)]^2 e_{a,2}^2(i).\end{aligned}\tag{10.4}$$

At this point, it is common in the combination literature to adopt some sort of supervisor separation principle to decouple the analysis of the component filters from that of the overall combination. Additionally, the variance of the supervisor is assumed to vanish at steady-state to simplify the derivations [33–35, 39, 110].

A.3 (*Steady-state supervisor separation principle*) At steady-state, $\eta(i)$ varies slowly compared to the coefficients error $\tilde{\mathbf{w}}_{n,i}$ and the *a priori* errors $e_{a,n}(i)$, so that their expected values can be separated, as in $\mathbb{E}[\eta(i)\tilde{\mathbf{w}}_{n,i}] = \mathbb{E} \eta(i) \mathbb{E} \tilde{\mathbf{w}}_{n,i}$ and $\mathbb{E}[\eta(i)e_{a,n}(i)] = \mathbb{E} \eta(i) \mathbb{E} e_{a,n}(i)$, as $i \rightarrow \infty$.¹

A.4 (*Small variance assumption*) The variance of $\eta(i)$ becomes negligible as $i \rightarrow \infty$, so that $\mathbb{E} \eta^2(i) \approx [\mathbb{E} \eta(i)]^2$.

Both assumptions are supported by simulations and have been successfully applied to the steady-state analysis of both affine [35, 39, 110] and convex [33, 34] supervisors. Moreover, notice that for supervisors as in (10.2), A.4 implies that f' and/or the difference between the output of the components vanish at steady-state. This is indeed the case for the sigmoidal activation function and sufficiently small μ_a , although it has been observed to remain accurate for a larger range of step sizes and different f [33–35, 39, 110]. When coefficients feedback is used, A.4 is further supported by the fact that it reduces the supervising parameters variance (see Section 10.3).

Hence, under A.3 and A.4, (10.4) becomes

$$\begin{aligned}\zeta(i) &= \bar{\eta}^2 \zeta_1(i) \\ &\quad + 2\bar{\eta}[1 - \bar{\eta}(i-1)] \zeta_{12}(i) \\ &\quad + [1 - \bar{\eta}]^2 \zeta_2(i), \quad \text{as } i \rightarrow \infty,\end{aligned}\tag{10.5}$$

where $\bar{\eta} = \lim_{i \rightarrow \infty} \mathbb{E} \eta(i)$ is the steady-state value of the mean supervising parameter.

A similar relation can be obtained for the MSD by subtracting (5.5) from $\mathbf{w}_i^o$ and

¹This assumption can also be stated in terms of the conditional independence of the variables given $\eta_n(i)$ [128]. Assumption A.10 in the transient analysis in Section 10.3 is stated in this way.

Figure 42: Component filters behavior under cyclic coefficients feedback (2-LMS). **White input data:** $M = 7$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.005$, and $\mu_2 = 0.05$. $L = 1$: $\mu_a = 200$ (convex supervisor); $L = 50$: $\mu_a = 300$ (convex supervisor); $L \rightarrow \infty$: $\mu_a = 200$ (convex supervisor).

taking the expected value of its squared Euclidian norm under A.3 and A.4:

$$\begin{aligned}
 \mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 &= \bar{\eta}^2 \mathbb{E} \|\tilde{\mathbf{w}}_{1,i}\|^2 \\
 &\quad + 2\bar{\eta}[1 - \bar{\eta}] \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \tilde{\mathbf{w}}_{2,i} \\
 &\quad + [1 - \bar{\eta}]^2 \mathbb{E} \|\tilde{\mathbf{w}}_{2,i}\|^2, \quad \text{as } i \rightarrow \infty.
 \end{aligned} \tag{10.6}$$

10.1.2 Cyclic feedback and steady-state

For relations (10.5) and (10.6) to hold, the quantities involved must have steady-state values, i.e., all error and supervisor statistics involved need to converge to fixed values as $i \rightarrow \infty$. With cyclic coefficients feedback, however, this does not occur even if the combination is stable (does not diverge). Indeed, when $1 < L < \infty$, the topology varies with time, alternating between a parallel-independent and a coefficients feedback structure. Hence, the combination does not reach a stationary form as $i \rightarrow \infty$ and constantly operates in transient regime (Figure 42b). In contrast, the limiting cases $L \rightarrow \infty$ and $L = 1$ have static topologies, implying the components converge as $i \rightarrow \infty$ (Figures 42a and 42c).

To account for the $1 < L < \infty$ situation without resorting to a full transient analysis (as in Section 10.3), notice that the limiting conditions analyses provide steady-state performance bounds for the general cyclic case. Also, recall that the global coefficients are only fed back cyclically so as not to disturb the supervisor adaptation. Hence, only the asymptotic performance for small L are of interest in practice, so that the following approximation holds:

A.5 (Small cycle period) For small cycle period, the steady-state performance of the

Figure 43: Steady-state of parallel combinations with cyclic coefficients feedback for different cycle periods. (a) **LMS + LMS (white input data)**: $M = 5$, $\sigma_u^2 = 1/M$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.07$, $\mu_2 = 0.01$, $\mu_\eta = 0.7$, $\eta_{\min} = -0.15$ (affine supervisor); (b) **LMS + RLS (correlated input data)**: $M = 5$, $\sigma_u^2 = 1/M$, $\sigma_v^2 = 10^{-2}$, $\gamma = 0.8$, $\mathbf{Q} = 2 \cdot 10^{-6} \cdot \mathbf{R}_u^{-1}$, $\mu_1 = 0.04$, $\lambda_{\text{RLS}} = 0.92$, $\epsilon_{\text{RLS}} = 10^{-6}$, and $\mu_a = 100$ (convex supervisor).

overall combination is the same as if $L = 1$.

This assumption can be motivated by observing that at steady-state and for reduced cycle periods, the component filters are unable to significantly alter their coefficients between feedbacks. The same goes for the supervising parameter. Note that the converse also holds: as the cycle period becomes larger, the components become unable to fully take advantage of the global coefficients and the combination starts behaving as if the AFs were independent. Finally, simulations suggest that A.5 is a valid approximation for a significant range of L , as illustrated in Figures 43a and 43b, where the dashed lines indicate a 1 dB gap in the steady-state EMSE. Notice that for cycle periods lower than 30, the error incurred from the use of A.5 is less than 1 dB. In view of these arguments, the following analyses are derived for combinations where $L \rightarrow \infty$ and $L = 1$.

10.1.3 Variance and covariance relations for $L \rightarrow \infty$

Parallel-independent combinations ($L \rightarrow \infty$) involving components such as LMS, RLS, and LMF filters have been analyzed in the literature for both affine and convex supervising rules [33–36, 38–40, 123, 128, 137]. Nevertheless, the following derivations provide a unified framework for the analysis of this structure and account for a wider variety of components by means of the general AF model in (10.1).

From the feedback relation (5.13), $L \rightarrow \infty \Rightarrow \mathbf{w}_{n,a} = \mathbf{w}_{n,i-1}$, for all i , so that (10.1)

simplifies to

$$\mathbf{w}_{n,i} = \mathbf{w}_{n,i-1} + \mu_n \mathbf{u}_i^T g_n[e_n(i)], \quad (10.7)$$

where $e_n(i) = d(i) - \mathbf{u}_i \mathbf{w}_{n,i-1}$ is the n -th component output estimation error. Subtracting (10.7) from $\mathbf{w}_i^o$ and noticing that the nonstationarity model (1.3) implies that $\mathbf{w}_i^o - \mathbf{w}_{n,i-1} = \tilde{\mathbf{w}}_{n,i-1} + \mathbf{q}_i$ gives

$$\tilde{\mathbf{w}}_{n,i} = \tilde{\mathbf{w}}_{n,i-1} + \mathbf{q}_i - \mu_n \mathbf{u}_i^T g_n, \quad (10.8)$$

in which the argument of g_n was omitted for clarity. Taking the expected value of the squared Euclidian norm of (10.8) yields

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_{n,i}\|^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{n,i-1}\|^2 + \text{Tr}(\mathbf{Q}) \\ &\quad - 2\mu_n \mathbb{E} e_{a,n}(i) g_n + \mu_n^2 \mathbb{E} \|\mathbf{u}_i\|^2 g_n^2, \end{aligned} \quad (10.9)$$

where $\mathbb{E} \|\mathbf{q}_i\|^2 = \text{Tr}(\mathbf{Q})$ and all terms depending linearly on $\mathbf{q}_i$ vanish due to A.2. Similarly, cross-multiplying (10.8) for $n = 1, 2$ and taking the expectation gives

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \tilde{\mathbf{w}}_{2,i} &= \mathbb{E} \tilde{\mathbf{w}}_{1,i-1}^T \tilde{\mathbf{w}}_{2,i-1} + \text{Tr}(\mathbf{Q}) \\ &\quad - \mu_1 \mathbb{E} e_{a,2}(i) g_1 - \mu_2 \mathbb{E} e_{a,1}(i) g_2 \\ &\quad + \mu_1 \mu_2 \mathbb{E} \|\mathbf{u}_i\|^2 g_1 g_2. \end{aligned} \quad (10.10)$$

To proceed, notice that, since there is no feedback, all of the above recursions depend uniquely on local quantities (i.e., $\mathbf{w}_{n,i-1}$). In other words, the components are uncoupled and operate as if there were no combination. Local steady-state is therefore a necessary condition for global steady-state to be achieved for an arbitrary η . This implies that Def. 10.1 and 10.2 are equivalent. Hence, variance and covariance relations can be derived for each component filter individually using only local steady-state assumptions, since they guarantee that the global steady-state analysis obtained by merging the components results is valid. Explicitly, under Def. 10.1, the relations in (10.5), (10.9), and (10.10) become, as $i \rightarrow \infty$,

$$\zeta = \bar{\eta}^2 \zeta_1 + 2\bar{\eta}[1 - \bar{\eta}] \zeta_{12} + [1 - \bar{\eta}]^2 \zeta_2 \quad (10.11a)$$

$$\mu_n \mathbb{E} \|\mathbf{u}_i\|^2 g_n^2[e_n(i)] + \mu_n^{-1} \text{Tr}(\mathbf{Q}) = 2 \mathbb{E} e_{a,n}(i) g_n[e_n(i)] \quad (10.11b)$$

$$\mu_1 \mu_2 \mathbb{E} \|\mathbf{u}_i\|^2 g_1[e_1(i)] g_2[e_2(i)] + \text{Tr}(\mathbf{Q}) = \mu_1 \mathbb{E} e_{a,2}(i) g_1[e_1(i)] + \mu_2 \mathbb{E} e_{a,1}(i) g_2[e_2(i)]. \quad (10.11c)$$

Notice that (10.11b) is equivalent to the variance relation deduced in [18, Thm. 6.4.1], which is used to derive the steady-state performance of several adaptive algorithms. In

fact, quantities ζ_1 and ζ_2 could be replaced by any of the well-known formulae for the mean-square behavior of AFs [18, 19]. The covariance relation (10.11c), on the other hand, introduces a unified framework upon which the cross-EMSE of several AFs can be evaluated. In summary, (10.11) allows the steady-state global EMSE to be evaluated for combinations without feedback involving a myriad of component filters.

10.1.4 Variance and covariance relations for $L = 1$

Although the previous section provided a new view on established results, the following analysis applies to the novel coefficients feedback topology. Indeed, for $L = 1$, the feedback relation (5.13) reduces to $\mathbf{w}_{n,a} = \mathbf{w}_{i-1}$ for all i , i.e., the global coefficients are fed back to the components at every iteration. Explicitly, (10.1) becomes

$$\mathbf{w}_{n,i} = \mathbf{w}_{i-1} + \mu_n \mathbf{u}_i^T g_n[e(i)]. \quad (10.12)$$

Recall that $e(i) = d(i) - \mathbf{u}_i \mathbf{w}_{i-1}$ is the global output estimation error.

As in Section 10.1.3, (10.12) is subtracted from $\mathbf{w}_i^o$ to yield

$$\tilde{\mathbf{w}}_{n,i} = \tilde{\mathbf{w}}_{i-1} + \mathbf{q}_i - \mu_n \mathbf{u}_i^T g_n[e(i)], \quad (10.13)$$

where the global coefficients error relation $\mathbf{w}_i^o - \mathbf{w}_{i-1} = \tilde{\mathbf{w}}_{i-1} + \mathbf{q}_i$ was deduced from (1.3). Finally, recursions for the variance and covariance of $\tilde{\mathbf{w}}_{n,i}$ can be expressed as

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_{n,i}\|^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \text{Tr}(\mathbf{Q}) \\ &\quad - 2\mu_n \mathbb{E} e_a(i) g_n[e(i)] \\ &\quad + \mu_n^2 \mathbb{E} \|\mathbf{u}_i\|^2 g_n^2[e(i)] \end{aligned} \quad (10.14a)$$

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \tilde{\mathbf{w}}_{2,i} &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \text{Tr}(\mathbf{Q}) \\ &\quad - \mu_1 \mathbb{E} e_a(i) g_1[e(i)] \\ &\quad - \mu_2 \mathbb{E} e_a(i) g_2[e(i)] \\ &\quad + \mu_1 \mu_2 \mathbb{E} \|\mathbf{u}_i\|^2 g_1[e(i)] g_2[e(i)]. \end{aligned} \quad (10.14b)$$

Once again, all terms depending linearly on $\mathbf{q}_i$ vanished due to A.2.

Notice that the recursions in (10.14) differ from their counterparts (10.9) and (10.10) in that they depend on $\tilde{\mathbf{w}}_{i-1}$, which is, in turn, a function of the $\tilde{\mathbf{w}}_{n,i-1}$ and $\eta(i-1)$. It is clear, then, that (10.14a) and (10.14b) are effectively coupled and cannot be solved for each component individually as in Section 10.1.3. However, all component filters update the global coefficients $\mathbf{w}_{i-1}$, which indicates that global steady-state implies local

steady-state (although the asymptotic behavior in this case is different from the one in Section 10.1.3). Once again, Def. 10.1 and 10.2 are therefore equivalent and the global performance of the combination can be analyzed directly.

Proceeding, substitute (10.14) in the MSD expression (10.6) to get

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \text{Tr}(\mathbf{Q}) \\ &\quad - 2\bar{\eta}\mu_1 \mathbb{E} e_a(i)g_1 - 2(1-\bar{\eta})\mu_2 \mathbb{E} e_a(i)g_2 \\ &\quad + \bar{\eta}\mu_1 \mathbb{E} \left\{ \|\mathbf{u}_i\|^2 g_1 [\bar{\eta}\mu_1 g_1 + (1-\bar{\eta})\mu_2 g_2] \right\} \\ &\quad + (1-\bar{\eta})\mu_2 \mathbb{E} \left\{ \|\mathbf{u}_i\|^2 g_2 [\bar{\eta}\mu_1 g_1 + (1-\bar{\eta})\mu_2 g_2] \right\}, \end{aligned}$$

which at global steady-state yields the variance relation for combinations with coefficients feedback. Explicitly, as $i \rightarrow \infty$,

$$\begin{aligned} &\bar{\eta}\mu_1 \mathbb{E} \left\{ \|\mathbf{u}_i\|^2 g_1 [\bar{\eta}\mu_1 g_1 + (1-\bar{\eta})\mu_2 g_2] \right\} \\ &\quad + (1-\bar{\eta})\mu_2 \mathbb{E} \left\{ \|\mathbf{u}_i\|^2 g_2 [\bar{\eta}\mu_1 g_1 + (1-\bar{\eta})\mu_2 g_2] \right\} + \text{Tr}(\mathbf{Q}) = \\ &\quad 2\bar{\eta}\mu_1 \mathbb{E} e_a(i)g_1 + 2(1-\bar{\eta})\mu_2 \mathbb{E} e_a(i)g_2. \quad (10.15) \end{aligned}$$

Due to the feedback, a single variance relation for the global error is obtained, in contrast to the variance and covariance relations required for each component filters in parallel-independent combinations. In fact, the combination can now be interpreted as a single AF whose optimization problem and solution is determined by the components and the supervising parameter η . Hence, (10.15) is effectively a generalization of the variance relation for stand-alone AFs [18, Thm. 6.4.1]. The introduction of coefficients feedback has, therefore, allowed for an explicit interpretation of energy transfer between global *a priori* and *a posteriori* errors.

Indeed, one can obtain, through similar derivations, a global ECR for parallel combinations with coefficients feedback:

$$\|\mathbf{w}_i^o - \mathbf{w}_i\|^2 + \frac{e_a^2(i)}{\|\mathbf{u}_i\|^2} = \|\mathbf{w}_i^o - \mathbf{w}_{i-1}\|^2 + \frac{e_p^2(i)}{\|\mathbf{u}_i\|^2}. \quad (10.16)$$

Note that the global error quantities are now defined as $e_a(i) = \eta e_{a,1}(i) + (1-\eta)e_{a,2}(i)$ and $e_p(i) = \eta e_{p,1}(i) + (1-\eta)e_{p,2}(i)$ and that $\mathbf{w}_i = \eta \mathbf{w}_{1,i} + (1-\eta)\mathbf{w}_{2,i}$ are the global coefficients.

10.1.5 Supervisor analysis

The steady-state and tracking performance of combinations with and without feedback can be evaluated for arbitrary values of the supervising parameter using (10.11) and (10.15). Suffices, then, to determine its steady-state value. To do so, consider supervising rules of the form (10.2). Then, derive an approximate recursion for $E a(i)$ and its stationary point to evaluate $\bar{\eta}$. Notice that similar results exist for parallel-independent combinations using convex and affine supervisors [33, 35]. However, the expressions obtained here are valid for a larger class of activation functions.

