A Novel Family of Adaptive Filtering Algorithms Based on the Logarithmic Cost

Muhammed O. Sayin, N. Denizcan Vanli, and Suleyman Serdar Kozat, *Senior Member, IEEE*

***Abstract*—We introduce a novel family of adaptive filtering algorithms based on a relative logarithmic cost inspired by the "competitive methods" from the online learning literature. The competitive or regret based approaches stabilize or improve the convergence performance of adaptive algorithms through relative cost functions. The new family elegantly and gradually adjusts the conventional cost functions in its optimization based on the error amount. We introduce important members of this family of algorithms such as the least mean logarithmic square (LMLS) and least logarithmic absolute difference (LLAD) algorithms. However, our approach and analysis are generic such that they cover other well-known cost functions as described in the paper. The LMLS algorithm achieves comparable convergence performance with the least mean fourth (LMF) algorithm and enhances the stability performance significantly. The LLAD and least mean square (LMS) algorithms demonstrate similar convergence performance in impulse-free noise environments while the LLAD algorithm is robust against impulsive interferences and outperforms the sign algorithm (SA). We analyze the transient, steady-state and tracking performance of the introduced algorithms and demonstrate the match of the theoretical analyses and simulation results. We show the enhanced stability performance of the LMLS algorithm and analyze the robustness of the LLAD algorithm against impulsive interferences. Finally, we demonstrate the performance of our algorithms in different scenarios through numerical examples.**

***Index Terms*—Logarithmic cost function, robustness against impulsive noise, stable adaptive method.**

I. Introduction

ADAPTIVE filtering applications such as channel equalization, noise removal or echo cancellation utilize a certain statistical measure of the error signal[1] e_t denoting the difference between the desired signal d_t and the estimation output $\hat{d}_t$. Usually, the mean square error $E[e_t^2]$ is used as the cost function due to its mathematical tractability and relative ease of analysis. The least mean square (LMS) and normalized least mean square (NLMS) algorithms are the members of this class [1]. Different powers of the error are commonly used as the cost function in order to provide stronger convergence or steady-state performance than the least-squares algorithms under certain settings [1].

The least mean fourth (LMF) algorithm and its family use the even powers of the error as the cost function, i.e., $E[e_t^{2n}]$ [2]. This family achieves a better trade-off between the transient and the steady-state performance, however, has stability issues [3]–[5]. The stability of the LMF algorithm depends on the input and noise power, and the initial value of the adaptive filter weights [6], [7]. On the contrary, the stability of the conventional LMS algorithm depends only on the input power for a given step-size [1]. The normalized filters improve the performance of the algorithms under certain settings by removing dependency to the input statistics in the updates [8]. However, note that the normalized least mean fourth (NLMF) algorithm does not solve the stability issues [6], [7]. In [6], the author proposes the stable NLMF algorithm, which might also be derived through the proposed relative logarithmic error cost framework as shown in this paper.

The performance of the least-squares algorithms degrades severely when the input and desired signal pairs are perturbed by impulsive interferences, e.g., in applications involving high power noise signals [9]. The impulsive noise consists of relatively short duration, infrequent, high amplitude noise pulses. In this context, we define *robustness* as the insensitivity of the algorithms against the impulsive interferences encountered in the practical applications and provide a theoretical framework [10]. Note that, the algorithms using lower-order measures of the error in their cost function are usually relatively less sensitive to such perturbations. For example, the well-known sign algorithm (SA) uses the L_1 norm of the error and is robust against impulsive interferences since its update involves only the sign of e_t. However, the SA usually exhibits slower convergence performance especially for highly correlated input signals [11].

The mixed-norm algorithms minimize a combination of different error norms in order to achieve improved convergence performance [12], [13]. For example, [13] combines the robust L_1 norm and the more sensitive but better converging L_2 norm through a mixing parameter. Even though the combination parameter brings in an extra degree of freedom, the design of the mixed norm filters requires the optimization of the mixing parameter based on *a priori* knowledge of the input and noise statistics. On the contrary, the mixture of experts algorithms adaptively combine different algorithms and provide improved performance irrespective of the environment statistics [14]–[17]. However, these mixture approaches require to operate several

Manuscript received September 16, 2013; revised March 12, 2014; accepted June 09, 2014. Date of publication June 27, 2014; date of current version August 06, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Raviv Raich. This work was supported in part by the Outstanding Researcher Programme of Turkish Academy of Sciences and TUBITAK Projects under Contracts 112E161 and 113E517.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara 06800, Turkey (e-mail: sayin@ee.bilkent.edu.tr; vanli@ee.bilkent.edu.tr; kozat@ee.bilkent.edu.tr).

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.2014.2333559

[1]Time index appears as a subscript.

different algorithms in parallel, which may be infeasible in different applications [18]. In [19], the authors propose an adaptive combination of the L_1 and L_2 norms of the error in parallel, however, the resulting algorithm demonstrates impulsive perturbations on the learning curves. This happens since the impulsive interferences severely degrade the algorithmic updates. In general, the samples contaminated with impulses contain little useful information [10]. Hence, the robust algorithms need to be less sensitive only against large perturbations on the error and can be as sensitive as the conventional least squares algorithms for small error values. The switched-norm algorithms switch between the L_1 and L_2 norms based on the error amount such as the robust Huber filter [20]. This approach combines the better convergence of L_2 and the robustness of L_1 together in a discrete manner with a breaking point in the cost function, however, requires optimization of certain parameters as detailed in this paper.

In this paper, we are inspired from the recent developments in the computational learning theory related to the "competitive" or the "regret" based approaches [21]–[23]. In these approaches, the well known adaptive or online learning algorithms are "stabilized" or "improved" by using a relative cost measure, i.e., the regret. Hence, we mitigate the stability or convergence issues of the well known adaptive algorithms by introducing a relative cost measure. To this end, we use *diminishing return* functions, e.g., the logarithm function, as a normalization (or a regularization) term, i.e., as a subtracting term, in the cost function. We particularly choose the logarithm function as the normalizing diminishing return function [24] in our cost definitions since the logarithmic function is differentiable and results efficient and mathematically tractable adaptive algorithms. As demonstrated in this paper, by using such a relative cost measure, we adjust the conventional cost functions elegantly and gradually in its optimization based on the error amount. We intrinsically use the higher-order statistics of the error for small perturbations. For larger error values, the introduced algorithms seek to minimize the conventional cost functions, due to the decreasing weight of the logarithmic term with the increased error amount. In this sense, the new framework is also akin to a continuous generalization of the switched norm algorithms, hence greatly improves the convergence performance of the mixed-norm methods as shown in this paper.

Our main contributions include: 1) We propose the least mean logarithmic square (LMLS) algorithm, which achieves a similar trade-off between the transient and the steady-state performance of the LMF algorithm, and as stable as the LMS algorithm; 2) We propose the least logarithmic absolute difference (LLAD) algorithm, which significantly improves the convergence performance of the SA while exhibiting comparable performance with the SA in the impulsive noise environments; 3) We analyze the transient, the steady-state and the tracking performance of the introduced algorithms; 4) We demonstrate the extended stability bound on the step-sizes with the logarithmic error cost framework; 5) We introduce an impulsive noise framework and analyze the robustness of the LLAD algorithm in the impulsive noise environments; 6) We demonstrate the significantly improved convergence performances of the introduced algorithms in several different scenarios in our simulations.

