Steady-State MSE Performance Analysis of Mixture Approaches to Adaptive Filtering

Suleyman Serdar Kozat, *Member, IEEE*, Alper Tunga Erdogan, *Member, IEEE*, Andrew C. Singer, *Fellow, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—In this paper, we consider mixture approaches that adaptively combine outputs of several parallel running adaptive algorithms. These parallel units can be considered as diversity branches that can be exploited to improve the overall performance. We study various mixture structures where the final output is constructed as the weighted linear combination of the outputs of several constituent filters. Although the mixture structure is linear, the combination weights can be updated in a highly nonlinear manner to minimize the final estimation error such as in Singer and Feder 1999; Arenas-Garcia, Figueiras-Vidal, and Sayed 2006; Lopes, Satorius, and Sayed 2006; Bershad, Bermudez, and Tourneret 2008; and Silva and Nascimento 2008. We distinguish mixture approaches that are convex combinations (where the linear mixture weights are constrained to be nonnegative and sum up to one) [Singer and Feder 1999; Arenas-Garcia, Figueiras-Vidal, and Sayed 2006], affine combinations (where the linear mixture weights are constrained to sum up to one) [Bershad, Bermudez, and Tourneret 2008] and, finally, unconstrained linear combinations of constituent filters [Kozat and Singer 2000]. We investigate mixture structures with respect to their final mean-square error (MSE) and tracking performance in the steady state for stationary and certain nonstationary data, respectively. We demonstrate that these mixture approaches can greatly improve over the performance of the constituent filters. Our analysis is also generic such that it can be applied to inhomogeneous mixtures of constituent adaptive branches with possibly different structures, adaptation methods or having different filter lengths.**

***Index Terms*—Adaptive filtering, affine mixtures, combination methods, convex mixtures, diversity gain, least mean squares (LMS), linear mixtures, recursive least squares (RLS), tracking.**

I. Introduction

In adaptive filtering applications, there are various design choices to be made that affect convergence and tracking performance. Among these, we can list the selection of the order of the adaptive filter [1], and the choice of the adaptation algorithm and its parameters [6], [7]. The lack of *a priori* structural and statistical information about the data model that relates the observations to the desired signals would make the selection process more difficult. As a robust solution to this problem, a method based on combining the outputs of various adaptive filtering branches corresponding to different design choices has recently been proposed [1]–[6], [8], [9]. According to this approach, parallel adaptive branches can be perceived as alternative hypotheses about the data model as well as different diversity sources, which can be used to achieve better performance than the individual branches. Depending on the availability of intelligent combining approaches and computational/hardware resources, such structures may be well-suited for a wide variety of adaptive filtering applications, especially for those involving nonstationary data models.

In this paper, we study different mixing strategies in which the final outputs are formed as the weighted linear combination of the outputs of several constituent algorithms for stationary and certain nonstationary data. In its most general form, the outputs of the constituent algorithms may be combined using a nonlinear method rather than a weighted linear mixture. However, the linear mixture is most commonly used due to its tractability and adequate accuracy in modeling [1]–[5], [10], [11]. We note that although the final combination structure is often linear,[1] the update on the linear combination weights can be highly nonlinear [1], [2], [4], [10].

We distinguish mixture approaches that are convex combinations (where the linear mixture weights are constrained to be nonnegative and sum up to one) [1], [2], [10], [11], affine combinations (where the linear mixture weights are constrained to sum up to one) [4] and, finally, unconstrained linear combinations of constituent filters [6]. We study several different methods to update these mixture weights under the given constraints, such as training the convex combination weights using a stochastic gradient update after a nonlinear variable transformation [2].

At first sight, constraining the mixture weights, such as imposing convex constraints, would result in lower target diversity gains than could be achieved by a less constrained (such as affine combination) or unconstrained mixture approach at steady state. However, we point out that the combination weights are also updated through an adaptive process and that impacts the overall performance. As an example, as we demonstrate here, although an optimal affine (or unconstrained) combination of the con-

Manuscript received May 26, 2009; accepted April 07, 2010. Date of publication May 06, 2010; date of current version July 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hideaki Sakai. This work is supported in part by TUBITAK Career Award, Contract 104E073, Contract 108E195, and the Turkish Academy of Sciences GEBIP Program. The work of A. H. Sayed was supported in part by NSF Grants ECS-0601266, ECCS-0725441, and CCF-094236.

S. S. Kozat and A. T. Erdogan are with the Electrial and Electronics Engineering Department, Koc University, Istanbul 34450, Turkey (e-mail: skozat@ku.edu.tr; aerdogan@ku.edu.tr).

A. C. Singer is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana IL 61801 USA (e-mail: acsinger@illinois.edu).

A. H. Sayed is with the Department of Electrical Engineering, University of California at Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2010.2049650

[1]However, for the framework investigated in this paper, the linear weights used for combination cannot have components from the outputs of the constituent algorithm. Hence, these are truly linear combinations, i.e., $y(t) = x^3(t) = [x(t)]^2 x(t)$ is not a linear model in $x(t)$.

stituent algorithms may yield a better target diversity gain than a convex combination in steady state, there may be excess gradient noise [or excess mean-square error (MSE)] due to the training of these affine weights based on a stochastic gradient approach [4]. In comparison, this additional MSE due to gradient noise, which is counteracting against the diversity gain, is not present in certain convex updates such as the one used in [2] due to the sigmoid nonlinearity. A similar tradeoff is also present for tracking performance: Although it may be favorable to use a Hessian-based approach such as the recursive least squares (RLS) algorithm in terms of reducing the excess MSE in the final combination stage, the least mean squares (LMS) (or gradient) based approaches are shown to have better tracking properties under certain conditions [7], [12].

The combination methods proposed for adaptive filtering can be considered as the descendants of some earlier work in adaptive control [13]–[15], computational learning theory [11], [16], [17], investment [18] and universal source coding [19], [20]. In particular, in computational learning theory such methods have been a major focus of research, in which several Bayesian algorithms have been proposed under the mixture-of-experts framework (see, for example, [11] and [17]). The objective here is to combine the outputs of several different adaptive filters running in parallel on a given task, with the goal of achieving performance better than or at least as good as the best constituent algorithm, for all sequences. This is usually accomplished by exploiting the time-dependent nature of the best choice among constituent filters [10], [11]. In computational learning theory, the performance of such combination algorithms are often measured through the excess loss, i.e., "the regret," with respect to the best constituent algorithm in a deterministic sense. The literature typically reports that the bounds are deterministic, i.e., they are guaranteed to hold under the given data models for each individual sequence of outcomes, and these results are given with respect to the best algorithm in the combination. It has been demonstrated that the final algorithm can often do even better than the best constituent algorithm in the competition class [1], [20], [21]. Although deterministic bounds are guaranteed to hold, they usually require certain assumptions on the underlying signals or regression parameters, such as boundedness [10], [22], [23] or upper limits on the norms of regression vectors [11], [24], respectively. The results for deterministic data do not hold in a fairly general stochastic context, e.g., if the observation sequence is a stationary Gaussian process, the boundedness assumption is invalidated.

Within the context of adaptive filtering, an adaptive convex combination algorithm based on Bayesian mixture strategies was introduced in [1] to merge a finite number of adaptive branches using RLS updates based on their accumulated past performance. This algorithm was shown to asymptotically achieve the performance of the best algorithm in the mixture for any bounded but arbitrary real-valued data in [1]. This convex mixture was then extended to more general combination structures in [6], where both the convex combination as well as unconstrained linear combination of multiple order LMS algorithms were studied.

An alternative algorithm that adapts a convex combination of two adaptive filters using stochastic gradient methods was studied in [2]. Although the analysis given for the convex combination of two filters was generic [2], the results were then specialized to the case of two LMS filters with different learning rates: one with a comparably smaller and the other with a comparably larger learning rate, were combined. Hence, the combination approach enjoyed fast converge in the start of the adaptation and smaller excess MSE at the steady state. The algorithm introduced in [2] was shown to be universal, such that it achieves the performance of the best constituent algorithm (of the two) under the given data model. This combination is shown to outperform even the best constituent filter if the cross-correlation between the *a priori* errors of the constituent filters are sufficiently small [2]. This approach was extended to a combination of multiple adaptive algorithms of the same length in [8], [9], and [25] and of the different lengths (along the same lines of [1]) in [3] and [25]. Recently, the convex combination of [2] was used in [5] for combination of adaptive filters using different adaptation rules, such as the RLS update, in a tracking context.

Motivated by the results of [2], the work of [4] relaxed the convex combination constraint and used affine combination for merging two adaptive branches. The authors demonstrated that under certain circumstances the optimal affine combination is different than a convex combination, and the affine combination algorithm introduced (which is not realizable) will have a final MSE that is better than the MSEs of both constituent algorithms. To realize this optimal affine combination of two filters using LMS updates, the authors introduced a stochastic gradient update (without performance analysis) and a variable transformation method. In this paper, we first extend this stochastic gradient update to a combination of multiple filters and then provide the steady-state analysis for stationary and certain nonstationary data.

In this article, we consider a more general adaptive combination framework consisting of various alternatives for the merging of multiple (two or more) branches. We provide a general framework for linear combination methods and investigate several different methods to update combination weights. The adaptive branches involved are allowed to be inhomogeneous, i.e., the filters are not constrained to have equal lengths or to use the same update. We first provide the maximal achievable diversity gains for each combination strategy in steady state. We then produce theoretical analysis for all structures and adaptive methods in steady state for stationary and non-stationary data in a tracking context. After deriving final MSEs for all structures, including the excess MSEs, we also provide comparison between these methods through simulations. We demonstrate that we can improve over the existing methods [2], [4], [5] in terms of steady-state MSE by using unconstrained methods for both stationary and nonstationary data. Our methods are generic and can be readily extended to blind algorithms as was done in [5]. In comparison to [6] where only the LMS update was used to train the combination weights, we provide an extension where more general combination structures (such as unconstrained, convex and affine combinations) and adaptation schemes (such as LMS, RLS and various gradient search algorithms based on constraint set parametrization) are used for the combination stage.

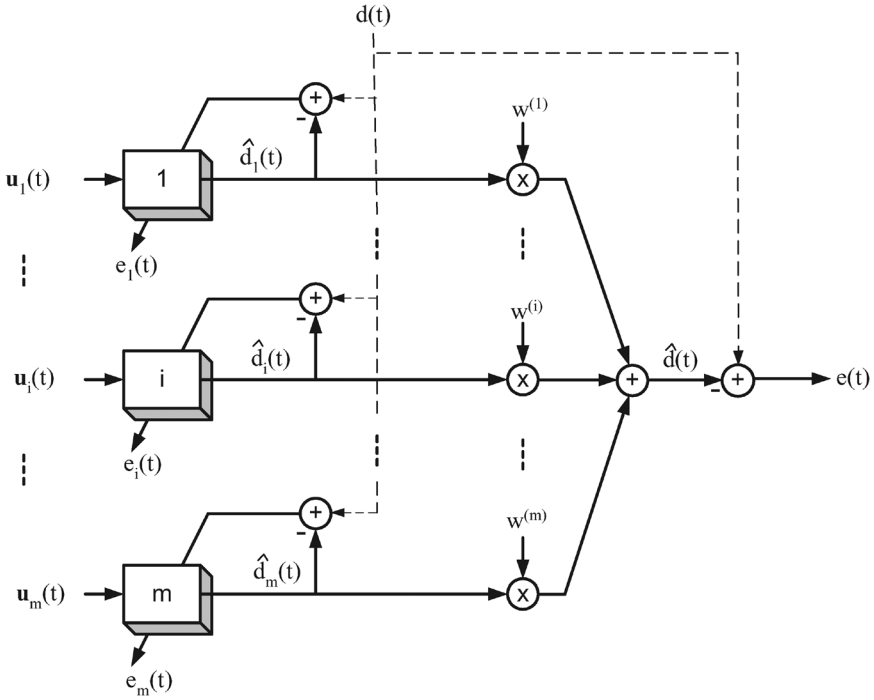

Fig. 1. Linear mixture of outputs of m adaptive filters.

The organization of the article is as follows. In Section II, we first provide the general mixture structure for the combination of outputs of several parallel adaptive filters. We then investigate three different combination structures including unconstrained linear, affine and convex combinations. We provide the corresponding diversity gains, i.e., the achievable final minimum MSEs by these three combination structures based on the cross-correlations between the excess errors of the constituent filters. As an illustrative and widely studied example, we specialize these results to the case where the constituent filters are of the same length. We then continue to investigate adaptive methods to update the mixture weights. For unconstrained and affine combinations, we study both the RLS and LMS updates and provide the final MSE, as well as the excess MSEs due to using adaptive methods, in Section IV and Section V, respectively. In Section V-A, we introduce the stochastic gradient based algorithm to train the affine weights, which is an extension of the algorithm used in [4] for the combination of only two LMS filters (without performance analysis). For the convex combination of several branches, we study the convex combination approach of [3], which is an extension of [2], in Section VI. As an illustrative example, in Section IV, we specialize the results to the case where each constituent filter has the same length and uses additive updates (such as the RLS update or the LMS update) [7]. We conclude the paper with numerical examples for different combination structures and corresponding remarks.