Start by obtaining from (1.1) and (5.5) the relations

$$\begin{aligned} y_1(i) - y_2(i) &= e_{a,2}(i) - e_{a,1}(i) \\ e(i) &= \eta(i-1) e_{a,1}(i) + [1 - \eta(i-1)] e_{a,2}(i) + v(i), \end{aligned}$$

which applied to the generic supervisor expression (10.2) yields

$$\begin{aligned} a(i) &= a(i-1) + \mu_a \left\{ [1 - f(a)] f'(a) e_{a,2}^2(i) \right. \\ &\quad + [2f(a) - 1] f'(a) e_{a,1}(i) e_{a,2}(i) - f(a) f'(a) e_{a,1}^2(i) \\ &\quad \left. + f'(a) [e_{a,2}(i) - e_{a,1}(i)] v(i) \right\}, \end{aligned} \quad (10.17)$$

where all $f(\cdot)$ are evaluated at $a(i-1)$. The dependence on the activation function is stressed in (10.17) by writing f instead of η . Then, proceeding similarly to [128], all terms dependent on f in (10.17) are approximated by their first-order Taylor expansions around $\bar{a}(i-1) = E a(i-1)$. Explicitly, (10.17) becomes

$$\begin{aligned} a(i) &\approx a(i-1) + \mu_a \left\{ \mathcal{A}_2 e_{a,2}^2(i) + \mathcal{A}_{12} e_{a,1}(i) e_{a,2}(i) \right. \\ &\quad \left. - \mathcal{A}_1 e_{a,1}^2(i) + \mathcal{A}_v [e_{a,2}(i) - e_{a,1}(i)] v(i) \right\} \end{aligned} \quad (10.18)$$

$$\begin{aligned} \mathcal{A}_1 &\approx \bar{f} \bar{f}' + [(\bar{f}')^2 + \bar{f} \bar{f}''] (a - \bar{a}) \\ \mathcal{A}_2 &\approx (1 - \bar{f}) \bar{f}' + [-(\bar{f}')^2 + (1 - \bar{f}) \bar{f}''] (a - \bar{a}) \\ \mathcal{A}_{12} &\approx (2\bar{f} - 1) \bar{f}' + [2(\bar{f}')^2 + (2\bar{f} - 1) \bar{f}''] (a - \bar{a}) \\ \mathcal{A}_v &\approx \bar{f}' + \bar{f}'' (a - \bar{a}) \end{aligned}$$

where f'' is the second derivative of f and for any function h : $\bar{h} = h[\bar{a}(i-1)]$. Notice that the $\mathcal{A}$ are functions only of the auxiliary parameter a . Moreover, the expected value of their second terms vanishes since $E(a - \bar{a}) = E a - \bar{a} = \bar{a} - \bar{a} = 0$. Then, using the

supervisor separation principle A.3 and the fact that the data model assumes that $v(i)$ is zero mean and independent of $u(j)$ for all i, j , the expected value of (10.18) becomes

$$a(i) = a(i-1) + \mu_a [(1 - \bar{f})e_{a,2}^2(i) + (2\bar{f} - 1)e_{a,1}(i)e_{a,2}(i) - \bar{f}e_{a,1}^2(i)] \bar{f}' \quad (10.19)$$

This expression reaches a stationary point ($\bar{a}(i) = \bar{a}(i-1) = \bar{a}$) if and only if the following relation holds

$$\underbrace{[\Delta\zeta_2 - (\Delta\zeta_1 + \Delta\zeta_2)f(\bar{a})]}_{\mathcal{P}} f'(\bar{a}) = 0, \text{ as } i \rightarrow \infty, \quad (10.20)$$

where $\Delta\zeta_n = \zeta_n - \zeta_{12}$. Given that f' is infinitely supported by assumption [see (10.2)], (10.20) can only be met when $\mathcal{P}$ vanishes, i.e., for

$$\bar{\eta} = f(\bar{a}) = \frac{\Delta\zeta_2}{\Delta\zeta_1 + \Delta\zeta_2}. \quad (10.21)$$

Although (10.21) is independent of the topology [as long as the supervising rule follows the form (10.2)], it requires the knowledge of the $\Delta\zeta_n$ values at steady-state. This not an issue in parallel-independent structures, since they are readily available from (10.11). In contrast, expressing these quantities in combinations with coefficients feedback is intricate, as the local (10.14) and global (10.15) coefficients error recursions are coupled. To address this matter, notice that (10.21) is the supervising parameter value which minimizes the global MSE at steady-state [33,35]. In other words, it is the optimal supervising parameter in the global MSE sense. Hence, $\bar{\eta}$ can be determined, in the coefficients feedback case, using

A.6 (*Steady-state mean supervisor*) As in parallel-independent combinations, the steady-state value of the mean supervising parameter for combinations with coefficients feedback ($L = 1$) is the optimal one in the global MSE sense, i.e., it is the value of $\bar{\eta}$ which minimizes ζ .

Similar assumptions have been used in the analysis of parallel-independent combinations [33], which sometimes take the more stringent assumption that the mean value of the supervising parameter is optimal for all i [108]. What is more, $\bar{\eta}$ could be evaluated using the transient analysis developed in Section 10.3. This approach, however, is more demanding and yields similar results to using A.6.

10.2 Steady-state and tracking performance of specific combinations

In this section, the results from Section 10.1 are used to analyze the performance of combinations involving different component filters. Hence, new assumptions are required to evaluate the expected values in (10.11) and (10.15), specially since the assumptions adopted so far were mostly related to the supervisor analysis (A.3, A.4, and A.5). Notice that the approximations used in the sequel are the same as to those commonly used in the literature on stand-alone AFs [18, 19]. Additionally, the derivations for the steady-state values of the local EMSE ζ_n in combinations without feedback are omitted for brevity. If necessary, they can be found in [18].

10.2.1 2 · LMS

Steady-state results for the parallel-independent combination of two LMS filters are available in the literature [33–35], although not necessarily under the energy conservation argument from Section 10.1. They are therefore presented here for completeness. On the other hand, the results for coefficients feedback topologies is novel. To get a 2 · LMS combination, take

$$g_n[x(i)] = x(i), \quad \text{for } n = 1, 2. \quad (10.22)$$

10.2.1.1 Without feedback

From the data model (1.1), notice that $e_n(i) = e_{a,n}(i) + v(i)$. Then, using (10.22), the covariance relation in (10.11c) becomes

$$\begin{aligned} \mu_1 \mu_2 \mathbb{E} \|\mathbf{u}_i\|^2 [e_{a,1}(i) + v(i)][e_{a,2}(i) + v(i)] + \text{Tr}(\mathbf{Q}) = \\ \mu_1 \mathbb{E} e_{a,2}(i) [e_{a,1}(i) + v(i)] \\ + \mu_2 \mathbb{E} e_{a,1}(i) [e_{a,2}(i) + v(i)], \quad \text{as } i \rightarrow \infty. \end{aligned} \quad (10.23)$$

In order to proceed with the derivations, take the commonly adopted assumption:

A.7 (*Data separation principle*) $\|\mathbf{u}_i\|^2$ is independent of $e_{a,n}(i)$ —and consequently $e_a(i)$ —as $i \rightarrow \infty$.

The same assumption is used in the steady-state analysis of stand-alone LMS filters [18]. Given the expected value of all terms linearly dependent on $v(i)$ will vanish (A.1), the

steady-state values of the EMSE and cross-EMSE for this combination are expressed as

$$\zeta_n^{\{2\text{-LMS}\}} = \frac{\mu_n \sigma_v^2 \text{Tr}(\mathbf{R}_u) + \mu_n^{-1} \text{Tr}(\mathbf{Q})}{2 - \mu_n \text{Tr}(\mathbf{R}_u)} \quad (10.24a)$$

$$\zeta_{12}^{\{2\text{-LMS}\}} = \frac{\mu_1 \mu_2 \sigma_v^2 \text{Tr}(\mathbf{R}_u) + \text{Tr}(\mathbf{Q})}{\mu_1 + \mu_2 - \mu_1 \mu_2 \text{Tr}(\mathbf{R}_u)}. \quad (10.24b)$$

Using the combination expression (10.11a) and evaluating the supervising parameter through (10.21), the well-known closed-form expression for the steady-state of parallel-independent combinations of LMS filters is recovered [33–35].

10.2.1.2 With feedback

For g_n as in (10.22) and noticing that the global output estimation error can be expressed as $e(i) = e_a(i) + v(i)$, the global variance relation (10.15) becomes

$$\begin{aligned} [\bar{\eta}\mu_1 + (1 - \bar{\eta})\mu_2]^2 \mathbb{E} \|\mathbf{u}_i\|^2 [e_a(i) + v(i)]^2 + \text{Tr}(\mathbf{Q}) = \\ 2[\bar{\eta}\mu_1 + (1 - \bar{\eta})\mu_2] \mathbb{E} e_a(i) [e_a(i) + v(i)], \quad \text{as } i \rightarrow \infty. \end{aligned}$$

Again, a closed-form relation is obtained using A.7. Explicitly,

$$\zeta^{f\{2\text{-LMS}\}} = \frac{\bar{\mu} \text{Tr}(\mathbf{R}_u) \sigma_v^2 + \bar{\mu}^{-1} \text{Tr}(\mathbf{Q})}{2 - \bar{\mu} \text{Tr}(\mathbf{R}_u)}, \quad (10.25)$$

where $\bar{\mu} = \bar{\eta}\mu_1 + (1 - \bar{\eta})\mu_2$ is the steady-state *equivalent step size* of the combination.

The stationary steady-state expressions (10.24) and (10.25) are compared to simulations in Figures 44 and 45. Notice that they are indeed accurate for $L = 1$ and $L \rightarrow \infty$. For $0 < L < \infty$, they hold as performance bounds and can be used to approximate the combination behavior.

Interestingly, (10.25) is equivalent to the expression for the steady-state of a stand-alone LMS filter with step size $\bar{\mu}$ [18, Lemma 7.5.1], as expected from the observations in Chapter 8. Its performance is therefore better than that of the parallel-independent combination, specially in non-stationary scenarios. Indeed, as argued in [129], a parallel-independent combination of LMS filters can only match the tracking performance of a stand-alone LMS filter with optimal step size μ^o [18, Lemma 7.5.1] if at least one of its component has step size μ^o . In contrast, combinations with coefficients feedback are able to adjust $\bar{\mu} = \mu^o$ for any selection of component step sizes². Moreover, since supervising rules of the form (10.2) minimize the global MSE, the supervising parameter will indeed

²This is actually the case for affine supervisors. Convex supervisors, on the other hand, require that $\mu_1 \leq \mu^o \leq \mu_2$.

Figure 44: Steady-state analysis of a $2 \cdot \text{LMS}$ combination. **Correlated input data:** $M = 10$, $\sigma_u^2 = 1$, $\gamma = 0.8$, $\sigma_v^2 = 10^{-3}$, $\mu_1 = 0.01$, and $\mu_2 = 0.002$. $\mathbf{L} \rightarrow \infty$: $\mu_a = 1000$ (convex supervisor); $\mathbf{L} = \mathbf{1}$: $\mu_\eta = 0.002$, $\beta = 0.9$, $\epsilon = 10^{-3}$, and $\eta_{\min} = -0.2$ (normalized affine supervisor); $\mathbf{1} < \mathbf{L} < \infty$: $L = 3$, $\mu_\eta = 0.7$, $\beta = 0.9$, $\epsilon = 10^{-1}$, and $\eta_{\min} = -0.2$ (normalized affine supervisor).

Figure 45: Tracking analysis of a $2 \cdot \text{LMS}$ combination [non-stationary $\mathbf{w}^o$: $\sigma_q^2 = 10^{-4}$ for $i \in [1, 3000)$, $\sigma_q^2 \rightarrow 10^{-5}$ for $i \in [3000, 6000)$ and $\sigma_q^2 \rightarrow 10^{-6}$ for $i \in [6000, 10000)$]. **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $Q = \sigma_q^2 \mathbf{I}$, $\mu_1 = 0.08$, and $\mu_2 = 0.005$. $\mathbf{L} \rightarrow \infty$: $\mu_\eta = 0.3$, $\beta = 0.9$, and $\epsilon = 10^{-2}$ (normalized convex supervisor); $\mathbf{1} < \mathbf{L} < \infty$: $L = 10$, $\mu_\eta = 0.3$, $\beta = 0.9$, and $\epsilon = 10^{-2}$ (normalized convex supervisor).

track μ^o (as illustrated in Figure 31 from Chapter 8).

10.2.2 LMS + LMF

Chapter 8 showed that the unsupervised (fixed η) LMS + LMF combination with coefficients feedback is equivalent to the LMMN algorithm from [28]. Hence, the following results can also be applied for that algorithm. To analyze this combination, define the

error nonlinearities as

$$g_1[x(i)] = x(i) \quad \text{and} \quad g_2[x(i)] = x^3(i). \quad (10.26)$$

10.2.2.1 Without feedback

Substituting the g_n in (10.26) in the covariance relation (10.11c) yields

$$\begin{aligned} \mu_1 \mu_2 \mathbb{E} \|\mathbf{u}_i\|^2 [e_{a,1}(i) + v(i)][e_{a,2}(i) + v(i)]^3 + \text{Tr}(\mathbf{Q}) = \\ \mu_1 \mathbb{E} e_{a,2}(i)[e_{a,1}(i) + v(i)] \\ + \mu_2 \mathbb{E} e_{a,1}(i)[e_{a,2}(i) + v(i)]^3, \quad \text{as } i \rightarrow \infty. \end{aligned} \quad (10.27)$$

To render the derivations more tractable, assume [18]

A.8 (*Higher-order moments approximation*) At steady-state, the $e_{a,n}$ are small enough so that their higher-order terms are negligible. Hence, $\mathbb{E} e_{a,1}^k(i) e_{a,2}^\ell(i) \approx 0$ for $k + \ell \geq 3$.

Then, under the supervisor separation principle A.7 and taking the component filters to be stable and, therefore, asymptotically unbiased ($\mathbb{E} e_{a,n}(i) \rightarrow 0$ as $i \rightarrow \infty$) [18] one finds

$$\begin{aligned} \mathbb{E} \|\mathbf{u}_i\|^2 [e_{a,1}(i) + v(i)][e_{a,2}(i) + v(i)]^3 &\approx \text{Tr}(\mathbf{R}_u) [3\zeta_{12}\sigma_v^2 + 3\sigma_v^2\zeta_2 + \xi_v^4] \\ \mathbb{E} e_{a,1}(i)[e_{a,2}(i) + v(i)]^3 &\approx 3\sigma_v^2\zeta_{12}, \end{aligned} \quad (10.28)$$

where $\xi_v^4 = \mathbb{E} v^4(i)$. Using these results, (10.27) can be rearranged to read

$$\zeta_{12}^{\{\text{LMS+LMF}\}} = \frac{\mu_1 \mu_2 (3\sigma_v^2 \zeta_2 + \xi_v^4) \text{Tr}(\mathbf{R}_u) + \text{Tr}(\mathbf{Q})}{\mu_1 + 3\mu_2 \sigma_v^2 - 3\mu_1 \mu_2 \sigma_v^2 \text{Tr}(\mathbf{R}_u)}, \quad (10.29)$$

where

$$\zeta_2^{\{\text{LMS+LMF}\}} = \frac{\mu_2 \xi_v^6 \text{Tr}(\mathbf{R}_u) + \mu_2^{-1} \text{Tr}(\mathbf{Q})}{6\sigma_v^2 - 15\mu_2 \xi_v^4 \text{Tr}(\mathbf{R}_u)} \quad (10.30)$$

with $\xi_v^6 = \mathbb{E} v^6(i)$ [18, Lemma 7.8.1]. Together with the LMS filter EMSE expression (10.24a) and the supervising parameter from (10.21), these formulae provide all the necessary quantities to evaluate the steady-state global EMSE using (10.5). These results match those in [90], although they arise from the more general framework in Section 10.1.

Figure 46: Steady-state analysis of an LMS + LMF combination. **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10$ (uniform), $\mu_{\text{LMS}} = 0.005$, and $\mu_{\text{LMF}} = 1.5 \cdot 10^{-5}$. $L \rightarrow \infty$: $\mu_a = 0.5$, $\beta = 0.9$, and $\epsilon = 1$ (normalized convex supervisor); $1 < L < \infty$: $L = 500$, $\mu_a = 0.35$, $\beta = 0.8$, and $\epsilon = 1$ (normalized convex supervisor).

10.2.2.2 With feedback

For g_n as in (10.26), the global variance relation (10.15) yields

$$\begin{aligned} \mathbb{E} \|\mathbf{u}_i\|^2 \left\{ \bar{\eta} \mu_1 [e_a(i) + v(i)] + (1 - \bar{\eta}) \mu_2 [e_a(i) + v(i)]^3 \right\}^2 + \text{Tr}(\mathbf{Q}) = \\ 2\bar{\eta} \mu_1 \mathbb{E} e_a(i) [e_a(i) + v(i)] \\ + 2(1 - \bar{\eta}) \mu_2 \mathbb{E} e_a(i) [e_a(i) + v(i)]^3, \quad \text{as } i \rightarrow \infty. \end{aligned}$$

Using A.7 and A.8 to obtain $\mathbb{E}[e_a(i) + v(i)]^4 \approx 6\sigma_v^2\zeta + \xi_v^4$ in addition to the relations in (10.28), the global EMSE at steady-state of the LMS + LMF combination with coefficients feedback can be estimated using

$$\zeta^{\{ \text{LMS+LMF} \}} = \frac{\mathcal{D}}{\mathcal{N}} \quad (10.31)$$

$$\begin{aligned} \mathcal{D} &= \bar{\eta} \mu_1 \text{Tr}(\mathbf{R}_u) [\bar{\eta} \mu_1 \sigma_v^2 + (1 - \bar{\eta}) \mu_2 \xi_v^4] \\ &\quad + (1 - \bar{\eta}) \mu_2 \text{Tr}(\mathbf{R}_u) [\bar{\eta} \mu_1 \xi_v^4 + (1 - \bar{\eta}) \mu_2 \xi_v^6] \\ &\quad + \text{Tr}(\mathbf{Q}) \\ \mathcal{N} &= 2[\bar{\eta} \mu_1 + 3(1 - \bar{\eta}) \mu_2 \sigma_v^2] \\ &\quad - \bar{\eta} \mu_1 \text{Tr}(\mathbf{R}_u) [\bar{\eta} \mu_1 + 6(1 - \bar{\eta}) \mu_2 \sigma_v^2] \\ &\quad - (1 - \bar{\eta}) \mu_2 \text{Tr}(\mathbf{R}_u) [6\bar{\eta} \mu_1 \sigma_v^2 + 15(1 - \bar{\eta}) \mu_2 \xi_v^4] \end{aligned}$$

The steady-state models from this section are validated in Figures 46 and 47. To motivate the use of the LMF algorithm, the simulations use a uniformly distribution noise.