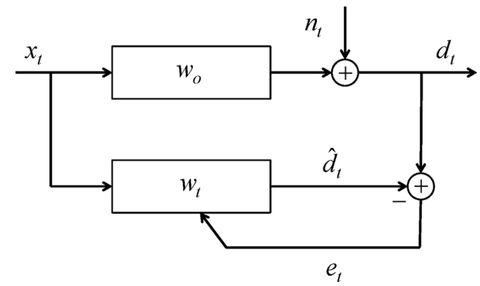


Fig. 1. Generic system identification framework.

We organize the paper as follows. In Section II, we introduce the relative logarithmic error cost framework. In Section III, the important members of the novel family are derived. We analyze the transient, the steady-state and the tracking performances of those members in Section IV. In Section V, we compare the stability bound on the step-sizes and robustness of the proposed algorithms. In Section VI, we provide the numerical examples demonstrating the improved performance of the conventional algorithms in the new logarithmic error cost framework. We conclude the paper in Section VII with several remarks.

Notation: Bold lower (or upper) case letters denote the vectors (or matrices). For a vector $\mathbf{a}$ (or matrix $\mathbf{A}$), $\mathbf{a}^T$ (or $\mathbf{A}^T$) is its ordinary transpose. $\|\cdot\|$ and $\|\cdot\|_{\mathbf{A}}$ denote the L_2 norm and the weighted L_2 norm with the matrix $\mathbf{A}$, respectively (provided that $\mathbf{A}$ is positive-definite). $|\cdot|$ is the absolute value operator. We work with real data for notational simplicity. For a random variable x (or vector $\mathbf{x}$), $E[x]$ (or $E[\mathbf{x}]$) represents its expectation. Here, $\text{Tr}(\mathbf{A})$ denotes the trace of the matrix $\mathbf{A}$ and $\nabla_{\mathbf{x}} f(\mathbf{x})$ is the gradient operator.

II. Cost Function With Logarithmic Error

We consider the system identification framework shown in Fig. 1, where we denote the input signal by $\mathbf{x}_t$ and the desired signal by d_t. Here, we observe an unknown vector[2] $\mathbf{w}_o \in \mathbb{R}^p$ through a linear model

$$d_t = \mathbf{w}_o^T \mathbf{x}_t + n_t,$$

where n_t represents the noise. We define the error signal as $e_t \triangleq d_t - \hat{d}_t = d_t - \mathbf{w}_t^T \mathbf{x}_t$ where $\mathbf{w}_t$ is the weight vector of the adaptive filter. In this framework, adaptive filtering algorithms estimate the unknown system vector $\mathbf{w}_o$ through the minimization of a certain cost function. The gradient descent methods usually employ convex and uni-modal cost functions in order to converge to the global minimum of the error surfaces, e.g., the mean square error $E[e_t^2]$ [1]. The different powers of e_t [2], [11] or a linear combination of different error powers [12], [13] are also widely used. The authors in [25] demonstrate that the optimum error nonlinearity function could be expressed as a linear combination of different orders of the error, i.e., e_t, for certain combination coefficients; and a mixture of the LMS algorithm and the LMF family of algorithms can approximate the

[2]Although we assume a time invariant unknown system vector here, we also provide the tracking performance analysis for certain non-stationary models later in the paper.

optimum error nonlinearity better than any of the individual algorithms. However, we emphasize that mixture coefficients are time varying since they depend on the variance of the *a priori* error. Hence, an algorithm combining different orders of the error terms with combination weights changing in time based on the error might better approximate the optimum error nonlinearity as shown in this paper.

In this framework, we introduce a normalized error cost function using the logarithm function given by

$$J(e_t) \triangleq F(e_t) - \frac{1}{\alpha}\ln(1+\alpha F(e_t)), \tag{1}$$

where $\alpha > 0$ is a design parameter and $F(e_t)$ is a conventional cost function of the error signal e_t, e.g., $F(e_t) = E[e_t^2]$ or $F(e_t) = E[|e_t|]$. By Maclaurin series of the natural logarithm for $|\alpha F(e_t)| \leq 1$, (1) yields

$$\begin{aligned} J(e_t) &= F(e_t) - \frac{1}{\alpha}\left(\alpha F(e_t) - \frac{\alpha^2}{2}F^2(e_t) + \cdots\right) \\ &= \frac{\alpha}{2}F^2(e_t) - \frac{\alpha^2}{3}F^3(e_t) + \cdots, \end{aligned} \tag{2}$$

which is an infinite combination of the conventional cost functions for small values of $F(e_t)$. We emphasize that the cost function (2) yields to the second power of the cost function $F(e_t)$ for small values of the error while for relatively large error values, the cost function $J(e_t)$ resembles $F(e_t)$ as follows:

$$F(e_t) - \frac{1}{\alpha}\ln(1+\alpha F(e_t)) \to F(e_t) \text{ as } e_t \to \infty.$$

Hence, the new methods use mainly the combination of $F^2(e_t)$ or $F(e_t)$ cost functions based on the error amount. Note that the objective functions $F^2(e_t)$, e.g., $E[e_t^2]^2$, and $F(e_t^2)$, e.g., $E[e_t^4]$, yields the same stochastic gradient update after removing the expectation in this paper. Hence, the proposed logarithmic cost function yields error nonlinearities combining different order of the error terms where the combination weights depend on the error signal. Particularly, through the new family of algorithms we could better approximate the optimum error nonlinearity even during the transient region [1], [26].

The switched norm algorithms also combine two different norms into a single update in a discrete manner based on the error amount. As an example, the Huber objective function combining L_1 and L_2 norms of the error is defined as [20]

$$\rho(e_t) \triangleq \begin{cases} \frac{1}{2}e_t^2 & \text{for } |e_t| \leq \gamma, \\ \gamma|e_t| - \frac{1}{2}\gamma^2 & \text{for } |e_t| > \gamma, \end{cases} \tag{3}$$

where $\gamma > 0$ denotes the cut-off value. As an illustration, in Fig. 2, we compare $|e_t|$, the Huber objective function for $\gamma = 1$, and the introduced cost (1) with $F(e_t) = |e_t|$ for $\alpha = 1$ (i.e., $|e_t| - \ln(1+|e_t|)$). From this plot, we observe that the logarithm based cost function is less steep for small perturbations on the error while both the logarithmic cost and absolute difference cost functions exhibit comparable steepness for relatively larger error values. Furthermore, this new family intrinsically combines the benefits of using lower and higher-order measures of the error into a single adaptation algorithm. Our algorithms provide comparable convergence rate with a conventional algorithm minimizing the cost function $F(e_t)$ and achieve smaller steady-state mean square errors through the use of the higher-order statistics for small perturbations of the error.