A. Notation

In this paper, all vectors are column vectors and represented by boldface lowercase letters. Matrices are represented by boldface capital letters. For presentation purposes, we work only with real data. Given a vector $\boldsymbol{w}$: $w^{(i)}$ denotes the ith individual entry of $\boldsymbol{w}$; $\boldsymbol{w}^T$ is the transpose of $\boldsymbol{w}$; $||\boldsymbol{w}||_1 \triangleq \sum_i |w^{(i)}|$ is the l_1 norm; $||\boldsymbol{w}|| \triangleq \sqrt{\boldsymbol{w}^T\boldsymbol{w}}$ is the l_2 norm; and $||\boldsymbol{w}||_{\boldsymbol{R}} \triangleq \sqrt{\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}}$ is the weighted l_2 norm for a positive definite matrix $\boldsymbol{R}$. For a vector $\boldsymbol{w}$, $\mathrm{diag}(\boldsymbol{w})$ represents a diagonal matrix formed using the entries of $\boldsymbol{w}$. For a real number a, $|a|$ is the absolute value; $(a)^+ = a$ if $a \geq 0$, $(a)^+ = 0$ if $a < 0$. For a vector $\boldsymbol{w}$, $(\boldsymbol{w})^+$ represents a vector, where each entry of $\boldsymbol{w}$ is given by $(w^i)^+$. For a symmetric square matrix, $\boldsymbol{R} \in \mathbb{R}^{m\times m}$, $\rho_i(\boldsymbol{R})$, $i = 1,\ldots,m$, are the eigenvalues sorted in descending order. Special vectors (or matrices with an abuse of notation) $\mathbf{1}$ and $\mathbf{0}$ denote vectors (or matrices) of all ones or zeros, respectively, where the size of the vector (or the matrix) is understood from the context.

II. Model Description and Combination Methods

The generic model we consider in this paper consists of two parts. In the first part, we have m adaptive algorithms running in parallel to estimate a desired signal $d(t)$ as seen in Fig. 1. Each algorithm updates a weight vector, $\boldsymbol{w}_i(t) \in \mathbb{R}^{s_i}$, s_i is an integer, $i = 1,\ldots,m$, and produces an estimate $\boldsymbol{w}_i^T(t)\boldsymbol{u}_i(t)$ using the input vector process $\boldsymbol{u}_i(t) \in \mathbb{R}^{s_i}$. The estimation error for each algorithm is given by $e_i(t) \triangleq d(t) - \hat{d}_i(t)$. For each filter, the optimal weight vector that minimizes the MSE is given by

$$\boldsymbol{w}_{o,i} \triangleq \boldsymbol{R}_i^{-1}\boldsymbol{p}_i,$$

$i = 1,\ldots,m$, where $\boldsymbol{R}_i \triangleq E[\boldsymbol{u}_i(t)\boldsymbol{u}_i^T(t)]$ and $\boldsymbol{p}_i \triangleq E[\boldsymbol{u}_i(t)d(t)]$ for wide-stationary data. We emphasize that this wide-sense stationary model is extended to nonstationary data models in Section VII. Using this optimal weight vector, we define

$$e_{o,i}(t) \triangleq d(t) - \boldsymbol{w}_{o,i}^T\boldsymbol{u}_i(t),$$

where $E[e_{o,i}(t)\boldsymbol{u}_i(t)] = 0$ by orthogonality [7]. For each algorithm, *a priori*, *a posteriori* and estimation errors are defined as

$$\begin{aligned} e_{a,i}(t) &\triangleq [\boldsymbol{w}_{o,i} - \boldsymbol{w}_i(t)]^T \boldsymbol{u}_i(t) \\ e_{p,i}(t) &\triangleq [\boldsymbol{w}_{o,i} - \boldsymbol{w}_i(t+1)]^T \boldsymbol{u}_i(t) \\ e_i(t) &= d(t) - \boldsymbol{w}_i^T(t)\boldsymbol{u}_i(t), \end{aligned}$$

respectively. Clearly, based on these definitions, we can write

$$e_i(t) = e_{a,i}(t) + e_{o,i}(t),$$

$i = 1, \ldots, m$. We further define $J_i(t) \triangleq E[e_i^2(t)]$, $J_i = \lim_{t\to\infty} J_i(t)$ (MSE) and $J_{\text{ex},i}(t) = E[e_{a,i}^2(t)]$, $J_{\text{ex},i} = \lim_{t\to\infty} J_{\text{ex},i}(t)$ (EMSE), when these limits exist. We define the cross-correlation between *a priori* errors as $J_{\text{ex},ij}(t) = E[e_{a,i}(t)e_{a,j}(t)]$ and $J_{\text{ex},ij} = \lim_{t\to\infty} J_{\text{ex},ij}(t)$, when the limit exists. If we define for each filter the "clean" part of the desired signal as $g_i(t) \triangleq \boldsymbol{w}_{o,i}^T \boldsymbol{u}_i(t)$, then

$$\begin{aligned} \hat{d}_i(t) &= \boldsymbol{w}_i^T \boldsymbol{u}(t) = \boldsymbol{w}_{o,i}^T \boldsymbol{u}(t) - [\boldsymbol{w}_{o,i} - \boldsymbol{w}_i(t)]^T \boldsymbol{u}(t) \\ &= g_i(t) - e_{a,i}(t). \end{aligned} \tag{1}$$

The second stage of the model is the mixing stage. In order to obtain the final output, we combine outputs of the constituent filters using a linear combiner as

$$\hat{d}(t) = \boldsymbol{w}^T(t)\boldsymbol{y}(t)$$

where $\boldsymbol{y}(t) \triangleq [\hat{d}_1(t), \ldots, \hat{d}_m(t)]^T$ and $\boldsymbol{w}(t) \in \mathbb{R}^m$. Similar to the constituent algorithms, we consider only linear combinations in the final output. The final estimation error is given by

$$e(t) \triangleq d(t) - \hat{d}(t).$$

Using (1), we have

$$\boldsymbol{y}(t) = \begin{bmatrix} g_1(t) - e_{a,1}(t) \\ \vdots \\ g_m(t) - e_{a,m}(t) \end{bmatrix}.$$

With these definitions, the autocorrelation matrix for the input of the combination stage is given by $\boldsymbol{R}(t) \triangleq E[\boldsymbol{y}(t)\boldsymbol{y}^T(t)]$, $\boldsymbol{R} \triangleq \lim_{t\to\infty} \boldsymbol{R}(t)$ and the cross-correlation vector is given by $\boldsymbol{p}(t) \triangleq E[\boldsymbol{y}(t)d(t)]$, $\boldsymbol{p} \triangleq \lim_{t\to\infty} \boldsymbol{p}(t)$, when the limits exist.

To calculate $\boldsymbol{R}$, we observe that for any filter pairs i and j:

$$\begin{aligned} &\lim_{t\to\infty} E\left[(g_i(t) - e_{a,i}(t))\,(g_j(t) - e_{a,j}(t))\right] \\ &\quad = \lim_{t\to\infty} \{E\left[g_i(t)g_j(t)\right] - E\left[g_j(t)e_{a,i}(t)\right] \\ &\qquad\qquad - E\left[g_i(t)e_{a,j}(t)\right] + E\left[e_{a,i}(t)e_{a,j}(t)\right]\} \\ &\quad = E\left[g_i(t)g_j(t)\right] + \lim_{t\to\infty} E\left[e_{a,i}(t)e_{a,j}(t)\right] \\ &\quad = \sigma_{g,ij}^2 + J_{\text{ex},ij}, \end{aligned}$$

$\sigma_{g,ij}^2 \triangleq \boldsymbol{w}_{o,i}^T E[\boldsymbol{u}_i(t)\boldsymbol{u}_j^T(t)]\boldsymbol{w}_{o,j}$, where in the third line we use a separation assumption similar to the one used in [2] and [7] that $E[g_i(t)e_{a,j}(t)] = E[g_i(t)]E[e_{a,j}(t)]$ for all i, j, in the limit as $t \to \infty$. This assumption is plausible as the filters converge as in [7]. With this result, by orthogonality we have

$$\begin{aligned} \boldsymbol{R} &= \begin{bmatrix} J_{\text{ex},1} + \sigma_{g,11}^2 & \ldots & J_{\text{ex},1m} + \sigma_{g,1m}^2 \\ \ldots & \ldots & \ldots \\ J_{\text{ex},1m} + \sigma_{g,1m}^2 & \ldots & J_{\text{ex},m} + \sigma_{g,mm}^2 \end{bmatrix} \\ &= \boldsymbol{G} + \boldsymbol{J} \end{aligned}$$

where $\boldsymbol{G}^{(i,j)} = \boldsymbol{G}^{(j,i)} \triangleq \sigma_{g,ij}^2$, $\boldsymbol{J}^{(i,j)} = \boldsymbol{J}^{(j,i)} \triangleq J_{\text{ex},ij}$ (when $i \neq j$) and $\boldsymbol{G}^{(i,i)} \triangleq \sigma_{g,ii}^2$, $\boldsymbol{J}^{(i,i)} \triangleq J_{\text{ex},i}$. Hence, the steady-state autocorrelation matrix $\boldsymbol{R}$ is formed as the sum of two positive semi-definite matrices: $\boldsymbol{G} = \lim_{t\to\infty} E\{[g_1(t), \ldots, g_m(t)]^T[g_1(t), \ldots, g_m(t)]\}$ for the stationary part of the data and $\boldsymbol{J} = \lim_{t\to\infty} E\{[e_{a,1}(t), \ldots, e_{a,m}(t)]^T[e_{a,1}(t), \ldots, e_{a,m}(t)]\}$ due to excess errors in modeling.

For the cross-correlation vector, since $\lim_{t\to\infty} E[d(t)\{g_i(t) - e_{a,i}(t)\}] = \sigma_{g,ii}^2$, by orthogonality we have

$$\boldsymbol{p} = \begin{bmatrix} \sigma_{g,11}^2 \\ \vdots \\ \sigma_{g,mm}^2 \end{bmatrix}.$$

We observe that for different structures and adaptation methods on the constituent filters, we would have different correlation matrices $\boldsymbol{R}$ and cross-correlation vectors $\boldsymbol{p}$. Given this generic setup, we consider three different combination structures including: unconstrained linear combination (linear combination), affine combination and convex combination. We first analyze the optimal diversity gains that can be achieved by these structures by formulating minimum MSE (MMSE) levels corresponding to the optimal combination weights for each structure. The adaptation algorithm alternatives corresponding to these structures will be investigated in Sections IV, V, and VI.

III. Mixture Structures and Final MMSEs

In this section, we obtain the optimal performance levels corresponding to the three different combination structures mentioned above. The derivations are generic such that one needs to only provide $\boldsymbol{R}$ and $\boldsymbol{p}$ for any constituent filter structure or the adaptation algorithm used. However, as an interesting and widely studied special case that was investigated for the combination of two filters in [2], [4], and [5] (or multiple filters in [8] and [9]), we also consider the scenario in which all of the constituent filters have the same order as a specific example. We note that, although here we consider wide-sense stationary data models, we extend our models to include nonstationarity in a tracking context in Section VII.

A. Linear Combination

For fixed $\boldsymbol{w}$ in the combination stage, the final MSE is given by

$$\lim_{t\to\infty} E\left[e^2(t)\right] = \sigma_d^2 - \boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p} + (\boldsymbol{w} - \boldsymbol{w}_o)^T\boldsymbol{R}(\boldsymbol{w} - \boldsymbol{w}_o) \tag{2}$$

where $\sigma_d^2 \triangleq E[d^2(t)]$ and $\boldsymbol{w}_o \triangleq \boldsymbol{R}^{-1}\boldsymbol{p}$. If the unconstrained linear combination scheme is used, then the optimal linear weight vector that minimizes the final MSE in (2) and the corresponding final MSE are given by

$$\boxed{\begin{array}{l}\boldsymbol{w}_o = \boldsymbol{R}^{-1}\boldsymbol{p}, \\ J_{\min} = \sigma_d^2 + \boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p}\end{array}} \tag{3}$$

where we assumed that $\boldsymbol{R}$ is invertible. Here, assuming asymptotic stationarity of $\boldsymbol{y}(t)$

$$e_o \triangleq d(t) - \boldsymbol{w}_o^T\boldsymbol{y}(t),$$

$J_{\min} = \lim_{t\to\infty} E[e_o^2(t)]$ and $\lim_{t\to\infty} E[e_o(t)\boldsymbol{y}(t)] = \boldsymbol{0}$.