Figure 47: Tracking analysis of the LMMN using an LMS + LMF combination (non-stationary $\mathbf{w}^o$: $Q = 10^{-4}\mathbf{I}$). **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10$ (uniform), $\mu_{\text{LMS}} = 0.002$, and $\mu_{\text{LMF}} = 10^{-4}$; $\mathbf{L} \rightarrow \infty$: $\mu_a = 7$ (convex supervisor); $\mathbf{L} = \mathbf{1}$ (LMMN): $\eta(i) = \eta^o = -1.34$ for all i .

In Figure 46, a combination using a large feedback cycle period is compared to a parallel-independent combination. Although the former displays an improved convergence behavior, they both achieve the same steady-state performance. Figure 47 shows a tracking analysis of the LMMN using an LMS + LMF combination with coefficients feedback. The optimal supervisor was evaluated from (10.31) and used as the mixing parameter δ in the LMMN recursion (2.8). Notice that its value is negative and that the combination exploits the cross-correlation between the AFs to yield almost a 1 dB gain over the best component.

10.2.3 2 · RLS

Recall from the beginning of Section 10.1 that, as it stands, the error nonlinearity AF recursion in (10.1) does not account for RLS filters. To analyze combinations involving this algorithm, replace the scalar step size μ_n by $\mathbf{P}_{n,i}$ defined in (2.6) and take $g_n[x(i)] = x(i)$ in (10.1) to get

$$\mathbf{w}_{n,i} = \mathbf{w}_{n,a} + \mathbf{P}_{n,i} \mathbf{u}_i^T [d(i) - \mathbf{u}_i \mathbf{w}_{n,a}]. \quad (10.32)$$

In contrast to Sections 10.1.3 and 10.1.4, analyses involving AF of the form (10.32) require the use of weighted variance and covariance relations. To do so, define the weighted norm of a vector $\mathbf{x}$ as $\|\mathbf{x}\|_{\mathbf{\Gamma}}^2 = \mathbf{x}^T \mathbf{\Gamma} \mathbf{x}$, where $\mathbf{\Gamma}$ is a symmetric positive-definite matrix.

10.2.3.1 Without feedback

Subtracting (10.32) from $\mathbf{w}_i^o$ and using $\mathbf{w}_{n,a} = \mathbf{w}_{n,i-1}$ yields

$$\tilde{\mathbf{w}}_{n,i} = \tilde{\mathbf{w}}_{n,i-1} + \mathbf{q}_i - \mathbf{P}_{n,i} \mathbf{u}_i^T e_n(i). \quad (10.33)$$

Taking the expected value of the product and cross-product of (10.33) weighted by some symmetric positive-definite matrix $\mathbf{\Gamma}$ then yields

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_{n,i}\|_{\mathbf{\Gamma}}^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{n,i-1}\|_{\mathbf{\Gamma}}^2 + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - 2 \mathbb{E} \tilde{\mathbf{w}}_{n,i-1}^T \mathbf{\Gamma} \mathbf{P}_{n,i} \mathbf{u}_i^T e_n(i) \\ &\quad + \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{P}_{n,i}^T \mathbf{\Gamma} \mathbf{P}_{n,i}}^2 e_n^2(i) \end{aligned} \quad (10.34a)$$

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{2,i} &= \mathbb{E} \tilde{\mathbf{w}}_{1,i-1}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{2,i-1} + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - \mathbb{E} \mathbf{u}_i \mathbf{P}_{1,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{2,i-1} e_1(i) \\ &\quad - \mathbb{E} \mathbf{u}_i \mathbf{P}_{2,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{1,i-1} e_2(i) \\ &\quad + \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{P}_{1,i}^T \mathbf{\Gamma} \mathbf{P}_{2,i}} e_1(i) e_2(i). \end{aligned} \quad (10.34b)$$

To proceed, the following approximation is used as in the analysis of stand-alone RLS filters [18]

$$\mathbf{A.9} \text{ (Steady-state } \mathbf{P}_{n,i}) \quad \mathbb{E} \mathbf{P}_{n,i} \approx [\mathbb{E} \mathbf{P}_{n,i}^{-1}]^{-1} = (1 - \lambda_n) \mathbf{R}_u^{-1}, \text{ as } i \rightarrow \infty.$$

Moreover, notice that the local and global steady-state definitions Def. 10.1 and Def. 10.2 can be extended to weighted norms whenever the weighting matrix is deterministic and constant. Therefore, recalling that $e_n(i) = e_{a,n}(i) + v(i)$, choosing $\mathbf{\Gamma} = \mathbf{R}_u$ in (10.34) yields

$$\zeta_n^{2\text{-RLS}} = \frac{(1 - \lambda_n) M \sigma_v^2 + (1 - \lambda_n)^{-1} \text{Tr}(\mathbf{Q} \mathbf{R}_u)}{2 - (1 - \lambda_n) M} \quad (10.35a)$$

$$\zeta_{12}^{2\text{-RLS}} = \frac{(1 - \lambda_1)(1 - \lambda_2) M \sigma_v^2 + \text{Tr}(\mathbf{Q} \mathbf{R}_u)}{2 - \lambda_1 - \lambda_2 - (1 - \lambda_1)(1 - \lambda_2) M}. \quad (10.35b)$$

As in the previous sections, (10.35) can be used in conjunction with the supervising parameter value from (10.21) into the global relation (10.5) to estimate the steady-state global EMSE of this combination.

10.2.3.2 With feedback

In the feedback case, a procedure similar to Section 10.1.4 can be followed, albeit using weighted norms. Hence, before proceeding, a weighted counterpart of the MSD relation (10.6) must be obtained. Explicitly,

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_i\|_{\mathbf{\Gamma}}^2 &= \bar{\eta}^2 \mathbb{E} \|\tilde{\mathbf{w}}_{1,i}\|_{\mathbf{\Gamma}}^2 \\ &\quad + 2\bar{\eta}(1 - \bar{\eta}) \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{2,i} \\ &\quad + (1 - \bar{\eta})^2 \mathbb{E} \|\tilde{\mathbf{w}}_{2,i}\|_{\mathbf{\Gamma}}^2. \end{aligned} \quad (10.36)$$

Then, (10.32) can be subtracted from $\mathbf{w}_i^o$ taking $\mathbf{w}_{n,a} = \mathbf{w}_{i-1}$ to get

$$\tilde{\mathbf{w}}_{n,i} = \tilde{\mathbf{w}}_{i-1} + \mathbf{q}_i - \mathbf{P}_{n,i} \mathbf{u}_i^T e(i),$$

which under A.2 leads to the following weighted variance and covariance relations:

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_{n,i}\|_{\mathbf{\Gamma}}^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|_{\mathbf{\Gamma}}^2 + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - 2 \mathbb{E} \tilde{\mathbf{w}}_{i-1}^T \mathbf{\Gamma} \mathbf{P}_{n,i} \mathbf{u}_i^T e(i) \\ &\quad + \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{P}_{n,i}^T \mathbf{\Gamma} \mathbf{P}_{n,i}}^2 e^2(i) \end{aligned} \quad (10.37a)$$

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{2,i} &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|_{\mathbf{\Gamma}}^2 + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - \mathbb{E} \mathbf{u}_i \mathbf{P}_{1,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{i-1} e(i) \\ &\quad - \mathbb{E} \mathbf{u}_i \mathbf{P}_{2,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{i-1} e(i) \\ &\quad + \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{P}_{1,i}^T \mathbf{\Gamma} \mathbf{P}_{2,i}}^2 e^2(i). \end{aligned} \quad (10.37b)$$

Finally, using the weighted counterpart of the global steady-state in Definition 10.2 and the approximation A.9, (10.37) can be substituted in (10.36) with $\mathbf{\Gamma} = \mathbf{R}_u$ to get

$$\zeta^{f\{2\text{-RLS}\}} = \frac{(1 - \bar{\lambda})M\sigma_v^2 + (1 - \bar{\lambda})^{-1} \text{Tr}(\mathbf{Q}\mathbf{R}_u)}{2 - (1 - \bar{\lambda})M}, \quad (10.38)$$

where $\bar{\lambda} = \bar{\eta}\lambda_1 + (1 - \bar{\eta})\lambda_2$. Notice that (10.38) is identical to the expression that would be obtained in the analysis of a stand-alone RLS with forgetting factor $\bar{\lambda}$ [18, Lemma 7.10.1]. The overall combination, however, is not equivalent to a variable forgetting factor algorithm (Figure 48a). Indeed, it amounts to an RLS filter whose inverse Hessian approximation is $\mathbf{P}_i = \eta(i)\mathbf{P}_{1,i} + [1 - \eta(i)]\mathbf{P}_{2,i}$, which is different from some $\mathbf{P}_{\text{eq},i}$ updated as in (2.6) using $\lambda_{\text{eq}} = \eta(i)\lambda_1 + [1 - \eta(i)]\lambda_2$. Nevertheless, the λ_{eq} of the combination displays the expected behavior, switching to a lower forgetting factor upon an abrupt change in the system (Figure 48b).

Figure 48: Steady-state analysis of a 2 · RLS combination ($w^o \rightarrow -w^o$ at $i = 1000$). **Correlated input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\gamma = 0.95$, $\lambda_1 = 0.92$, $\lambda_2 = 0.99$, and $\epsilon = 10^{-6}$. **$L \rightarrow \infty$:** $\mu_a = 2$, $\beta = 0.9$, and $\epsilon = 10^{-3}$ (normalized convex supervisor); **$1 < L < \infty$:** $L = 30$, $\mu_a = 2.3$, $\beta = 0.9$, and $\epsilon = 10^{-4}$ (normalized convex supervisor); **Stand-alone RLS with equivalent forgetting factor:** $\lambda(i) = \eta(i)\lambda_1 + [1 - \eta(i)]\lambda_2$, where $\eta(i)$ is the instantaneous supervising parameter of the combination with coefficients feedback.

10.2.4 LMS + RLS

Notice that derivations similar to those in Section 10.2.3 can be used to analyze LMS + RLS combinations by simply replacing the matrix $\mathbf{P}_{1,i}$ by $\mu_1 \mathbf{I}$.

10.2.4.1 Without feedback

The parallel-independent case has been thoroughly studied in [34]. The results are reproduced here for reference. Indeed, formulae for the LMS and the RLS steady-state EMSE required to evaluate (10.5) are found in (10.24a) and (10.35a), respectively. The steady-state cross-EMSE is obtained from [34]:

$$\zeta_{12}^{\{\text{LMS+RLS}\}} = \mu_1(1 - \lambda_2)\sigma_v^2 \text{Tr}(\mathbf{\Sigma}^{-1} \mathbf{R}_u) + \text{Tr}(\mathbf{Q} \mathbf{\Sigma}^{-1} \mathbf{R}_u), \quad (10.39)$$

where $\mathbf{\Sigma} = \mu_1 \mathbf{R}_u + (1 - \lambda_2) \mathbf{I}$.

10.2.4.2 With feedback

For the case with coefficients feedback, replace $\mathbf{P}_{1,i}$ by $\mu_1 \mathbf{I}$ in (10.38) to get

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_{1,i}\|_{\mathbf{\Gamma}}^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|_{\mathbf{\Gamma}}^2 + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - 2\mu_1 \mathbb{E} \tilde{\mathbf{w}}_{i-1}^T \mathbf{\Gamma} \mathbf{u}_i^T [e_a(i) + v(i)] \\ &\quad + \mu_1^2 \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{\Gamma}}^2 [e_a(i) + v(i)]^2 \end{aligned} \quad (10.40a)$$

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_{2,i}\|_{\mathbf{\Gamma}}^2 &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|_{\mathbf{\Gamma}}^2 + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - 2 \mathbb{E} \tilde{\mathbf{w}}_{i-1}^T \mathbf{\Gamma} \mathbf{P}_{2,i} \mathbf{u}_i^T [e_a(i) + v(i)] \\ &\quad + \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{P}_{2,i}^T \mathbf{\Gamma} \mathbf{P}_{2,i}}^2 [e_a(i) + v(i)]^2 \end{aligned} \quad (10.40b)$$

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_{1,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{2,i} &= \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|_{\mathbf{\Gamma}}^2 + \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) \\ &\quad - \mu_1 \mathbb{E} u_i \mathbf{\Gamma} \tilde{\mathbf{w}}_{i-1} [e_a(i) + v(i)] \\ &\quad - u_i \mathbb{E} \mathbf{P}_{2,i}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{i-1} [e_a(i) + v(i)] \\ &\quad + \mu_1 \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{\Gamma} \mathbf{P}_{2,i}} [e_a(i) + v(i)]^2. \end{aligned} \quad (10.40c)$$

Using (10.40) in the weighted MSD relation (10.36), the overall variance relation of the combination can be expressed as

$$\begin{aligned} 2 \mathbb{E} \mathbf{u}_i \mathbf{M}^T \mathbf{\Gamma} \tilde{\mathbf{w}}_{i-1} e_a(i) &= \text{Tr}(\mathbf{Q}\mathbf{\Gamma}) + \bar{\eta} \mu_1 \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{\Gamma} \mathbf{M}}^2 [e_a(i) + v(i)]^2 \\ &\quad + (1 - \bar{\eta}) \mathbb{E} \|\mathbf{u}_i\|_{\mathbf{M}^T \mathbf{\Gamma} \mathbf{P}_{2,i}}^2 [e_a(i) + v(i)]^2, \end{aligned} \quad (10.41)$$

where $\mathbf{M} = \bar{\eta} \mu_1 \mathbf{I} + (1 - \bar{\eta}) \mathbf{P}_{2,i}$. Choosing $\mathbf{\Gamma} = \mathbf{R}_u \mathbf{H}^{-1}$, for $\mathbf{H} = \bar{\eta} \mu_1 \mathbf{R}_u + (1 - \bar{\eta})(1 - \lambda_2) \mathbf{I}$, and using the data separation principle A.7 and the approximation A.9, (10.41) reduces to a closed-form relation for the global EMSE at steady-state:

$$\zeta^{f\{\text{LMS+RLS}\}} = \frac{\text{Tr}(\mathbf{H}) \sigma_v^2 + \text{Tr}(\mathbf{Q} \mathbf{R}_u \mathbf{H}^{-1})}{2 - \text{Tr}(\mathbf{H})}. \quad (10.42)$$

Figure 49 shows the accuracy of the expression in (10.42). It also illustrates well the performance improvements achieved at almost no additional cost when using coefficients feedback. Indeed, taking the parallel-independent LMS+RLS in the same scenario as [34] and embedding a cyclic coefficients feedback ($L = 25$) can improve tracking behavior by up to 2 dB.

Figure 49: Tracking analysis of an LMS + RLS combination. **Correlated input data:** $M = 5$, $\sigma_u^2 = 1/M$, $\sigma_v^2 = 10^{-2}$, $\gamma = 0.8$, $\mu_{\text{LMS}} = 0.04$, $\lambda_{\text{RLS}} = 0.92$, and $\epsilon = 10^{-6}$. **Parallel-independent:** $\mu_a = 100$ (convex supervisor); **cyclic coefficients feedback:** $L = 25$ and $\mu_a = 100$ (convex supervisor).

10.3 Transient Performance

Sections 10.1 and 10.2 revealed that coefficients feedback can be used to improve the steady-state and tracking of parallel combinations. Nevertheless, the initial motivation of this topology was to address the convergence stagnation issue of parallel-independent structures (Section 5.2.1), which is clearly a transient phenomenon. Hence, to analyze this effect, the current section examines the overall convergence behavior of a 2·LMS combination with cyclic coefficients feedback. In doing so, a transient supervisor model is derived, allowing the impact of cyclic feedback on the supervisor variance to be investigated. Notice that, although analyses of convex and affine supervising rules for parallel-independent topologies have been carried out [39, 110, 123, 128], the following derivations are valid for stochastic gradient supervisors with arbitrary activation functions. Indeed, the approach from [128] is extended to any supervising rule of the form (10.2), of which the convex (5.8) and the affine supervisors (5.10) are special cases.

In the sequel, a stationary scenario is adopted ($\mathbf{q}_i = \mathbf{0} \Rightarrow \mathbf{Q} = \mathbf{0}$), such that the coefficients errors are redefined as $\tilde{\mathbf{w}}_i = \mathbf{w}^o - \mathbf{w}_i$ and $\tilde{\mathbf{w}}_{n,i} = \mathbf{w}^o - \mathbf{w}_{n,i}$ and the *a priori* errors become $e_a(i) = \mathbf{u}_i \tilde{\mathbf{w}}_{i-1}$ and $e_{a,n}(i) = \mathbf{u}_i \tilde{\mathbf{w}}_{n,i-1}$. Again, an assumption is used to decouple the analyses of the supervisor and the component filters. Assumption A.3 is therefore extended for all i :

A.10 (*Transient supervisor separation principle*) The supervising parameter varies slowly enough that $\mathbb{E}[\eta(i)e_{a,m}(i)e_{a,n}(i) | \eta(i)] = \mathbb{E}\eta(i) \mathbb{E}[e_{a,m}(i)e_{a,n}(i)]$, $m, n = 1, 2$.

Notice that A.10 is stated in terms of the conditional independence of the RVs $\eta(i)$ and $e_{a,n}(i)$. Here, however, no assumptions are made on the supervising parameter variance (as in A.4) since it is part of the model derived below.

Under A.10 and for $\mathbf{q}_i = \mathbf{0}$, the global EMSE relation (10.4) can be rearranged to read

$$\zeta(i) = [\mathbb{E} \eta^2(i-1)] [\Delta\zeta_1(i) + \Delta\zeta_2(i)] - 2 [\mathbb{E} \eta(i-1)] \Delta\zeta_2(i) + \zeta_2(i), \quad (10.43)$$

where $\Delta\zeta_n(i) = \zeta_n(i) - \zeta_{12}(i)$. The following two sections are then dedicated to determining the values necessary to evaluate (10.43) for all i . Namely, Section 10.3.1 devises recursions for the component filters EMSE and cross-EMSE $[\zeta_n(i)$ and $\zeta_{12}(i)]$ and Section 10.3.2 analyzes the supervising parameters moments $\mathbb{E} \eta^2(i)$ and $\mathbb{E} \eta(i)$. The interpretation of the model is deferred to Section 10.3.3, which collects all results into a complete transient analysis, so that the effect of cyclic coefficients feedback on components and supervisor can be studied simultaneously.

