[21] I. S. Chong and A. Ortega, "Dynamic voltage scaling algorithms for power constrained motion estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Apr. 2007, vol. 2, pp. 101–104.

[22] G. V. Varatkar, S. Narayanan, N. R. Shanbhag, and D. L. Jones, "Trends in energy-efficiency and robustness using stochastic sensor network-on-a-chip," in *Proc. 18th ACM Great Lakes Symp. VLSI (GLSVLSI)*, 2008, pp. 351–354.

[23] S. Appadwedula, M. Goel, N. R. Shanbhag, D. L. Jones, and K. Ramchandran, "Total system energy minimization for wireless image transmission," *J. VLSI Signal Process.*, vol. 27, no. 1-2, pp. 99–117, Feb. 2001.

# Unbiased Model Combinations for Adaptive Filtering

Suleyman S. Kozat, Andrew C. Singer, Alper Tunga Erdogan, and Ali H. Sayed

***Abstract*—In this paper, we consider model combination methods for adaptive filtering that perform unbiased estimation. In this widely studied framework, two adaptive filters are run in parallel, each producing unbiased estimates of an underlying linear model. The outputs of these two filters are combined using another adaptive algorithm to yield the final output of the system. Overall, we require that the final algorithm produce an unbiased estimate of the underlying model. We later specialize this framework where we combine one filter using the least-mean squares (LMS) update and the other filter using the least-mean fourth (LMF) update to decrease cross correlation in between the outputs and improve the overall performance. We study the steady-state performance of previously introduced methods as well as novel combination algorithms for stationary and nonstationary data. These algorithms use stochastic gradient updates instead of the variable transformations used in previous approaches. We explicitly provide steady-state analysis for both stationary and nonstationary environments. We also demonstrate close agreement with the introduced results and the simulations, and show for this specific combination, more than 2 dB gains in terms of excess mean square error with respect to the best constituent filter in the simulations.**

***Index Terms*—Adaptive filtering, gradient projection, least-mean fourth, least-mean square, mixture methods.**

## I. Introduction

We investigate unbiased mixture methods to combine outputs of two adaptive filtering algorithms operating in stationary and nonstationary environments. The objective is to achieve a steady-state mean-square error (MSE) better than, or at least as good as, each individual adaptive branch by exploiting the cross correlation structure between them through an adaptive combining scheme. We may achieve an unbiased output through the use of the convex or affine combination constraints on the combination weights. We focus on steady-state results for stationary and certain nonstationary data models, however, the transient analysis of the algorithms can be derived using similar methods. Furthermore, although we only use stochastic gradient updates to train the combination weights, one can extend these algorithms to other methods, such as those based on Newton or quasi-Newton updates.

The structure we consider consists of two stages [1], [2]. In the first stage, we have two adaptive filters, working in parallel, to model a desired signal. These adaptive filters have the same length, however, each may use a different adaptation algorithm. We also require that these constituent filters produce unbiased estimates of the underlying model. The desired signal has a random walk formulation to represent both stationary and nonstationary environments [3]. A more precise problem formulation is given in Section II. The second stage of the model is the combination stage. Here, the outputs of the adaptive filters in the first stage are linearly combined to yield the final output. We only consider combination methods that produce unbiased final estimates of the underlying model. A sufficient condition to satisfy this requirement is to assume that the second stage coefficients sum up to one at all times, i.e., affine combinations. In addition to unbiasedness, the combination coefficients can be further constrained to be nonnegative, which corresponds to the case of convex combination. We consider both of these cases.

The framework where multiple adaptive algorithms are combined using an unbiased linear combination with the goal of improving the overall performance has recently attracted wide interest [1], [4], and [5], following the result in [1] that the convex combinations can improve the resulting MSE performance. The requirement on unbiasedness may be motivated from some problem-specific constraints as well as implementation related issues. The combination weights are usually trained using stochastic gradient updates, either after a sigmoid nonlinearity transformation to satisfy convex constraints [1], [4] or after a variable transformation to satisfy affine constraints [5]. There are also Bayesian inspired methods that have extensive roots in machine learning literature [2]. The methods in [1], [2], [4], and [5] combine filters using least-mean squares (LMS) or recursive least squares (RLS) updates (or unsupervised updates). As demonstrated in [1] and [4], mixtures of two filters using the LMS or RLS updates (or a combination of the two) with the convex methods yield combination structures that converge to the best algorithm among the two for stationary data. As demonstrated in [1], the cross correlation between *a priori* errors of the two LMS filters (or LMS and RLS filters in [4]) remains sufficiently high that it limits the combination performance and the optimal convex combination solution converges to only selecting one of the two outputs.

In this paper, we first quantify the achievable gains using convex or affine constrained combination weights in steady-state for stationary and nonstationary data. We also provide the optimal combination weights to yield these gains. We next demonstrate that the update given in [5, Eq. (45)] (which tries to simulate the unrealizable optimal affine combiner) is a stochastic gradient update with a single tap input regressor and derive its steady-state MSE for both stationary and nonstationary environments. Here, we refrain from making variable transformations and directly adapt the combination weights using stochastic gradient updates. However, to preserve convexity or affinity, after each update, we project each updated mixture weight vector back to the convex or affine space. These methods update the weights directly instead of using variable transformations [1], [4]. As a by product of our analysis, we demonstrate that the update in [5, Eq. (45)] is also a stochastic gradient projection update. As a specific example,

Manuscript received May 22, 2009; accepted March 15, 2010. Date of publication April 05, 2010; date of current version July 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Danilo P. Mandic. This work was supported in part by a TUBITAK Career Award, Contract Nos. 104E073 and 108E195, and by 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-0942936.

S. S. Kozat and A. T. Erdogan are with the EEE Department, Koc University, Istanbul (e-mail: skozat@ku.edu.tr; alperdogan@ku.edu.tr).

A. C. Singer is with the Department of ECE, University of Illinois at Urbana-Champaign, Urbana IL 61801 USA (e-mail: acsinger@ad.uiuc.edu).

A. H. Sayed is with the Department of Electrical Engineering, University of California, 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.2010.2047639

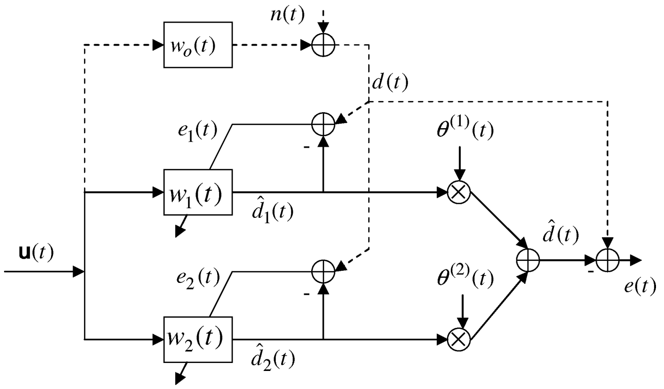


Fig. 1. A mixture of two adaptive filters working in parallel to model a desired signal.

we consider the case of a combination of two adaptive branches using the LMS and the least-mean fourth (LMF) algorithms respectively and derive the steady-state MSE. By this specific combination, we achieve sufficient decorrelation of the outputs so that in both stationary and nonstationary environments the final steady-state MSE of the combination is better than the MSE of the best constituent filter. In [1] and [4], it was shown that for combination of two LMS filters (or LMS and RLS filters) such a performance improvement over the constituent filters is not possible in stationary environments but only achievable in certain nonstationary environments.

We first introduce the basic problem and derive the optimal affine and convex combinations along with their corresponding MSE performance in Section II. We then continue to introduce several different methods to train the combination weights and provide the corresponding steady-state MSEs in Section III. In Section IV, we specialize our results to the case where we combine an LMS and an LMF adaptive filters. We also derive the corresponding cross correlation between the *a priori* errors of the LMS and LMF filters. We conclude the correspondence with simulations and some remarks.