As an illustrative example, suppose that the constituent filters are of the same length, such that each $\boldsymbol{w}_i(t) \in \mathbb{R}^s$ for an integer s. Assume further that each $g_i(t)$ can be represented by a common $g_i(t) \triangleq g(t), \forall i$, $\boldsymbol{u}_i(t) = \boldsymbol{u}(t), \forall i$, and define $\sigma_g^2 \triangleq E[g^2(t)] = \boldsymbol{w}_o^T E[\boldsymbol{u}(t)\boldsymbol{u}^T(t)]\boldsymbol{w}_o$. For each term in $\boldsymbol{R}$, we have $\boldsymbol{R}^{(i,j)} = \lim_{t\to\infty} E[\{g(t) - e_{a,i}(t)\}\{g(t) - e_{a,j}(t)\}] = \sigma_g^2 + J_{\text{ex},ij}$ (and $\boldsymbol{R}^{(i,i)} = \sigma_g^2 + J_{\text{ex},i}$). For the cross-correlation vector, since $\lim_{t\to\infty} E[d(t)\{g(t) - e_{a,i}(t)\}] = \sigma_g^2$, we have

$$\boldsymbol{R} = \boldsymbol{J} + \sigma_g^2 \boldsymbol{1}\,\boldsymbol{1}^T$$

and

$$\boldsymbol{p} = \sigma_g^2\boldsymbol{1}$$

where $\boldsymbol{J}^{(i,j)} = J_{\text{ex},ij}$ (and $\boldsymbol{J}^{(i,i)} = J_{\text{ex},i}$). Using the matrix inversion lemma in (3), we obtain

$$\boxed{\begin{array}{l}\boldsymbol{w}_o = \frac{\boldsymbol{J}^{-1}\boldsymbol{1}}{\sigma_g^{-2} + (\boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1})} \\ J_{\min} = \sigma_n^2 + \frac{1}{\sigma_g^{-2} + (\boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1})}\end{array}} \tag{4}$$

where $\sigma_n^2 \triangleq \min_{\boldsymbol{w}} E[(d(t) - \boldsymbol{w}^T\boldsymbol{u}(t))^2]$, i.e., the linear MMSE and $\boldsymbol{J}$ is invertible. Clearly, $J_{\min} \geq \sigma_n^2$ and $J_{\min}$ approaches σ_n^2 for nearly singular $\boldsymbol{J}$ and for small σ_g^2 values assuming that $\boldsymbol{J}$ does not depend on σ_g^2. Furthermore, we observe that unless σ_g^{-2} is equal to zero, then $\boldsymbol{1}^T\boldsymbol{w}_o \neq 1$, hence the optimal weight combination is not affine. However, the optimal linear combination would be nearly affine if σ_g^{-2} is negligible compared to $(\boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1})$. As an example, for most commonly used adaptive methods (such as the RLS update [5], the LMS update [2]), since the excess terms $J_{\text{ex},i}$, $J_{\text{ex},ij}$, are proportional to σ_n^2, we have σ_n^2/σ_g^2 term in the denominator of $\boldsymbol{w}_o$ (4). Hence, for large SNR, it can be shown after some algebra that we get $\boldsymbol{1}^T\boldsymbol{w}_o \approx 1$, i.e., an affine mixture, for the combination of these adaptive methods.

B. Affine Combination

When the mixture weights are constrained to be affine, we find the final MMSE by solving

$$\min_{\boldsymbol{w}} \left\{\sigma_d^2 - \boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p} + (\boldsymbol{w} - \boldsymbol{w}_o)^T\boldsymbol{R}(\boldsymbol{w} - \boldsymbol{w}_o)\right\}$$

subject to $\boldsymbol{1}^T\boldsymbol{w} = 1$. If we define the Lagrangian

$$\min_{\boldsymbol{w}} \left\{\sigma_d^2 - \boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p} + (\boldsymbol{w} - \boldsymbol{w}_o)^T\boldsymbol{R}(\boldsymbol{w} - \boldsymbol{w}_o) + \lambda(\boldsymbol{1}^T\boldsymbol{w} - 1)\right\},$$

then the optimal affine combination weights are given by

$$\boxed{\boldsymbol{w}_o^a \triangleq \boldsymbol{w}_o - \frac{(\boldsymbol{1}^T\boldsymbol{w}_o - 1)}{\boldsymbol{1}^T\boldsymbol{R}^{-1}\boldsymbol{1}}\boldsymbol{R}^{-1}\boldsymbol{1}.} \tag{5}$$

For this optimal affine combination in (5), the final MSE is given by

$$J_{\min}^a = \sigma_d^2 - \boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p} + \frac{(\boldsymbol{1}^T\boldsymbol{w}_o - 1)^2}{\boldsymbol{1}^T\boldsymbol{R}^{-1}\boldsymbol{1}}.$$

Hence, the difference between final MMSEs of the optimal unconstrained linear combination and the affine combination is given by

$$\boxed{J_{\min}^a - J_{\min} = \frac{(\boldsymbol{1}^T\boldsymbol{w}_o - 1)^2}{\boldsymbol{1}^T\boldsymbol{R}^{-1}\boldsymbol{1}}.}$$

Consider the same illustrative example that led to (4) such that $\boldsymbol{p} = \sigma_g^2\boldsymbol{1}$ and $\boldsymbol{R} = [\boldsymbol{J} + \sigma_g^2\boldsymbol{1}\,\boldsymbol{1}^T]$. In this case, we have

$$\boxed{\begin{array}{l}\boldsymbol{w}_o^a = \frac{\boldsymbol{J}^{-1}\boldsymbol{1}}{\boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1}}, \\ \quad = \frac{\boldsymbol{w}_o}{\|\boldsymbol{w}_o\|_1}, \\ J_{\min}^a = \sigma_n^2 + \frac{1}{\boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1}}.\end{array}} \tag{6}$$

Hence, for this case

$$J_{\min}^a - J_{\min} = \frac{\sigma_g^{-2}}{\left(\sigma_g^{-2} + \boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1}\right)(1 + \boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1})} \tag{7}$$

which is approximately equal to $1/(1 + \boldsymbol{1}^T\boldsymbol{J}^{-1}\boldsymbol{1})$ for small σ_g^2.

C. Convex Combination

When the combination is constrained to be convex, we solve

$$\min_{\boldsymbol{w}} \left\{\sigma_d^2 - \boldsymbol{p}^T R^{-1}\boldsymbol{p} + (\boldsymbol{w} - \boldsymbol{w}_o)^T\boldsymbol{R}(\boldsymbol{w} - \boldsymbol{w}_o)\right\} \tag{8}$$

subject to $\boldsymbol{1}^T\boldsymbol{w} = 1$ and $w^{(i)} \geq 0$, $i = 1, \ldots, m$, where $\boldsymbol{w} = [w^{(1)}, \ldots, w^{(m)}]^T$, i.e., we have a (convex) quadratic minimization problem over the unit (or standard) simplex $\Delta = \{\boldsymbol{w} | \boldsymbol{1}^T\boldsymbol{w} = 1, w^{(i)} \geq 0, i = 1, \ldots, m\}$, which is the intersection of the plane corresponding to affinity constraints and the nonnegative orthant.

We note that the cost function in (8) can be rewritten as

$$J^c = J_{\min} + \|\boldsymbol{w} - \boldsymbol{w}_o\|_{\boldsymbol{R}}^2$$

which is in terms of the weighted norm of $(\boldsymbol{w} - \boldsymbol{w}_o)$. Therefore, ignoring the constant term $J_{\min}$ we can rewrite the optimization problem in (8) as

$$\min_{\boldsymbol{w}} \|\boldsymbol{w} - \boldsymbol{w}_o\|_{\boldsymbol{R}}^2 \tag{9}$$

$$\text{subject to } \boldsymbol{w} \in \Delta \tag{10}$$

which is the projection of $\boldsymbol{w}_o$ to the unit simplex with respect to the weighted norm. We can further write

$$\begin{aligned} J^c &= J_{\min} + \|\boldsymbol{w} - \boldsymbol{w}_o^a + \boldsymbol{w}_o^a - \boldsymbol{w}_o\|_{\boldsymbol{R}}^2 \\ &= J_{\min} + \|\boldsymbol{w}_o^a - \boldsymbol{w}_o\|_{\boldsymbol{R}}^2 + \|\boldsymbol{w} - \boldsymbol{w}_o^a\|_{\boldsymbol{R}}^2 \\ &\quad + 2\,(\boldsymbol{w} - \boldsymbol{w}_o^a)^T\,\boldsymbol{R}\,(\boldsymbol{w}_o^a - \boldsymbol{w}_o) \\ &= J_{\min}^a + \|\boldsymbol{w} - \boldsymbol{w}_o^a\|_{\boldsymbol{R}}^2 - 2\frac{(\boldsymbol{1}^T\boldsymbol{w}_o - 1)}{\boldsymbol{1}^T\boldsymbol{R}^{-1}\boldsymbol{1}}\,(\boldsymbol{w} - \boldsymbol{w}_o^a)^T\,\boldsymbol{1}. \end{aligned}$$

Since over the constraint set Δ, $(\boldsymbol{w} - \boldsymbol{w}_o^a)^T\boldsymbol{1} = 0$, and ignoring the constant term $J_{\min}^a$ we can reformulate the optimization problem (9) corresponding to the best convex combination coefficients as

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \|\boldsymbol{w} - \boldsymbol{w}_o^a\|_{\boldsymbol{R}}^2 \\ \text{subject to} \quad & \boldsymbol{w} \in \Delta \end{aligned}$$

which is the projection of $\boldsymbol{w}_o^a$ to the unit simplex with respect to the weighted norm. When the weighting matrix $\boldsymbol{R} = \boldsymbol{I}$, i.e., the projection is with respect to the standard Euclidean norm, the corresponding projection can be obtained in a finite number of steps using the algorithm suggested in [26]. However, for a more general positive definite $\boldsymbol{R}$, the formulation of the projection point and the corresponding $J_{\min}^c$ is more involved. However, we can start with the following basic observation:

- if $\boldsymbol{w}_o^a \in \Delta$, i.e., $\boldsymbol{w}_o^a$ consists of only nonnegative elements, then $\boldsymbol{w}_o^c = \boldsymbol{w}_o^a$;
- otherwise, $\boldsymbol{w}_o^c$ is at the boundary of the constraint set Δ.

Using the results of [27], we can make more explicit statements:

1) $m = 2$ Case: When there are two adaptive branches to be combined, the projection task is rather simple:

$$\boldsymbol{w}_o^c = \begin{cases} \boldsymbol{w}_o^a & \boldsymbol{w}_o^a \geq 0 \\ [0 \quad 1]^T & w_o^{a(1)} < 0 \\ [1 \quad 0]^T & w_o^{a(2)} < 0 \end{cases} \tag{11}$$

where $\boldsymbol{w} \geq 0$ means all the entries of the vector is nonnegative. As an example, we present the case given in the first line of (11) in Fig. 2(a), where we plot the combination weights in $\mathbb{R}^2$. The cases given in the second and third line of (11) are illustrated in Fig. 2(b) with the level set of the cost function as the ellipse segment corresponding to the minimum achievable cost. The excess MSE corresponding to the convex combination approach, relative to the affine combination approach, is given by

$$J_{\min}^c - J_{\min}^a = \begin{cases} 0 & \boldsymbol{w}_o^a \geq 0 \\ \left|w_o^{a(1)}\right|^2 \zeta & w_o^{a(1)} < 0 \\ \left|w_o^{a(2)}\right|^2 \zeta & w_o^{a(2)} < 0 \end{cases} \tag{12}$$

where $\zeta = R_{11} + R_{22} - 2R_{12}$ and R_{ij} is the element of $\boldsymbol{R}$ in row i and column j.