10.3.1 Component filters analysis

Before proceeding with the derivations of the local errors statistics recursions, assume that

A.11 (*Data independence assumptions*) $\{\mathbf{u}_i\}$ constitutes an i.i.d. sequence of vectors independent of $v(j)$, for all i, j . Consequently, $\{\mathbf{u}_i, \tilde{\mathbf{w}}_j\}$, $\{d(i), d(j)\}$, and $\{\mathbf{u}_i, d(j)\}$ are independent for $i > j$.

Note that in some cases (e.g., when $u(i)$ are i.i.d. and the regressor is updated in parallel), A.11 is exact. Even when it is not the case, this assumption is common in transient analyses [18, 19] and yields good match with simulations (see Section 10.3.3).

Under A.11, it holds from the definition of the *a priori* error that

$$\begin{aligned} \zeta_n(i) &= \mathbb{E} \mathbf{u}_i \tilde{\mathbf{w}}_{n,i-1} \tilde{\mathbf{w}}_{n,i-1}^T \mathbf{u}_i^T = \text{Tr}(\mathbf{R}_u \mathbf{K}_{n,i-1}) \\ \zeta_{12}(i) &= \mathbb{E} \mathbf{u}_i \tilde{\mathbf{w}}_{1,i-1} \tilde{\mathbf{w}}_{2,i-1}^T \mathbf{u}_i^T = \text{Tr}(\mathbf{R}_u \mathbf{K}_{12,i-1}) \\ \zeta(i) &= \mathbb{E} \mathbf{u}_i \tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T \mathbf{u}_i^T = \text{Tr}(\mathbf{R}_u \mathbf{K}_{i-1}), \end{aligned} \quad (10.44)$$

where $\mathbf{K}_{n,i} = \mathbb{E} \tilde{\mathbf{w}}_{n,i} \tilde{\mathbf{w}}_{n,i}^T$, $\mathbf{K}_{12,i} = \mathbf{K}_{21,i}^T = \mathbb{E} \tilde{\mathbf{w}}_{1,i} \tilde{\mathbf{w}}_{2,i}^T$, and $\mathbf{K}_i = \mathbb{E} \tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^T$, are covariance and cross-covariance matrices of the coefficients errors. Due to the linearity of the trace operator [154], it also holds that $\Delta\zeta_n(i) = \text{Tr}(\mathbf{R}_u \Delta\mathbf{K}_{n,i-1})$ with $\Delta\mathbf{K}_{n,i} = \mathbf{K}_{n,i} - \mathbf{K}_{12,i}$. For the 2 · LMS combination, these matrices can be derived from the general AF recur-

sion (10.1), with $g_n[x(i)] = x(i)$, and the coefficients feedback relation (5.13). Defining $\mathbf{K}_{n,a} = \mathbb{E} \tilde{\mathbf{w}}_{n,a} \tilde{\mathbf{w}}_{n,a}^T$:

$$\begin{aligned} \mathbf{K}_{n,i} &= \mathbf{K}_{n,a} - \mu_n \mathbf{K}_{n,a} \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i - \mu_n \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{n,a} \\ &\quad + \mu_n^2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{n,a} \mathbf{u}_i^T \mathbf{u}_i + \mu_n^2 \sigma_v^2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \end{aligned} \quad (10.45a)$$

$$\begin{aligned} \mathbf{K}_{12,i} &= \mathbf{K}_{12,a} - \mu_2 \mathbf{K}_{12,a} \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i - \mu_1 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{12,a} \\ &\quad + \mu_1 \mu_2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{12,a} \mathbf{u}_i^T \mathbf{u}_i + \mu_1 \mu_2 \sigma_v^2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \end{aligned} \quad (10.45b)$$

$$\mathbf{K}_{n,a} = \delta_L(i) \mathbf{K}_{i-1} + [1 - \delta_L(i)] \mathbf{K}_{n,i-1} \quad (10.46a)$$

$$\mathbf{K}_{12,a} = \delta_L(i) \mathbf{K}_{i-1} + [1 - \delta_L(i)] \mathbf{K}_{12,i-1} \quad (10.46b)$$

First, consider the case upon feedback, when $i = rL$, $r \in \mathbb{N}$, in (10.46). Then, (10.45) yields, for $n = 1, 2$,

$$\begin{aligned} \mathbf{K}_{n,i} &= \mathbf{K}_{i-1} - \mu_n [\mathbf{K}_{i-1} \mathbf{R}_u + \mathbf{R}_u \mathbf{K}_{i-1}] \\ &\quad + \mu_n^2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{i-1} \mathbf{u}_i^T \mathbf{u}_i + \mu_n^2 \sigma_v^2 \mathbf{R}_u \end{aligned} \quad (10.47a)$$

$$\Delta \mathbf{K}_{1,i} = (\mu_2 - \mu_1) [\mathbf{K}_{i-1} \mathbf{R}_u - \mu_1 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{i-1} \mathbf{u}_i^T \mathbf{u}_i - \mu_1 \sigma_v^2 \mathbf{R}_u] \quad (10.47b)$$

$$\Delta \mathbf{K}_{2,i} = (\mu_1 - \mu_2) [\mathbf{R}_u \mathbf{K}_{i-1}^T - \mu_2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{K}_{i-1} \mathbf{u}_i^T \mathbf{u}_i - \mu_2 \sigma_v^2 \mathbf{R}_u] \quad (10.47c)$$

To account for arbitrary coloring of the input signal, define the eigenvalue decomposition (EVD) of the input regressor covariance matrix as $\mathbf{R}_u = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$, where $\mathbf{Q}$ is an orthonormal eigenvector matrix and $\mathbf{\Lambda} = \text{diag}\{\lambda_m\}$ is a diagonal matrix of eigenvalues λ_m , $m = 1, \dots, M$. Given that the trace operator is invariant to similarity transformations [154], the following relations hold from (10.44):

$$\begin{aligned} \zeta_n(i) &= \text{Tr}(\mathbf{\Lambda} \bar{\mathbf{K}}_{n,i-1}) \\ \zeta_{12}(i) &= \text{Tr}(\mathbf{\Lambda} \bar{\mathbf{K}}_{12,i-1}) \\ \Delta \zeta_n(i) &= \text{Tr}(\mathbf{\Lambda} \Delta \bar{\mathbf{K}}_{n,i-1}) \end{aligned} \quad (10.48)$$

where for any matrix $\mathbf{H}$: $\bar{\mathbf{H}} = \mathbf{Q}^T \mathbf{H} \mathbf{Q}$. Then, (10.47) can be transformed into

$$\begin{aligned} \bar{\mathbf{K}}_{n,i} &= \bar{\mathbf{K}}_{i-1} - \mu_n [\bar{\mathbf{K}}_{i-1} \mathbf{\Lambda} + \mathbf{\Lambda} \bar{\mathbf{K}}_{i-1}] \\ &\quad + \mu_n^2 \mathbb{E} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i \bar{\mathbf{K}}_{i-1} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i + \mu_n^2 \sigma_v^2 \mathbf{\Lambda} \\ \Delta \bar{\mathbf{K}}_{1,i} &= (\mu_2 - \mu_1) [\bar{\mathbf{K}}_{i-1} \mathbf{\Lambda} - \mu_1 \mathbb{E} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i \bar{\mathbf{K}}_{i-1} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i - \mu_1 \sigma_v^2 \mathbf{\Lambda}] \\ \Delta \bar{\mathbf{K}}_{2,i} &= (\mu_1 - \mu_2) [\mathbf{\Lambda} \bar{\mathbf{K}}_{i-1}^T - \mu_2 \mathbb{E} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i \bar{\mathbf{K}}_{i-1} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i - \mu_2 \sigma_v^2 \mathbf{\Lambda}] \end{aligned}$$

where $\bar{\mathbf{u}}_i = \mathbf{u}_i \mathbf{Q}$. Since $\bar{\mathbf{u}}_i$ is Gaussian (according to the data model from Section 1.6.1) and its covariance matrix is diagonal (namely, $\mathbf{E} \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i = \mathbf{\Lambda}$), the fourth-order moment relation [18] can be used to get

$$\begin{aligned} \bar{\mathbf{K}}_{n,i} &= \bar{\mathbf{K}}_{i-1} - \mu_n [\bar{\mathbf{K}}_{i-1} \mathbf{\Lambda} + \mathbf{\Lambda} \bar{\mathbf{K}}_{i-1}] \\ &\quad + \mu_n^2 [\mathbf{\Lambda} \text{Tr}(\bar{\mathbf{K}}_{i-1} \mathbf{\Lambda}) + 2\mathbf{\Lambda} \bar{\mathbf{K}}_{i-1} \mathbf{\Lambda}] \\ &\quad + \mu_n^2 \sigma_v^2 \mathbf{\Lambda} \end{aligned} \quad (10.49a)$$

$$\begin{aligned} \Delta \bar{\mathbf{K}}_{1,i} &= (\mu_2 - \mu_1) \{ \bar{\mathbf{K}}_{i-1} \mathbf{\Lambda} \\ &\quad - \mu_1 [\mathbf{\Lambda} \text{Tr}(\bar{\mathbf{K}}_{i-1} \mathbf{\Lambda}) + 2\mathbf{\Lambda} \bar{\mathbf{K}}_{i-1} \mathbf{\Lambda}] \\ &\quad - \mu_1 \sigma_v^2 \mathbf{\Lambda} \} \end{aligned} \quad (10.49b)$$

$$\begin{aligned} \Delta \bar{\mathbf{K}}_{2,i} &= (\mu_1 - \mu_2) \{ \mathbf{\Lambda} \bar{\mathbf{K}}_{i-1}^T \\ &\quad - \mu_2 [\mathbf{\Lambda} \text{Tr}(\bar{\mathbf{K}}_{i-1} \mathbf{\Lambda}) + 2\mathbf{\Lambda} \bar{\mathbf{K}}_{i-1} \mathbf{\Lambda}] \\ &\quad - \mu_2 \sigma_v^2 \mathbf{\Lambda} \} \end{aligned} \quad (10.49c)$$

Recursions for the error statistics required to evaluate (10.44) upon feedback can be obtained from (10.48) and (10.49). For instance, when the input signal is white ($\mathbf{R}_u = \sigma_u^2 \mathbf{I}$), the local EMSE recursions are expressed by

$$\zeta_n(i) = [1 - 2\mu_n \sigma_u^2 + \mu_n^2 (M + 2) \sigma_u^4] \zeta(i - 1) + \mu_n^2 M \sigma_u^4 \sigma_v^2 \quad (10.50a)$$

$$\begin{aligned} \Delta \zeta_n(i) &= (\mu_m - \mu_n) \sigma_u^2 [1 - \mu_n (M + 2) \sigma_u^2] \zeta(i - 1) \\ &\quad + \mu_n (\mu_n - \mu_m) M \sigma_u^4 \sigma_v^2, \end{aligned} \quad (10.50b)$$

where $\zeta(i - 1)$ is evaluated using (10.43).

Now, when there is no feedback ($i \neq rL$ in (10.46)), the closed form recursion for the coefficients error covariance matrices in (10.45) reduce to those well-known in the literature for stand-alone AFs and parallel-independent combinations [19, 39, 123]. Explicitly,

$$\begin{aligned} \bar{\mathbf{K}}_{n,i} &= \bar{\mathbf{K}}_{n,i-1} - \mu_n [\bar{\mathbf{K}}_{n,i-1} \mathbf{\Lambda} + \mathbf{\Lambda} \bar{\mathbf{K}}_{n,i-1}] \\ &\quad + \mu_n^2 [2\mathbf{\Lambda} \bar{\mathbf{K}}_{n,i-1} \mathbf{\Lambda} + \text{Tr}(\mathbf{\Lambda} \bar{\mathbf{K}}_{n,i-1}) \mathbf{\Lambda}] \\ &\quad + \mu_n^2 \sigma_v^2 \mathbf{\Lambda} \end{aligned} \quad (10.51a)$$

$$\begin{aligned} \bar{\mathbf{K}}_{12,i} &= \bar{\mathbf{K}}_{12,i-1} - \mu_1 \mathbf{\Lambda} \bar{\mathbf{K}}_{12,i-1} - \mu_2 \bar{\mathbf{K}}_{12,i-1} \mathbf{\Lambda} \\ &\quad + \mu_1 \mu_2 [2\mathbf{\Lambda} \bar{\mathbf{K}}_{12,i-1} \mathbf{\Lambda} + \text{Tr}(\mathbf{\Lambda} \bar{\mathbf{K}}_{12,i-1}) \mathbf{\Lambda}] \\ &\quad + \mu_1 \mu_2 \sigma_v^2 \mathbf{\Lambda}. \end{aligned} \quad (10.51b)$$

Considering, once again, that the input data is white, the components EMSE and cross-

EMSE yield, from (10.51),

$$\begin{aligned}\zeta_n(i) &= [1 - 2\mu_n\sigma_u^2 + \mu_n^2(M+2)\sigma_u^4]\zeta_n(i-1) \\ &\quad + \mu_n^2 M\sigma_u^4\sigma_v^2\end{aligned}\tag{10.52a}$$

$$\begin{aligned}\Delta\zeta_n(i) &= [1 - \mu_n\sigma_u^2]\Delta\zeta_n(i-1) \\ &\quad - \sigma_u^2[1 - \mu_n(M+2)\sigma_u^2][\mu_n\zeta_n(i-1) - \mu_m\zeta_{12}(i-1)] \\ &\quad + \mu_n(\mu_n - \mu_m)M\sigma_u^4\sigma_v^2\end{aligned}\tag{10.52b}$$

10.3.2 Supervisor analysis

Similar to the transient model derivations in [128], the activation function is approximated by a first-order Taylor expansion around $\bar{a}(i-1) = \mathbb{E} a(i-1)$:

$$\eta(i-1) = f[a(i-1)] \approx \bar{f} + \bar{f}' \cdot [a(i-1) - \bar{a}(i-1)].\tag{10.53}$$

Recall that, for any function $h(a)$, $\bar{h} = h[\bar{a}(i-1)]$. Using (10.53), the supervising parameters moments in (10.43) become

$$\mathbb{E} \eta(i-1) \approx \bar{f}\tag{10.54a}$$

$$\mathbb{E} \eta^2(i-1) \approx \bar{f}^2 + \sigma_a^2(i-1)[\bar{f}']^2,\tag{10.54b}$$

taking $\sigma_a^2(i) = \mathbb{E}[a(i) - \bar{a}(i)]^2$ as the variance of the auxiliary parameter a . Notice that under the small supervisor variance assumption A.4 (i.e., $\sigma_a^2 \approx 0$), (10.54b) reduces to the form used in the steady-state analyses. Moreover, (10.54) is expressed solely in terms of moments of $a(i-1)$, for which recursions are now obtained.

Recall that an approximate recursion for $a(i)$ was derived in Section 10.1.5 also by relying on first-order approximations of f . Namely,

$$\begin{aligned}a(i) \approx a(i-1) + \mu_a \{ &\mathcal{A}_2 e_{a,2}^2(i) + \mathcal{A}_{12} e_{a,1}(i) e_{a,2}(i) \\ &- \mathcal{A}_1 e_{a,1}^2(i) + \mathcal{A}_v [e_{a,2}(i) - e_{a,1}(i)] v(i) \}\end{aligned}\tag{10.55}$$

$$\begin{aligned}\mathcal{A}_1 &\approx \bar{f}\bar{f}' + [(\bar{f}')^2 + \bar{f}\bar{f}''](a - \bar{a}) \\ \mathcal{A}_2 &\approx (1 - \bar{f})\bar{f}' + [-(\bar{f}')^2 + (1 - \bar{f})\bar{f}''](a - \bar{a}) \\ \mathcal{A}_{12} &\approx (2\bar{f} - 1)\bar{f}' + [2(\bar{f}')^2 + (2\bar{f} - 1)\bar{f}''](a - \bar{a}) \\ \mathcal{A}_v &\approx \bar{f}' + \bar{f}''(a - \bar{a})\end{aligned}$$

The only distinction is that, instead of the steady-state supervisor separation principle

A.3, the more stringent A.10 is necessary to find mean and variance recursions that hold for all i . Additionally, the following assumption is required to evaluate *a priori* errors moments:

A.12 (*Jointly Gaussian a priori errors*) The *a priori* errors $\{e_{a,n}(i)\}$, $n = 1, 2$, are zero mean jointly Gaussian RVs, so that

$$\begin{aligned} \mathbb{E} e_{a,n}^4(i) &= 3\zeta_n^2(i) \\ \mathbb{E} e_{a,1}^k(i) e_{a,2}^\ell(i) &= \begin{cases} 0, & k + \ell = 3 \\ 3\zeta_1(i)\zeta_{12}(i), & k = 3, \ell = 1 \\ 3\zeta_2(i)\zeta_{12}(i), & k = 1, \ell = 3 \\ 2\zeta_{12}^2(i) + \zeta_1(i)\zeta_2(i), & k = \ell = 2 \end{cases} \end{aligned}$$

This assumption is accurate for small step-sizes and long filters (large M) even when the inputs are not normally distributed due to the central limit theorem [18]. Finally, evaluating the expected value of the approximate recursion (10.55) and its square, one gets

$$\bar{a}(i) \approx \bar{a}(i-1) + \mu_a[(1 - \bar{f})\Delta\zeta_2 - \bar{f}\Delta\zeta_1]\bar{f}' \quad (10.56a)$$

$$\sigma_a^2(i) \approx [1 + 2\mu_a\mathcal{B}_1 + \mu_a^2\mathcal{B}_2]\sigma_a^2(i-1) + \mu_a^2\mathcal{B}_v \quad (10.56b)$$

$$\begin{aligned} \mathcal{B}_1 &= [(1 - \bar{f})\Delta\zeta_2 - \bar{f}\Delta\zeta_1]\bar{f}'' - [\Delta\zeta_1 + \Delta\zeta_2](\bar{f}')^2 \\ \mathcal{B}_2 &= 3(\bar{f}')^4(\Delta\zeta_1 + \Delta\zeta_2)^2 + (\bar{f}'')^2[\zeta_2(\Delta\zeta_1 + \Delta\zeta_2) + 2\Delta\zeta_2^2] \\ &\quad + 3\bar{f}(\bar{f}'')^2(\Delta\zeta_1 + \Delta\zeta_2)[\bar{f}\Delta\zeta_1 - (2 - \bar{f})\Delta\zeta_2] \\ &\quad + 6(\bar{f}')^2\bar{f}''(\Delta\zeta_1 + \Delta\zeta_2)[\bar{f}\Delta\zeta_1 - (1 - \bar{f})\Delta\zeta_2] \\ &\quad + (\bar{f}'')^2(\Delta\zeta_1 + \Delta\zeta_2)\sigma_v^2 \\ \mathcal{B}_v &= (\bar{f}')^2\{\Delta\zeta_2^2 + (\Delta\zeta_1 + \Delta\zeta_2) \times \\ &\quad [2\bar{f}^2(\Delta\zeta_1 + \Delta\zeta_2) - 4\bar{f}\Delta\zeta_2 + \zeta_2 + \sigma_v^2]\}. \end{aligned}$$

In contrast to [128], (10.56) is valid for any activation function as long as it is strictly increasing. In fact, the result for the convex supervisor from [128] is recovered for an appropriate choice of f . For the affine supervisor, however, this analysis differs from the previous literature. Indeed, [108] assumes that the mean supervisor parameter is identical to the optimal one, while [110] relies on a different scheme and does not explicitly evaluate these quantities. Finally, (10.56) does not match the model in [39] due to algebraic

mistakes in its derivations. Indeed, [39, Eq. (48)] can result in the estimation of negative supervisor variances.