## II. Problem Description

The system we consider has two parts as shown in Fig. 1. The first part contains two constituent adaptive filters, running in parallel to model the desired signal $d(t)$. The desired signal is given by $d(t) = \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) + n(t)$, where $\boldsymbol{u}(t) \in \mathbb{R}^m$ is a zero mean stationary vector process with $\boldsymbol{Q} \triangleq E[\boldsymbol{u}(t)\boldsymbol{u}^T(t)]$, $n(t)$ is an i.i.d. noise process independent of $\boldsymbol{u}(t)$ with $\sigma_n^2 \triangleq E[n^2(t)]$ and $\boldsymbol{w}_o(t) \in \mathbb{R}^m$ is an unknown system vector.[1] We assume a widely used [3] random walk model on $\boldsymbol{w}_o(t)$ such that $\boldsymbol{w}_o(t+1) - \boldsymbol{w}_o(0) = \beta[\boldsymbol{w}_o(t) - \boldsymbol{w}_o(0)] + \boldsymbol{q}(t)$, where $\boldsymbol{q}(t) \in \mathbb{R}^m$ is an i.i.d. zero mean vector process with covariance matrix $E[\boldsymbol{q}(t)\boldsymbol{q}^T(t)] = \boldsymbol{\Phi}$, $\boldsymbol{w}_o(0)$ is the initial weight vector as well as the mean of this process. We observe that $\boldsymbol{\Phi} = \boldsymbol{0}$ and $\beta = 1$ corresponds to the stationary case. Usually, $0 \ll |\beta| \leq 1$. Each filter updates a weight vector $\boldsymbol{w}_1(t) \in \mathbb{R}^m$ and $\boldsymbol{w}_2(t) \in \mathbb{R}^m$ and produces estimates, $\hat{d}_i(t) = \boldsymbol{w}_i^T(t)\boldsymbol{u}(t)$, $i = 1, 2$, respectively. For each filter we also define estimation, *a priori* and a posteriori errors as

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

Hence, for each filter we have

$$\hat{d}_i(t) = \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) - e_{i,a}(t) \tag{1}$$

and $e_i(t) = e_{i,a}(t) + n(t)$. We also have $J_i(t) \triangleq E[e_i^2(t)]$, $J_{\text{ex},i}(t) \triangleq E[e_{i,a}^2(t)]$ and their limiting values (if they exist) $J_i \triangleq \lim_{t\to\infty} J_i(t)$, $J_{\text{ex},i} \triangleq \lim_{t\to\infty} J_{\text{ex},i}(t)$, respectively. We further have $J_{\text{ex}12}(t) \triangleq E[e_{1,a}(t)e_{2,a}(t)]$ and (if it exists) $J_{\text{ex}12} \triangleq \lim_{t\to\infty} J_{\text{ex}12}(t)$.

The second part of the system is the mixture stage. Here, the outputs of the two constituent filters are combined to produce the final output as $\hat{d}(t) = \boldsymbol{\theta}^T(t)\boldsymbol{y}(t)$, where $\boldsymbol{y}(t) \triangleq [\hat{d}_1(t)\hat{d}_2(t)]^T$. We also update $\boldsymbol{\theta}(t)$ with an adaptive algorithm, where possible candidates are given in Section III. We note that for the mixture stage, the correlation matrix $R(t) \triangleq E[\boldsymbol{y}(t)\boldsymbol{y}^T(t)]$ and the cross correlation vector $\boldsymbol{p}(t) \triangleq E[\boldsymbol{y}(t)d(t)]$ are time varying. To obtain the limiting values, we observe that

$$\boldsymbol{y}(t) = \begin{bmatrix} \hat{d}_1(t) \\ \hat{d}_2(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) - e_{1,a}(t) \\ \boldsymbol{w}_o^T(t)\boldsymbol{u}(t) - e_{2,a}(t) \end{bmatrix}. \tag{2}$$

We further have

$$\begin{aligned} &\lim_{t\to\infty} E\left[\hat{d}_1(t)\hat{d}_2(t)\right] \\ &\quad = \lim_{t\to\infty} E\Big[\boldsymbol{w}_o^T(t)\boldsymbol{u}(t)\boldsymbol{w}_o^T(t)\boldsymbol{u}(t) - \boldsymbol{w}_o^T(t)\boldsymbol{u}(t)e_{1,a}(t) \\ &\qquad\qquad - \boldsymbol{w}_o^T(t)\boldsymbol{u}(t)e_{2,a}(t) + e_{1,a}(t)e_{2,a}(t)\Big] \\ &\quad = \sigma_c^2(\infty) + J_{\text{ex},12} \end{aligned} \tag{3}$$

where $\sigma_c^2(t) \triangleq \text{tr}(E\{\boldsymbol{Q}E[\boldsymbol{w}_o(t)\boldsymbol{w}_o^T(t)]\})$ and we use a separation assumption such that $E[\boldsymbol{u}(t)e_{i,a}(t)] = E[\boldsymbol{u}(t)]E[e_{i,a}(t)]$ similar to [3] to cancel the cross terms in (3). We note that $\lim_{t\to\infty}\sigma_c^2(t) = \text{tr}([\boldsymbol{\Phi}\boldsymbol{Q} + (1-\beta^2)\boldsymbol{w}_o(0)\boldsymbol{w}_o^T(0)\boldsymbol{Q}]/(1-\beta^2))$ when $|\beta| < 1$, and diverges when $|\beta| = 1$. With an abuse of notation, we preserve the time index as $t \to \infty$ in $\sigma_c^2(t)$ to study both cases together. Hence, by this derivation we obtain

$$\begin{aligned} \lim_{t\to\infty} \boldsymbol{R}(t) &= \lim_{t\to\infty} E\left[\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right] \\ &= \begin{bmatrix} J_{\text{ex},1} + \sigma_c^2(\infty) & J_{\text{ex},12} + \sigma_c^2(\infty) \\ J_{\text{ex},12} + \sigma_c^2(\infty) & J_{\text{ex},2} + \sigma_c^2(\infty) \end{bmatrix} \end{aligned} \tag{4}$$

and $\lim_{t\to\infty}\boldsymbol{p}(t) = [\sigma_c^2(\infty)\sigma_c^2(\infty)]^T$. We next present the corresponding optimal affine and convex mixture weights to minimize the steady-state MSE. These optimal convex and affine weights are targeted by the adaptive algorithms applied on $\boldsymbol{\theta}(t)$ in Section III.

### A. Affine and Convex Combinations

Under the aforementioned analysis of $\boldsymbol{R}(t)$ and $\boldsymbol{p}(t)$, for given $\boldsymbol{\theta} \in \mathbb{R}^2$, the steady-state MSE (if the limit exists) in the limit is given by

$$\begin{aligned} \lim_{t\to\infty} E\left[\left(d(t) - \boldsymbol{\theta}^T\boldsymbol{y}(t)\right)^2\right] = \lim_{t\to\infty} \Big\{ &\sigma_d^2(t) - \boldsymbol{p}^T(t)\boldsymbol{R}^{-1}(t)\boldsymbol{p}(t) \\ &+ [\boldsymbol{\theta} - \boldsymbol{\theta}_o(t)]^T \boldsymbol{R}(t)\,[\boldsymbol{\theta} - \boldsymbol{\theta}_o(t)]\Big\}, \end{aligned}$$

where $\boldsymbol{\theta}_o(t) \triangleq \boldsymbol{R}^{-1}(t)\boldsymbol{p}(t)$ and $\sigma_d^2(t) \triangleq E[d^2(t)] = \sigma_c^2(t) + \sigma_n^2$. If the combination weights are constrained to be affine, then the optimal affine weights that minimize the final MSE are given as the solution to the following convex quadratic minimization problem:

$$\begin{aligned} \boldsymbol{\theta}_o^a \triangleq \lim_{t\to\infty} \arg\min_{\boldsymbol{\theta}^a} \Big\{ &\sigma_d^2(t) - \boldsymbol{p}^T(t)\boldsymbol{R}^{-1}(t)\boldsymbol{p}(t) \\ &+ [\boldsymbol{\theta}^a - \boldsymbol{\theta}_o(t)]^T \boldsymbol{R}(t)\,[\boldsymbol{\theta}^a - \boldsymbol{\theta}_o(t)]\Big\} \end{aligned}$$

[1] All vectors are column vectors, represented by boldface lowercase letters, $(\cdot)^T$ is the transpose operation and $\|\cdot\|$ is the $l_2$-norm. For a vector $\boldsymbol{w}$, $w^{(i)}$ is the $i$th entry. Matrices are represented with boldface capital letters. For a matrix $\boldsymbol{R}$, $\text{tr}(\boldsymbol{R})$ is the trace. Also, the vector or matrix $\mathbf{1}$ (or $\mathbf{0}$) represents a vector or a matrix of all ones (or zeros) where the size is understood from the context.