2) $m = 3$ Case: When there are three adaptive branches to be combined, we can define the optimal convex combination weights in terms of the following six polyhedral regions:

$$\begin{aligned} P_i &= \left\{\boldsymbol{w} | \boldsymbol{e}_i^T\boldsymbol{w} \leq 0, \boldsymbol{h}_{ij}^T\boldsymbol{w} \geq 0 \quad 1 \leq j \neq i \leq 3\right\}, \quad i = 1, \ldots 3 \\ P_{ij} &= \left\{\boldsymbol{w} | \boldsymbol{h}_{ij}^T\boldsymbol{w} \leq 0, \boldsymbol{h}_{ji}^T\boldsymbol{w} \leq 0\right\}, \quad 1 \leq i \leq 2, i \leq j \leq 3 \end{aligned}$$

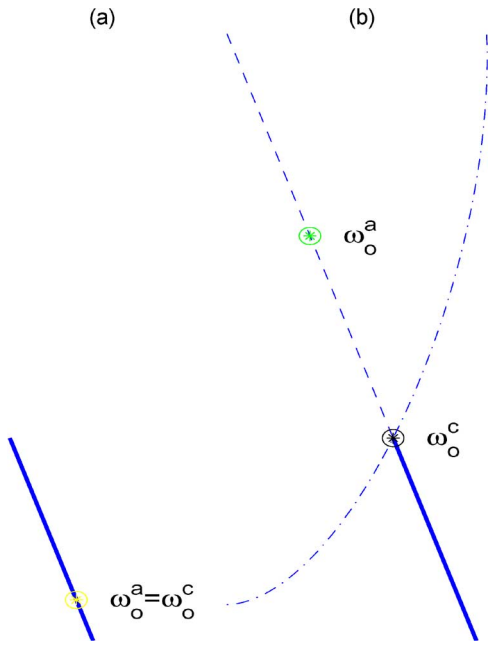

Fig. 2. Optimal mixture weights for affine versus convex combination of two adaptive filters.

where

$$\begin{aligned} \boldsymbol{h}_{12}^T &= [R_{33} + R_{12} - R_{13} - R_{23} \quad R_{22} + R_{33} - 2R_{23} \quad 0] \\ \boldsymbol{h}_{13}^T &= [R_{22} + R_{13} - R_{23} - R_{12} \quad 0 \quad R_{22} + R_{33} - 2R_{23}] \\ \boldsymbol{h}_{21}^T &= [R_{11} + R_{33} - 2R_{13} \quad R_{33} + R_{12} - R_{13} - R_{23} \quad 0] \\ \boldsymbol{h}_{23}^T &= [0 \quad R_{11} + R_{23} - R_{12} - R_{13} \quad R_{11} + R_{33} - 2R_{13}] \\ \boldsymbol{h}_{31}^T &= [R_{11} + R_{22} - 2R_{12} \quad 0 \quad R_{33} + R_{13} - R_{12} - R_{23}] \\ \boldsymbol{h}_{32}^T &= [0 \quad R_{11} + R_{22} - 2R_{12} \quad R_{11} + R_{23} - R_{12} - R_{13}] \end{aligned}$$

and $\boldsymbol{e}_i$ is the unit vector for the ith coordinate axis. Based on these definitions, the expression for the optimal convex combination coefficients and corresponding relative MSE levels

$$\boldsymbol{w}_o^c = \begin{cases} \boldsymbol{w}_o^a - \boldsymbol{w}_o^{a(i)}\boldsymbol{q}_i & \boldsymbol{w}_o^a \in P_i \\ \boldsymbol{e}_3 & \boldsymbol{w}_o^a \in P_{12} \\ \boldsymbol{e}_2 & \boldsymbol{w}_o^a \in P_{13} \\ \boldsymbol{e}_1 & \boldsymbol{w}_o^a \in P_{23} \\ \boldsymbol{w}_o^a & \text{otherwise} \end{cases}$$

$$J_{\min}^c - J_{\min}^a = \begin{cases} \left|\boldsymbol{w}_o^{a(i)}\right|^2 \boldsymbol{q}_i^T\boldsymbol{R}\boldsymbol{q}_i & \boldsymbol{w}_o^a \in P_i \\ \|\boldsymbol{w}_o^a - \boldsymbol{e}_3\|_{\boldsymbol{R}}^2 & \boldsymbol{w}_o^a \in P_{12} \\ \|\boldsymbol{w}_o^a - \boldsymbol{e}_2\|_{\boldsymbol{R}}^2 & \boldsymbol{w}_o^a \in P_{13} \\ \|\boldsymbol{w}_o^a - \boldsymbol{e}_1\|_{\boldsymbol{R}}^2 & \boldsymbol{w}_o^a \in P_{23} \\ 0 & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} \boldsymbol{q}_1 &= \left[1 \quad \frac{R_{13} + R_{23} - R_{33} - R_{12}}{R_{22} + R_{33} - 2R_{23}} \quad \frac{R_{12} + R_{23} - R_{22} - R_{13}}{R_{22} + R_{33} - 2R_{23}}\right]^T \\ \boldsymbol{q}_2 &= \left[\frac{R_{23} + R_{13} - R_{33} - R_{12}}{R_{11} + R_{23} - 2R_{13}} \quad 1 \quad \frac{R_{21} + R_{13} - R_{11} - R_{23}}{R_{11} + R_{33} - 2R_{13}}\right]^T \\ \boldsymbol{q}_3 &= \left[\frac{R_{23} + R_{12} - R_{22} - R_{13}}{R_{11} + R_{22} - 2R_{12}} \quad \frac{R_{13} + R_{12} - R_{11} - R_{23}}{R_{11} + R_{22} - 2R_{12}} \quad 1\right]^T. \end{aligned}$$

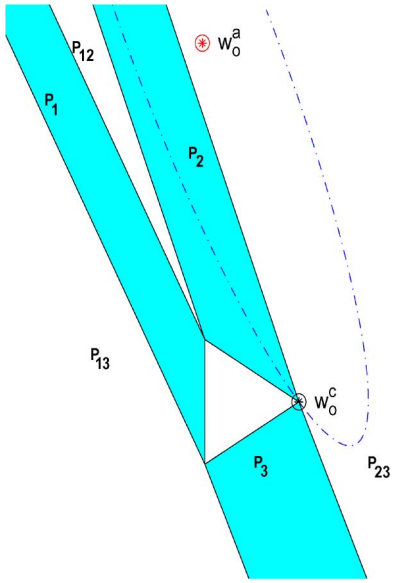

Fig. 3. Optimal mixture weights for affine versus convex combination. Three adaptive branches are combined in the figure.

As an example, we illustrate the projection problem corresponding to obtaining optimal convex combination coefficients from affine combination coefficients in Fig. 3, where we plot the corresponding combination weights and polyhedral regions in $\mathbb{R}^3$. In this figure, the unit simplex corresponding to the convex combination weights and the polyhedral partitions that specify whether the affine combination weights are mapped to the sides or the corners of the unit simplex are shown on the plane corresponding to the affine constrained combination. An example case, where $\boldsymbol{w}_o^a$ is in P_{23} and mapped to $\boldsymbol{e}_1$ is shown, where the level set of the weighted distance cost function is also drawn to demonstrate the nature of projection.

3) $m \geq 3$*:* The polyhedral partitioning approach introduced for $m = 3$ case can be extended for more general m values. This leads to rather complicated expressions for large values of m. However, irrespective of the value of m chosen, due to the inclusion ordering of the corresponding constraint sets, we can always write

$$J_{\min} \leq J_{\min}^a \leq J_{\min}^c. \tag{13}$$

In the following sections, we introduce adaptive methods to train the combination weights. We first investigate training the unconstrained linear combination weights using the LMS update and then the RLS update in Section IV. We next continue using the LMS and the RLS updates to train the affine combination weights in Section V. When the combination weights are constrained to be convex, we use the stochastic gradient update after a variable transformation using the sigmoid nonlinearity from [2] and [3] in Section VI.

IV. Adaptive Methods to Update Unconstrained Linear Mixture Weights

In this section, we first use the LMS update to adapt the unconstrained linear combination weights in order to minimize the mean of the overall quadratic estimation error $e^2(t)$. We then use the RLS update instead of the LMS update.

A. Adapting Linear Mixture Weights Using the LMS Update

Given that $\boldsymbol{y}(t) = [\hat{d}_1(t), \ldots, \hat{d}_m(t)]^T$, the LMS update on the combination weights is derived using the gradient of the instantaneous squared error $e^2(t)$ to update the combination weights, i.e.,

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \frac{1}{2}\mu_{\text{lin}} \nabla_w \, e^2(t).$$

Applying the LMS update to the outputs of the constituent filters yields

$$\begin{aligned} \hat{d}(t) &= \boldsymbol{w}^T(t)\boldsymbol{y}(t), \\ e(t) &= d(t) - \hat{d}(t) \qquad (14) \\ \boldsymbol{w}(t+1) &= \boldsymbol{w}(t) + \mu_{\text{lin}} e(t)\boldsymbol{y}(t) \qquad (15) \end{aligned}$$

where $\mu_{\text{lin}} > 0$ is the learning rate of the mixture update. For this LMS update, using the optimal weight vector (3) that minimizes the final MSE, we define *a priori*, *a posteriori*, and orthogonal errors as

$$\begin{aligned} e_a(t) &\triangleq [\boldsymbol{w}_o - \boldsymbol{w}(t)]^T \boldsymbol{y}(t) \qquad (16) \\ e_p(t) &\triangleq [\boldsymbol{w}_o - \boldsymbol{w}(t+1)]^T \boldsymbol{y}(t) \qquad (17) \\ e_o(t) &= d(t) - \boldsymbol{w}_o^T \boldsymbol{y}(t) \qquad (18) \end{aligned}$$

where we also have from (17) and (18) $e(t) = e_o(t) + e_a(t)$. We next present the relation between *a priori* and *a posteriori* errors using (15), (16) and (17), $e_p(t) = e_a(t) - \mu_{\text{lin}} e(t) \|\boldsymbol{y}(t)\|^2$, as well as the energy conservation relation [7]

$$\begin{aligned} E\left[\|\boldsymbol{w}_o - \boldsymbol{w}(t+1)\|^2\right] + E\left[\frac{e_a^2(t)}{\|\boldsymbol{y}(t)\|^2}\right] \\ = E\left[\|\boldsymbol{w}_o - \boldsymbol{w}(t)\|^2\right] + E\left[\frac{e_p^2(t)}{\|\boldsymbol{y}(t)\|^2}\right] \qquad (19) \end{aligned}$$

which yields

$$\begin{aligned} E\left[\|\boldsymbol{w}_o - \boldsymbol{w}(t+1)\|^2\right] + \mu_{\text{lin}} E\left[\|\boldsymbol{y}(t)\|^2 e^2(t)\right] \\ = E\left[\|\boldsymbol{w}_o - \boldsymbol{w}(t)\|^2\right] + 2E\left[e_a(t)e(t)\right]. \end{aligned}$$

As shown in the Appendix, $\lim_{t\to\infty} E[\boldsymbol{w}(t)] = \boldsymbol{w}_o$, and if we assume that in the limit $\lim_{t\to\infty} E[\|\boldsymbol{w}_o - \boldsymbol{w}(t+1)\|^2] = \lim_{t\to\infty} E[\|\boldsymbol{w}_o - \boldsymbol{w}(t)\|^2]$, i.e., the update is mean-square convergent, then we get the variance relation [7]: $\lim_{t\to\infty} \mu_{\text{lin}} E[\|\boldsymbol{y}(t)\|^2 e^2(t)] = 2E[e_a(t)e(t)]$, as $t \to \infty$. We note that in the limit assuming asymptotic stationarity, we only have $E[\boldsymbol{y}(t)e_o(t)] = \boldsymbol{0}$, however, we next make the assumption that $e_o(t)$ is independent of $\boldsymbol{y}(t)$ and $\boldsymbol{w}(t)$ (which is plausible as the filters are near convergence [7]). After this assumption and straightforward algebra, the steady-state MSE of the mixture stage algorithm is given as

$$\lim_{t\to\infty} E\left[e^2(t)\right] = J_{\min} + \frac{\mu_{\text{lin}} J_{\min} \text{tr}(\boldsymbol{R})}{2 - \mu_{\text{lin}} \text{tr}(\boldsymbol{R})}. \tag{20}$$

Thus, we observe that although the unconstrained linear combination theoretically could achieve the MMSE, i.e., $J_{\min}$, we

have excess MSE due to stochastic gradient update used in the adaptation.

B. Adapting Linear Mixture Weights Using the RLS Update

In this section, linear combination weights are trained using the RLS update to minimize the square error of the overall combination. If we write the desired signal as $d(t) = \boldsymbol{w}_o^T \boldsymbol{y}(t) + e_o(t)$, where $\lim_{t\to\infty} E[\boldsymbol{y}(t)e_o(t)] = \boldsymbol{0}$ by definition, the RLS update is given by

$$\begin{aligned}
\hat{d}(t) &= \boldsymbol{w}^T(t)\boldsymbol{y}(t) \\
e(t) &= d(t) - \hat{d}(t) \\
\boldsymbol{K}^{-1}(t+1) &= \lambda_{\text{lin}}^{-1}\left[\boldsymbol{K}^{-1}(t) - \frac{\lambda_{\text{lin}}^{-1}\boldsymbol{K}^{-1}(t)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\boldsymbol{K}^{-1}(t)}{1+\lambda_{\text{lin}}^{-1}\boldsymbol{y}^T(t)\boldsymbol{K}^{-1}(t)\boldsymbol{y}(t)}\right] \\
\boldsymbol{w}(t+1) &= \boldsymbol{w}(t) + \boldsymbol{K}^{-1}(t+1)e(t)\boldsymbol{y}(t)
\end{aligned} \tag{21}$$

where $0 < \lambda_{\text{lin}} \leq 1$ is the forgetting factor, $\boldsymbol{K}(t) = \sum_{l=1}^{t} \lambda_{\text{lin}}^{t-l}\boldsymbol{y}(l)\boldsymbol{y}^T(l) + \lambda_{\text{lin}}^{t}\epsilon\boldsymbol{I}$ is the estimated correlation matrix, $\boldsymbol{K}(0) = \epsilon\boldsymbol{I}$, ϵ is a small positive number and $\boldsymbol{I}$ is an appropriate sized identity matrix. We next define *a priori* error $e_a(t)$ and *a posteriori* error $e_p(t)$ for the mixture stage as in (16) and (17), respectively. Using the energy relation and the variance relation for the RLS algorithm derived in [7], and making the assumption that $\lim_{t\to\infty} E[\boldsymbol{K}^{-1}(t)] = \boldsymbol{R}^{-1}(1-\lambda_{\text{lin}})$, we can show that of [7, p. 265]

$$\lim_{t\to\infty} E\left[e^2(t)\right] = J_{\min} + J_{\min}\frac{(1-\lambda_{\text{lin}})m}{2-(1-\lambda_{\text{lin}})m}. \tag{22}$$