10.3.3 Complete transient analysis

Combining the results from Sections 10.3.1 and 10.3.2, a complete transient model for the $2 \cdot \text{LMS}$ combination with cyclic coefficients feedback can be obtained. To do so, the components model (10.49) and (10.51) are alternated depending on whether feedback occurs in the iteration. The full model is illustrated in the sequel for the i.i.d. input sequence case, although its generalization to arbitrarily colored $\{u(i)\}$ is straightforward.

Assuming the component coefficients and the supervisor are initialized with $\mathbf{w}_{n,-1} = \mathbf{0}$ and $a(-1) = 0$, the transient analysis start with $\zeta_1(-1) = \zeta_2(-1) = \zeta_{12}(-1) = \sigma_u^2 \|\mathbf{w}^o\|^2$ and iterates through:

- (i) Update the errors statistics using:
$$\begin{cases} (10.50), & i = rL \\ (10.52), & i \neq rL \end{cases}$$
- (ii) Evaluate the supervising parameter moments using (10.54)
- (iii) Evaluate the global EMSE using (10.43)
- (iv) Update the auxiliary variable statistics using (10.56)

The validity of this model is illustrated for a convex and affine supervisor in Figures 51 and 52. Notice that, due to the first-order approximation, the variance model in 51e is not as precise as that in 52, although both display the same tendencies as the simulations. Only when the supervising parameter is switching does the model overestimates the parameter variance to the point of affecting the global EMSE model.

The results in this section show the advantages of using of cyclic coefficients feedback in parallel combinations from a transient behavior viewpoint. Firstly, notice from (10.50) that feeding back the global coefficients makes all local EMSEs become functions of the global EMSE $\zeta(i-1)$. In doing so, the convergence stagnation issue is eradicated, since the difference between the components errors cannot grow very large (for appropriate cycle periods). This effect is evident when (b) and (c) in Figures 51 and 52. Indeed, upon feedback, the difference between the $\zeta_n(i)$ is proportional to the difference between the components step sizes.

Figure 50: Supervising parameter variance under different topologies (normalized convex supervisor). **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.05$, and $\mu_2 = 0.005$. **Parallel-independent:** $\mu_a = 1$, $\beta = 0.9$, and $\epsilon = 10^{-3}$; **coefficients leakage:** $\alpha_0 = 0.6$, $\bar{\alpha} = 0.98$, $\mu_a = 2.5$, $\beta = 0.9$, and $\epsilon = 10^{-3}$; **cyclically incremental:** $L = 10$, $\tau = 0.98$, $\mu_a = 3$, $\beta = 0.9$, and $\epsilon = 10^{-3}$; **coefficients feedback:** $L = 80$, $\mu_a = 4$, $\beta = 0.8$, and $\epsilon = 10^{-2}$.

Secondly, note that (10.46) implies that $\Delta\zeta_n(i-1)$ vanishes upon feedback, since $\zeta_n(i-1) = \zeta_{12}(i-1) = \zeta(i-1)$. Thus, the EMSE/cross-EMSE gap is reduced, which, from (10.56b), points to a considerable smaller in the supervising parameter variance compared to other topologies and, ultimately, an improved stability and performance (Figure 50). Nevertheless, if the cycle period is not correctly designed, the feedbacks can hinder the stability of the supervisor (52e).

Finally, the mean supervising parameter model (10.56a) explicitly reveals how coefficients feedback stalls the supervisor adaptation. This fact justifies the use of *cyclic* feedback, so that the combination can exploit the reduction in the supervisor variance and avoid convergence stagnation without hindering the tracking of the optimal supervisor (Figures 51c and 52c).

10.4 Concluding remarks

The analysis presented here for parallel combinations without and with feedback demonstrate the advantages and effects of increasing cooperation by means of a strongly connected topology. Firstly, coefficients feedback allows for the steady-state/tracking analyses of the parallel combination to be derived in a single step, following a path similar to that of stand-alone AFs. Parallel-independent structures, on the other hand, require that the component filters and the output of the combination be studied separately. Sec-

Figure 51: Transient analysis of a 2-LMS combination with a convex supervisor. **White input data:** $M = 10$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.05$, and $\mu_2 = 0.005$. **Parallel-independent:** $\mu_a = 170$; **cyclic coefficients feedback:** $L = 100$ and $\mu_a = 250$. Transient expression use $f(a) = [1 + e^{-a}]^{-1}$ and $a(i) \in [-4, 4]$.

only, the use of coefficients feedback strengthens the effect of the supervising parameters on the steady-state/tracking behavior. This is clear, for instance, in the LMS + RLS

Figure 52: Transient analysis of the 2-LMS combination with an affine supervisor. **White input data:** $M = 7$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-2}$, $\mu_1 = 0.05$, and $\mu_2 = 0.01$. **Parallel-independent:** $\mu_\eta = 1$ and $\eta_{\min} = -0.25$; **cyclic coefficients feedback:** $L = 60$, $\mu_\eta = 1.4$, and $\eta_{\min} = -0.25$. Transient expressions use $f(a) = a$ and $a(i) \in [-0.25, 1.25]$.

case, in which the supervising parameter value $\bar{\eta}$ influences the system disturbance covariance $\mathbf{Q}$ directly [see (10.41)], instead of through the weighting of the cross-EMSE [as

in (10.39)]. Finally, these analyses clarify the contribution of the feedback structure to the transient behavior, by showing how the combination contributes to improving the convergence rate of the component filters and by eliciting its variance reduction effect on the supervising parameters.

11 PERFORMANCE ANALYSIS OF INCREMENTAL COMBINATIONS

What is now proved was once only imagined.

-- William Blake, "The Marriage of Heaven and Hell"

Analyzing incremental combinations is inherently intricate due to the coupling of the component filters recursions. Thus, one cannot study the AFs individually to then combine the results. Moreover, it should be noted that although this combination bears strong resemblance to incremental ANs [3], the analysis of the latter assumes that the nodes' data pairs are independent, which is clearly not the case for combinations of AFs (unless specifically enforced by the data distribution strategy).

Although, the behavior of parallel combinations has been extensively studied in the literature and the unified analysis in Chapter 10, performance analyses of incremental combinations are still in early stages. Indeed, tracking and steady-state results for data sharing incremental combinations of two AFs were derived in [43] for a convexly constrained supervisor. However, Section 6.1 argued that such constraint can hinder the performance of incremental structures and the model proposed in [43] has limitations when it comes to accounting for arbitrary $\{\eta_n(i)\}$. Moreover, Chapter 9 motivates the use of incremental chains with $N > 2$ component filters. As for the transient behavior, mean convergence analyses of arbitrarily sized incremental chains of LMS filters was carried out in [45] and mean-square analyses for the $\{\text{LMS}\}^2$ is available from [44]. All these derivations consider data sharing strategies.

The following derivations present steady-state mean-square analysis for incremental combinations of N LMS filters using data sharing (Section 11.2). Transient performance of this topology is also studied by providing mean (Section 11.3.1) and mean-square (Section 11.3.3) convergence results for $\{\text{LMS}\}^N$ and $\{\text{LMS}\}^2$ combinations, respectively. These derivations consider deterministic supervising parameters, although not necessarily static or convex, so that their optimal values (Section 11.3.4) and unbiasedness constraint (Section 11.3.2) can be evaluated subsequently. Restrictive as it may seem, this is actually the most common application case of the incremental topology. In fact, recall that supervised incremental combinations, as they stand, may not even make sense since

they can be construed as unsupervised combinations of VSS AFs. Moreover, contrary to parallel structures, outstanding results can be achieved using fixed $\{\eta_n\}$ [43, 45, 47, 73].

Even though these analyses are carried out for LMS filters, they can be extended to account for other adaptive algorithms by mean of arguments similar to those in Chapter 10, albeit using AFs with data nonlinearities (see Section 2.2). Accounting for data buffering methods, however, is considerably more challenging.

11.1 Overall recursion

The analysis of incremental combinations begins by deriving an *overall recursion* between the global coefficients $\mathbf{w}_{i-1}$ and $\mathbf{w}_i$. To do so, the incremental combination of N LMS filters:

$$\begin{cases} \mathbf{w}_{0,i} = \mathbf{w}_{i-1} \\ \mathbf{w}_{n,i} = \mathbf{w}_{n-1,i} + \eta_n(i) \mu_n \mathbf{u}_i^T [d(i) - \mathbf{u}_i \mathbf{w}_{n-1,i}] \\ \mathbf{w}_i = \mathbf{w}_{N,i} \end{cases}$$

Combining these equations yields

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mathbf{u}_i^T \sum_{n=1}^N \eta_n(i) \mu_n e_n(i), \quad (11.1)$$

with $e_1(i) = e(i)$ and $e_n(i) = d(i) - \mathbf{u}_i \mathbf{w}_{n-1,i}$, $n = 2, \dots, N$.

The $\{e_n(i)\}$ in (11.1) depend on the intermediary coefficients $\{\mathbf{w}_{n-1,i}\}$ and need to be eliminated. To do, notice that $e_n(i)$ can be rewritten as a function of $e(i)$ as in

$$\begin{aligned} e_1(i) &= e(i) \\ e_2(i) &= d(i) - \mathbf{u}_i \mathbf{w}_{1,i} = \\ &= d(i) - \mathbf{u}_i [\mathbf{w}_{i-1} + \eta_1(i) \mu_1 \mathbf{u}_i^T e(i)] = \\ &= (1 - \eta_1(i) \mu_1 \|\mathbf{u}_i\|^2) e(i), \end{aligned}$$

which, by induction, lead to the general form

$$e_n(i) = \begin{cases} e(i), & n = 1 \\ \prod_{\ell=1}^{n-1} (1 - \eta_\ell \mu_\ell \|\mathbf{u}_i\|^2) e(i) & n = 2, \dots, N \end{cases}, \quad (11.2)$$

The iteration index of the supervising parameters were omitted for clarity. Substitut-

ing (11.2) into (11.1) yields

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \eta_1 \mu_1 \mathbf{u}_i^T e(i) + \sum_{n=2}^N \eta_n \mu_n \prod_{\ell=1}^{n-1} (1 - \eta_\ell \mu_\ell \|\mathbf{u}_i\|^2) \mathbf{u}_i^T e(i),$$

which can be rearranged in the clearer form

$$\mathbf{w}_i = \mathbf{w}_{i-1} + (\bar{\mu} + \mu') \mathbf{u}_i^T e(i) \quad (11.3a)$$

$$\bar{\mu} = \sum_{n=1}^N \eta_n \mu_n \quad (11.3b)$$

$$\mu' = \sum_{\ell=2}^N (-1)^{\ell-1} \|\mathbf{u}_i\|^{2(\ell-1)} \left[\sum_{\ell} \left(\{\eta_n \mu_n\} \right) \right], \quad \text{for } N \geq 2 \quad (11.3c)$$

where, by abuse of notation, $(\{\eta_n \mu_n\})$ represents the set of $\binom{N}{\ell}$ products of ℓ -combinations of $\{\eta_n \mu_n\}$. For instance, when $N = 3$ and $\ell = 2$ the set contains the elements $\eta_1 \eta_2 \mu_1 \mu_2$, $\eta_1 \eta_3 \mu_1 \mu_3$, and $\eta_2 \eta_3 \mu_2 \mu_3$ twice and the $\eta_n^2 \mu_n^2$ for $n = 1, 2, 3$. Notice that the iteration index of $\bar{\mu}$ and μ' were also omitted for conciseness. When $N = 1$, take $\mu' = 0$ instead of (11.3c).

Notice that (11.3) is algebraically similar to a VSS algorithm. In fact, taking $\mu' = 0$ in (11.3a) recovers the recursion of a parallel combination with coefficients feedback (see Chapter 8). Any effect the incremental topology has on performance can then be assigned to μ' , which exploit higher-order moments of the input data [see (11.3c)].

11.2 Steady-state performance

11.2.1 Energy conversation relation

As for stand-alone AFs and parallel combinations with coefficients feedback, an ECR can be derived for ring incremental topologies. This is a strong indication that there is a relation between the conservation of error energy and strongly connected topologies. To obtain an ECR for the $\{\text{LMS}\}^N$, subtract (11.3) from $\mathbf{w}^o$ to get¹

$$\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_{i-1} - (\bar{\mu} + \mu') \mathbf{u}_i^T e(i), \quad (11.4)$$

which multiplied by $\mathbf{u}_i$ gives

$$e_p(i) = e_a(i) - (\bar{\mu} + \mu') \|\mathbf{u}_i\|^2 e(i). \quad (11.5)$$

¹Since only steady-state performance is analyzed in this section, assume $\mathbf{q}_i = \mathbf{0}$.

Assuming $\mathbf{u}_i \neq \mathbf{0}$, (11.2) yields

$$e(i) = \frac{1}{(\bar{\mu} + \mu') \|\mathbf{u}_i\|^2} [e_p(i) - e_a(i)],$$

which together with the squared Euclidian norm of (11.4) gives the ECR

$$\|\tilde{\mathbf{w}}_i\|^2 + \frac{1}{\|\mathbf{u}_i\|^2} |e_a(i)|^2 = \|\tilde{\mathbf{w}}_{i-1}\|^2 + \frac{1}{\|\mathbf{u}_i\|^2} |e_p(i)|^2. \quad (11.6)$$

Once again, for $\mathbf{u}_i = \mathbf{0}$, this relation reduces to the trivial $\|\tilde{\mathbf{w}}_i\|^2 = \|\tilde{\mathbf{w}}_{i-1}\|^2$. Notice that (11.6) is indeed identical to (2.14) and (10.16), although the error quantities have different definitions.

11.2.2 Mean-square analysis

To determine the steady-state value of the EMSE of the $\{\text{LMS}\}^N$, first evaluate the expected value of the squared Euclidian norm of (11.4), as in

$$\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 = \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 - 2 \mathbb{E}(\bar{\mu} + \mu') e_a(i) e(i) + \mathbb{E}(\bar{\mu} + \mu')^2 \|\mathbf{u}_i\|^2 e^2(i), \quad (11.7)$$

Then, assuming the combination converges, one has $\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 = \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2$ in (11.7) as $i \rightarrow \infty$, which leads to the variance relation

$$\mathbb{E}(\bar{\mu} + \mu')^2 \|\mathbf{u}_i\|^2 e^2(i) = 2 \mathbb{E}(\bar{\mu} + \mu') e_a(i) e(i), \quad \text{as } i \rightarrow \infty. \quad (11.8)$$

Before proceeding, the following assumption is necessary to render the derivations tractable:

A.13 (*Data separation principle*) At steady-state, $\|\mathbf{u}_i\|^{2\ell}$ is independent of $|e_a(i)|^2$ for $1 \leq \ell \leq N$.

Assumption A.13 is more restrictive than the one usually adopted in the analysis of stand-alone AFs, which only assumes independence for $\ell = 1$ [18]. Here, however, the higher order moments in μ' require a more stringent hypothesis. As is the case for the original separation principle, results based on A.13 are expected to be accurate only when adaptation is slow. Indeed, Figure 53 illustrates the accuracy of A.13 for different step sizes and moment orders. As the number of component filters (N and, consequently, ℓ) grows, the range of step sizes ($\eta_n \mu_n$) for which the assumption is valid decreases (as shown in Figure 54).

From the data model in Section 1.6.1, one has $e(i) = e_a(i) + v(i)$, where the noise is

Figure 53: Separation principle A.13. Comparison between $E \|\mathbf{u}_i\|^2 e_a^2(i)$ and $E \|\mathbf{u}_i\|^2 E e_a^2(i)$ in an $\{\text{LMS}\}^5$ combination with white input data, $M = 10$, $\sigma_u^2 = 1$, and $\sigma_v^2 = 10^{-3}$.

considered independent of the input $u(i)$ as per A.1. Hence, (11.8) becomes, under A.13,

$$\zeta^{\{\text{LMS}\}^N} = \frac{E [(\bar{\mu} + \mu')^2 \|\mathbf{u}_i\|^2] \sigma_v^2}{E [2(\bar{\mu} + \mu') - (\bar{\mu} + \mu')^2 \|\mathbf{u}_i\|^2]}, \quad (11.9)$$

where the value of the expressions in square brackets depend only on even moments of the input signal $u(i)$. Indeed, defining $m_u^{(p)} = E \|\mathbf{u}_i\|^{2p}$ and expanding (11.9) yields

$$\zeta^{\{\text{LMS}\}^N} = \frac{[\bar{\mu}^2 \text{Tr}(\mathbf{R}_u) + 2\bar{\mu}\mathcal{M}_2 + \mathcal{M}_3] \sigma_v^2}{2(\bar{\mu} + \mathcal{M}_1) - [\bar{\mu}^2 \text{Tr}(\mathbf{R}_u) + 2\bar{\mu}\mathcal{M}_2 + \mathcal{M}_3]} \quad (11.10)$$

$$\begin{aligned}
\mathcal{M}_1 &= \mathbb{E} \mu' = \sum_{\ell=2}^N (-1)^{\ell-1} m_u^{(\ell-1)} \left[\sum \binom{\{\eta_n \mu_n\}}{\ell} \right] \\
\mathcal{M}_2 &= \mathbb{E} \mu' \|\mathbf{u}_i\|^2 = \sum_{\ell=2}^N (-1)^{\ell-1} m_u^{(\ell)} \left[\sum \binom{\{\eta_n \mu_n\}}{\ell} \right] \\
\mathcal{M}_3 &= \mathbb{E} \mu'^2 \|\mathbf{u}_i\|^2 = \sum_{\ell=2}^N m_u^{(2\ell-1)} \left[\sum \binom{\{\eta_n \mu_n\}}{\ell} \right]^2 \\
&\quad + \sum_{q=2}^N \sum_{r=2}^N (-1)^{q+r} m_u^{(q+r-1)} \left[\sum \binom{\{\eta_n \mu_n\}}{q} \right] \left[\sum \binom{\{\eta_n \mu_n\}}{r} \right]
\end{aligned}$$

Notice that, when $\mathcal{M}_1 = \mathcal{M}_2 = \mathcal{M}_3 = 0$, (11.10) takes the form of the steady-state EMSE expression for a stand-alone LMS filter with step size $\bar{\mu}$. Hence, the $\mathcal{M}$ values are misadjustment terms that determine the impact of having more than one AF in the incremental chain.