such that $(\boldsymbol{\theta}^a)^T \mathbf{1} = 1, \boldsymbol{\theta}^a \in \mathbb{R}^2$. The optimal affine weights can be shown to be

$$\begin{aligned}\boldsymbol{\theta}_o^a &= \lim_{t\to\infty}\left\{\boldsymbol{\theta}_o(t) + \left[1 - \boldsymbol{\theta}_o^T(t)\mathbf{1}\right]\frac{\boldsymbol{R}^{-1}(t)\mathbf{1}}{\mathbf{1}^T\boldsymbol{R}^{-1}(t)\mathbf{1}}\right\}\\ &= \frac{1}{J_{\mathrm{ex},1} + J_{\mathrm{ex},2} - 2J_{\mathrm{ex},12}}\begin{bmatrix}J_{\mathrm{ex},2} - J_{\mathrm{ex},12}\\ J_{\mathrm{ex},1} - J_{\mathrm{ex},12}\end{bmatrix} \qquad (5)\end{aligned}$$

using (4) and $\boldsymbol{p} = [\sigma_c^2(t)\ \sigma_c^2(t)]^T$. We also define $J_o^a \triangleq \lim_{t\to\infty} E[(d(t) - (\boldsymbol{\theta}_o^a)^T\boldsymbol{y}(t))^2] = \sigma_n^2 + (J_{\mathrm{ex}1}J_{\mathrm{ex}2} - J_{\mathrm{ex}12}^2)/(J_{\mathrm{ex}1} + J_{\mathrm{ex}2} - 2J_{\mathrm{ex}12})$, i.e., the MMSE of the affine combination.

For the convexity constraint on the combining weights, we have a convex quadratic minimization problem with linear constraints,

$$\begin{aligned}\boldsymbol{\theta}_o^c \triangleq \lim_{t\to\infty}\arg\min_{\boldsymbol{\theta}^c}\Big\{J^c(\boldsymbol{\theta}^c) &\triangleq \sigma_d^2(t) - \boldsymbol{p}^T(t)\boldsymbol{R}^{-1}(t)\boldsymbol{p}(t)\\ &+ [\boldsymbol{\theta}^c - \boldsymbol{\theta}_o(t)]^T\boldsymbol{R}(t)[\boldsymbol{\theta}^c - \boldsymbol{\theta}_o(t)]\Big\} \qquad (6)\end{aligned}$$

such that $\boldsymbol{\theta}^c \in \mathcal{B}$, where $\mathcal{B} = \{\boldsymbol{z} : \boldsymbol{z}^T\mathbf{1} = 1, z^{(1)} \geq 0, z^{(2)} \geq 0\}$ is the unit simplex. Since $\mathcal{B}$ is included inside the set corresponding to the affine combination weights, we have $J_o^a \leq J_o^c$.

After some algebra as in [6], we can rewrite the cost function in (6) as

$$J^c(\boldsymbol{\theta}^c) = J_o^a + (\boldsymbol{\theta}^c - \boldsymbol{\theta}_o^a)^T\boldsymbol{R}(t)(\boldsymbol{\theta}^c - \boldsymbol{\theta}_o^a) \qquad (7)$$

$t \gg 1$ for any $\boldsymbol{\theta}^c$ in the unit simplex, i.e., $\boldsymbol{\theta}^c \in \mathcal{B}$. Therefore, by ignoring the constant term in (7), we can view the problem of finding $\boldsymbol{\theta}_o^c$ as that of projecting $\boldsymbol{\theta}_o^a$ onto the unit simplex $\mathcal{B}$ with respect to the $\boldsymbol{R}(t)$-weighted 2-norm. In other words, the problem of finding the best $\boldsymbol{\theta}_o^c$ can be posed as $\boldsymbol{\theta}_o^c = \arg\min_{\boldsymbol{\theta}^c}\|\boldsymbol{\theta}_o^a - \boldsymbol{\theta}^c\|_{\boldsymbol{R}(t)}^2$ s.t. $\boldsymbol{\theta}^c \in \mathcal{B}$. For the solution, we observe that the unit simplex $\mathcal{B}$ is the line segment between $[1\ 0]^T$ and $[0\ 1]^T$, yielding two cases to consider

- Case $\boldsymbol{\theta}_o^a \in \mathcal{B}$: In this case, we simply have $\boldsymbol{\theta}_o^c = \boldsymbol{\theta}_o^a$ and $J_o^a = J_o^c$.
- Case $\boldsymbol{\theta}_o^a \notin \mathcal{B}$: This case occurs when one of the components of $\boldsymbol{\theta}_o^a$ is strictly negative and the other is positive. Without loss of generality, we let the first component be negative. Then, for any vector $\boldsymbol{\theta}^c$ in the unit simplex, due to the constraint that $\mathbf{1}^T\boldsymbol{\theta}_o^a = \mathbf{1}^T\boldsymbol{\theta}^c = 1$, we can write $\boldsymbol{\theta}^c = \boldsymbol{\theta}_o^a + \nu[1\ -1]^T$. Thus, the cost function in (7) would be equivalent to $\|\nu[-1\ 1]^T\|_{\boldsymbol{R}(t)}^2 = \nu^2[-1\ 1]\boldsymbol{R}(t)[-1\ 1]^T$, which implies that the cost increases when the magnitude of $\nu$ increases. Therefore, the smallest value which makes $\boldsymbol{\theta}^c = \boldsymbol{\theta}_o^a + \nu[1\ -1]^T$ feasible (i.e., in $\mathcal{B}$) would be the optimal choice for $\nu$, which is equivalent to $\nu = -\theta_o^{a(1)}$. This would be equivalent to the choice $\boldsymbol{\theta}_o^c = [0\,1]$, i.e., a corner point of $\mathcal{B}$. As a result, $J_o^c \leq \min\{J_1, J_2\}$ and the cost increase for the convex combination relative to the affine combination case can be written as

$$J_o^c - J_o^a = \begin{cases}\left(\boldsymbol{\theta}e_o^{a(1)}\right)^2\zeta & \boldsymbol{\theta}_o^{a(1)} < 0\\ \left(\boldsymbol{\theta}e_o^{a(2)}\right)^2\zeta & \boldsymbol{\theta}_o^{a(2)} < 0\\ 0 & \text{otherwise}\end{cases}$$

$$\text{where } \zeta \triangleq [-1 \quad 1]\boldsymbol{R}(t)\begin{bmatrix}-1\\ 1\end{bmatrix} = J_{\mathrm{ex},1} + J_{\mathrm{ex},2} - 2J_{\mathrm{ex},12}. \qquad (8)$$

We observe that when the combinations are constrained to be unbiased, i.e., affine or convex, the optimal weights depend on the relative value of the cross correlation between *a priori* errors relative to *a priori* error variances of each constituent filter. We point out that for a combination of two filters each using the LMS update with a different learning rate, it was illustrated in [1] that $\boldsymbol{\theta}_o^a$ is out of reach of the convex combination and $\boldsymbol{\theta}_o^c \neq \boldsymbol{\theta}_o^a$ (except on the simplex boundary) in stationary context, i.e., $\boldsymbol{\Phi} = \mathbf{0}$ and $\beta = 1$. In the next section, we investigate different methods for training the combination weights that preserve either the convex or affine constraints.

## III. Methods to Update the Combination Weights

In this section, we study four different adaptive methods to train the combination weights and their respective steady-state MSEs. These include previously introduced (Section III-A) [5], previously investigated (Section III-D) [1] as well as new combination methods. We present them all for completeness.