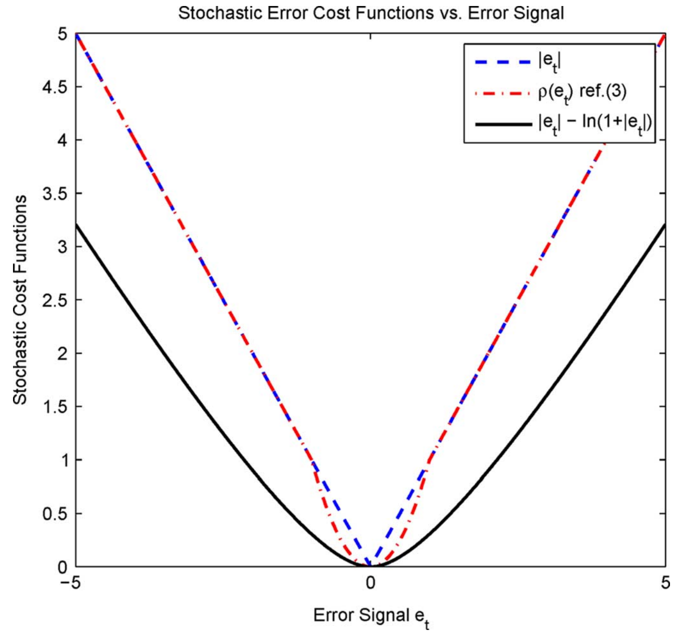


Fig. 2. A comparison of the stochastic cost functions illustrating the decreased steepness of the absolute difference algorithm in the logarithmic error cost framework for relatively small error.

We note that the robust Huber cost definition, i.e., (3), uses a piecewise-function combining two different algorithms based on the comparison of the error with the cut-off value γ. On the contrary, the logarithm based cost function $J(e_t)$ intrinsically combines the functions with different order of powers in a continuous manner into a single update and avoids possible anomalies that might arise due to the breaking point in the cost function as shown later in the paper.

As a side note, in the context of online learning, authors in [27] demonstrate that the LMS algorithm can achieve the optimal deterministic performance in hindsight through the step size decreasing in time. However, such an algorithm cannot respond to the changing statistical profiles since the steps diminish in time. In the new logarithmic cost definition, we inspire from [27] in order to attain more gradual steps for small error values in any conventional algorithm through the diminishing return property of the logarithm function. In addition, the proposed cost function can respond to the changes in the statistical profiles or to the possible outliers instantly. Particularly, the proposed algorithm achieves smaller error in time and takes more gradual steps. If the error gets larger due to a statistical profile change or an outlier, the algorithm can respond immediately by taking relatively steeper steps.

Remark 2.1: In [28], the authors propose a stochastic cost function using the logarithm function as follows

$$J_{[28]}(e_t) \triangleq \frac{1}{2\gamma}\ln\left(1+\gamma\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)^2\right).$$

Note that the cost function $J_{[28]}(e_t)$ is the subtracted term in (1) for $F(e_t) = \frac{e_t^2}{\|\mathbf{x}_t\|^2}$. The Hessian matrix of $J_{[28]}(e_t)$ is given by

$$\mathbf{H}\left(J_{[28]}(e_t)\right) = \frac{\mathbf{x}_t\mathbf{x}_t^T}{\|\mathbf{x}_t\|^2\left(1+\gamma\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)^2\right)} \times \left(1 - \frac{2\gamma e_t^2}{\|\mathbf{x}_t\|^2\left(1+\gamma\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)^2\right)}\right).$$

We emphasize that $\mathbf{H}(J_{[28]}(e_t))$ is positive semi-definite provided that $\gamma(\frac{e_t}{\|\mathbf{x}_t\|})^2 \leq 1$, thus, the parameter γ should be chosen carefully to be able to efficiently use the gradient descent algorithms.

On the contrary, the new cost function in (1) is a convex function enabling the use of the diminishing return property [24] of the logarithm function for stable and robust updates. The Hessian matrix of $J(e_t)$ is given by

$$\mathbf{H}\left(J(e_t)\right) = \mathbf{H}\left(F(e_t)\right)\frac{\alpha F(e_t)}{1+\alpha F(e_t)} + \frac{\alpha \nabla_{\mathbf{w}} F(e_t) \nabla_{\mathbf{w}} F(e_t)^T}{\left(1+\alpha F(e_t)\right)^2},$$

which is positive semi-definite provided that $\mathbf{H}(F(e_t))$ is a positive semi-definite matrix and $F(e_t)$ is a non-negative cost function.

In the following, we demonstrate that the optimal solution for the relative logarithmic error cost function, i.e., $J(e_t)$, is the same with the cost function $F(e_t)$. The first gradient of (1) is given by

$$\nabla_{\mathbf{w}} J(e_t) = \nabla_{\mathbf{w}} F(e_t)\frac{\alpha F(e_t)}{1+\alpha F(e_t)},$$

which yields zero if $\nabla_{\mathbf{w}} F(e_t)$ is a zero vector or $F(e_t)$ is zero. The optimal solution for the cost function $F(e_t)$ minimizes $F(e_t)$ and is obtained by

$$\nabla_{\mathbf{w}=\mathbf{w}_o} F(e_t) = \mathbf{0}.$$

Since $F(e_t)$ is a non-negative convex function, the global minimum and the value yielding zero gradient coincide if the latter exits. Hence, the optimal solution for the relative logarithmic error cost function is the same with the cost function $F(e_t)$ since the Hessian matrix of the logarithmic cost definition is positive semi-definite. As an example, the mean-square error cost function $F(e_t) = E[e_t^2]$ in the logarithmic cost framework, i.e., $J(e_t) = E[e_t^2] - \alpha^{-1}\ln(1+\alpha E[e_t^2])$, yields to the Wiener solution $\mathbf{w}_o = E[\mathbf{x}_t\mathbf{x}_t^T]^{-1}E[\mathbf{x}_t d_t]$.

Remark 2.2: Instead of a logarithmic normalization term, it is also possible to use various functions having diminishing returns property in order to provide stability and robustness to the conventional algorithms. For example, one can choose the cost function as

$$J_{\arctan}(e_t) \triangleq F(e_t) - \frac{1}{\alpha}\arctan(\alpha F(e_t)) \tag{4}$$

and the Taylor series expansion of the second term in (4) around $F(e_t) = 0$ is given by

$$\frac{1}{\alpha}\arctan\left(1+\alpha F(e_t)\right) = F(e_t) - \frac{\alpha^2}{3}F^3(e_t) + \cdots.$$

Thus, the resulting algorithm combines the algorithms using mainly $F^3(e_t)$ (for small perturbations on the error) and $F(e_t)$. We note that the algorithms using (4) are also as stable as $F(e_t)$, however, they behave like minimizing the higher-order measures, i.e., $F^3(e_t)$, for small error values.

In the next section, we propose important members of this novel adaptive filter family.