V. Adaptive Methods to Update Affine Mixture Weights

When the weights are constrained to be affine, we can use the following parametrization involving $m-1$ unconstrained weights:

$$\begin{aligned}
w^{(i)}(t) &= z^{(i)}(t), \quad i = 1,\ldots,m-1 \\
w^{(m)}(t) &= 1 - \sum_{i=1}^{m-1} z^{(i)}(t).
\end{aligned}$$

Here, the $m-1$ dimensional vector $\boldsymbol{z}(t) \triangleq [z^{(1)}(t), \ldots, z^{(m-1)}(t)]^T$ is the unconstrained weight vector. Hence, we transformed the constrained optimization problem into an unconstrained quadratic optimization problem. We note that when we combine just two filters, this update corresponds to the stochastic gradient update given in (45) of [4]. Observing that $e(t) = d(t) - \boldsymbol{w}^T(t)\boldsymbol{y}(t)$ and if we use $\boldsymbol{z}(t)$ as our weight vector, we have

$$\begin{aligned}
e(t) &= d(t) - \left[\hat{d}_1(t),\ldots,\hat{d}_{m-1}(t)\right]\boldsymbol{z}(t) - \left(1-\boldsymbol{1}^T\boldsymbol{z}(t)\right)\hat{d}_m(t) \\
&= \left[d(t) - \hat{d}_m(t)\right] - \boldsymbol{z}^T(t)\vec{\delta}(t)
\end{aligned}$$

where $\boldsymbol{\delta}(t) \triangleq [\hat{d}_1(t) - \hat{d}_m(t), \ldots, \hat{d}_{m-1}(t) - \hat{d}_m(t)]^T$. Hence, we have an adaptive filter problem with $[d(t) - \hat{d}_m(t)]$ as the desired signal and $\boldsymbol{\delta}(t)$ as the input vector. We next define $\boldsymbol{\Gamma}(t) = E[\boldsymbol{\delta}(t)\boldsymbol{\delta}^T(t)]$, $\boldsymbol{\Gamma} = \lim_{t\to\infty}\boldsymbol{\Gamma}(t)$ and $\boldsymbol{\gamma}(t) = E[\boldsymbol{z}(t)(d(t) - \hat{d}_m(t))]$, $\boldsymbol{\gamma} = \lim_{t\to\infty}\boldsymbol{\gamma}(t)$ when the limits exist. For this affine combination, the final MMSE is reached when $\boldsymbol{z}_o = \boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}$, which, along with $w^{(m)}(t) = 1 - \sum_{i=1}^{m-1} z^{(i)}(t)$, should be equal to $\boldsymbol{w}_o^a$, since $\boldsymbol{w}_o^a$ is the optimal affine combination weight vector. The final MMSE of this filter is given by $J_{\min}^a$.

In terms of adaptation strategies for the reduced dimensional unconstrained parametrization, we first look at LMS update versions in Section IV-A, which is along the same lines as the algorithm for the two branch version in [4]. Next, we train the affine combination weights using the RLS update as in IV-B.

A. Adapting Affine Mixture Weights Using the LMS Update

Since we transformed affine constrained weights $\boldsymbol{w}(t)$ into unconstrained weights $\boldsymbol{z}(t)$, we apply the LMS update directly on $\boldsymbol{z}(t)$. Similar analysis for the MSE can be done as in Section IV-A by using $(d(t) - \hat{d}_m(t))$ instead of $d(t)$ and $\boldsymbol{\delta}(t)$ instead of $\boldsymbol{y}(t)$. Hence, we will only provide a brief explanation, update equations and the final MSE.

The update to minimize the variance of $e(t)$ is given by

$$\begin{aligned}
\boldsymbol{z}(t+1) &= \boldsymbol{z}(t) - \frac{1}{2}\mu_{\text{aff}}\nabla_z\, e^2(t) \\
&= \boldsymbol{z}(t) + \mu_{\text{aff}}e(t)\boldsymbol{\delta}(t),
\end{aligned}$$

by new definitions of desired signal and input vectors. We next define *a priori*, *a posteriori* and estimation errors as in (16), (17), and (18), but using $\boldsymbol{z}(t)$ and $\boldsymbol{z}_o = \boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}$ instead of $\boldsymbol{y}(t)$ and $\boldsymbol{w}_o$, respectively. After these new definitions, we derive energy conservation and variance relations [7] to obtain

$$\lim_{t\to\infty} E\left[e^2(t)\right] = J_{\min}^a + \frac{\mu_{\text{aff}}J_{\min}^a\text{tr}(\boldsymbol{\Gamma})}{2-\mu_{\text{aff}}\text{tr}(\boldsymbol{\Gamma})}. \tag{23}$$

This expression is the final MSE when the affine combination coefficients are trained using the LMS update. In order to compare the excess MSE due to training in this case to the excess MSE expressions for the unconstrained case in (20), we note that

$$\boldsymbol{\delta}(t) = \boldsymbol{\Phi}\boldsymbol{y}(t) \qquad \text{where} \quad \boldsymbol{\Phi} \triangleq [\boldsymbol{I}_{m-1} \quad -\boldsymbol{1}], \tag{24}$$

and therefore, $\boldsymbol{\Gamma} = \boldsymbol{\Phi}\boldsymbol{R}\boldsymbol{\Phi}^T$. If we partition $\boldsymbol{R}$ in the form

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{12}^T & \boldsymbol{R}_{22} \end{bmatrix} \tag{25}$$

where $\boldsymbol{R}_{11} \in \mathbb{R}^{m-1\times m-1}$, $\boldsymbol{R}_{12} \in \mathbb{R}^{m-1\times 1}$ and $\boldsymbol{R}_{22} \in \mathbb{R}$, then we have $\boldsymbol{\Gamma} = \boldsymbol{R}_{11} - \boldsymbol{R}_{12}\boldsymbol{1}^T - \boldsymbol{1}\boldsymbol{R}_{12}^T + \boldsymbol{R}_{22}\boldsymbol{1}\boldsymbol{1}^T$. Therefore, $\text{tr}(\boldsymbol{\Gamma}) = \text{tr}(\boldsymbol{R}) + (m-2)\boldsymbol{R}_{22} - 2\boldsymbol{1}^T\boldsymbol{R}_{12}$. As a result, any inequality relations between the traces of $\boldsymbol{R}$ and $\boldsymbol{\Gamma}$ would depend on the correlation among the adaptive filtering branches.

B. Adapting Affine Mixture Weights Using the RLS Update

We next apply the RLS update on $\boldsymbol{z}(t)$ using the desired signal $d(t) - \hat{d}_m(t)$ and the input vector $\boldsymbol{\delta}(t)$. The RLS update for the affine weights are given as

$$\boldsymbol{z}(t+1) = \boldsymbol{z}(t) + \boldsymbol{\Lambda}^{-1}(t+1)e(t)\boldsymbol{\delta}(t),$$

with $e(t) = d(t) - \hat{d}_m(t) - \boldsymbol{z}^T(t)\boldsymbol{\delta}(t)$ and

$$\boldsymbol{\Lambda}^{-1}(t+1) = \lambda_{\text{aff}}^{-1}\left[\boldsymbol{\Lambda}^{-1}(t) - \frac{\lambda_{\text{aff}}^{-1}\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{\delta}(t)\boldsymbol{\delta}^T(t)\boldsymbol{\Lambda}^{-1}(t)}{1+\lambda_{\text{aff}}^{-1}\boldsymbol{\delta}^T(t)\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{\delta}(t)}\right]$$

where $0 < \lambda_{\text{aff}} \leq 1$ is the forgetting factor, $\boldsymbol{\Lambda}(t) = \sum_{l=1}^{t} \lambda_{\text{aff}}^{t-l} \boldsymbol{\delta}(l)\boldsymbol{\delta}^T(l) + \lambda_{\text{aff}}^{t} \epsilon \boldsymbol{I}$ is the estimated correlation matrix, $\boldsymbol{\Lambda}(0) = \epsilon I$, ϵ is a small positive number. Following the same lines as in Section IV-B, we derive the final MSE of this algorithm as

$$\lim_{t\to\infty} E\left[e^2(t)\right] = J_{\min}^a + J_{\min}^a \frac{(1-\lambda_{\text{aff}})(m-1)}{2-(1-\lambda_{\text{aff}})(m-1)}. \quad (26)$$

VI. An Adaptive Method to Update Convex Mixture Weights

In this section, we study the convex combination approach using the stochastic gradient update illustrated in [3], which can be considered as a multiple filter extension of [2].

A. Adapting Convex Mixture Weights Using a Stochastic Gradient Update

In this section, we consider the algorithm in [3] which is a multiple order extension of the algorithm in [2]. The results derived here for the algorithm of [3] can be readily extended to [2] or other variable transformations as used in [3]. In [2], [3], the optimization problem with convex combination constraints (i.e., the unit simplex constraint set) is transformed into an unconstrained optimization problem by a change of variables. Then, a stochastic gradient update is applied to this new set of unconstrained variables to minimize the final estimation error. Here, the convex combination weights $w^{(i)}(t)$, $\mathbf{1}^T\boldsymbol{w}(t) = 1$, are reparametrized using

$$w^{(i)}(t) = \frac{e^{-z^{(i)}(t)}}{\sum_{k=1}^{m} e^{-z^{(k)}(t)}} \quad (27)$$

such that $z^{(i)}(t)$ are the unconstrained variables. We denote these unconstrained weights by $\boldsymbol{z}(t) \triangleq [z^{(1)}(t), \ldots, z^{(m)}(t)]^T$. The unconstrained weights $\boldsymbol{z}(t)$ are trained using a stochastic gradient update to minimize the instantaneous squared error $e^2(t)$ such that

$$\boldsymbol{z}(t+1) = \boldsymbol{z}(t) - \frac{1}{2}\mu_{\text{cvx}} \nabla_z\, e^2(t)$$

where $\mu_{\text{cvx}} > 0$. However, unlike [8] we use a single μ_{cvx} instead of using different μ_{cvx}'s for each dimension $i = 1, \ldots, m$ to be consistent with the other adaptive updates considered in this paper. Hence, the final update is given by

$$\begin{aligned}\boldsymbol{z}(t+1) &= \boldsymbol{z}(t) - \mu_{\text{cvx}} \left\{\nabla_z \boldsymbol{w}(t)\right\}^T \left[e(t) \nabla_{\boldsymbol{w}}\, e(t)\right] \\ &= \boldsymbol{z}(t) + \mu_{\text{cvx}} \left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}\left(\boldsymbol{w}(t)\right)\right\} \left[e(t)\boldsymbol{y}(t)\right]\end{aligned}$$

where $\nabla_z \boldsymbol{w}(t) = \{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}(\boldsymbol{w}(t))\}$ and $\nabla_{\boldsymbol{w}} e(t) = -\boldsymbol{y}(t)$. For convergence analysis, we uphold the assumption introduced in [3] such that the variance of $\boldsymbol{z}(t)$ is zero at the convergence such that $E[\boldsymbol{z}(t)] \approx \boldsymbol{z}(t)$, then as $t \to \infty$

$$\boldsymbol{z}(t+1) = \boldsymbol{z}(t) + \mu_{\text{cvx}} \left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}\left(\boldsymbol{w}(t)\right)\right\} E\left[e(t)\boldsymbol{y}(t)\right].$$

Since at convergence $\lim_{t\to\infty} \boldsymbol{z}(t+1) = \lim_{t\to\infty} \boldsymbol{z}(t)$, the minimizing $\boldsymbol{z}(t)$ should satisfy

$$\begin{aligned}&\left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}\left(\boldsymbol{w}(t)\right)\right\} E\left[e(t)\boldsymbol{y}(t)\right] \\ &\quad = \left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}\left(\boldsymbol{w}(t)\right)\right\} \left[\boldsymbol{p} - \boldsymbol{R}\boldsymbol{w}(t)\right] \\ &\quad = \left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}\left(\boldsymbol{w}(t)\right)\right\} \boldsymbol{R}\left[\boldsymbol{w}_o - \boldsymbol{w}(t)\right] \\ &\quad = \mathbf{0}.\end{aligned}$$

The converged $\boldsymbol{z}(t)$ satisfies,

$$\left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}\left(\boldsymbol{w}(t)\right)\right\} \bar{\boldsymbol{R}}\left[\boldsymbol{w}_o - \boldsymbol{w}(t)\right] = \mathbf{0}, \quad (28)$$

with the added constraints on $\boldsymbol{w}(t)$ such that $\boldsymbol{w}(t)^T\mathbf{1} = 1$ and $w^{(i)} \geq 0$, $i = 1, \ldots, m$.