### A. Stochastic Gradient Update on Affine Weights

To constrain the combination weights to be affine, we can use a variable transformation such that we do not constrain $\theta^{(1)}(t)$ and define $\theta^{(2)}(t) = 1 - \theta^{(1)}(t)$. We next use a gradient update on $\theta^{(1)}(t)$ as

$$e(t) = d(t) - \theta e^{(1)}(t)\hat{d}_1(t) - \left[1 - \theta e^{(1)}(t)\right]\hat{d}_2(t) \qquad (9)$$

$$\begin{aligned}\theta e^{(1)}(t+1) &= \theta e^{(1)}(t) - \frac{\mu}{2}\nabla_{\theta e^{(1)}}\left(e^2(t)\right)\\ &= \theta e^{(1)}(t) + \mu e(t)\left[\hat{d}_1(t) - \hat{d}_2(t)\right]. \qquad (10)\end{aligned}$$

This update requires $O(1)$ computational complexity. We note that this update corresponds to the affine update given in (45) of [5]. We observe from (9) and (10) that this update corresponds to the ordinary LMS update on $\theta^{(1)}(t)$ with the desired signal as $[d(t) - \hat{d}_2(t)]$ and one-tap input regressor $[\hat{d}_1(t) - \hat{d}_2(t)]$ [7]. However, using (1) and the definition of $d(t)$, we get $d(t) - \hat{d}_2(t) = n(t) + e_{2,a}(t)$ and $\hat{d}_1(t) - \hat{d}_2(t) = e_{2,a}(t) - e_{1,a}(t)$. A simple derivation invoking the separation assumptions (or independence assumptions) [3] yields that this LMS update converges to a stationary point such that $\lim_{t\to\infty}E[\boldsymbol{\theta}(t)] = \boldsymbol{\theta}_o^a$ when $\mu < 1/(J_{\mathrm{ex},1}(t) + J_{\mathrm{ex},2}(t) - 2J_{\mathrm{ex},12}(t))$ and the steady-state MSE of this combination is given by

$$\lim_{t\to\infty}E\left[e^2(t)\right] = J_o^a + \frac{\mu J_o^a(J_{\mathrm{ex},1} + J_{\mathrm{ex},2} - 2J_{\mathrm{ex},12})}{2 - \mu(J_{\mathrm{ex},1} + J_{\mathrm{ex},2} - 2J_{\mathrm{ex},12})} \qquad (11)$$

since $J_{\mathrm{ex},1} + J_{\mathrm{ex},2} - 2J_{\mathrm{ex},12} = \lim_{t\to\infty}E\{[e_{2,a}(t) - e_{1,a}(t)]^2\}$ is the power of the one tap input regressor and $J_o^a$ is the MMSE of this one tap filter. We note that the effect of $\boldsymbol{w}_o^T(t)\boldsymbol{u}(t)$ cancels out, hence for any value of $\beta$, even if $|\beta| = 1$ such that $\sigma_c^2(t)$ diverges for nonzero $\boldsymbol{\Phi}$, (11) holds due to the affine constraint. One can also perform transient analysis of this update through analogous means, since this is an ordinary LMS update with time varying correlation matrix and cross correlation vector. It can be shown that the particular algorithm converges in MSE sense for

$$0 < \mu < 1/\left(J_{\mathrm{ex},1}(t) + J_{\mathrm{ex},2}(t) - 2J_{\mathrm{ex},12}(t)\right) \qquad (12)$$

for small enough $\mu$.

### B. Stochastic Gradient Projection Update on Affine Weights

In stochastic gradient projection update, at each iteration, the combination weights are updated using the ordinary stochastic gradient update (which is the LMS update). However, at each iteration, since after the gradient update, the updated weight vector can end up outside the affine space, we project this vector back to the affine surface. The update is given by

$$\begin{aligned}\boldsymbol{\theta}(t+1) &= \mathcal{P}_{\mathcal{A}}\left[\boldsymbol{\theta}(t) - \frac{\mu}{2}\nabla_{\boldsymbol{\theta}}e^2(t)\right] = \mathcal{P}_{\mathcal{A}}\left[\boldsymbol{\theta}(t) + \mu e(t)\boldsymbol{y}(t)\right]\\ &= \boldsymbol{\theta}(t) + \frac{\mu}{2}e(t)\begin{bmatrix}e_{2,a}(t) - e_{1,a}(t)\\ -\left(e_{2,a}(t) - e_{1,a}(t)\right)\end{bmatrix} \qquad (13)\end{aligned}$$

where $e(t) = d(t) - \boldsymbol{\theta}^T(t)\boldsymbol{y}(t)$, $\mathcal{P}_{\mathcal{A}}[\cdot]$ is the projection operator to the affine space and $\mu > 0$ is the learning rate. The last equality in (13) is derived using the definition of the projection operation to the affine space as: $\forall\boldsymbol{x} \in \mathbb{R}^2\mathcal{P}_{\mathcal{A}}[\boldsymbol{x}] \triangleq \arg\min_{\boldsymbol{z}\in\mathcal{A}}\|\boldsymbol{x} - \boldsymbol{z}\| =$

$\boldsymbol{x}+((1-\mathbf{1}^T\boldsymbol{x})/\mathbf{1}^T\mathbf{1})\mathbf{1}$, where $\mathcal{A}$ is the affine space, i.e., $\mathcal{A} \triangleq \{\boldsymbol{z} \in \mathbb{R}^2 : \boldsymbol{z}^T\mathbf{1} = 1\}$. However, if we focus on each constituent coefficient, e.g., the first coefficient, then we observe from (13) that $\theta^{(1)}(t+1) = \theta^{(1)}(t) + (\mu/2)e(t)[e_{2,a}(t) - e_{1,a}(t)]$, which is equivalent to (10) (except 1/2 scaling in front of $\mu$). Hence, owing to the affine constraint, this demonstrates that the gradient projected update is equivalent to the stochastic gradient update with affine constraints such that

$$\lim_{t\to\infty} E\left[e^2(t)\right] = J_o^a + \frac{\mu J_o^a(J_{\text{ex},1} + J_{\text{ex},2} - 2J_{\text{ex},12})}{4 - \mu(J_{\text{ex},1} + J_{\text{ex},2} - 2J_{\text{ex},12})}. \quad (14)$$

### C. Stochastic Gradient Projection Update on Convex Weights

In this section, we again use a gradient update on the combination weights $\boldsymbol{\theta}(t)$. However, to preserve convexity, we use a projection onto the constraint set $\mathcal{B}$ following the gradient update. Therefore, the resulting update equation can be written

$$\boldsymbol{\theta}(t+1) = \mathcal{P}_{\mathcal{B}}\left[\boldsymbol{\theta}(t) - \frac{\mu}{2}\nabla_{\boldsymbol{\theta}} e^2(t)\right] = \mathcal{P}_{\mathcal{B}}\left[\boldsymbol{\theta}(t) + \mu e(t)\boldsymbol{y}(t)\right] \quad (15)$$

where $e(t) = d(t) - \boldsymbol{\theta}^T(t)\boldsymbol{u}(t)$, $\mathcal{P}_{\mathcal{B}}[\cdot]$ is the projection operator to the unit simplex. The projection operator to the unit simplex is defined as: $\forall \boldsymbol{x} \in \mathbb{R}^2$, $\mathcal{P}_{\mathcal{B}}[\boldsymbol{x}] \triangleq \arg\min_{\boldsymbol{z}\in\mathcal{B}} \|\boldsymbol{x} - \boldsymbol{z}\|$, where $\mathcal{B} = \{\boldsymbol{z} \in \mathbb{R}^2 : \boldsymbol{z}^T\mathbf{1} = 1, z^{(1)} \geq 0, z^{(2)} \geq 0\}$. For the combination of two branches, this projection operator can be written more explicitly as

$$\begin{aligned} \mathcal{P}_{\mathcal{B}}[\boldsymbol{x}] &= \mathcal{P}_{\mathcal{A}}[\boldsymbol{x}] \quad \text{if} \quad \mathcal{P}_{\mathcal{A}}[\boldsymbol{x}] \geq 0; \\ \mathcal{P}_{\mathcal{B}}[\boldsymbol{x}] &= \boldsymbol{e}_1 \quad \text{if} \quad \boldsymbol{e}_2^T\mathcal{P}_{\mathcal{A}}[\boldsymbol{x}] < 0; \\ \mathcal{P}_{\mathcal{B}}[\boldsymbol{x}] &= \boldsymbol{e}_2 \quad \text{if} \quad \boldsymbol{e}_1^T\mathcal{P}_{\mathcal{A}}[\boldsymbol{x}] < 0 \end{aligned} \quad (16)$$

where $\boldsymbol{e}_i$ is the unit vector for $i$th coordinate. We can show by using independence or separation assumptions [3] and using the nonexpansive property of the projection [8] (Prop.2.1.3(c)), this projected update algorithm converges in the mean to $\boldsymbol{\theta}_o^c$ for sufficiently small $\mu$ when $\boldsymbol{\theta}_o^c$ is in the relative interior of the simplex. We observe that unlike (13), (16) cannot be written in a simple closed form. However, if we follow along the lines that yielded (14) and approximate the progress of the convex weights with the progress of the affine weights (which is true in the interior region of the simplex), then we can give the steady-state MSE as

$$\lim_{t\to\infty} E\left[e^2(t)\right] = J_o^c + \frac{\mu J_o^c(J_{\text{ex},1} + J_{\text{ex},2} - 2J_{\text{ex},12})}{4 - \mu(J_{\text{ex},1} + J_{\text{ex},2} - 2J_{\text{ex},12})}. \quad (17)$$

Note that (17) corresponds to the final MSE of a stochastic gradient update algorithm converging to $\boldsymbol{\theta}_o^c$ with MMSE $J_o^c$ and updating a weight vector with a single tap. The computational complexity of this combination algorithm is only $O(1)$ per output sample.