Although (11.10) is already in closed form, determining the $m_u^{(p)}$ analytically can be intricate, specially for large p . Nevertheless, (11.10) can be simplified assuming the step sizes are small:

A.14 (*Small step sizes*) The step size $\{\mu_n\}$ are small enough that $m_u^{(p)}(\eta_n \mu_n)^\ell \approx 0$ for any p and $\ell > 2$.

As the number of filters in the combination grows, A.14 needs to be valid for larger p , which requires even smaller net step sizes $\{\eta_n \mu_n\}$.

Under A.14, the misadjustment terms reduce to

$$\begin{aligned}
\mathcal{M}_1 &= -m_u^{(1)} \left[\sum \binom{\{\eta_n \mu_n\}}{2} \right] \\
\mathcal{M}_2 &= -m_u^{(2)} \left[\sum \binom{\{\eta_n \mu_n\}}{2} \right] \\
\mathcal{M}_3 &= 0
\end{aligned}$$

which only depend on second and fourth moments of the input. For the special case of Gaussian data, $\eta_n \mu_n = \mu$, and small step size, (11.10) becomes:

$$\zeta^{\{\text{LMS}\}^N} = \frac{[N\mu \text{Tr}(\mathbf{\Lambda}) - \mathcal{M}_0] \sigma_v^2}{2[1 - \mathcal{K}/N\mu \text{Tr}(\mathbf{\Lambda})] - [N\mu \text{Tr}(\mathbf{\Lambda}) - \mathcal{M}_0]} \quad (11.11)$$

where $\mathcal{M}_0 = 2[\mathbf{\Lambda} \text{Tr}(\mathbf{\Lambda}) + 2\mathbf{\Lambda}^2] \mathcal{K} \mu^2$, $\mathbf{\Lambda}$ is the eigenvalue matrix of $\mathbf{R}_u = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$, and $\mathcal{K} = \binom{N}{2}$, the binomial coefficient, for $N \geq 2$, and $\mathcal{K} = 0$, for $N = 1$.

This last expression makes the effect of the incremental topology on the steady-state

Figure 54: Steady-state model validation. **White input data:** $M = 10$, $\sigma_u^2 = 1$, and $\sigma_v^2 = 10^{-3}$. Higher-order moments of $u(i)$ for the complete model (11.10) were evaluated using Monte Carlo simulations with 10^6 realizations.

error very clear. Indeed, as before, (11.11) for $N = 1$ reduces to the well-known expression for the steady-state EMSE of a stand-alone LMS filter with step size μ . As the number of component filters (N) grows, so does $\mathcal{M}_0$ and, consequently, the misadjustment. Therefore, increasing the length of an AF chain degrades its steady-state behavior. Nevertheless, observations advocate that incremental combinations with more AFs display a faster convergence rate, which suggests a trade-off similar to that in stand-alone adaptive algorithms. This effect, which motivated the parallel-incremental topology in Section 7.2, is quantified in (11.11). Lastly, notice that as $\mathcal{M}_0$ become larger, it will dominate both the numerator and the denominator of (11.11), at which point its impact on the value of the EMSE will continuously decrease. Hence, the effect on the misadjustment of adding filters to the combination becomes less expressive as N grows.

These observations are illustrated in Figure 54 which compares numerical results to those from (11.10) and (11.11). The moments of $u(i)$ required in $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ were evaluated using Monte Carlo simulations with 10^{-6} realizations. Notice that, as N grows, the range of step sizes for which the models are valid decrease, to the point where (11.11) breaks down (yields negative EMSEs). The divergence of (11.10) for larger N is mostly due to errors in the estimation of moment with higher order ($p > 10$).

11.3 Transient performance

11.3.1 Mean convergence analysis

Before proceeding with the derivations of the mean global coefficients error recursion, consider the following assumption commonly used in the transient analysis of stand-alone AFs [18, 19]. This same assumption was successfully used in the analysis of the parallel combination in Chapter 10.

A.15 (*Data independence assumptions*) $\{\mathbf{u}_i\}$ constitutes an i.i.d. sequence of vectors independent of $v(j)$ for all i, j . Consequently, $\{\mathbf{u}_i, \tilde{\mathbf{w}}_j\}$, $\{d(i), d(j)\}$, and $\{\mathbf{u}_i, d(j)\}$ are independent for $i > j$.

To obtain the desired recursion, take the expected value of the coefficients error equation (11.4) considering the data model from Section 1.6.1 and A.15. Explicitly,

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_i &= \mathbb{E} \tilde{\mathbf{w}}_{i-1} - \mathbb{E}(\bar{\mu} + \mu') \mathbf{u}_i^T (\mathbf{u}_i \tilde{\mathbf{w}}_{i-1}) \\ &= \mathbb{E} \tilde{\mathbf{w}}_{i-1} - (\bar{\mu} \mathbf{R}_{\mathbf{u}} + \mathbb{E} \mu' \mathbf{u}_i^T \mathbf{u}_i) \mathbb{E} \tilde{\mathbf{w}}_{i-1} \\ &= [\mathbf{I} - \bar{\mu} \mathbf{R}_{\mathbf{u}} - \mathbb{E} \mathbf{u}_i^T \mu' \mathbf{u}_i] \mathbb{E} \tilde{\mathbf{w}}_{i-1}, \end{aligned}$$

Then, use the definition of μ' from (11.3c) to obtain

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_i &= [\mathbf{I} - \mathbb{E} \bar{\mu} \mathbf{R}_{\mathbf{u}} + \mathcal{M}] \mathbb{E} \tilde{\mathbf{w}}_{i-1} \\ \mathcal{M} &= \sum_{\ell=2}^N \mathbb{E} (-\mathbf{u}_i^T \mathbf{u}_i)^\ell \left[\sum \binom{\{\eta_n \mu_n\}}{\ell} \right]. \end{aligned} \quad (11.12)$$

Once again, although (11.12) is a closed-form recursion, it involves higher-order moments of $\mathbf{u}_i$ which can be challenging to compute for large N . Adopting the same small step sizes assumption as in the steady-state derivations (A.14), the terms for $\ell > 2$ can be eliminated to approximate (11.12) using

$$\mathcal{M}' = \sum_{m \neq n} \mu_m \mu_n \eta_m \eta_n \mathbb{E} u_i^T u_i u_i^T u_i. \quad (11.13)$$

For Gaussian inputs, (11.13) can be evaluated analytically using the same transformation as in Section 10.3:

$$\begin{aligned} \mathbb{E} \tilde{\mathbf{w}}_i &= \mathbf{A} \cdot \mathbb{E} \tilde{\mathbf{w}}_{i-1} \\ \mathbf{A} &= \mathbf{I} - \bar{\mu} \mathbf{\Lambda} + \sum_{m \neq n} \mu_m \mu_n \eta_m \eta_n [\mathbf{\Lambda} \text{Tr}(\mathbf{\Lambda}) + 2\mathbf{\Lambda}^2], \end{aligned} \quad (11.14)$$

Figure 55: Mean convergence analysis of an $\{\text{LMS}\}^N$. **White input data:** $M = 50$, $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, and $\eta_n = 1$ (unsupervised). The higher-order moments in the complete model (11.12) were evaluate using Monte Carlo simulations with 10^4 realizations.

where $\bar{\mathbf{w}}_i = \mathbf{Q}\tilde{\mathbf{w}}_i$ and $\mathbf{Q}$ and $\mathbf{\Lambda}$ come from the EVD $\mathbf{R}_u = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. For $N = 2$, as in [41,43], this recursion is exact. Figure 55 shows that the approximation in (11.14) (red curves) is accurate even for a large number of components as long as the step sizes are kept small enough. On the other hand, (11.12) remains valid.

Comparing (11.14) [or (11.12)] with the mean coefficients error recursion for a single LMS with step size μ , explicitly [18]

$$\mathbf{E} \bar{\mathbf{w}}_i = [\mathbf{I} - \mu\mathbf{\Lambda}] \mathbf{E} \bar{\mathbf{w}}_i, \quad (11.15)$$

it is clear that the incremental combination is able to improve the convergence rate by operating as though a larger step size was used ($\bar{\mu}$). However, the reprocessing of the same data pair yields ever smaller gains in convergence speed, so that the net effect of the $\{\text{LMS}\}^N$ combination is not equivalent to a single LMS filter with step size $\bar{\mu}$. This difference is quantified by $\mathcal{M}$ (or approximated by $\mathcal{M}'$), which depends on higher-order moments of the input and on the component filters step sizes.

Finally, notice from (11.14) that the mean transient performance depends directly on the eigenvalues of $\mathbf{R}_u$, in contrast to the mean coefficients error recursion of the NLMS or the APA where $\mathbf{\Lambda}$ appears normalized by some data nonlinearity which reduces the eigenvalue spread [18,19,61]. Correlated inputs can therefore impair the convergence rate of the incremental combination, which motivates the use of data buffering techniques (see Chapter 9) [45,47,73].

11.3.2 Unbiasedness constraint

When discussing incremental supervising rules (Section 6.1), it was suggested that imposing a convexity constraint on the supervising parameters of incremental combinations is unwarranted. In fact, it can hinder the capability of the combination to exploit the component filters and degrade the overall convergence rate [44, 45]. The embedding of this restriction in [41, 43] was inspired by its use in parallel topologies, where it guarantees that the combination preserves the unbiasedness of the component filters (see Section 5.1) [33, 35, 90]. Using the results from last section, the following derivations evaluate the unbiasedness constraint of the $\{\text{LMS}\}^N$ and show that it is not equivalent to the convexity constraint of parallel combinations. For tractability, the expression for Gaussian inputs in (11.14) is considered.

To guarantee that the mean overall coefficients error vanishes as $i \rightarrow \infty$, suffices that the eigenvalues of $\mathbf{A}$ in (11.14) have magnitudes smaller than one [18, 19]. Indeed, given that $\mathbf{Q}$ is an isometry, $\bar{\mathbf{w}}_i \rightarrow \mathbf{0} \Leftrightarrow \tilde{\mathbf{w}}_i \rightarrow \mathbf{0}$ [154]. Before proceeding, assuming that the net step size of all filters are the same, i.e., that $\eta_n \mu_n = \eta \mu$ for all $n = 1, \dots, N$. It will be shown momentarily (Section 11.3.4) that this condition is necessary to achieve the maximum convergence rate. Under this condition, the unbiasedness constraint of the $\{\text{LMS}\}^N$ can be obtained from $\mathbf{A}$ as

$$\left| 1 - N(\eta\mu)\lambda_m + \binom{N}{2}(\eta\mu)^2 [\lambda_m \text{Tr}(\mathbf{\Lambda}) + 2\lambda_m^2] \right| < 1, \quad \text{for all } m = 1, \dots, M. \quad (11.16)$$

This expression can be simplified by noticing that for $N > 1$, the absolute value in (11.16) is unnecessary because its argument is positive for any value of $\eta\mu$. Indeed, its discriminant is

$$\begin{aligned} \Delta &= N^2\lambda_m^2 - 4\binom{N}{2} [\lambda_m \text{Tr}(\mathbf{\Lambda}) + 2\lambda_m^2] \\ &= N^2\lambda_m^2 - 2N(N-1) [\lambda_m \text{Tr}(\mathbf{\Lambda}) + 2\lambda_m^2] \\ &= -(3N+4)N\lambda_m^2 - 2N(N-1)\lambda_m \text{Tr}(\mathbf{\Lambda}) < 0, \end{aligned}$$

since $N \in \mathbb{N}$, $N > 1$, $\text{Tr}(\mathbf{\Lambda}) = \sum \lambda_m$, and $\lambda_m > 0$, assuming the covariance matrix of the input data is full rank. If not, then (11.16) becomes non-negative instead of positive.

Hence, the unbiasedness condition (11.16) is equivalent to

$$\binom{N}{2}(\eta\mu)^2 [\lambda_m \text{Tr}(\mathbf{\Lambda}) + 2\lambda_m^2] - N(\eta\mu)\lambda_m < 0. \quad (11.17)$$

To solve (11.17), notice that it can only be satisfied if $\eta\mu > 0$, in which case some

straightforward manipulation leads to

$$0 < \eta\mu < \frac{2}{(N-1)[\text{Tr}(\mathbf{R}_u) + 2\lambda_{\max}]}, \quad \text{for } N \geq 2, \quad (11.18)$$

where $\lambda_{\max} = \max_m \lambda_m$ is the largest eigenvalue of $\mathbf{R}_u$. What is more, if the net step sizes are different, (11.18) is still valid taking $\eta\mu = \max_n \eta_n \mu_n$.

The mean stability condition in (11.18) describes effects similar to those in stand-alone AFs. Whenever the length of the filter (M) or the variance of the input signal (σ_u^2) grows, the equivalent step size $\eta\mu$ has to decrease in order to maintain the combination stable. Moreover, when more component filters are used (large N), their step sizes must also be smaller. Finally, notice that for $N = 2$, (11.18) becomes very similar to the mean-square stability convergence of a stand-alone LMS filter with step size $\eta\mu$. Indeed, for the white input case $\mathbf{R}_u = \sigma_u^2 \mathbf{I}$, (11.18) reduces to

$$0 < \eta\mu < \frac{2}{(M+2)\sigma_u^2}, \quad (11.19)$$

which is identical to the mean-square convergence condition for the LMS filter in [18, Thm. 23.2].

11.3.3 Mean-square convergence analysis

Mean-square transient analysis are challenging for large incremental combinations due to the accumulated effect of several AFs. Hence, this section presents derivations for $N = 2$ LMS filters (as in [41, 43]). In this case, the quantities from (11.3) become $\mu' = \eta_1 \mu_1 \eta_2 \mu_2$ and $\bar{\mu} = \eta_1 \mu_1 + \eta_2 \mu_2$. As in the previous section, adopt A.14. Furthermore, consider the input data $u(i)$ to be i.i.d. ($\mathbf{R}_u = \sigma_u^2 \mathbf{I}$).

Note that under these assumptions, the MSD and EMSE are expressed as

$$\begin{aligned} \text{MSD}(i) &= \text{Tr}(\mathbf{K}_{i-1}) \\ \text{EMSE}(i) &= \zeta(i) = \sigma_u^2 \cdot \text{MSD}(i), \end{aligned} \quad (11.20)$$

where, once again, the coefficients error covariance matrix is denoted by $\mathbf{K}_i = \text{E} \tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^T$. Then, to obtain a transient performance model for the $\{\text{LMS}\}^2$, all that is required is a recursion for $\mathbf{K}_i$. This recursion can be obtained from (11.4) as

$$\begin{aligned} \mathbf{K}_i &= \mathbf{K}_{i-1} - \text{E} [\bar{\mu} - \mu' \|\mathbf{u}_i\|^2] [\tilde{\mathbf{w}}_{i-1} \mathbf{u}_i + \mathbf{u}_i^T \tilde{\mathbf{w}}_{i-1}^T] e(i) \\ &\quad + \text{E} [\bar{\mu} - \mu' \|\mathbf{u}_i\|^2]^2 \mathbf{u}_i^T \mathbf{u}_i \|e(i)\|^2. \end{aligned} \quad (11.21)$$

Employing the relation $e(i) = e_a(i) + v(i)$, all terms linearly dependent on $v(i)$ vanish due to A.1, yielding

$$\begin{aligned}
\mathbf{K}_i = & \mathbf{K}_{i-1} - \bar{\mu} \mathbb{E} [\tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T \mathbf{u}_i^T \mathbf{u}_i + \mathbf{u}_i^T \mathbf{u}_i \tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T] \\
& + \mu' \mathbb{E} [\tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i + \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T] \\
& + \bar{\mu}^2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T \mathbf{u}_i^T \mathbf{u}_i \\
& - 2\bar{\mu}\mu' \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T \mathbf{u}_i^T \mathbf{u}_i \\
& + (\mu')^2 \mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \tilde{\mathbf{w}}_{i-1} \tilde{\mathbf{w}}_{i-1}^T \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \\
& + \mathbb{E} [\bar{\mu}^2 \mathbf{u}_i^T \mathbf{u}_i - 2\bar{\mu}\mu' \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i + (\mu')^2 \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i] \sigma_v^2.
\end{aligned} \tag{11.22}$$

Although the data independence assumption A.15 will render (11.22) more tractable by separating the expectations, higher-order moments of $\mathbf{u}_i$ still need to be evaluated. This can be done in closed form by assuming that $\mathbf{u}_i$ is a Gaussian vector. Therefore, under the initial i.i.d. input assumption one gets

$$\begin{aligned}
\mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{W} \mathbf{u}_i^T \mathbf{u}_i &= \sigma_u^4 [\text{Tr}(\mathbf{W}) \mathbf{I} + 2\mathbf{W}] \\
\mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \mathbf{W} \mathbf{u}_i^T \mathbf{u}_i &= (M+4) \sigma_u^6 [\text{Tr}(\mathbf{W}) \mathbf{I} + 2\mathbf{W}] \\
\mathbb{E} \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i \mathbf{W} \mathbf{u}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{u}_i &= (M+4)(M+6) \sigma_u^8 [\text{Tr}(\mathbf{W}) \mathbf{I} + 2\mathbf{W}],
\end{aligned} \tag{11.23}$$

where $\mathbf{W}$ is some deterministic symmetric matrix. These expressions are obtained by directly evaluating the vector products, applying Isserlis' theorem [155] element-by-element, and rearranging the results in matrix form. A detailed derivation of the fourth-order moment is found in [18].