It can be shown that $\boldsymbol{w}_o^a$ given in (5) satisfies (28) if all entries of $\boldsymbol{w}_o^a$ are nonnegative, i.e., $\boldsymbol{w}_o^a = \boldsymbol{w}_o^c$. Clearly, since the original minimization problem is convex and the sigmoid transformation in (27) is uniformly decaying (or increasing), the minimal point $\boldsymbol{w}^c$ of the original cost function in (8) is also the minimal point for the transformed cost function. Although the minimal point is unique for the original cost function, since the sigmoid (27) is many-to-one mapping, there are infinite number of points in $\boldsymbol{z}(t)$ domain that correspond to $\boldsymbol{w}^c$ and achieves $J_{\min}^c$, e.g., for any τ, $\boldsymbol{z} + \tau\mathbf{1}$ map to the same vector where $\boldsymbol{z}$ maps. We also note that, $\boldsymbol{w}(t) = [0, \ldots, 1, \ldots, 0]^T$, i.e., any vector $\boldsymbol{w}(t)$ where all the entries except a single entry is equal to one, also satisfies (28) such that $\{\boldsymbol{w}(t)\boldsymbol{w}(t)^T - \text{diag}(\boldsymbol{w}(t))\} = \mathbf{0}$. These points are the saddle points of the sigmoid cost [28].

VII. Tracking Analysis for Adaptive Combination Methods

In this section, we investigate the tracking performance of the combination methods introduced in this paper in a nonstationary environment. After a general comment, we study a particular model for the statistics of the desired data, commonly used to model nonstationarity in tracking analysis [7]. We note that the derivations in the previous sections solely relied on the auto- and cross-correlations between *a priori* errors of the constituent filters. Hence, for nonstationary environments, in the cases where $\lim_{t\to\infty} \boldsymbol{R}^{-1}(t)\boldsymbol{p}(t)$ exists, the previous analysis will still hold, since the definitions of *a priori* errors have not changed. Based on the new values of the optimal weight vectors and converged statistics, one only needs to change the previous results for the final MSEs accordingly given in (20), (22), (23), and (26).

As a widely studied illustrative example [2], [5], [7], we consider the case where the constituent filters have the same length, i.e., $\hat{d}_i(t) = \boldsymbol{w}_i^T(t)\boldsymbol{u}(t)$, $\boldsymbol{w}_i(t), \boldsymbol{u}(t) \in \mathbb{R}^s$. Furthermore, in the generation of the desired signal $d(t)$, we assume a random walk model [7] for $\boldsymbol{w}_o(t)$ such that

$$\boldsymbol{w}_o(t+1) - \boldsymbol{w}_o(0) = \alpha\left[\boldsymbol{w}_o(t) - \boldsymbol{w}_o(0)\right] + \boldsymbol{q}(t).$$

Here, $d(t) = \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) + n(t)$, where $\boldsymbol{q}(t) \in \mathbb{R}^s$ is an i.i.d. zero mean vector process with covariance matrix $E[\boldsymbol{q}(t)\boldsymbol{q}^T(t)] = \boldsymbol{Q}$, $\boldsymbol{w}_o(0)$ is the initial weight vector (as well as the mean of this vector process), $\boldsymbol{u}(t) \in \mathbb{R}^s$ is a stationary input vector process and $n(t)$ is an i.i.d. zero mean noise process with variance σ_n^2

and $\boldsymbol{q}(t)$, $\boldsymbol{u}(t)$ and $n(t)$ are mutually independent. Usually, $0 \ll |\alpha| \leq 1$. In this case, *a priori*, *a posteriori* and estimation errors for the constituent filters are given as

$$\begin{aligned} e_{a,i}(t) &\triangleq [\boldsymbol{w}_o(t) - \boldsymbol{w}_i(t)]^T \boldsymbol{u}(t) \\ e_{p,i}(t) &\triangleq [\boldsymbol{w}_o(t) - \boldsymbol{w}_i(t+1)]^T \boldsymbol{u}(t) \\ e_i(t) &= e_{a,i}(t) + n(t). \end{aligned}$$

Under this data model, it can be shown that limits for the autocorrelation matrix of *a priori* errors, $\boldsymbol{J}$, exists for most commonly used adaptive methods, including the LMS update [2], [7], the RLS update and several unsupervised updates [5] (when $|\alpha| = 1$, and can be readily extended to the case when $|\alpha| < 1$ using results from [7]). By these definitions, as the input to the combination stage, we have

$$\boldsymbol{y}(t) = \begin{bmatrix} \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) - e_{a,1}(t) \\ \vdots \\ \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) - e_{a,m}(t) \end{bmatrix}.$$

The variance of clean part of the desired signal, $g(t) \triangleq \boldsymbol{w}_o^T(t)\boldsymbol{u}(t)$, is given as

$$\sigma_g^2(t) = \operatorname{tr}\left\{E\left[\boldsymbol{w}_o(t)\boldsymbol{w}_o^T(t)\right] E\left[\boldsymbol{u}(t)\boldsymbol{u}^T(t)\right]\right\}$$

where

$$\begin{aligned} E\left[\boldsymbol{w}_o(t)\boldsymbol{w}_o^T(t)\right] = \alpha^2 E\left[\boldsymbol{w}_o(t-1)\boldsymbol{w}_o^T(t-1)\right] \\ +(1-\alpha^2)\boldsymbol{w}_o(0)\boldsymbol{w}_o^T(0) + \boldsymbol{Q}, \end{aligned}$$

i.e., the variance $\sigma_g^2(t)$ is time-varying, unlike previous sections. When $|\alpha| < 1$,

$$\lim_{t\to\infty} E\left[\boldsymbol{w}_o(t)\boldsymbol{w}_o^T(t)\right] = \frac{\boldsymbol{Q} + (1-\alpha^2)\boldsymbol{w}_o(0)\boldsymbol{w}_o^T(0)}{1-\alpha^2}.$$

However, when $|\alpha| = 1$, $E[\boldsymbol{w}_o(t)\boldsymbol{w}_o^T(t)]$ diverges. Hence, the variance of $g(t)$ is increasing, yielding the covariance matrix of $\boldsymbol{y}(t)$ to be unbounded when $|\alpha| = 1$. Hence, we need to consider the two cases, $|\alpha| = 1$ and $|\alpha| < 1$, separately for tracking analysis.

When $|\alpha| = 1$, the cross-correlation matrix for the *a priori* errors, $\boldsymbol{J}$, can be shown to be convergent [5]. For affine or convex combinations, since $\mathbf{1}^T\boldsymbol{w}(t) = 1$, i.e., the estimation is unbiased, we have

$$\begin{aligned} e(t) &= d(t) - \boldsymbol{w}^T(t)\boldsymbol{y}(t) \\ &= g(t) + n(t) \\ &\quad - \left(\boldsymbol{w}^T(t)\mathbf{1}g(t) - \boldsymbol{w}^T(t)\left[e_{a,1}(t), \ldots, e_{a,m}(t)\right]^T\right) \\ &= g(t) + n(t) - \left(g(t) - \left[e_{a,1}(t), \ldots, e_{a,m}(t)\right]\boldsymbol{w}(t)\right) \\ &= n(t) + \left[e_{a,1}(t), \ldots, e_{a,m}(t)\right]\boldsymbol{w}(t). \end{aligned}$$

Thus, as demonstrated in [2], [5], the effect of the unboundedness of $g(t)$ need not affect the convergence or final MSE analysis. For unconstrained linear combinations, although $\boldsymbol{w}_o(t)$ can be shown to be convergent, we note that from (4)

$$\mathbf{1}^T\boldsymbol{w}_o(t) = \frac{\mathbf{1}^T\boldsymbol{J}^{-1}\mathbf{1}}{\sigma_g^{-2}(t) + (\mathbf{1}^T\boldsymbol{J}^{-1}\mathbf{1})},$$

yielding as $t \to \infty$, $\mathbf{1}^T\boldsymbol{w}_o(t) \approx 1$ since $\sigma_g^2(t)$ diverges. Hence, the optimal unconstrained linear combination coincides with optimal affine combination.

For values $|\alpha| < 1$, the variance of $g(t)$ is convergent and one can derive the corresponding $\boldsymbol{J}$ and $\boldsymbol{G}$ values as in [7]. After this point the derivations exactly follow along lines similar to those given in previous sections with the updated cross-correlation matrix and vectors.

VIII. Simulations

In this section, we demonstrate the performance of mixture algorithms through simulations using both stationary and nonstationary data. We observe that the combination structures provide improved performance over the constituent filters, especially under low SNR conditions both in stationary and nonstationary environments in these simulations. We also observed close agreement between the simulations and the theoretical results introduced in this paper under different scenarios and algorithmic parameters.

The first set of experiments involve system identification with different order linear filters as the constituent algorithms. To observe the accurateness of the results introduced in (3), (5), (20), (22), (23), and (26) under different algorithmic parameters and SNRs, the desired signal as well as the system parameters are selected as follows. First a third-order linear filter, $\boldsymbol{w}_o = [0.32, -0.48, -0.23]^T$, is chosen, where each entry is selected randomly from $[-1, 1]$. The underlying signal is generated using the data model $d(t) = \tau\boldsymbol{w}_o^T\boldsymbol{u}(t) + n(t)$, where $\boldsymbol{u}(t)$ is an i.i.d. Gaussian vector process with zero mean and unit variance entries, i.e., $E[\boldsymbol{u}(t)\boldsymbol{u}^T(t)] = \boldsymbol{I}$, $n(t)$ is an i.i.d. Gaussian noise process with zero mean and variance $E[n^2(t)] = 0.01$, and τ is a positive scalar to control SNR. Hence, the SNR of the desired signal is given by $\mathrm{SNR} \triangleq 10\log(E[\tau^2(\boldsymbol{w}_o^T\boldsymbol{u}(t))^2]/0.01) = 10\log(\tau^2\|\boldsymbol{w}_o\|^2/0.01)$. By changing τ, we simulate the performance of the combination algorithms under different SNRs. We select the constituent algorithms as linear filters from first-order through fifth-order, all using the LMS update to train their weight vectors $\boldsymbol{w}_i(t) \in \mathbb{R}^i$, $i = 1, \ldots, 5$. Here, each filter produces $\hat{d}_i(t) = \boldsymbol{w}_i^T(t)\boldsymbol{u}_i(t)$, where $\boldsymbol{u}_i(t) \in \mathbb{R}^i$. The input vector processes that are fed to the constituent filters and to the desired system model are generated as follows. First, a fifth order i.i.d. Gaussian vector process with zero mean and unit variance, such that $\boldsymbol{u}_5(t) \in \mathbb{R}^5$, is generated. The ith constituent filter uses the first i entries of this vector process as its input, i.e., $\boldsymbol{u}_i(t) = [u^{(1)}(t), \ldots, u^{(i)}(t)]^T$, where $\boldsymbol{u}_5(t) = [u^{(1)}(t), \ldots, u^{(5)}(t)]^T$. Hence, for the desired signal, $d(t)$, generation, we use $\boldsymbol{u}(t) = \boldsymbol{u}_3(t)$. The learning rate of the LMS update for each constituent filter is set to $\mu_i = 0.1$, $i = 1, \ldots, 5$. In Fig. 4, we plot the final excess MSEs corresponding to the mixture methods investigated in this paper, i.e., $\lim_{t\to\infty} E[e^2(t)] - E[n^2(t)]$, with respect to the learning parameters and forgetting factors of the mixture algorithms. The simulations are done over 2×10^5 samples, averaged over 100 independent trials. The final MSEs are calculated by averaging the last 7000 samples of each iteration. Fig. 4(a), (b), and (c) shows the excess MSE versus the algorithmic parameters of the corresponding mixture algorithms. The x axis displays