III. Proposed Algorithms

Based on the gradient of $J(e_t)$ we obtain the general steepest descent update as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mu\nabla_{\mathbf{w}} F(e_t)\frac{\alpha F(e_t)}{1+\alpha F(e_t)},$$

where $\mu > 0$ is the step size and $\alpha > 0$ is the design parameter. If we assume that after removing the expectation to generate stochastic gradient updates $F(e_t)$ yields $f(e_t)$, e.g., $F(e_t) = E[f(e_t)]$, then the general stochastic gradient update is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu\mathbf{x}_t\frac{\partial f(e_t)}{\partial e_t}\frac{\alpha f(e_t)}{1+\alpha f(e_t)}. \tag{5}$$

In the following subsections, we introduce algorithms improving the performance of the conventional algorithms such as the LMS (i.e., $f(e_t) = e_t^2$), SA (i.e., $f(e_t) = |e_t|$) and normalized updates.

A. The Least Mean Logarithmic Square (LMLS) Algorithm

For $F(e_t) = E[e_t^2]$, the stochastic gradient update yields

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t + \mu\mathbf{x}_t e_t\frac{\alpha e_t^2}{1+\alpha e_t^2} \\ &= \mathbf{w}_t + \mu\frac{\alpha\mathbf{x}_t e_t^3}{1+\alpha e_t^2}.\end{aligned} \tag{6}$$

Note that we incorporate the multiplier '2' from the gradient $\nabla_{e_t} e_t^2 = 2e_t$ into the step-size μ. The algorithm (6) resembles a least-mean fourth update for the small error values while it behaves like the least-mean square algorithm for large perturbations on the error. This provides smaller steady-state mean square error thanks to the fourth-order statistics of the error for small perturbations and stability of the least-squares algorithms for large perturbations. Hence, the LMLS algorithm intrinsically combines the least mean-square and least-mean fourth algorithms based on the error amount instead of mixed LMF + LMS algorithms [12] that need artificial combination parameter in the cost definition.

B. The Least Logarithmic Absolute Difference (LLAD) Algorithm

The SA utilizes $F(e_t) = E[|e_t|]$ as the cost function, which provides robustness against impulsive interferences [1]. However, the SA has slower convergence rate since the L_1 norm is the smallest possible error power for a convex cost function. In the logarithmic cost framework, for $F(e_t) = E[|e_t|]$, (5) yields

$$\begin{aligned} \mathbf{w}_{t+1} &= \mathbf{w}_t + \mu \mathbf{x}_t \mathrm{sign}(e_t) \frac{\alpha |e_t|}{1+\alpha|e_t|} \\ &= \mathbf{w}_t + \mu \frac{\alpha \mathbf{x}_t e_t}{1+\alpha|e_t|}. \end{aligned} \quad (7)$$

The algorithm (7) combines the LMS algorithm and SA into a single robust algorithm with improved convergence performance. We note that in Section V we calculate the optimum α_{opt} in order to achieve better convergence performance than the SA in the impulsive noise environments.

C. Normalized Updates

We introduce normalized updates with respect to the regressor signal in order to provide independence from the input data correlation statistics under certain settings. We define the new objective function as

$$J_{\mathrm{new}}(e_t) \triangleq F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right) - \frac{1}{\alpha}\ln\left(1+\alpha F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)\right),$$

for example $F(\frac{e_t}{\|\mathbf{x}_t\|}) = E[\frac{e_t^2}{\|\mathbf{x}_t\|^2}]$. The Hessian matrix of the new cost function $J_{\mathrm{new}}(e_t)$ is also positive semi-definite provided that the Hessian matrix of $F(\frac{e_t}{\|\mathbf{x}_t\|})$ is positive semi-definite.

The steepest-descent update is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mu \nabla_{\mathbf{w}} F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right) \frac{\alpha F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)}{1+\alpha F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)}.$$

For $F(\frac{e_t}{\|\mathbf{x}_t\|}) = E[(\frac{e_t}{\|\mathbf{x}_t\|})^2]$, we get the normalized least mean logarithmic square (NLMLS) algorithm given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu \frac{\alpha \mathbf{x}_t e_t^3}{\|\mathbf{x}_t\|^2 \left(\|\mathbf{x}_t\|^2 + \alpha e_t^2\right)}. \quad (8)$$

We point out that (8) is also proposed as the stable normalized least mean-fourth algorithm in [6].

For $F(\frac{e_t}{\|\mathbf{x}_t\|}) = E[\frac{|e_t|}{\|\mathbf{x}_t\|}]$, we obtain the normalized least logarithmic absolute difference (NLLAD) algorithm as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu \frac{\alpha \mathbf{x}_t e_t}{\|\mathbf{x}_t\|(\|\mathbf{x}_t\| + \alpha|e_t|)}.$$

Remark 3.1: Through the logarithmic cost function, in general, we can enhance the convergence performance for any conventional cost function (in such cases, a typical choice for the design parameter is $\alpha = 1$). In the environments where the higher order error measure cannot converge (e.g., in the impulsive noise environment the LMS algorithm with mean-square error cost does not usually converge) the performance of the corresponding logarithmic cost-induced algorithm (the LLAD algorithm using the mean absolute difference of the error in the logarithmic cost framework) may degrade with respect to the performance of the associated conventional algorithm. In the definition of the new cost function, we introduce the design parameter α, i.e., an additional freedom of dimension, such that through the optimization of α we can still achieve enhanced convergence performance even in such unrealistic environments. Actually, even in such unrealistic conditions the proposed algorithms provide enhanced convergence performance for the algorithms with the cost function $F(e_t)^2$. As an example, the LMLS algorithms overcome the stability issues of the LMF algorithm and the LLAD algorithm provides robustness to the LMS algorithm. Moreover, we can improve the convergence performance of the logarithmic cost-induced algorithms further by optimizing α with prior information about the environment.

Remark 3.2: We note that the proposed algorithms enhance the convergence performance of the conventional algorithms, e.g., the LMS algorithm and the SA, through similar computational cost. In Table I, we tabulate the detailed computational cost of the proposed algorithms, where we compare the estimated computational cost per iteration for the SA, LLAD, LMS, LMLS and LMF algorithms for real valued data in terms of multiplications, additions, divisions and sign evaluations.

In the next section, we analyze the transient, steady-state and tracking performance of the introduced algorithms.

IV. Performance Analysis

The performance analysis will be carried out by following the energy-conversation approach of [1], [25], [29]. We define *a priori* estimation error and the weighted form as

$$e_{a,t} \triangleq \mathbf{x}_t^T \tilde{\mathbf{w}}_t \text{ and } e_{a,t}^{\mathbf{\Sigma}} \triangleq \mathbf{x}_t^T \mathbf{\Sigma} \tilde{\mathbf{w}}_t,$$

where $\tilde{\mathbf{w}}_t \triangleq \mathbf{w}_o - \mathbf{w}_t$ and $\mathbf{\Sigma}$ is a symmetric positive definite weighting matrix. Note that different choice of $\mathbf{\Sigma}$ leads to the different performance measures of the algorithm [1]. The general weighted-energy recursion for an adaptive filter with error nonlinearity [26] is given by

$$\begin{aligned} E\left[\|\tilde{\mathbf{w}}_{t+1}\|_{\mathbf{\Sigma}}^2\right] &= E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}}^2\right] - \mu 2 E\left[e_{a,t}^{\mathbf{\Sigma}} g(e_t)\right] \\ &\quad + \mu^2 E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2 g^2(e_t)\right]. \end{aligned} \quad (9)$$

where $g(e_t)$ is the nonlinear error function. For the proposed algorithms, $g(e_t)$ is defined as

$$g(e_t) \triangleq \frac{\partial f(e_t)}{\partial e_t} \frac{\alpha f(e_t)}{1+\alpha f(e_t)}. \quad (10)$$