### D. Stochastic Gradient Update with Convex Weights

Here, the combination weights are trained using a gradient update after a variable transformation using a sigmoid nonlinearity [1]. The update for the combination weights to minimize the final estimation error is given as

$$\theta e^{(1)}(t) = \frac{1}{1 + e^{-a(t)}} \quad (18)$$

$\theta^{(2)}(t) = 1 - \theta^{(1)}(t)$, where $a(t)$ is trained using the stochastic gradient update

$$\begin{aligned} a(t+1) &= a(t) - \frac{\mu}{2}\nabla_a e^2(t) \\ &= a(t) + \mu e(t)\left[1 - \theta e^{(1)}(t)\right] \\ &\quad \times \theta e^{(1)}(t)\left[\hat{d}_1(t) - \hat{d}_2(t)\right]. \end{aligned} \quad (19)$$

In [1], it has been shown under several assumptions that (18) converges to the optimal convex combination weights $\boldsymbol{\theta}_o^c$ and $\lim_{t\to\infty} E[e^2(t)] = J_o^c$. This argument assumed that the stochastic gradient noise does not propagate to the final MSE due to the sigmoid nonlinearity. Hence, the final MSE of (18) is the MMSE $J_o^c$, without any additional terms coming from the stochastic gradient noise. As also emphasized in [1], this MSE equality is accurate when the convex combination converges to one of the constituent filters, i.e., when $\boldsymbol{\theta}_o^c = [0\ 1]^T$ or $\boldsymbol{\theta}_o^c = [1\ 0]^T$, so that $\theta^{(1)}(t)$ (or $\theta^{(2)}(t)$) becomes small enough to attenuate the propagation of the stochastic gradient noise through the sigmoid in (19). As shown in the simulations, the stochastic gradient noise may not be completely eliminated through the sigmoid nonlinearity in certain cases, such as $\boldsymbol{\theta}_o^c \approx [1/2\ 1/2]^T$, such that the final MSE of (18) is not equal to $J_o^c$.

## IV. Combinations of Two Specific Adaptive Filters

In this section, we study the case where we have specific adaptive algorithms to update each of the constituent filter weights. To simplify the notation, only for this section, we assume that $\beta = 1$, i.e., $\boldsymbol{w}_o(t+1) = \boldsymbol{w}_o(t) + \boldsymbol{q}(t)$. Suppose the first constituent filter updates its weight vector $\boldsymbol{w}_1(t)$ using the LMS update as

$$e_1(t) = d(t) - \boldsymbol{w}_1^T(t)\boldsymbol{u}(t) \quad (20)$$

$$\boldsymbol{w}_1(t+1) = \boldsymbol{w}_1(t) + \mu_1 e_1(t)\boldsymbol{u}(t) \quad (21)$$

where $\mu_1 > 0$. We then have

$$e_{1,a}(t) = \left[\boldsymbol{w}_o(t) - \boldsymbol{w}_1(t)\right]^T \boldsymbol{u}(t) \quad (22)$$

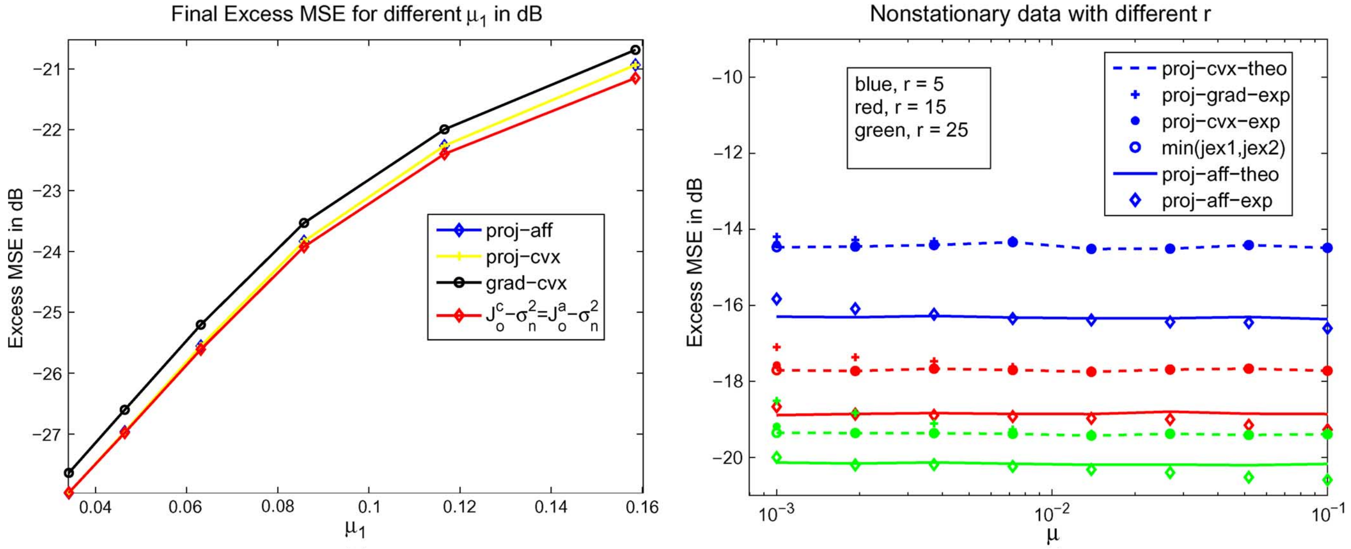
$$\begin{aligned} e_{1,p}(t) &= \left[\boldsymbol{w}_o(t) - \boldsymbol{w}_1(t+1)\right]^T \boldsymbol{u}(t) \\ &= e_{1,a}(t) - \mu_1\|\boldsymbol{u}(t)\|^2 e_1(t) \end{aligned} \quad (23)$$

where $\hat{d}_1(t) = \boldsymbol{w}_1^T(t)\boldsymbol{u}(t)$ and $e_1(t) = e_{1,a}(t) + n(t)$. With these definitions and using the separation assumption such that $E[\|\boldsymbol{u}(t)\|^2 e_{1,a}(t)] = E[\|\boldsymbol{u}(t)\|^2]E[e_{1,a}(t)]$, the converged mean-square *a priori* error of this filter using the LMS update is given by ([3, ch. 6])

$$\begin{aligned} J_{\text{ex},1} = \lim_{t\to\infty} E\left[e_{1,a}^2(t)\right] &= \frac{\mu_1\sigma_n^2\text{tr}(\boldsymbol{Q}) + \mu_1^{-1}\boldsymbol{\Phi}}{2 - \mu_1\text{tr}(\boldsymbol{Q})} \\ &\approx \frac{\mu_1\sigma_n^2\text{tr}(\boldsymbol{Q}) + \mu_1^{-1}\boldsymbol{\Phi}}{2} \end{aligned} \quad (24)$$

where the approximation is accurate for small $\mu_1$.