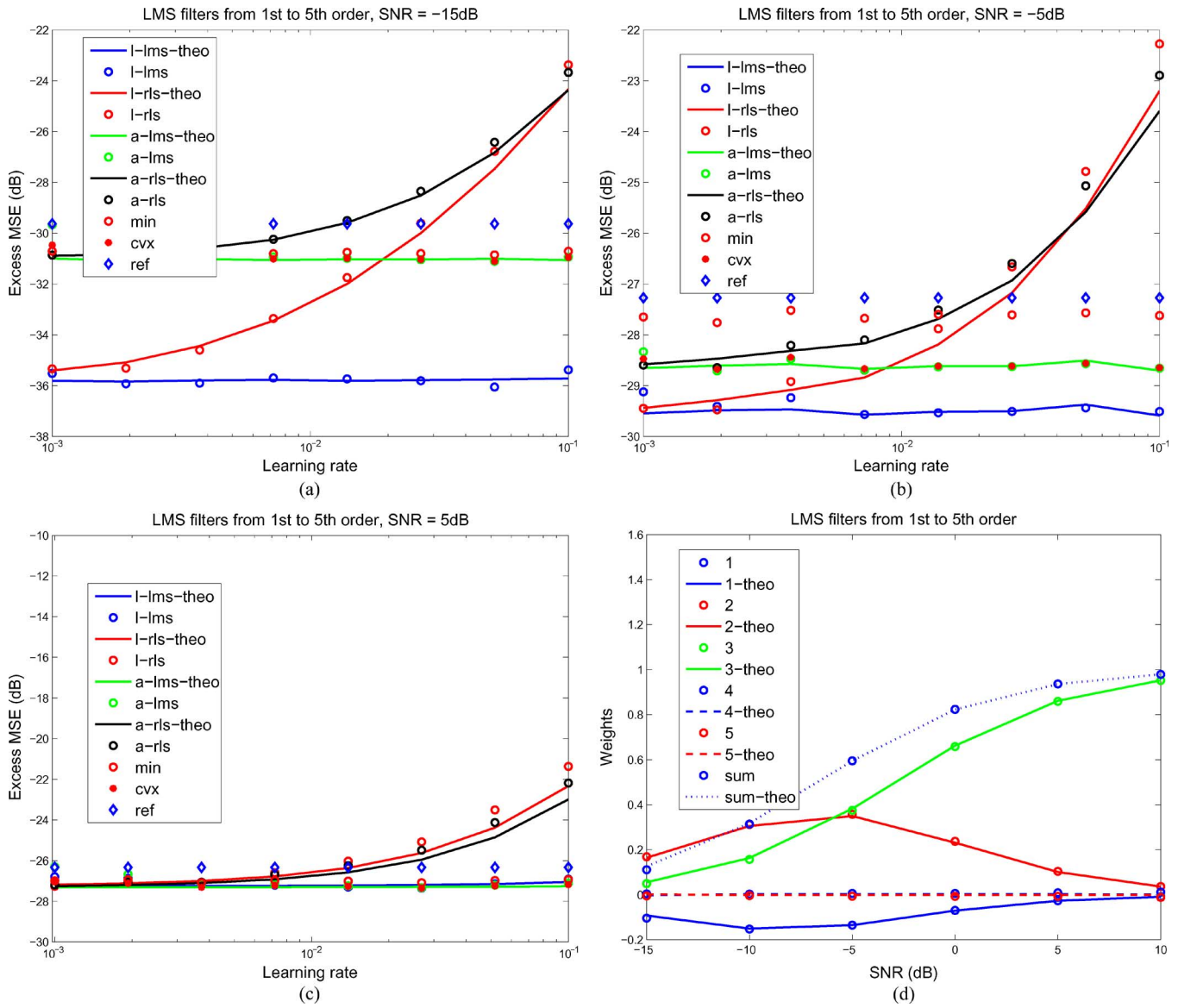

Fig. 4. System identification with constituent filters from first-order to fifth-order linear filters with a third-order desired signal. Here, the x axis represents learning rates μ for the LMS based mixture methods. The forgetting factors of the RLS based algorithms are defined as one minus the learning rate of the LMS based mixture algorithms. The y axis is the excess MSE in dB. (a) $\mathrm{SNR} = -15$ dB. (b) $\mathrm{SNR} = -5$ dB. (c) $\mathrm{SNR} = 5$ dB. (d) The estimated final weights $\lim_{t\to\infty} E[\boldsymbol{w}(t)]$ for the unconstrained linear filter trained using the RLS update and the theoretical $\boldsymbol{w}_o$ estimated using the cross-correlation matrix $\boldsymbol{R}$ from the observed data with respect to SNR.

the corresponding learning rates for $\mu_{\text{lin}} \in [10^{-3}, 10^{-1}]$ ("l-lms") and $\mu_{\text{aff}} \in [10^{-3}, 10^{-1}]$ ("a-lms"). We point out that since the combination algorithms work on possibly highly nonstatitonary outputs generated by the first stage adaptive filters, the final combination may have convergence issues for relatively larger learning rates used in the mixture stage. For display purposes, the x axis also represents the forgetting factors for the RLS based mixture methods, where corresponding forgetting factors are defined as $\lambda \triangleq 1 - \mu$ for each learning rate μ, i.e., $\lambda_{\text{lin}} \in [1 - 10^{-1}, 1 - 10^{-3}]$ ("l-rls") and $\lambda_{\text{aff}} \in [1 - 10^{-1}, 1 - 10^{-3}]$ ("a-rls"). To guarantee the convergence of the convex constrained algorithm studied in Section VI for these simulations, we have selected the learning parameters for the stochastic gradient method of [3] as $\mu_{\text{cvx}} \in [50 \times 10^{-3}, 50 \times 10^{-1}]$ ("cvx"), i.e., the learning parameters are selected 50 times larger than the unconstrained or affine constrained algorithms. Note that for presentation purposes, we plot the steady-state MSE of the convex constrained method with the same x axis range; however, the learning parameters are 50 times the value given in the x-axis. In the same plot, we also show the steady-state MSE for the algorithm from [15] ("ref"), where the "analysis interval" is selected as 40. For the RLS based algorithms, we set $\boldsymbol{K}(0) = 10^{-2}\boldsymbol{I}$, however, note that the value of $\boldsymbol{K}(0)$ does not affect the final results (guaranteed that the mixture stage converges) since we plot the steady-state MSEs after convergence. We repeat the same experiment under three different SNRs including, $\mathrm{SNR} = -15$ dB, -5 dB, and 5 dB. To get the corresponding theoretical results in (20), (22), (23), and (26), we calculate the corresponding $\boldsymbol{R}$ and $\boldsymbol{p}$ from these simulations. In these plots,

we observe a close agreement between the introduced results and the simulations. Note that the results are more accurate for smaller values of learning rates (or larger values of forgetting factors), since it is well known that the theoretical derivations involving the LMS (or the RLS) update based algorithms are more accurate for small μ (or large λ) with the assumptions used for the derivations [7]. To get more accurate results for larger values of μ (or smaller values of λ), one can change the corresponding derivations accordingly [7]. In Fig. 4, we also plot the final excess MSE of the best constituent filter with the smallest excess final MSE ("min"). We observe that the unconstrained linear combination methods outperform the other mixture methods for low SNR values. We observe that the "cvx" algorithm outperforms "ref" algorithm, since it can exploit diversity as explained in Section VI-A. Since the Bayesian inspired method of [15] do not explicitly seek to minimize the final MSE (unlike the unconstrained, affine constrained methods or convex constrained method of [2]), this algorithm provides inferior steady-state performance compared to the other methods for these simulations.

In Fig. 4(d), we plot the estimated final weights $\lim_{t\to\infty} E[\boldsymbol{w}(t)]$ for the unconstrained linear filter trained using the RLS update (averaged over last 5000 samples) as well as the theoretical $\boldsymbol{w}_o$ estimated using the cross-correlation matrix $\boldsymbol{R}$ and $\boldsymbol{p}$ from the observed data with respect to SNR. In this figure, we plot the sum of the weights, i.e., $\sum_i \lim_{t\to\infty} E[w^{(i)}(t)]$, as well as the combination weights for all constituent filters. We observe that for low SNR values, sum of the weights differ greatly from 1, i.e., the affine mixture. This is also the main reason that we observe high performance gains in Fig. 4 for low SNR values with respect to affine and convex combination methods, since the unconstrained methods can scale the corresponding coefficients towards zero for low SNR values. We also observe a close agreement between the theoretical results and the simulations.

To test the accurateness of the separation assumptions used heavily in the derivations for unconstrained and affine constrained methods, we also plot in Fig. 5, the normalized difference $\|E[\|\boldsymbol{y}(t)\|^2 e_a^2(t)] - E[\|\boldsymbol{y}(t)\|^2]E[e_a^2(t)]\|^2 / \sqrt{E[\|\boldsymbol{y}(t)\|^2]E[e_a^2(t)]}$ in the steady state for unconstrained and affinely constrained methods with the same algorithmic parameters as in Fig. 4 under $\mathrm{SNR} = -15, -5$, and 5 dB. We observe that in the convergence, the separation assumption is fairly accurate for these algorithms.

In the next set of experiments, we consider the system identification task under the nonstationary model discussed in Section VII. Here, the desired signal is generated as $d(t) = \tau \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) + n(t)$, where $\boldsymbol{w}_o(t+1) - \boldsymbol{w}_o(0) = \beta[\boldsymbol{w}_o(t-1) - \boldsymbol{w}_o(0)] + \boldsymbol{q}(t)$, the initial value is selected as $\boldsymbol{w}_o(0) = [-0.12, 0.63]^T$, $E[\boldsymbol{q}(t)\boldsymbol{q}^T(t)] = c\boldsymbol{I}$ and $n(t)$ is an i.i.d. Gaussian process with zero mean and $\sigma_n^2 = 0.01$. The input regressor $\boldsymbol{u}(t)$ is an i.i.d. Gaussian vector process with zero mean and unit variance entries. As the constituent filters, we combine outputs of two adaptive filters of length 2, the first one using the LMS update with learning rate $\mu_{\mathrm{LMS}} = 0.03$ and the second one using the least-mean fourth (LMF) update with learning rate $\mu_{\mathrm{LMF}} = 0.5$. For the correlation matrix of $\boldsymbol{q}(t)$ and β, we selected $c = 0.1$ and $\beta = 0.93$. We also

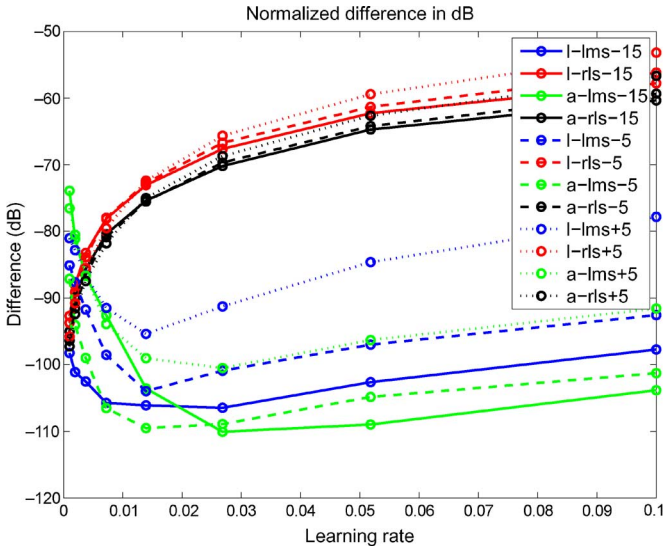

Fig. 5. Normalized difference $\|E[\|\boldsymbol{y}(t)\|^2 e_a^2(t)] - E[\|\boldsymbol{y}(t)\|^2]E[e_a^2(t)]\|^2 / \sqrt{E[\|\boldsymbol{y}(t)\|^2]E[e_a^2(t)]}$ in steady state for unconstrained and affine constrained methods with the same algorithmic parameters as in Fig. 2 under $\mathrm{SNR} = -15$, -5, and 5 dB.

set τ to yield $\mathrm{SNR} = -15$ dB. All parameter values have been selected in order to have the final mixture to converge to $\lim_{t\to\infty} E[\boldsymbol{w}(t)] = [0.24; 0.25]^T$, i.e., both filters have nearly equal contribution in the final mixture and the sum of the weights, 0.49, is away from the affine or convex mixture. We then simulate all the mixture algorithms, using the same setup as in the first set of experiments, under different learning rates and forgetting factors. The results are displayed in Fig. 6(a). The simulations are done with 4×10^4 samples over 70 independent trials. The final MSEs are averaged over the last 7000 samples. We observe from these plots that even under a nonstationary data model in a tracking context, the results introduced in (20), (22), (23), and (26) accurately describe the final MSEs, especially for small learning rates (or large forgetting factors). We next repeat the same experiment for $\beta = 0.995$ and display the results in Fig. 6(b). With this value of β, we have $\lim_{t\to\infty} E[\boldsymbol{w}(t)] = [0.32; 0.19]^T$. For this configuration, we again observe close agreement among the simulations and the introduced results.

IX. Conclusion

In this paper, we investigated adaptive linear mixture approaches in terms of their final MSE in the steady state for stationary and nonstationary environments. Our analysis is generic such that the mixtures can be constructed based on several different adaptive filters each having a different adaptation method, structure or length. We demonstrated the performance gains when we use unconstrained linear, affine and convex combination weights, and provided adaptive methods to achieve these results. We show that by using these mixture approaches, we can greatly improve upon the performance of the constituent filters by exploiting the cross-correlation information between the constituent filter outputs and biasing the combination weights toward zero for low SNR.