TABLE I
A COMPARISON OF THE ESTIMATED COMPUTATIONAL COST PER ITERATION FOR THE PROPOSED ALGORITHMS AND THE CONVENTIONAL ALGORITHMS (p IS THE FILTER LENGTH)

Algorithm	$h_G(e_t)$	$h_U(e_t)$
LMF	$3\sigma_e^2$	$15\sigma_e^6$
LMLS	$1-2\lambda\left(1-\sqrt{\pi\lambda}\exp(\lambda)\text{erfc}(\sqrt{\lambda})\right)$	$\sigma_e^2\left(1-2\lambda(\lambda+2)+\lambda(2\lambda+5)\sqrt{\pi\lambda}\exp(\lambda)\text{erfc}\left(\sqrt{\lambda}\right)\right)$
LMS	1	σ_e^2
LLAD	$\frac{1}{\sigma_e}\sqrt{\frac{2}{\pi}}\left(1-\sqrt{\kappa\pi}+\kappa\frac{\pi\text{erfi}(\sqrt{\kappa})-\text{Ei}(\kappa)}{\exp(\kappa)}\right)$	$1-2\kappa+2\sqrt{\frac{\kappa}{\pi}}\left(1+(\kappa-1)\frac{\pi\text{erfi}(\sqrt{\kappa})-\text{Ei}(\kappa)}{\exp(\kappa)}\right)$
SA	$\frac{1}{\sigma_e}\sqrt{\frac{2}{\pi}}$	1

In the subsequent analysis of (9), we use the following assumptions:

Assumption 1: The observation noise n_t is zero-mean independently and identically distributed (i.i.d.) Gaussian random variable and independent from $\mathbf{x}_t$. The regressor signal $\mathbf{x}_t$ is also zero-mean i.i.d. Gaussian random variable with the autocorrelation matrix $\mathbf{R} \triangleq E[\mathbf{x}_t\mathbf{x}_t^T]$.

Assumption 2: The *a priori* estimation error $e_{a,t}$ has Gaussian distribution and it is jointly Gaussian with the weighted *a priori* estimation error $e_{a,t}^{\mathbf{\Sigma}}$ for any constant matrix $\mathbf{\Sigma}$. The assumption is reasonable for long filters, i.e., p is large, sufficiently small step size μ and by the Assumption 1 [26].

Assumption 3: The random variables $\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2$ and $g^2(e_t)$ are uncorrelated, which enables the following split as

$$E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2 g^2(e_t)\right] = E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2\right]E\left[g^2(e_t)\right].$$

We next analyze the transient behavior of the new algorithms.

A. Transient Analysis

The Assumptions 1 and 2 imply that the estimation error $e_t = e_{a,t} + n_t$ has Gaussian distribution since it is the addition of two independent Gaussian random variable. Hence, we use the Lemma 1 and (14) from [29] to get:

Lemma 1: Under Assumptions 1–3, we have

$$E\left[e_{a,t}^{\mathbf{\Sigma}} g(e_t)\right] = E\left[e_{a,t}^{\mathbf{\Sigma}} e_t\right]\frac{E[e_t g(e_t)]}{E\left[e_t^2\right]}. \quad (11)$$

Proof: The proof of Lemma 1 follows from the Price's result [30], [31]. That is, for any Borel function $g(b)$ we can write

$$E[xg(y)] = \frac{E[xy]}{E[y^2]}E[yg(y)],$$

where x and y are zero-mean jointly Gaussian random variables [32]. Hence by Assumptions 1–3, we obtain (11) and the proof is concluded. □

The weighted-error recursion (9) could be written as follows [26]

$$E\left[\|\tilde{\mathbf{w}}_{t+1}\|_{\mathbf{\Sigma}}^2\right] = E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}}^2\right] - \mu 2h_G(e_t)E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma R}}^2\right] + \mu^2 E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2\right]h_U(e_t), \quad (12)$$

where

$$h_G(e_t) \triangleq \frac{E[e_t g(e_t)]}{E[e_t^2]},\ h_U(e_t) \triangleq E[g^2(e_t)].$$

We, next, evaluate $h_G(e_t)$ and $h_U(e_t)$ functions of the LMLS and the LLAD algorithms.

1) $h_G(e_t)$ Function of the LMLS Algorithm: We have

$$h_G(e_t) = \frac{1}{\sigma_e^2}E\left[\frac{\alpha e_t^4}{1+\alpha e_t^2}\right], = \frac{1}{\sigma_e^2}\left(\sigma_e^2 - \alpha^{-1} + \alpha^{-1}E\left[\frac{1}{1+\alpha e_t^2}\right]\right), \quad (13)$$

where $\sigma_e^2 = E[e_t^2]$ and the first line of the equation follows according to the definition of $g(e_t)$ in (10). Since e_t has Gaussian distribution by the Assumptions 1 and 2, we obtain the last term in (13) as follows

$$E\left[\frac{1}{1+\alpha e_t^2}\right] = \frac{1}{\sqrt{2\pi}\sigma_e}\int_{-\infty}^{\infty}\frac{1}{1+\alpha e_t^2}\exp\left(-\frac{e_t^2}{2\sigma_e^2}\right)de_t = \frac{1}{\sqrt{2\alpha\pi}\sigma_e}\int_{-\infty}^{\infty}\frac{\exp}{1+u^2}du = \frac{1}{\sqrt{2\alpha\pi}\sigma_e}\pi\exp(\lambda)\,\text{erfc}\,(\sqrt{\lambda}), \quad (14)$$

where $u \triangleq \sqrt{\alpha}e_t$, $\lambda \triangleq \frac{1}{2\alpha\sigma_e^2}$, and the third line follows from [33] with $\text{erfc}(\cdot)$ denoting the complementary error function. Hence, putting (14) in (13), we obtain $h_G(e_t)$ for the LMLS update as

$$h_G(e_t) = 1 - 2\lambda\left(1-\sqrt{\pi\lambda}\exp(\lambda)\,\text{erfc}\,(\sqrt{\lambda})\right).$$