If the second constituent filter also updates its weight vector $\boldsymbol{w}_2(t)$ with the same LMS update as in (20), (21), but with $\mu_2 > 0$, then the converged mean square *a priori* error is given by $J_{\text{ex},2} = (\mu_2\sigma_n^2\text{tr}(\boldsymbol{Q}) + \mu_2^{-1}\boldsymbol{\Phi})/(2 - \mu_2\text{tr}(\boldsymbol{Q}))$. For the combination of these two filters each running the LMS update with learning rates $\mu_1$ and $\mu_2$, the converged cross correlation between *a priori* errors is given by $J_{\text{ex},12} = (\mu_c\sigma_n^2\text{tr}(\boldsymbol{Q}) + 2\boldsymbol{\Phi}/(\mu_1 + \mu_2))/(2 - \mu_c\text{tr}(\boldsymbol{Q}))$, where $\mu_c \triangleq (2\mu_1\mu_2/(\mu_1 + \mu_2))$ as given in [1]. Without loss of generality if we assume that $\mu_2 > \mu_1$, then $\mu_2 > \mu_c > \mu_1$, i.e., the "learning rate" of the excess error is always between the learning rates of the constituent filters. For stationary environments such that $\boldsymbol{\Phi} = \mathbf{0}$, since $\mu_2 > \mu_c > \mu_1$, we have $J_{\text{ex},2} > J_{\text{ex},12} > J_{\text{ex},1}$. Because of this strict ordering, when used in (5), one element of $\boldsymbol{\theta}_o^a$ will be negative, yielding $\boldsymbol{\theta}_o^a \neq \boldsymbol{\theta}_o^c$. For this case $\boldsymbol{\theta}_o^c = [1\ 0]^T$ from Section II-A, i.e., the convex combination is unable to use all the cross-correlation information between constituent filter outputs to reduce the final estimation error and only converges to the one best in stationary environments.



(a) (b)

Fig. 2. System identification with seventh-order filter. (a) Here, $r = 2, \sigma_n^2 = 0.1, \mathrm{tr}(\boldsymbol{Q}) = 1$ and $\mathrm{tr}(\boldsymbol{\Phi}) = \mathbf{0}$. For the algorithm: of Section III-B $\mu = 0.04$ “proj-aff”, of Section III-C $\mu = 0.04$ “proj-cvx”, of Section III-D $\mu = 30$ “grad-cvx”. (b) The x axis is the learning rate for the second stage for all algorithms, $\mu = 10^{-3}, \ldots, 10^{-1}, \mathrm{tr}(\boldsymbol{\Phi}) = 10^{-4}, \mathrm{tr}(\boldsymbol{Q}) = 1, \mu_1 = 0.001, \mu_2 = r\mu_1, \sigma_n^2 = 0.05$, 700 hundred iterations, $7 \times 10^5$ samples, $\beta = 1$.

To be able to find a combination structure where we have better control over $J_{\mathrm{ex},12}$ both in stationary and nonstationary environments, we now introduce a combination of two filters; the first running the LMS update and the second using the least mean-fourth (LMF) update [3]. In this case, the second constituent filter updates its weight vector $\boldsymbol{w}_2(t)$ using the LMF update as

$$\begin{aligned} e_2(t) &= d(t) - \boldsymbol{w}_2^T(t)\boldsymbol{u}(t) \\ \boldsymbol{w}_2(t+1) &= \boldsymbol{w}_2(t) + \mu_2 e_2^3(t)\boldsymbol{u}(t) \end{aligned} \tag{25}$$

where $\mu_2 > 0$. We then have

$$e_{2,a}(t) = [\boldsymbol{w}_o - \boldsymbol{w}_2(t)]^T \boldsymbol{u}(t) \tag{26}$$

$$\begin{aligned} e_{2,p}(t) &= [\boldsymbol{w}_o - \boldsymbol{w}_2(t+1)]^T \boldsymbol{u}(t) \\ &= e_{2,a}(t) - \mu_2 \|\boldsymbol{u}(t)\|^2 e_2^3(t) \end{aligned} \tag{27}$$

where $\hat{d}_2(t) = \boldsymbol{w}_2^T(t)\boldsymbol{u}(t)$ and $e_2(t) = e_{2,a}(t) + n(t)$. With these definitions and using the separation assumption, the converged mean-square *a priori* error of this filter using the LMF update is given by ([3, ch. 6])

$$\begin{aligned} J_{\mathrm{ex},2} = \lim_{t\to\infty} E\left[e_{2,a}^2(t)\right] &= \frac{\mu_2 E\left[n^6(t)\right]\mathrm{tr}(\boldsymbol{Q}) + \mu_2^{-1}\mathrm{tr}(\boldsymbol{\Phi})}{6\sigma_n^2 - 15\mu_2 E\left[n^4(t)\right]\mathrm{tr}(\boldsymbol{Q})} \\ &\approx \frac{\mu_2 E\left[n^6(t)\right]\mathrm{tr}(\boldsymbol{Q}) + \mu_2^{-1}\mathrm{tr}(\boldsymbol{\Phi})}{6\sigma_n^2} \end{aligned} \tag{28}$$

where the approximation is accurate for small $\mu_2$.

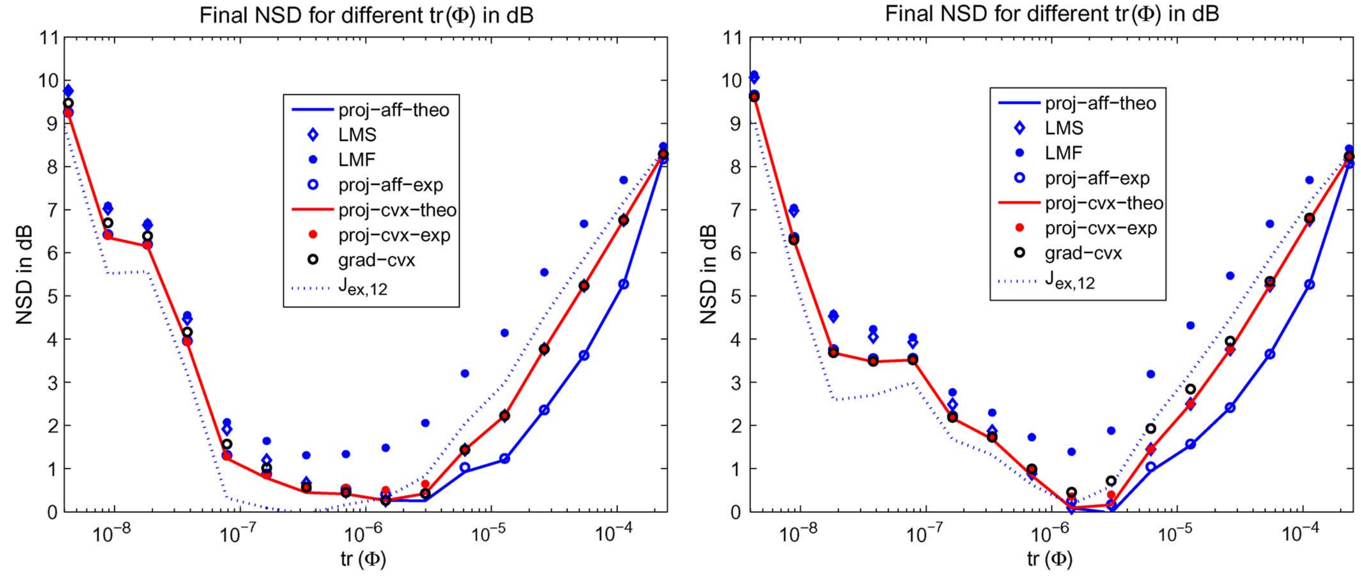
As shown in the Appendix, for this combination, the converged cross correlation is given by

$$J_{\mathrm{ex},12} = \frac{\mu_1\mu_2\mathrm{tr}(\boldsymbol{Q})\left\{3\sigma_n^2 E\left[e_{2,a}^2(t)\right] + E\left[n^4(t)\right]\right\} + \mathrm{tr}(\boldsymbol{\Phi})}{\mu_1 + 3\mu_2\sigma_n^2 - 3\mu_1\mu_2\mathrm{tr}(\boldsymbol{Q})\sigma_n^2}. \tag{29}$$

Hence, even in the stationary case such that $\boldsymbol{\Phi} = \mathbf{0}$, by arranging $\mu_1$, $\mu_2$, we can obtain a cross correlation $J_{\mathrm{ex},12}$ either between the $J_{\mathrm{ex},i}$'s or less than both of $J_{\mathrm{ex},i}$'s and simulate both of the cases where $\boldsymbol{\theta}_o^c = \boldsymbol{\theta}_o^a$ and $\boldsymbol{\theta}_o^c \neq \boldsymbol{\theta}_o^a$. As an example, suppose $n(t)$ is Gaussian and we choose $\mathrm{tr}(\boldsymbol{Q}) = 1$. Further, for $\mu_1$, $\mu_2 \ll 1$ and $\mu_2 = r\ \mu_1$ (for some scaling $r > 0$), we can simplify $J_{\mathrm{ex},12}$ such that the condition $J_{\mathrm{ex},12} < J_{\mathrm{ex},1}$ and $J_{\mathrm{ex},12} < J_{\mathrm{ex},2}$ is satisfied when $1/(15\sigma_n^2) < r < 1/(3\sigma_n^2)$. Hence, the combination of the LMS and LMF filters can provide, even in the stationary case, a wide range of $J_{\mathrm{ex},12}$ to fully exploit the diversity among constituent branches.