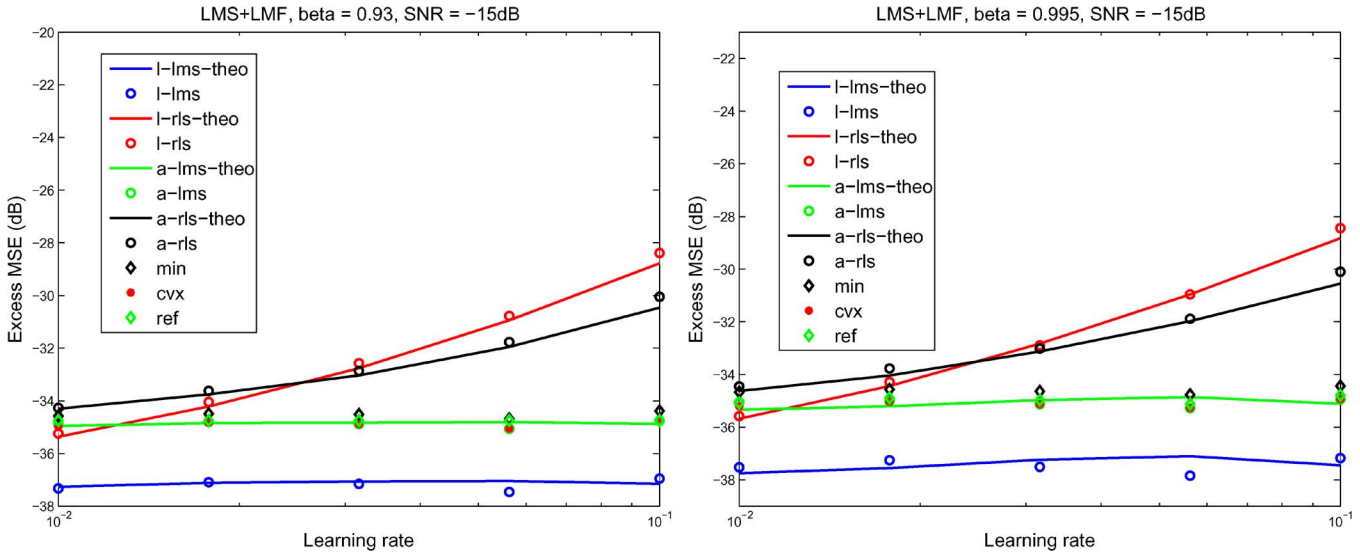

Fig. 6. Mixture of an LMS filter and an LMF filter in tracking context. Here, the x axis represents learning rates μ for the LMS based mixture methods. The forgetting factors of the RLS based algorithms are defined as one minus the learning rate of the LMS based mixture algorithms. The y axis is the excess MSE in dB. (a) $\beta = 0.93$, $\mathrm{SNR} = -15$ dB; (b) $\beta = 0.995$.

APPENDIX

Since for the unconstrained linear combination, $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \mu_{\mathrm{lin}} e(t)\boldsymbol{y}(t)$, we get

$$\begin{aligned}\boldsymbol{w}(t+1) &= \boldsymbol{w}(t) + \mu_{\mathrm{lin}} e(t)\boldsymbol{y}(t) \\ &= \left(\boldsymbol{I} - \mu_{\mathrm{lin}}\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right) + \mu_{\mathrm{lin}}\boldsymbol{y}(t)d(t).\end{aligned}$$

Defining, $\boldsymbol{w}_o(t) \triangleq \boldsymbol{R}^{-1}(t)\boldsymbol{p}(t)$, taking the expectation of both sides and assuming independence of $\boldsymbol{w}(t)$ and $\boldsymbol{y}(t)$, yields

$$\begin{aligned}\boldsymbol{w}(t+1) - \boldsymbol{w}_o(t+1) = &\left(\boldsymbol{I} - \mu_{\mathrm{lin}}\boldsymbol{R}(t)\right)\left(\boldsymbol{w}(t) - \boldsymbol{w}_o(t)\right) \\ &+ \left(\boldsymbol{w}_o(t) - \boldsymbol{w}_o(t+1)\right). \quad (29)\end{aligned}$$

Hence, $\lim_{t\to\infty} E[\boldsymbol{w}(t)] = \boldsymbol{w}_o$, provided that $\mu_{\mathrm{lin}} < 2/\lambda_{\max}(\boldsymbol{R}(t))$, where $\lambda_{\max}(\boldsymbol{R}(t))$ is the maximum eigenvalue of $\boldsymbol{R}(t)$.

REFERENCES

[1] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2685–2699, Oct. 1999.

[2] J. Arenas-Garcia, 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, no. 3, pp. 1078–1090, Mar. 2006.

[3] C. G. Lopes, E. Satorius, and A. H. Sayed, "Adaptive carrier tracking for direct-to-earth Mars communications," in *Proc. 40th Asilomar Conf. Signals, Systems, Computers*, 2006, pp. 1042–1046.

[4] N. J. Bershad, J. C. M. Bermudez, and J. Tourneret, "An affine combination of two lms adaptive filters-transient mean-square analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1853–1864, May 2008.

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

[6] S. S. Kozat and A. C. Singer, "Multi-stage adaptive signal processing algorithms," in *Proc. SAM Signal Process. Workshop*, 2000, pp. 380–384.

[7] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[8] J. Arenas-Garcia, V. Gomez-Verdejo, M. Martinez-Ramon, and A. R. Figueiras-Vidal, "Separate-variable adaptive combination of LMS adaptive filters for plant identification," in *Proc. 13th IEEE Int. Workshop Neural Networks Signal Processing*, 2003, pp. 239–248.

[9] J. Arenas-Garcia, M. Martinez-Ramon, V. Gomez-Verdejo, and A. R. Figueiras-Vidal, "Multiple plant identifier via adaptive LMS convex combination," in *Proc. IEEE Int. Symp. Intel. Signal Processing*, 2003, pp. 137–142.

[10] V. Vovk, "A game of prediction with expert advice," *J. Comput. System Sciences*, vol. 56, pp. 153–173, 1998.

[11] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[12] E. Eweda, "Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels," *IEEE Trans. Signal Process.*, vol. 42, no. 11, pp. 2937–2944, Nov. 1994.

[13] S. R. Kulkarni and P. J. Ramadge, "On the performance and complexity of a class of hybrid controller switching policies," in *Lecture Notes Control Inf. Sci.*, 1996, pp. 248–261.

[14] P. Anderson, "Adaptive forgetting in recursive identification through multiple models," *Int. J. Control*, vol. 42, pp. 1175–1193, 1985.

[15] M. Niedzwiecki, "Identification of nonstationary stochastic systems using parallel estimation schemes," *IEEE Trans. Autom. Control*, vol. 35, no. 3, pp. 329–334, 1990.

[16] N. Cesa-Bianchi, P. M. Long, and M. K. Warmuth, "Worst-case quadratic loss bounds for prediction using linear functions and gradient descent," *IEEE Trans. Neural Netw.*, vol. 7, no. 3, pp. 604–619, 1996.

[17] V. Vovk, "Aggregating strategies," in *Proc. COLT*, 1990, pp. 371–383.

[18] T. Cover and E. Ordentlich, "Universal portfolios with side-information," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 348–363, 1996.

[19] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[20] S. S. Kozat and A. C. Singer, "Universal switching linear least squares prediction," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 189–204, Jan. 2008.

[21] S. S. Kozat, A. C. Singer, and G. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3730–3745, Jul. 2007.

[22] V. Vovk, "Competitive on-line statistics," *Int. Stat. Rev.*, vol. 69, pp. 213–248, 2001.

[23] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, 1994.

[24] D. Haussler and J. Kivinen, "Additive versus exponentiated gradient updates for linear prediction," *J. Inf. Comput.*, vol. 132, no. 1, pp. 1–64, 1997.

[25] J. Arenas-Garcia, V. Gomez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filters," *IEEE Trans. Instrum. Meas.*, vol. 54, pp. 2239–2249, Dec. 2005.
[26] H. J. H. Tuenter, "The minimum l_2 distance projection onto the canonical simplex: A simple algorithm," *Algo Res. Quart.*, vol. 4, pp. 53–55, Dec. 2001.
[27] A. T. Erdogan, S. S. Kozat, and A. C. Singer, "Comparison of convex combination and affine combination of adaptive filters," presented at the IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Taipei, Taiwan, R. O. C., 2009.
[28] D. P. Bertsekas, *Nonlinear Programming*. Singapore: Athena Scientific, 1999.

Suleyman Serdar Kozat (M'04) was born in Ankara, Turkey. He received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1998, receiving a full-time scholarship from the university during his undergraduate studies, and the M.S. and Ph.D. degrees from the Electrical and Computer Engineering Department in the Signal Processing Group at the University of Illinois at Urbana Champaign, Urbana, IL, in 2001 and 2004, respectively.

Until 2007, he was with IBM Research as a full-time Research Staff Member in Speech Technologies Group, T. J. Watson Research Center, Yorktown, NY. He is currently an Assistant Professor at the Electrical Engineering Department in Koc University, Istanbul, Turkey. His research interests include machine learning, signal processing, communications, and statistical signal processing.

Dr. Kozat is currently serving as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING and has served as a reviewer for IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is a member of the IEEE, the Signal Processing Society, the IEEE Information Theory Society.

Alper Tunga Erdogan (M'00) was born in Ankara, Turkey, in 1971. He received the B.S. degree from the Middle East Technical University, Ankara, Turkey, in 1993 and the M.S. and Ph.D. degrees from Stanford University, CA, in 1995 and 1999, respectively.

He was a Principal Research Engineer in Globespan-Virata Corporation (formerly Excess Bandwidth and Virata Corporations) from September 1999 to November 2001. In January 2002, he joined the Electrical and Electronics Engineering Department of Koc University, Istanbul, Turkey, , where he is currently an Associate Professor. His research interests include wireless, fiber and wireline communications, adaptive signal processing, optimization, system theory and control, and information theory.

Prof. Erdogan is the recipient of TUBITAK Career Award in 2005, the Werner Von Siemens Excellence Excellence Award in 2007, and the TUBA GEBIP Outstanding Young Scientist Award in 2008. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Andrew C. Singer (S'92–M'96–SM'05–F'10) received the S.B., S.M., and Ph.D. degrees, all in electrical engineering and computer science, from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1996, respectively.

During the academic year 1996, he was a Postdoctoral Research Affiliate in the Research Laboratory of Electronics at MIT. From 1996 to 1998, he was a Research Scientist at Sanders, A Lockheed Martin Company in Manchester, New Hampshire, where he designed algorithms, architectures, and systems for a variety of DOD applications. Since 1998, he has been on the faculty of the Department of Electrical and Computer Engineering (ECE) at the University of Illinois at Urbana-Champaign, where he is currently a Professor in the ECE Department and a Research Professor in the Coordinated Science Laboratory. His research spans algorithms and architectures for statistical signal processing and communication systems, as well as information theory and machine learning. In 2005, he was appointed as the Director of the Technology Entrepreneur Center (TEC) in the College of Engineering creating and overseeing courses and activities attended by over 1000 students annually. In 2000, he also co-founded Intersymbol Communications, Inc., a venture-funded fabless semiconductor IC company, based in Champaign Illinois, which brought an MLSE-based electronic dispersion compensation chip-set to the optical communications market. In 2007, Intersymbol Communications, Inc., was acquired by Finisar Corporation, a publicly traded optical communications company (NASDAQ: FNSR). He serves on the Board of Directors of Dx Photonics, a medical imaging technology company and of Mimosa Acoustics, an audiology diagnostic device company. He also serves on the Board of Advisors to EnterpriseWorks, the technology incubator facility for the University of Illinois and as an expert witness to the digital signal processing and optical communications industries.

Dr. Singer was a Hughes Aircraft Masters Fellow and was the recipient of the Harold L. Hazen Memorial Award for Excellence in Teaching in 1991. He received the National Science Foundation CAREER Award inn 2000, the Xerox Award for Outstanding Faculty Award in 2001, and was named a Willett Faculty Scholar in 2002. He has received numerous awards and honors, including Best Paper Awards from the IEEE Solid State Circuits Society in 2006 for his paper in the IEEE JOURNAL OF SOLID-STATE CIRCUITS, and in 2008 from the IEEE Signal Processing Society for his paper in the *IEEE Signal Processing Magazine*. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and is a member of the MIT Educational Council, and of Eta Kappa Nu and Tau Beta Pi.

Ali H. Sayed (F'01) is Professor and Chairman of Electrical Engineering at the University of California, Los Angeles (UCLA), where he directs the Adaptive Systems Laboratory (www.ee.ucla.edu/asl). He has authored or coauthored several books, including *Adaptive Filters* (Wiley, 2008), *Fundamentals of Adaptive Filtering* (Wiley, 2003), and *Linear Estimation* (Prentice-Hall, 2000).

Dr. Sayed has served on the editorial boards of several journals, including as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He has published widely in the areas of adaptive systems, statistical signal processing, estimation theory, adaptive and cognitive networks, and signal processing for communications. His work has received several awards, including the 1996 IEEE Donald G. Fink Award, the 2003 Kuwait Prize, the 2005 Terman Award, and the 2002 Best Paper Award and 2005 Young Author Best Paper Award from the IEEE Signal Processing Society. He has served as a 2005 Distinguished Lecturer of the same society and as General Chairman of ICASSP 2008. He is serving as the Vice-President at Publications of the IEEE Signal Processing Society.