2) $h_G(e_t)$ Function of the LLAD Algorithm: Correspondingly, we have

$$h_G(e_t) = \frac{1}{\sigma_e^2}E\left[\frac{\alpha e_t^2}{1+\alpha|e_t|}\right], = \frac{1}{\sigma_e^2}\left(E[|e_t|] - \alpha^{-1} + \alpha^{-1}E\left[\frac{1}{1+\alpha|e_t|}\right]\right), \quad (15)$$

where the first line follows according to the definition of $g(e_t)$ in (10). Through the Assumptions 1 and 2, we obtain the last term in (15) as follows

$$E\left[\frac{1}{1+\alpha|e_t|}\right] = \frac{1}{\sqrt{2\pi}\sigma_e}\int_{-\infty}^{\infty}\frac{1}{1+\alpha|e_t|}\exp\left(-\frac{e_t^2}{2\sigma_e^2}\right)de_t$$
$$= \frac{1}{\sqrt{2\pi}\alpha\sigma_e}\int_{-\infty}^{\infty}\frac{1}{1+|u|}\exp\left(-\kappa u^2\right)du$$
$$= \frac{1}{\sqrt{2\pi}\alpha\sigma_e}\frac{\pi\mathrm{erfi}(\sqrt{\kappa})-\mathrm{Ei}(\kappa)}{\exp(\kappa)}, \quad (16)$$

where $u \triangleq \alpha e_t$, and $\kappa \triangleq \frac{1}{2\alpha^2\sigma_e^2}$, and the third line follows from [33] with $\mathrm{erfi}(z) = -j\mathrm{erf}(jz)$ denoting the imaginary error function and $\mathrm{Ei}(x)$ denoting the exponential integral, i.e.,

$$\mathrm{Ei}(x) = -\int_{-x}^{\infty}\frac{\exp(-t)}{t}dt. \quad (17)$$

We note that $\mathrm{Ei}(x)$ is the Cauchy principal value of (17) due to the singularity of the integrand at $t=0$ [33]. Therefore, putting (16) in (15), we obtain $h_G(e_t)$ for the LLAD update as

$$h_G(e_t) = \frac{1}{\sigma_e}\sqrt{\frac{2}{\pi}}\left(1-\sqrt{\kappa\pi}+\kappa\frac{\pi\mathrm{erfi}(\sqrt{\kappa})-\mathrm{Ei}(\kappa)}{\exp(\kappa)}\right).$$

3) $h_U(e_t)$ Function of the LMLS Algorithm: We have

$$h_U(e_t) = E\left[\frac{\alpha^2 e_t^6}{(1+\alpha e_t^2)^2}\right]$$
$$= E\left[-\alpha^2\frac{\partial}{\partial\alpha}\left(\frac{e_t^4}{1+\alpha e_t^2}\right)\right]$$
$$= -\alpha^2\frac{\partial}{\partial\alpha}\left(E\left[\frac{e_t^4}{1+\alpha e_t^2}\right]\right),$$

where in the last line we applied the interchange of integration and differentiation property since $\theta(e_t,\alpha) \triangleq \frac{e_t^4}{1+\alpha e_t^2}$ and $\frac{\partial\theta(e_t,\alpha)}{\partial\alpha}$ are both continuous in $\mathbb{R}^2$. Hence, we have $h_U(e_t)$ for the LMLS algorithm as

$$h_U(e_t) = -\alpha^2\frac{\partial}{\partial\alpha}\left(\alpha^{-1}E\left[\frac{\alpha e_t^4}{1+\alpha e_t^2}\right]\right)$$
$$= -\alpha^2\frac{\partial}{\partial\alpha}\left(\alpha^{-1}\sigma_e^2 h_G(e_t)\right)$$
$$= \sigma_e^2\left(1-2\lambda(\lambda+2)\right.$$
$$\left.+\ \lambda(2\lambda+5)\sqrt{\pi\lambda}\exp(\lambda)\mathrm{erfc}\left(\sqrt{\lambda}\right)\right).$$

4) $H_U(e_t)$ Function of the LLAD Algorithm: Correspondingly, we have

$$h_U(e_t) = E\left[\frac{\alpha^2 e_t^2}{(1+\alpha|e_t|)^2}\right]$$
$$= E\left[-\alpha^2\frac{\partial}{\partial\alpha}\left(\frac{|e_t|}{1+\alpha|e_t|}\right)\right]$$
$$= -\alpha^2\frac{\partial}{\partial\alpha}\left(E\left[\frac{|e_t|}{1+\alpha|e_t|}\right]\right),$$

TABLE II
$h_G(e_t)$ AND $h_U(e_t)$ CORRESPONDING TO THE STOCHASTIC COSTS e_t^2 AND $|e_t|$, WHERE $\sigma_e^2 = E[e_t^2]$ AND $\lambda = \frac{1}{2\alpha\sigma_e^2} = \alpha\kappa$

Algorithm	×	+	/	sign
SA	$2p$	$2p$		1
LLAD	$2p+1$	$2p+1$	1	1
LMS	$2p+1$	$2p$		
LMLS	$2p+3$	$2p+1$	1	
LMF	$2p+3$	$2p$		

where in the last line we applied the interchange of integration and differentiation property since $\theta(e_t,\alpha) \triangleq \frac{|e_t|}{1+\alpha|e_t|}$ and $\frac{\partial\theta(e_t,\alpha)}{\partial\alpha}$ are both continuous in $\mathbb{R}^2$. Therefore, we obtain $h_U(e_t)$ for the LLAD algorithm as

$$h_U(e_t) = -\alpha^2\frac{\partial}{\partial\alpha}\left(\alpha^{-1}E\left[\frac{\alpha|e_t|}{1+\alpha|e_t|}\right]\right)$$
$$= -\alpha^2\frac{\partial}{\partial\alpha}\left(\alpha^{-1}\left(1-E\left[\frac{1}{1+\alpha|e_t|}\right]\right)\right)$$
$$= -\alpha^2\frac{\partial}{\partial\alpha}\left(\alpha^{-1}\left(1-\frac{1}{\sqrt{2\pi}\alpha\sigma_e}\frac{\pi\mathrm{erfi}(\sqrt{\kappa})-\mathrm{Ei}(\kappa)}{\exp(\kappa)}\right)\right)$$
$$= 1-2\kappa+2\sqrt{\frac{\kappa}{\pi}}\left(1+(\kappa-1)\frac{\pi\mathrm{erfi}\left(\sqrt{\kappa}\right)-\mathrm{Ei}(\kappa)}{\exp(\kappa)}\right),$$

where the third line follows from (16).

We tabulate the evaluated results with the results for the LMS algorithm, LMF algorithm and SA in Table II.

Using (12) and the evaluated $h_G(e_t)$ and $h_U(e_t)$ functions, in the following we construct the learning curves for the new algorithms.

i) For the white regression data for which $\mathbf{R} = \sigma_x^2\mathbf{I}$, the time-evolution of the mean square deviation (MSD) $E[\|\tilde{\mathbf{w}}_t\|^2]$ is given by

$$E\left[\|\tilde{\mathbf{w}}_{t+1}\|^2\right] = \left(1-\mu 2\sigma_x^2 h_G(e_t)\right)E\left[\|\tilde{\mathbf{w}}_t\|^2\right]+\mu^2 p\sigma_x^2 h_U(e_t),$$

where the right hand side only depends on $E[\|\tilde{\mathbf{w}}_t\|^2]$.