Using (11.23), the trace of (11.22), under A.15, evaluates to

$$\text{MSD}(i+1) = A \cdot \text{MSD}(i) + b \cdot M \sigma_u^2 \sigma_v^2, \tag{11.24}$$

$$\begin{aligned}
A &= 1 - 2\bar{\mu}\sigma_u^2 + (\bar{\mu}^2 + 2\mu')(M+2)\sigma_u^4 \\
&\quad - 2\bar{\mu}\mu'(M+4)(M+2)\sigma_u^6 \\
&\quad + (\mu')^2(M+6)(M+4)(M+2)\sigma_u^8 \\
b &= \bar{\mu}^2 - 2\bar{\mu}\mu'(M+2)\sigma_u^2 \\
&\quad + (\mu')^2(M+4)(M+2)\sigma_u^4
\end{aligned}$$

with $\text{MSD}(0) = \|\mathbf{w}^o\|^2$ assuming $\mathbf{w}_{-1} = 0$.

This expression can be further simplified assuming the step sizes are small enough. Explicitly, assuming that $[\mu_1 \text{Tr}(\mathbf{R}_u)]^\ell \approx 0$ for $\ell > 2$, taking $\mu_1 \geq \mu_2$ without loss of

Figure 56: Incremental mean-square analysis of a $\{\text{LMS}\}^2$. **White input data:** $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\mu_1 = 0.005$, and $\mu_2 = 0.003$. (a) **Gaussian regressor** (11.24). **No constraint:** $\{1 + \exp[0.05(i - 150)]\}^{-1}$ (deterministic supervisor). (b) **Small step sizes** (11.25). **Convexly constrained:** $\{1 + \exp[0.05(i - 1100)]\}^{-1}$ (deterministic supervisor); **no constraint:** $\{1 + \exp[0.05(i - 700)]\}^{-1}$ (deterministic supervisor).

generality. Then, A and b in (11.24) can be replaced by

$$\begin{aligned} A' &= 1 - 2\bar{\mu}\sigma_u^2 + (M + 2)(\bar{\mu}^2 + 2\mu')\sigma_u^4 \\ b' &= \bar{\mu}^2. \end{aligned} \quad (11.25)$$

These models are compared to numerical simulations in Figure 56. They use a deterministic choice of supervising parameters $\eta(i)$, with convexity constraint ($\eta_1 = \eta$; $\eta_2 = 1 - \eta$) and without ($\eta_1 = \eta$; $\eta_2 = 1$). Notice that both (11.24) (Figure 56a) and (11.25) (Figure 56b) are accurate models of the transient behavior of $\{\text{LMS}\}^2$ combinations. In the former, only the “unconstrained” combination is shown because both curves overlap. Notice that, taking $\mu' = 0$ in (11.25) yields once more the mean-square expression from the transient analysis of a stand-alone LMS filter with step size $\bar{\mu}$ [18]. Hence, μ' represents a convergence rate loss due to reusing the same data.

11.3.4 Optimal supervisor

Using (11.24) and (11.25), it is possible to derive the sequence of supervising parameters $\{\eta_1(i), \eta_2(i)\}$ that results in the fastest local convergence. This is achieved by solving

$$\nabla_{\eta_1, \eta_2} \text{MSD}(i + 1) = 0, \quad (11.26)$$

since the expression for $\text{MSD}(i+1)$ is convex for a given $\text{MSD}(i)$. Expanding (11.26) yields

$$\begin{aligned}\frac{\partial}{\partial \eta_1} \text{MSD}(i+1) &= 2\sigma_u^2 \mu_1 [A_2 \cdot \text{MSD}(i) + b_2 \cdot M\sigma_v^2] = 0 \\ \frac{\partial}{\partial \eta_2} \text{MSD}(i+1) &= 2\sigma_u^2 \mu_2 [A_1 \cdot \text{MSD}(i) + b_1 \cdot M\sigma_v^2] = 0,\end{aligned}\tag{11.27}$$

with

$$\begin{aligned}A_n &= (M+2)\sigma_u^2 \left\{ \mu'(M+4)\sigma_u^2 [\eta_n \mu_n (M+6)\sigma_u^2 - 1] \right. \\ &\quad \left. + \eta_n \mu_n [1 - \bar{\mu}(M+4)\sigma_u^2] + \bar{\mu} \right\} - 1 \\ b_n &= (M+2)\sigma_u^2 \left\{ \eta_n \mu_n [\mu'(M+4)\sigma_u^2 - \bar{\mu}] - \mu' \right\} + \bar{\mu}\end{aligned}$$

for the Gaussian regressor model (11.24) and

$$\begin{aligned}A'_n &= (\bar{\mu} + \eta_n \mu_n)(M+2)\sigma_u^2 - 1 \\ b'_n &= \bar{\mu}\end{aligned}$$

for the small step sizes model (11.25). Due to the symmetry of the equations in (11.27), they can only be simultaneously satisfied if $\eta_1 \mu_1 = \eta_2 \mu_2$. Thus, the maximum local convergence rate is achieved when all component filters operate with the same net step size. Notice that this is the case of the DR-LMS adaptive algorithm. Using an incremental combination interpretation of this data reusing AF, it is therefore possible to show that by choosing a single value of μ for all its iterations over k [see (2.10)], it is in fact promoting a faster convergence.

Given that the closed form expression for the $\eta_n(i)$ in the Gaussian regressor case is intricate and the small step sizes approximation leads to accurate results (see Figure 56), the derivations proceed using A'_n and b'_n , leading to

$$\eta_n^o(i) \mu_n = \frac{\text{MSD}(i)}{3(M+2)\sigma_u^2 \text{MSD}(i) + 2M\sigma_v^2}, \quad n = 1, 2.\tag{11.28}$$

The result of using $\eta_n^o(i)$ in an incremental combination is illustrated in Figure 57. Note that (11.28) only serves analytical purposes and cannot be evaluated exactly in practice since it requires the knowledge of $\mathbf{w}^o$. Nevertheless, it can be used to limit the value of the incremental supervising parameter as in Section 6.1.

The result in (11.28) is fundamental to the argument in favor of unsupervised incremental combinations. Indeed, it bears an extreme resemblance to the expression for the optimal step size of a stand-alone LMS filter, which for white input signals reduces

Figure 57: Optimal supervisor for the incremental combination ($\{\text{LMS}\}^2$). **White input data:** $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\mu_1 = 0.005$, and $\mu_2 = 0.003$.

to [143]

$$\mu(i) = \frac{\text{MSD}(i)}{(M+2)\sigma_u^2 \text{MSD}(i) + M\sigma_v^2}. \quad (11.29)$$

Hence, the argument that a supervised incremental combination (at least in its current form) can be restated as an unsupervised combination of VSS algorithms is substantiated. Instead of designing more complex supervising rules for the incremental combination, it may therefore be more effective to exploit hybrid parallel-incremental topologies as in Chapter 7.

11.4 Concluding remarks

The performance analysis of incremental combinations is intricate due to the coupling of the component filters. Therefore, open areas of research remain, such as the convergence behavior of arbitrarily sized chains and analyses of DR combinations. Nevertheless, the results from this chapter bring a considerable advancement to their understanding. First, the misadjustment degradation effect can now be quantified and has been shown to depend, not only on the component filters step sizes, but also on the length of the incremental chain. Second, mean convergence analysis has shown that the use of convex/affine constraints on the supervising parameters can limit the achievable performance of the combination. Moreover, the unbiasedness constraint for the incremental topology bears a striking resemblance to mean-square stability results for stand-alone AFs. Finally, although limited to two filters, the mean-square transient analysis presented provides an expression for the value of the optimal supervising parameter at each iteration. Interestingly, in order to achieve the maximum convergence rate the net step sizes ($\eta_n \mu_n$) of all

component filters must be the same, a condition that is met by the DR-LMS stand-alone AF. A justification can now be provided for this natural choice.

12 CONCLUSION

*Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.*¹

-- Blaise Pascal, "Lettres Provinciales", 1657.

This dissertation set out to develop a more general and formal framework to study combinations of AFs and, though open questions remain, it has made significant developments towards this goal. Firstly, it proposed a formal definition of combinations of AFs, illustrating how this could be used to describe different structures in a unified manner (Chapter 3). To do so, it put forward a network-inspired view of combinations by means of graph-theoretic arguments. As a result, it built relations between combinations of AFs and both ANs and stand-alone adaptive algorithms.

This work then provided an extensive literature review (Chapter 4) that was used to support the novel general description from Chapter 3, create a taxonomy of combinations, and guide the analyses and investigations of the following chapters. In the sequel, motivated by the network approach to combinations of AFs, parallel (Chapter 5), incremental (Chapter 6), and hybrid (Chapter 7) topologies were detailed, along with their existing supervising rules. Simulations were used to illustrate the effect of the topology on performance as well as evidence the advantages of different supervisors.

Before proceeding performance analyses, two novel applications of combination of AFs were presented and illustrated. One, used combinations to understand and improve stand-alone AFs, proposing improved and robust VSS adaptive algorithms (Chapter 8). The other was related to designing low complexity combinations that could match or outperform established AFs, such as the APA (Chapter 9). Results in the latter lead to the introduction of the counter-intuitive concept of *combinations as a complexity-reduction technique*.

Finally, mean and mean-square analyses of parallel and incremental combinations were derived. Their results quantified and explained several effect observed in simulations throughout this work, such as the improved convergence rate and misadjustment degradation of incremental structures and the robustness and tracking gains of parallel combinations with coefficients feedback.

¹I have made this one longer than usual only because I have not had time to make it shorter.

13 SUGGESTIONS FOR FUTURE WORK

“Heavier-than-air flying machines are impossible.”

-- William Thomson (Lord Kelvin), 1895.

“There is not the slightest indication that nuclear energy will ever be obtainable.”

-- Albert Einstein, 1932.

“I think there is a world market for about four or five electronic computers.”

-- Thomas J. Watson, CEO of IBM, 1943.

This dissertation developed several aspects of combinations of AFs under a more general and formal framework. This approach has lead to improvements in their performance, novel applications, and conceptual and theoretical results. Nevertheless, a variety of open issues remain to be further investigated. The following list provides possible avenues for future research.

- *Extend the performance analysis of incremental combinations:* tracking analysis, transient performance of incremental chains of arbitrary length, improvements in the mean-square analysis results for larger N and μ_n , new analyses accounting for different adaptive algorithms, study of data buffering combinations;
- *Find relations between combinations of AFs and other algorithms:* Dichotomous Coordinate Descent (DCD) as a combination, relations to low rank and interpolated FIR (IFIR) AFs;
- *Extend combinations of AFs concepts to ANs (and vice-versa):* coefficients feedback, supervising rules, new topology results;
- *Reducing the complexity of combinations:* conditional component update, resources-dependent performance, data selective operation;
- *Study of combination topologies:* topology design, dynamical topologies, influence on supervising rule;
- *Unified analysis using $\mathcal{FOB}$ descriptions:* unified analysis of combinations with arbitrary topologies, component filters, supervising rules;

- *Combination as a complexity-reduction technique*: new combinations to compete with RLS, FTF;
- *Extend results to unsupervised and IIR AFs*.

REFERENCES

- [1] B. Widrow and M.E. Hoff Jr., “Adaptive switching circuits,” in *IRE WESCON Conference*, 1960, pp. 96–104.
- [2] J. Arenas-García, M. Martínez-Ramón, A. Navia-Vázquez, and A.R. Figueiras-Vidal, “Plant identification via adaptive combination of transversal filters,” *Signal Process.*, vol. 86[9], pp. 2430–2438, 2006.
- [3] C.G. Lopes and A.H. Sayed, “Incremental adaptive strategies over distributed networks,” *IEEE Trans. Signal Process.*, vol. 55[8], pp. 4064–4077, 2007.
- [4] C.G. Lopes and A.H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Trans. Signal Process.*, vol. 56[7], pp. 4064–4077, 2007.
- [5] V. Krishnamurthy and H.V. Poor, “Social learning and Bayesian games in multiagent signal processing: how do local and global decision makers interact?,” *IEEE Signal Processing Magazine*, vol. 30[3], pp. 43–57, 2013.
- [6] A.H. Sayed, *Adaptation, Learning, and Optimization over Networks*, NOW Publishers, 2014.
- [7] C.G. Lopes, L.F.O. Chamon, and V.H. Nascimento, “Towards spatially universal adaptive networks,” in *Global Conference on Signal and Information Processing (GlobalSip)*, 2014, (Invited paper).
- [8] S.M. Kay, *Fundamentals Of Statistical Signal Processing: Estimation theory*, vol. 1, Addison Wesley Longman, 2001.
- [9] S.M. Kay, *Fundamentals Of Statistical Signal Processing: Detection theory*, vol. 2, Addison Wesley Longman, 2001.
- [10] N. Merhav and M. Feder, “Universal prediction,” *IEEE Trans. Inf. Theory*, vol. 44[6], pp. 2124–2147, 1998.
- [11] N. Cesa-Bianchi and G. Lugosi, “On prediction of individual sequences,” *Annals of Statistics*, vol. 27[6], pp. 1865–1895, 1999.
- [12] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*, Cambridge University Press, 2006.
- [13] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013.
- [14] M. Kearns and L.G. Valiant, “Learning Boolean formulae or finite automata is as hard as factoring,” Tech. Rep., Harvard University Aiken Computation Laboratory, 1988.

- [15] M. Kearns and L.G. Valiant, “Cryptographic limitations on learning Boolean formulae and finite automata,” in *ACM Symposium on Theory of Computing*, 1989, pp. 433–444.
- [16] R.E. Schapire, “The strength of weak learnability,” *Machine Learning*, vol. 5, pp. 197–227, 1990.
- [17] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems?,” *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.
- [18] A.H. Sayed, *Adaptive filters*, Wiley-IEEE Press, 2008.
- [19] P.S.R. Diniz, *Adaptive filtering: Algorithms and practical implementation*, Springer, 3rd edition, 2008.
- [20] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 4 edition, 2001.
- [21] M.C. Tsakiris and C.G. Lopes, “A robust affine projection algorithm with feedback compensation of the condition number,” in *International Conference in Acoustic, Speech, and Signal Processing*, 2009, pp. 2057–2060.
- [22] B.A. Schnaufer and W.K. Jenkins, “New data-reusing LMS algorithms for improved convergence,” in *Asilomar Conf. on Signals, Syst. and Comput.*, 1993, pp. 1584–1588.
- [23] J. Apolinário Jr., M.L.R. Campos, and P.S.R. Diniz, “Convergence analysis of the binormalized data-reusing LMS algorithm,” *IEEE Trans. Signal Process.*, vol. 48[11], pp. 3235–3242, 2000.
- [24] R.A. Soni, K.A. Gallivan, and W.K. Jenkins, “Low-complexity data reusing methods in adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 52[2], pp. 394–405, 2004.
- [25] 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.
- [26] D.I. Pazaitis and A.G. Constantinides, “LMS+F algorithm,” *IET Electronics Letters*, vol. 31[7], pp. 1423–1424, 1995.
- [27] D.I. Pazaitis and A.G. Constantinides, “An intelligent LMS+F algorithm,” in *Signal Process. Workshop on Statistical Signal and Array Process.*, 1996, pp. 486–489.
- [28] J.A. Chambers, O. Tanrikulu, and A.G. Constantinides, “Least mean mixed-norm adaptive filtering,” *IET Electronics Letters*, vol. 30[19], pp. 1574–1575, 1994.
- [29] J. Chambers and A. Avlonitis, “A robust mixed-norm adaptive filter algorithm,” *IEEE Signal Process. Lett.*, vol. 4[2], pp. 46–48, 1997.
- [30] R.W. Harris, D.M. Chabries, and F.A. Bishop, “Variable step (VS) adaptive filter algorithm,” *IEEE Trans. Signal Process.*, vol. 34, pp. 309–316, 1986.
- [31] R.H. Kwong and E.W. Johnston, “A variable step size LMS algorithm,” *IEEE Trans. Signal Process.*, vol. 40[7], pp. 1633–1642, 1992.

- [32] H.-C. Shin, A.H. Sayed, and W.-J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, pp. 132–135, 2004.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] J. Arenas-García and A.R. Figueiras-Vidal, "Adaptive combination of IPNLMS filters for robust sparse echo cancellation," in *MLSP*, 2008, pp. 221–226.
- [37] M. Ferrer, M. de Diego, A. Gonzalez, and G. Piñero, "Convex combination of affine projection algorithms," in *European Signal Processing Conference (EUSIPCO)*, 2009, pp. 431–435.
- [38] 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.
- [39] R. Candido, M.T.M. Silva, and V.H. Nascimento, "Transient and steady-state analysis of the affine combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 58[8], pp. 4064–4078, 2010.
- [40] L.A. Azpicueta-Ruiz, M. Zeller, A.R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, "Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation," vol. 19[1], pp. 97–110, 2011.
- [41] W.B. Lopes and C.G. Lopes, "Incremental-cooperative strategies in combination of adaptive filters," in *ICASSP*, 2011, pp. 4132–4135.
- [42] L.F.O. Chamon, W.B. Lopes, and C.G. Lopes, "Combination of adaptive filters with coefficients feedback," in *ICASSP*, 2012, pp. 3785–3788.
- [43] W.B. Lopes and C.G. Lopes, "Incremental combination of RLS and LMS adaptive filters in nonstationary scenarios," *Accepted in ICASSP*, 2013.
- [44] L.F.O. Chamon and C.G. Lopes, "Transient performance of an incremental combination of LMS filters," in *European Signal Processing Conference (EUSIPCO)*, 2013, pp. 1–5.
- [45] 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.
- [46] L. Arévalo, R. Sampaio-Neto, J.A. Apolinário, and M.L.R. Campos, "Convex combination of three affine projections adaptive filters," in *ISWCS*, 2013.