## V. Simulations and Conclusion

The first set of experiments involve modeling the seventh-order filter introduced in [1], where $\boldsymbol{w}_o(t) = [0.25, -0.47, -0.37, 0.045, -0.18, 0.78, 0.147]^T$, $\sigma_n^2 = 0.1$ and $\boldsymbol{u}(t)$ are zero mean i.i.d. vectors distributed uniformly with $\mathrm{tr}(\boldsymbol{Q}) = 1$ and $\mathrm{tr}(\boldsymbol{\Phi}) = 0$, i.e., a stationary environment. In Fig. 2(a), we plot the excess MSE for all the methods introduced in this paper as well as $J_o^a = J_o^c$ with respect to $\mu_1$, over 150 iterations and $10^5$ samples. The learning rates of the combination algorithms are given in the caption of Fig. 2. For this case, $r = 2$ such that $1/(15\sigma_n^2) < r = 2 < 1/(3\sigma_n^2)$ such that $J_{\mathrm{ex},12}$ is less than both of the $J_{\mathrm{ex},i}$'s and $\boldsymbol{\theta}_o^c = \boldsymbol{\theta}_o^a$ due to (5) and Section II-A. We note that for such a combination where $\boldsymbol{\theta}_o^a$ is in the interior of the unit simplex, i.e., $\boldsymbol{\theta}_o^a = \boldsymbol{\theta}_o^c \approx [1/2\ 1/2]^T$, and according to (19), the gradient noise actually propagates through the sigmoid nonlinearity. Therefore the MSE of the sigmoid based algorithm, as well as MSEs of the other studied algorithms, are larger than $J_o^c = J_o^a$. To test the validity of (11), (14) and (17), we next simulate all the algorithms with different $\mu$ and $r$ values under nonstationary scenarios. We plot in Fig. 2(b), the excess MSE corresponding to all algorithms, the theoretical curves, i.e., “-theo”, and the minimum excess MSE of the constituent algorithms when $\mathrm{tr}(\boldsymbol{\Phi}) = 10^{-4}$, $\beta = 1$ and $\mathrm{tr}(\boldsymbol{Q}) = 1$. The x axis corresponds to different learning rates for all the algorithms. We observe close agreement with the theory and simulations. We note that the affine constrained algorithms provide more than 2 dB gains for $r = 5$. In the final set of experiments, we simulate the corresponding algorithms for varying $\mathrm{tr}(\boldsymbol{\Phi})$. For this figure, we use a relative figure of merit studied in [1], i.e., for any algorithm $i$, $\mathrm{NSD}_i \triangleq \text{excess MSE}_i / \min\{\min_{\mu_1} J_{\mathrm{ex},1}, \min_{\mu_2} J_{\mathrm{ex},2}\}$, where $J_{\mathrm{ex},1}$ and $J_{\mathrm{ex},2}$ are from (24) and (28), respectively, which is a relative performance with respect to the excess MSE of the best constituent



(a)

(b)

Fig. 3. System identification with seventh-order filter. (a) $\mu_1 = 0.005$, $r = 4$, for the algorithm of Section III-B $\mu = 0.02$ "proj-aff", of Section III-C $\mu = 0.02$ "proj-cvx", of Section III-D $\mu = 0.02$ "grad-cvx", over 200 iterations, $5 \times 10^5$ samples, $\beta = 1$, $\sigma_n^2 = 0.05$. (b) Here, $\beta = 1 - 10^{-4}$.

filter with the optimal learning parameter. We plot NSD values for all the algorithms as well as the theoretical curves in Fig. 3(a), when $\beta = 1$ and in Fig. 3(b) when $\beta = 1 - 10^{-4}$. We again observe a close agreement under different values of $\mathrm{tr}(\boldsymbol{\Phi})$ between the simulations and the theoretical results. For relatively larger values of $\mathrm{tr}(\boldsymbol{\Phi})$, the performance of the affine mixture is significantly better than the other methods since in that region $J_{\mathrm{ex},12}$ is between $J_{\mathrm{ex},1}$ and $J_{\mathrm{ex},2}$. Hence, while the convex mixtures only converge to the performance of the best filter, the affine methods can exploit the full cross correlation inbetween the *a priori* errors. The performance of the convex mixtures are better than the best constituent filter for relatively smaller values of $\mathrm{tr}(\boldsymbol{\Phi})$ where $J_{\mathrm{ex},12}$ is less than both $J_{\mathrm{ex},1}$ and $J_{\mathrm{ex},2}$ for the specific combination of the LMF and LMS filters.

In this paper, we investigated unbiased combination methods where we introduce methods to directly train, i.e., without any variable transformations, the constrained combination weights. We first quantified the achievable diversity gains and then provided the corresponding steady-state MSEs for all studied algorithms. We next specialized the combinations to a mixture of an LMS filter and an LMF filter, where we also derived the corresponding excess cross correlation between the two. The LMF filter and the LMS filter combination is special since we can adjust the excess correlation to yield the optimal affine combination to be the same as the optimal convex combination in stationary environments, which is not possible for any combination of LMS or RLS filters (with the same order) under the studied data model in [1], [4]. We observed more than 2 dB gain in terms of excess MSE with respect to the constituent filters under different nonstationary environments. We show a close agreement between the simulations and the theoretical results.

## Appendix

For the first filter using (21), (22), (23) and for the second filter using (25), (26), (27), and following the lines of [1], we have

$$[\boldsymbol{w}_o(t) - \boldsymbol{w}_i(t)] + e_{i,p}(t)\frac{\boldsymbol{u}(t)}{\|\boldsymbol{u}(t)\|^2} = [\boldsymbol{w}_o(t) - \boldsymbol{w}_i(t+1)] + e_{i,a}(t)\frac{\boldsymbol{u}(t)}{\|\boldsymbol{u}(t)\|^2} \qquad (30)$$

$i = 1, 2$, respectively. Multiplying the left-hand side (LHS) of (30) for $i = 1$ with $i = 2$ yields

$$[\boldsymbol{w}_o(t) - \boldsymbol{w}_1(t)]^T [\boldsymbol{w}_o(t) - \boldsymbol{w}_2(t)] + \frac{e_{1,p}(t)e_{2,p}(t)}{\|\boldsymbol{u}(t)\|^2} = [\boldsymbol{w}_o(t) - \boldsymbol{w}_1(t+1)]^T [\boldsymbol{w}_o(t) - \boldsymbol{w}_2(t+1)] + \frac{e_{1,a}(t)e_{2,a}(t)}{\|\boldsymbol{u}(t)\|^2}$$

after canceling the cross terms. Assuming convergence yields

$$E\left[\frac{e_{1,p}(t)e_{2,p}(t)}{\|\boldsymbol{u}(t)\|^2}\right] + \mathrm{tr}(\boldsymbol{\Phi}) = E\left[\frac{e_{1,a}(t)e_{2,a}(t)}{\|\boldsymbol{u}(t)\|^2}\right]. \qquad (31)$$