ii) For the correlated regression data, by the Cayley-Hamilton theorem, [26] uses the state-space recursion as

$$\mathcal{W}_{t+1} = \mathcal{A}\mathcal{W}_t + \mu^2\mathcal{Y}$$

where the vectors are defined as

$$\mathcal{W}_t \triangleq \begin{bmatrix} E\left[\|\tilde{\mathbf{w}}_t\|^2\right] \\ \vdots \\ E\left[\|\tilde{\mathbf{w}}_t\|^2_{\mathbf{R}^{p-1}}\right]\end{bmatrix}, \mathcal{Y} \triangleq h_U(e_t)\begin{bmatrix} E\left[\|\mathbf{x}_t\|^2\right] \\ \vdots \\ E\left[\|\mathbf{x}_t\|^2_{\mathbf{R}^{p-1}}\right]\end{bmatrix}.$$

The coefficient matrix $\mathcal{A}$ is given by

$$\mathcal{A} \triangleq \begin{bmatrix} 1 & -2\mu h_G(e_t) & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 2\mu c_0 h_G(e_t) & 2\mu c_1 h_G(e_t) & \cdots & 1+2\mu c_{p-1}h_G(e_t)\end{bmatrix}.$$

where the c_i's for $i \in \{0,1,\ldots,p-1\}$ are the coefficients of the characteristic polynomial of $\mathbf{R}$. Note that the top entry of the state vector $\mathcal{W}_t$ yields the time-evolution of the mean square deviation $E[\|\tilde{\mathbf{w}}_t\|^2]$ and the second entry gives the learning curves for the excess mean square error $E[e_{a,t}^2]$. Hence, despite $h_G(e_t)$ and $h_U(e_t)$ are functions of σ_e^2, we can calculate σ_e^2 at each iteration through $\sigma_e^2 = \sigma_{e_a}^2 + \sigma_n^2$.

In the following subsection, we analyze the steady-state excess mean square error (EMSE) and MSD of the LMLS and LLAD algorithms.

B. Steady-State Analysis

In [26], the authors demonstrate that the steady-state EMSE, i.e., $\zeta \triangleq E[e_{a,t}^2]$, of the adaptive filter with an error nonlinearity function is given by

$$\zeta = \frac{\mu}{2}\mathrm{Tr}(\mathbf{R})\frac{h_U(e_t)}{h_G(e_t)}. \tag{18}$$

In addition, by the Assumption 1, the steady-state MSD, i.e., $\eta \triangleq E[\|\tilde{\mathbf{w}}_t\|^2]$, yields

$$\eta = \frac{p}{\mathrm{Tr}(\mathbf{R})}\zeta,$$

where p denotes the filter length.

At the steady-state, we additionally use the following assumptions that yield intuitive and neat theoretical results.

Assumption 4: For the LMLS algorithm and for sufficiently small μ, the limits as $t \to \infty$ of functions $h_G(e_t)$ and $h_U(e_t)$ are given by

$$h_G(e_t) = \frac{1}{\sigma_e^2}E\left[\frac{\alpha e_t^4}{1+\alpha e_t^2}\right] \to \frac{\alpha}{\sigma_e^2}E\left[e_t^4\right],$$
$$h_U(e_t) = E\left[\frac{\alpha^2 e_t^6}{(1+\alpha e_t^2)^2}\right] \to \alpha^2 E\left[e_t^6\right].$$

Assumption 5: For the LLAD algorithm and for sufficiently small μ, the limits as $t \to \infty$ of functions $h_G(e_t)$ and $h_U(e_t)$ are given by

$$h_G(e_t) = \frac{1}{\sigma_e^2}E\left[\frac{\alpha e_t^2}{1+\alpha |e_t|}\right] \to \frac{\alpha}{\sigma_e^2}E\left[e_t^2\right],$$
$$h_U(e_t) = E\left[\frac{\alpha^2 e_t^2}{(1+\alpha |e_t|)^2}\right] \to \alpha^2 E\left[e_t^2\right].$$

For the LMLS algorithm, by the Assumption 4, (18) leads to

$$\zeta_{\mathrm{LMLS}} = \frac{\mu}{2}\alpha\mathrm{Tr}(\mathbf{R})\sigma_e^2\frac{E\left[e_t^6\right]}{E\left[e_t^4\right]}. \tag{19}$$

By Assumption 1 and 2, e_t is a Gaussian random variable and $\sigma_e^2 = \zeta + \sigma_n^2$, we have

$$\begin{aligned}\zeta_{\mathrm{LMLS}} &= \frac{\mu}{2}\alpha\mathrm{Tr}(\mathbf{R})\sigma_e^2\frac{15\sigma_e^6}{3\sigma_e^4},\\ &= \frac{5\mu}{2}\alpha\mathrm{Tr}(\mathbf{R})\left(\zeta_{\mathrm{LMLS}}+\sigma_n^2\right)^2.\end{aligned}$$

Hence, after some algebra, the EMSE and MSD for the LMLS algorithm are given by

$$\zeta_{\mathrm{LMLS}} = \frac{1-5\alpha\mu\mathrm{Tr}(\mathbf{R})\sigma_n^2 \pm \sqrt{1-10\alpha\mu\mathrm{Tr}(\mathbf{R})\sigma_n^2}}{5\alpha\mu\mathrm{Tr}(\mathbf{R})}, \tag{20}$$

where the smaller roots match with the simulations. Note that (20) for $\alpha = 1$ is the same with the EMSE of the LMF algorithm [1], [26].

Remark 4.1: In (20), let $\tilde{\mu} \triangleq \mu\alpha$, then

$$\zeta_{\mathrm{LMLS}} = \frac{1-5\tilde{\mu}\mathrm{Tr}(\mathbf{R})\sigma_n^2 \pm \sqrt{1-10\tilde{\mu}\mathrm{Tr}(\mathbf{R})\sigma_n^2}}{5\tilde{\mu}\mathrm{Tr}(\mathbf{R})}. \tag{21}$$

By (21), we could achieve similar steady-state convergence performance for different α by changing the step size μ, e.g., $\tilde{\mu} = \mu\alpha = \frac{\mu}{10}10\alpha$, however, note that larger α decreases the weight of the normalization term, i.e., the logarithm function, in the proposed cost definition (1).

Correspondingly, for the LLAD algorithm, by the Assumption 5, (18) yields

$$\begin{aligned}\zeta_{\mathrm{LLAD}} &= \frac{\mu}{2}\mathrm{Tr}(\mathbf{R})\sigma_e^2\alpha\frac{E[e_t^2]}{E[e_t^2]},\\ &= \frac{\mu\alpha}{2}\mathrm{Tr}(\mathbf{R})\sigma_e^2.\end{aligned}$$

Hence, by the Assumptions 1 and 2, the EMSE for the LLAD algorithm is given by

$$\zeta_{\mathrm{LLAD}} = \frac{\mu\alpha\mathrm{Tr}(\mathbf{R})\sigma_n^2}{2-\mu\alpha\mathrm{Tr}(\mathbf{R})}. \tag{22}$$

Note that (22) for $\alpha = 1$ is the same with the EMSE of the LMS algorithm [26]. Hence, for sufficiently small μ, the LLAD algorithm achieves analogous steady-state convergence performance with the LMS algorithm under the zero-mean Gaussian error signal assumption.