- [47] L.F.O. Chamon and C.G. Lopes, “There’s plenty of room at the bottom: Incremental combinations of sign-error LMS filters,” in *International Conference in Acoustic, Speech, and Signal Processing (ICASSP)*, 2014.
- [48] H.F. Ferro, L.F.O. Chamon, and C.G. Lopes, “FIR-IIR adaptive filters hybrid combination,” *IET Electronics Letters*, vol. 50[7], pp. 501–503, 2014.
- [49] J.H. Choia, S.H. Kima, and S.W. Kima, “Adaptive combination of affine projection and NLMS algorithms,” *Signal Processing*, vol. 100, pp. 64–70, 2014.
- [50] T. T. Adali and P.J. Schreier, “Optimization and estimation of complex-valued signals: Theory and applications in filtering and blind source separation,” *IEEE Signal Process. Mag.*, vol. 31[5], pp. 112–128, 2014.
- [51] K. Ogata, *Modern Control Engineering*, Prentice Hall, 2009.
- [52] Y.A. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*, Springer, 2006.
- [53] J. Benesty, D.R. Morgan, and M.M. Sondhi, “A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation,” *IEEE Trans. Speech Audio Process.*
- [54] F. Reed, P.L. Feintuch, and N.J. Bershad, “Time delay estimation using the LMS adaptive filter—Static behavior,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29[3], pp. 561–571, 1981.
- [55] P.L. Feintuch, N.J. Bershad, and F. Reed, “Time delay estimation using the LMS adaptive filter—Dynamic behavior,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29[3], pp. 571–576, 1981.
- [56] L. Qu and N.J. Bershad, “Comment on “time delay estimation using the LMS adaptive filter—Static behavior”,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33[6], pp. 1627–1627, 1985.
- [57] H.C. So and P.C. Ching, “Comparative study of five LMS-based adaptive time delay estimators,” *IEE Proc. Radar, Sonar and Nav.*, vol. 148[1], pp. 9–15, 2001.
- [58] A.A. Emadzadeh, C.G. Lopes, and J.L. Speyer, “Online time delay estimation of pulsar signals for relative navigation using adaptive filters,” in *ION Pos., Loc. and Nav. Symp.*
- [59] L.F.O. Chamon and C.G. Lopes, “Combination of adaptive filters for relative navigation,” in *European Signal Processing Conference (EUSIPCO)*, 2011, pp. 1771–1775.
- [60] B. Widrow and E. Walach, *Adaptive inverse control: a signal processing approach*, Wiley-IEEE Press, 2008.
- [61] H.-C. Shin and A.H. Sayed, “Mean-square performance of a family of affine projection algorithms,” *IEEE Trans. Signal Process.*, vol. 52[1], pp. 90–102, 2004.

- [62] S.G. Sankaran and A.A. Beex, “Normalized LMS algorithm with orthogonal correction factors,” in *Asilomar Conf. on Signals, Syst. and Comput.*, 1997, pp. 1670–1673.
- [63] D.G. Manolakis, V.K. Ingle, and S.M. Kogon, *Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing*, Artech House, 2005.
- [64] L. Pasquato and I. Kale, “Adaptive IIR filter initialization via hybrid FIR/IIR adaptive filter combination,” *IEEE Trans. Instrum. Meas.*, vol. 50[6], pp. 1830–1835, 2001.
- [65] M.T.M. Silva and J. Arenas-García, “A soft-switching blind equalization scheme via convex combination of adaptive filters,” *IEEE Trans. Signal Process.*, vol. 61[5], pp. 1171–1182, 2013.
- [66] K.D. Gilbert and K.L. Payton, “Competitive algorithm blending for enhanced source separation,” in *Asilomar Conference on Signals, Syst. and Comput.*, 2014.
- [67] E. Walach and B. Widrow, “The Least Mean Fourth (LMF) adaptive algorithm and its family,” *IEEE Trans. Inf. Theory*, vol. 30[2], pp. 275–283, 1984.
- [68] D.L. Duttweiler, “Proportionate normalized least-mean-square adaptation in echo cancelers,” *IEEE Trans. Speech Audio Process.*, vol. 8[5], pp. 508–518, 2000.
- [69] J. Benesty and S.L. Gay, “An improved PNLMS algorithm,” in *International Conference in Acoustic, Speech, and Signal Processing*. IEEE, 2002, pp. 1881–1884.
- [70] S. Shaffer and C.S. Williams, “Comparison of LMS, alpha LMS, and data reusing LMS algorithms,” in *Asilomar Conference on Circuits, Systems, and Computers*, 1983, pp. 260–264.
- [71] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Elec. Comm. Japan*, vol. J67-A, pp. 126–132, 1984.
- [72] D.P. Bertsekas, “A new class of incremental gradient methods for least squares problems,” *SIAM J. Optim.*, vol. 7[4], pp. 913–926, 1997.
- [73] 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.
- [74] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [75] Y.A. Huang and J. Benesty, Eds., *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Springer, 2004.
- [76] T. van Waterschoot and M. Moonen, “Fifty years of acoustic feedback control: State of the art and future challenges,” *Proc. IEEE*, vol. 99[2], pp. 288–327, 2011.
- [77] A.C. Singer and M. Feder, “Universal linear prediction by model order weighting,” *IEEE Trans. Signal Process.*, vol. 47[10], pp. 2685–2699, 1999.

- [78] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer, 2 edition, 2009.
- [79] A. Hać, *Wireless Sensor Network Designs*, John Wiley & Sons, 2003.
- [80] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.
- [81] P. van Mieghem, *Graph spectra for complex networks*, Cambridge University Press, 2011.
- [82] F. Vahid and T.D. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley & Sons, 2001.
- [83] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 3 edition, 1996.
- [84] R.B. Bramley, *Row projection methods for linear systems*, Ph.D. thesis, Computer Science Department, University of Illinois at Urbana-Champaign, Urbana, IL, 1989.
- [85] H.H. Bauschke and J.M. Borwein, “On projection algorithms for solving convex feasibility problems,” *SIAM Review*, vol. 38[3], pp. 367–426, 1996.
- [86] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1[3], pp. 123–231, 2014.
- [87] J. Benesty and T. Gänsler, “On data-reuse adaptive algorithms,” in *International Workshop on Acoustic Echo and Noise Control*, 2003, pp. 31–34.
- [88] S. Roy and J.J. Shynk, “Analysis of the data-reusing LMS algorithm,” in *Midwest Symposium on Circuits and Systems*, 1989, pp. 1127–1130.
- [89] Y. Zhang and J.A. Chambers, “Convex combination of adaptive filters for a variable tap-length LMS algorithm,” *IEEE Signal Process. Lett.*, vol. 13[10], pp. 628–631, 2006.
- [90] S.S. Kozat, A.C. Singer, A.T. Erdogan, and A.H. Sayed, “Unbiased model combinations for adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 58[8], pp. 4421–4427, 2010.
- [91] L.A. Azpicueta-Ruiz, A.R. Figueiras-Vidal, and J. Arenas-García, “A normalized adaptation scheme for the convex combination of two adaptive filters,” in *ICASSP*, 2008, pp. 3301–3304.
- [92] S.S. Kozat and A.C. Singer, “Further results in multistage adaptive filtering,” in *ICASSP*, 2002, pp. 1329–1332.
- [93] P. Andersson, “Adaptive forgetting in recursive identification through multiple models,” *International Journal of Control*, vol. 42[5], pp. 1175–1193, 1985.
- [94] M. Niedzwiecki, “Multiple-model approach to finite memory adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 40[2], pp. 470–473, 1992.

- [95] M. Niedzwiecki, "Identification of nonstationary stochastic systems using parallel estimation schemes," *IEEE Trans. Autom. Control*, vol. 35[3], pp. 329–334, 1990.
- [96] 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.
- [97] K.C. Ho, "A study of two adaptive filters in tandem," *IEEE Trans. Signal Process.*, vol. 48[6], pp. 1626–1636, 2000.
- [98] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2007.
- [99] P.S.R. Diniz, R.P. Braga, and S. Werner, "Set-membership affine projection algorithm for echo cancellation," in *IEEE International Symposium on Circuits and Systems*, 2006.
- [100] W.A. Martins, M.V.S. Lima, and P.S.R. Diniz, "Semi-blind data-selective equalizers for qam," in *IEEE Workshop on Signal Processing Advances in Wireless Communications*, 2008, pp. 501–505.
- [101] R.C. de Lamare and P.S.R. Diniz, "Blind adaptive interference suppression based on set-membership constrained constant-modulus algorithms with dynamic bounds," *IEEE Trans. Signal Process.*, vol. 61[5], pp. 1288–1301, 2013.
- [102] T. Strohmer and R. Vershynin, "A randomized solver for linear systems with exponential convergence," in *International Workshop on Randomization and Computation*, 2006, pp. 499–507.
- [103] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 25, pp. 262–278, 2009.
- [104] D. Needell, "Randomized Kaczmarz solver for noisy linear systems," *BIT Numerical Mathematics*, vol. 50[2], pp. 395–403, 2010.
- [105] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5[1], pp. 1–122, 2012.
- [106] J.C.M. Bermudez, N.J. Bershad, and J.-Y. Tournet, "Stochastic analysis of an error power ratio scheme applied to the affine combination of two LMS adaptive filters," *Signal Processing*, vol. 91[11], pp. 2615–2622, 2011.
- [107] M.A. Donmez, H.A. Inan, and S.S. Kozat, "Adaptive mixture methods based on bregman divergences," *Digital Signal Processing*, vol. 23[1], pp. 86–97, 2013.
- [108] R. Candido, M.T.M. Silva, and V.H. Nascimento, "Affine combinations of adaptive filters," in *Asilomar Conference on Signals, Syst. and Comput.*, 2008, pp. 236–240.
- [109] L.A. Azpicueta-Ruiz, M. Zeller, A.R. Figueiras-Vidal, and J. Arenas-García, "Least-squares adaptation of affine combinations of multiple adaptive filters," in *IEEE International Symposium on Circuits and Systems*, 2010, pp. 2976–2979.

- [110] S.S. Kozat, A.T. Erdogan, A.C. Singer, and A.H. Sayed, “Transient analysis of adaptive affine combinations,” *IEEE Trans. Signal Process.*, vol. 59[12], pp. 6227–6232, 2011.
- [111] A. Bhaya and E. Kaszkurewicz, *Control perspectives on numerical algorithms and matrix problems*, SIAM, 2006.
- [112] F.S. Cattivelli, C.G. Lopes, and A.H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *IEEE Trans. Signal Process.*, vol. 56[5], pp. 1865–1877, 2008.
- [113] F.S. Cattivelli and A.H. Sayed, “Distributed detection over adaptive networks using diffusion adaptation,” *IEEE Trans. Signal Process.*, vol. 59[5], pp. 1917–1932, 2011.
- [114] Sheng-Yuan Tu and A.H. Sayed, “Optimal combination rules for adaptation and learning over networks,” in *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*. IEEE, 2011, pp. 317–320.
- [115] A.H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z.J. Towfic, “Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior,” *IEEE Signal Process. Mag.*, vol. 30[3], pp. 155–171, 2013.
- [116] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Mag.*, vol. 30[3], pp. 83–98, 2013.
- [117] “Apache Spark: Lightning-fast cluster computing,” <http://spark.apache.org/>.
- [118] E. Ferrara Jr. and B. Widrow, “The time-sequenced adaptive filter,” *IEEE Trans. Circuits Syst.*, vol. 28[6], pp. 519–523, 1981.
- [119] H.W. Sorenson and D.L. Alspach, “Recursive Bayesian estimation using Gaussian sums,” *Automatica*, vol. 7, pp. 465–479, 1971.
- [120] J.T.-H. Lo, “Finite-dimensional sensor orbits and optimal nonlinear filtering,” *IEEE Trans. Inf. Theory*, vol. 18[5], pp. 583–588, 1972.
- [121] S.S. Kozat and A.C. Singer, “Multi-stage adaptive signal processing algorithms,” in *Sensor Array and Multichannel Signal Processing Workshop*, 2000, pp. 380–384.
- [122] M. Martinez-Ramon, J. Arenas-García, A. Navia-Vazquez, and A.R. Figueiras-Vidal, “An adaptive combination of adaptive filters for plant identification,” in *International Conference on Digital Signal Processing*, 2002, pp. 1195–1198.
- [123] M.T.M. Silva, V.H. Nascimento, and J. Arenas-García, “A transient analysis for the convex combination of two adaptive filters with transfer of coefficients,” in *ICASSP*, 2010, pp. 3842–3845.
- [124] A. Gonzalo-Ayuso, M.T.M. Silva, V.H. Nascimento, and J. Arenas-García, “Improving sparse echo cancellation via convex combination of two nlms filters with different lengths,” in *International Workshop on Machine Learning for Signal Processing (MLSP)*, 2012, pp. 1–6.

- [125] W. Gao, C. Richard, J.C.M. Bermudez, and J. Huang, “Convex combinations of kernel adaptive filters,” in *International Workshop on Machine Learning for Signal Processing (MLSP)*, 2014, pp. 1–5.
- [126] L.A. Azpicueta-Ruiz, M. Zeller, J. Arenas-García, and W. Kellermann, “Novel schemes for nonlinear acoustic echo cancellation based on filter combinations,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 193–196.
- [127] B. Jelfs, S. Javidi, P. Vayanos, and D. Mandic, “Characterisation of signal modality: Exploiting signal nonlinearity in machine learning and signal processing,” *Journal of Signal Processing Systems*, vol. 61[1], pp. 105–115, 2010.
- [128] V.H. Nascimento, M.T.M. Silva, R. Candido, and J. Arenas-García, “A transient analysis for the convex combination of adaptive filters,” in *SSP*, 2009, pp. 53–56.
- [129] V.H. Nascimento, M.T.M. Silva, L.A. Azpicueta-Ruiz, and J. Arenas-Garcia, “On the tracking performance of combinations of least mean squares and recursive least squares adaptive filters,” in *IEEE International Conference on Acoustics Speech and Signal Processing*, 2010, pp. 3710–3713.
- [130] T. Trump, “On a combination of m adaptive filters,” in *European Signal Processing Conference (EUSIPCO)*, 2013, pp. 1–5.
- [131] T.-K. Woo, “Fast hierarchical least mean square algorithm,” *IEEE Signal Process. Lett.*, vol. 8, pp. 289–291, 2001.
- [132] V.H. Nascimento, “Analysis of the hierarchical LMS algorithm,” *IEEE Signal Process. Lett.*, vol. 10[3], pp. 78–81, 2003.
- [133] 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.
- [134] L. Shi and Y. Lin, “Convex combination of adaptive filters under the maximum correntropy criterion in impulsive interference,” *IEEE Signal Process. Lett.*, vol. 21[11], pp. 1385–1388, 2014.
- [135] A. Zerguine, M. Bettayeb, and C.F.N. Cowan, “A hybrid LMS-LMF scheme for echo cancellation,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, pp. 2313–2316.
- [136] J. Arenas-García and A.R. Figueiras-Vidal, “Adaptive combination of stochastic gradient filters with different cost functions,” in *Learning’04 International Conference*, 2004, pp. 123–128.
- [137] C.G. Lopes, E. Satorius, P. Estabrook, and A.H. Sayed, “Efficient adaptive carrier tracking for Mars to Earth communications during entry, descent and landing,” in *SSP*, 2007, pp. 517–521.
- [138] M. Nekui and M. Atarodi, “A fast converging algorithm for network echo cancellation,” *IEEE Signal Process. Lett.*, vol. 11[4], pp. 427–430, 2004.

- [139] R.C. de Lamare and V.H. Nascimento, “Joint model-order and step-size adaptation with convex combinations of reduced-rank adaptive filters,” in *International Symposium on Wireless Communication Systems (ISWCS)*, 2010, pp. 394–398.
- [140] M. Lazaro-Gredilla, L.A. Azpicueta-Ruiz, A.R. Figueiras-Vidal, and J. Arenas-García, “Adaptively biasing the weights of adaptive filters,” *IEEE Trans. Signal Process.*, vol. 58[7], pp. 3890–3895, 2010.
- [141] K. Mayyas and T. Aboulnasr, “A robust variable step size LMS-type algorithm: Analysis and simulations,” in *ICASSP*, 1995, pp. 1408–1411.
- [142] V.J. Mathews and Z. Xie, “A stochastic gradient adaptive filter with gradient adaptive step size,” *IEEE Trans. Signal Process.*, vol. 41[6], pp. 2075–2087, 1993.
- [143] C.G. Lopes and J.C.M. Bermudez, “Evaluation and design of variable step size adaptive algorithms,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, pp. 3845–3848.
- [144] G.E.P. Box, J.S. Hunter, and W.G. Hunter, *Statistics for Experimenters: Design, Innovation, and Discovery*, Springer, 2 edition, 2005.
- [145] S.L. Gay and S. Tavathia, “The fast affine projection algorithm,” in *International Conference in Acoustic, Speech, and Signal Processing*. IEEE, 1995, pp. 3023–3026.
- [146] M.C. Tsakiris and P.A. Naylor, “Fast exact affine projection algorithm using displacement structure theory,” in *International Conference on Digital Signal Processing*, 2009.
- [147] A. Leon-Garcia, *Probability, Statistics, and Random Processes For Electrical Engineering*, Prentice Hall, 2008.
- [148] B. Hassibi, A.H. Sayed, and T. Kailath, “ H^∞ optimality of the LMS algorithm,” vol. 44[2], pp. 267–280, 1996.
- [149] 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.
- [150] Y.V. Zakharov, “Low-complexity implementation of the affine projection algorithm,” *IEEE Signal Process. Lett.*, vol. 15, pp. 557–560, 2008.
- [151] W.A. Sethares and C.R. Johnson, “A comparison of two quantized state adaptive algorithms,” vol. 37[1], pp. 138–143, 1989.
- [152] 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.
- [153] H.-J. Lo and D.V. Anderson, “A hardware-efficient implementation of the fast affine projection algorithm,” in *ICASSP*, 2009, pp. 581–584.
- [154] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 2013.

- [155] L. Isserlis, “On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables,” *Biometrika*, vol. 12[1/2], pp. 134–139, 1918.