Using (23) and (27) to replace $e_{1,p}(t)$ and $e_{2,p}(t)$ terms in (31) results

$$E\left[\mu_2 e_{1,a}(t)e_2^3(t) + \mu_1 e_{2,a}(t)e_1(t)\right] = E\left[\mu_1\mu_2 \|\boldsymbol{u}(t)\|^2 e_1(t)e_2^3(t)\right] + \mathrm{tr}(\boldsymbol{\Phi}). \qquad (32)$$

For the LHS of (32) using $e_i(t) = e_{a,i}(t) + n(t)$, $i = 1, 2$, we have

$$\begin{aligned} &E\left[\mu_2 e_{1,a}(t)\left\{e_{2,a}^3(t) + 3e_{2,a}^2(t)n(t) + 3e_{2,a}(t)n^2(t) + n^3(t)\right\}\right. \\ &\quad \left. + \mu_1 e_{2,a}(t)\left\{e_{1,a}(t) + n(t)\right\}\right] \\ &\approx 3\mu_2 E\left[e_{1,a}(t)e_{2,a}(t)\right]\sigma_n^2 + \mu_1 E\left[e_{1,a}(t)e_{2,a}(t)\right] \end{aligned} \qquad (33)$$

where we omitted third– and higher-order terms for $e_{i,a}(t)$ (or third– and higher-order cross terms) as in ([3, ch. 6]), used that $n(t)$ is i.i.d. and independent of $e_{i,a}(t)$ and assume that $\lim_{t\to\infty} E[e_{i,a}(t)] = 0$. For the right-hand side (RHS) of (32), we have

$$\begin{aligned} &E\left[\mu_1\mu_2 \|\boldsymbol{u}(t)\|^2 \left\{\left[e_{1,a}(t) + n(t)\right]\right.\right. \\ &\quad \times \left[e_{2,a}^3(t) + 3e_{2,a}^2(t)n(t)\right. \\ &\quad \left.\left.\left. + 3e_{2,a}(t)n^2(t) + n^3(t)\right]\right\}\right] + \mathrm{tr}(\boldsymbol{\Phi}) \\ &\approx \mu_1\mu_2 \mathrm{tr}(\boldsymbol{Q})\left\{3\sigma_n^2 E\left[e_{1,a}(t)e_{2,a}(t)\right] + 3\sigma_n^2 E\left[e_{2,a}^2(t)\right]\right. \\ &\quad \left. + E\left[n^4(t)\right]\right\} + \mathrm{tr}(\boldsymbol{\Phi}) \end{aligned} \qquad (34)$$

where we again omit third– and higher-orde terms and use the i.i.d. property of $n(t)$. Using (33) and (34) in (32), we get (29). □

## References

[1] 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, pp. 1078–1090, 2006.

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

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

[4] 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, 2008.

[5] 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, 2008.

[6] A. T. Erdogan, S. S. Kozat, and A. C. Singer, "Comparison of convex combination and affine combination of adaptive filters," in *Proc. ICASSP*, Taipei, Taiwan, 2009.

[7] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "A new least-squares adaptation scheme for the affine combination of two adaptive filters," in *Proc. Mach. Learn. Signal Process. Workshop*, 2008.

[8] D. P. Bertsekas, *Nonlinear Programming*. New York: Athena Scientific, 1999.

# A Quaternion Widely Linear Adaptive Filter

Clive Cheong Took and Danilo P. Mandic

***Abstract*—A quaternion widely linear (QWL) model for quaternion valued mean-square-error (MSE) estimation is proposed. The augmented statistics are first introduced into the field of quaternions, and it is demonstrated that this allows for capturing the complete second order statistics available. The QWL model is next incorporated into the quaternion least mean-square (QLMS) algorithm to yield the widely linear QLMS (WL-QLMS). This allows for a unified approach to adaptive filtering of both $\mathbb{Q}$-proper and $\mathbb{Q}$-improper signals, leading to improved accuracies compared to the QLMS class of algorithms. Simulations on both benchmark and real world data support the analysis.**

***Index Terms*—$\mathbb{Q}$-properness, quadrivariate processes, quaternion adaptive filtering, quaternion LMS (QLMS), quaternion second-order noncircularity, widely linear model, widely linear QLMS, Wiener model.**

## I. Introduction

Standard techniques employed in multichannel statistical signal processing typically do not fully cater for the "coupled" nature of the available information within the channels. Thus, most practical approaches operate based on channelwise processing, which is not optimal for general multivariate signals (where data channels are typically correlated). On the other hand, the quaternion domain $\mathbb{H}$ allows for the direct modeling of three- and four-dimensional signals, and its algebra naturally accounts for the coupling between the signal components.

The use of quaternions is rapidly gaining in popularity, as for instance, many multivariate problems based on vector sensors (motion body sensors, seismics, wind modeling) can be cast into the quaternion domain. The recent resurgence of quaternion valued signal processing stems from the potential advantages that special properties of quaternion algebra offer over real valued vector algebra in multivariate modeling. Applications of quaternions include those in vector sensing [1], machine learning [2], and adaptive filters [3].

Recent advances in complex valued signal processing have been based on the widely linear model proposed by Picinbono [4]. This model, together with the corresponding augmented complex statistics, has been successfully used to design enhanced algorithms in communications [5], [6] and adaptive filters [7]. These studies have shown that widely linear modeling and the associated augmented statistics offer theoretical and practical advantages over the standard complex models, and are applicable to the generality of complex signals, both circular and noncircular.

Models suitable for the processing of signals with rotation dependent distribution (noncircular) are lacking in the quaternion domain, and their development has recently attracted significant research effort [3]. Current second order algorithms operate based on only the quaternion valued covariance [1]–[3] and thus do not fully exploit the available statistical information. Advances in this direction include the work by Vakhania, who defined the concept of $\mathbb{Q}$-properness as the invariance of the distribution of a quaternion valued variable under some specific rotations around the angle of $\pi/2$ [8]. Amblard and Le Bihan relaxed the conditions of $\mathbb{Q}$-properness to an arbitrary axis and angle of rotation $\varphi$, that is, $q \overset{\Delta}{=} e^{\nu\varphi}q$ [9] for any pure unit quaternion $\nu$ (whose real part vanishes); where symbol $\overset{\Delta}{=}$ denotes equality in terms of probability density function (pdf).

Although these results provide an initial insight into the processing of general quaternionic signals, they are not straightforward to apply in the context of adaptive filtering applications. To this end, we first propose the quaternion widely linear model, specifically designed for the unified modeling of the generality of quaternion signals, both $\mathbb{Q}$-proper and $\mathbb{Q}$-improper. The benefits of such an approach are shown to be analogous to the benefits that the augmented statistics provides for complex valued data [7]. Next, the QWL model is incorporated into the quaternion LMS [3] to yield the widely linear QLMS (WL-QLMS), and its theoretical and practical advantages are demonstrated through analysis and simulations.

## II. Properties of Quaternion Random Vectors

### A. Quaternion Algebra

The quaternion domain, a non-commutative extension of the complex domain, provides a natural framework for the processing of three- and four-dimensional signals. A quaternion variable $q \in \mathbb{H}$ comprises a real part $\Re\{\cdot\}$ and a vector-part, also known as a pure quaternion $\Im\{\cdot\}$, consisting of three imaginary components, and can be expressed as

$$\begin{aligned} q &= \Re\{q\} + \Im\{q\} \\ &= \Re\{q\} + \imath\Im_{\imath}\{q\} + \jmath\Im_{\jmath}\{q\} + \kappa\Im_{k}\{q\} \\ &= q_a + \imath q_b + \jmath q_c + \kappa q_d \quad \in \mathbb{H}. \end{aligned} \tag{1}$$

The relationship between the orthogonal unit vectors, $\imath, \jmath, \kappa$ are given by

$$\begin{aligned} &\imath\jmath = \kappa \quad \jmath\kappa = \imath \quad \kappa\imath = \jmath \\ &\imath\jmath\kappa = \imath^2 = \jmath^2 = \kappa^2 = -1. \end{aligned} \tag{2}$$

Manuscript received December 04, 2009; accepted April 08, 2010. Date of publication April 15, 2010; date of current version July 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Isao Yamada.

The authors are with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: c.cheong-took@imperial.ac.uk; d.mandic@imperial.ac.uk).

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

Digital Object Identifier 10.1109/TSP.2010.2048323