In Fig. 3, we plot the theoretical and simulated MSD vs. step size for the LMLS and LLAD algorithms. In the system identification framework, we choose the regressor and noise signals as i.i.d. zero mean Gaussian with the auto-covariance matrix $\mathbf{C}_{\mathbf{x}} = \mathbf{I}_5$ and $\sigma_n^2 = 0.01$, respectively. The parameter of interest $\mathbf{w}_o \in \mathbb{R}^5$ is randomly chosen. We observe that the theoretical steady-state MSD matches with the simulation results generated through the ensemble average of the last 10^3 iterations of 10^5 (for the LMLS algorithm) and 10^4 (for the LLAD algorithm) iterations of 200 independent trials. In. Fig. 4, under the same configurations, we compare the simulated MSD curves generated through the ensemble average of 200 independent trials with the theoretical results for the step-size $\mu = 0.1$. We note that theoretical performance analyses accurately match the ensemble averaged results.

C. Tracking Performance

In this subsection, we investigate the tracking performance of the introduced algorithms in a non-stationary environment. We assume a random walk model [1] for $\mathbf{w}_{o,t}$ such that

$$\mathbf{w}_{o,t+1} = \mathbf{w}_{o,t} + \mathbf{q}_t, \tag{23}$$

where $\mathbf{q}_t \in \mathbb{R}^p$ is a zero-mean vector process with covariance matrix $E[\mathbf{q}_t\mathbf{q}_t^T] = \mathbf{Q}$. In this model, we assume that $\mathbf{q}_t$ is independent from the regression and noise signals. We note that

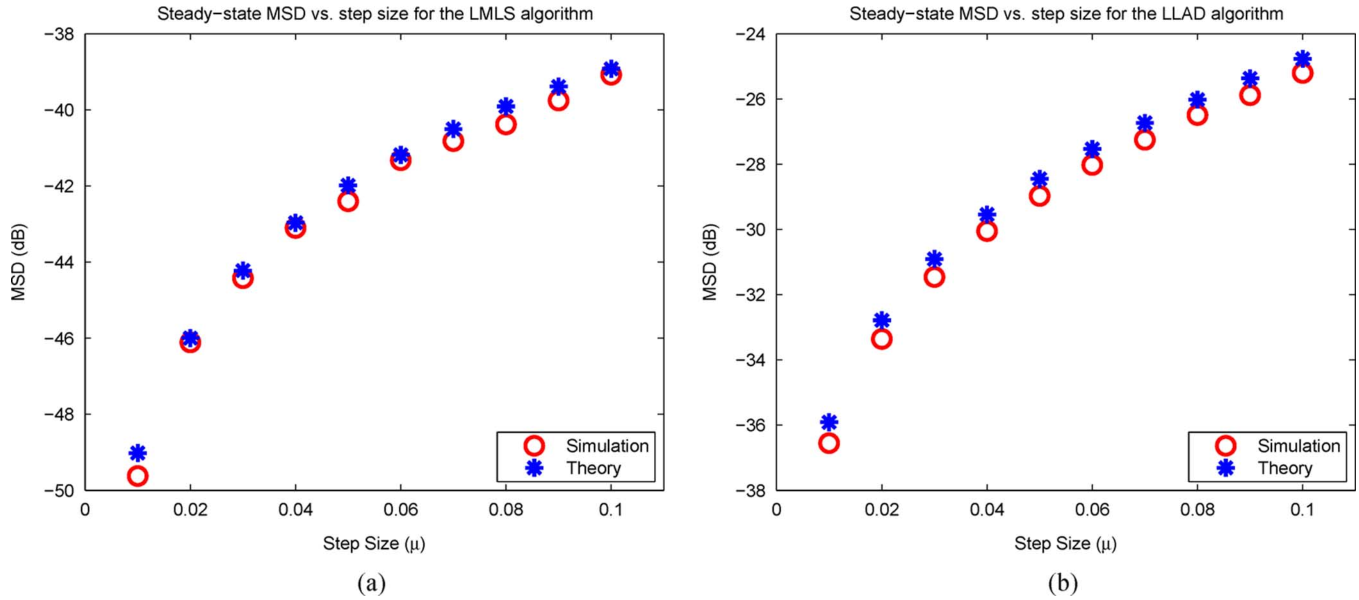


Fig. 3. Dependence of the steady-state MSD on the step size μ for the LMLS and LLAD algorithms. (a) The LMLS Algorithm, (b) The LLAD Algorithm.

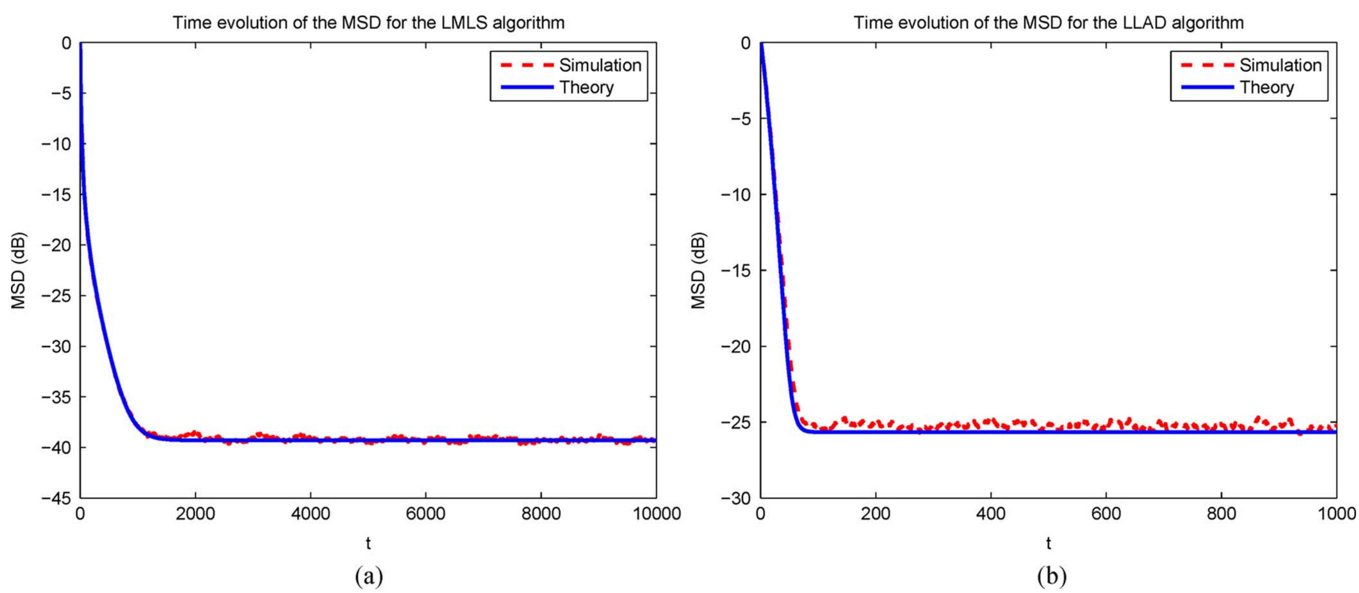


Fig. 4. Theoretical and simulated MSD for the LMLS anf LLAD algorithms, (a) The LMLS algorithm, (b) The LLAD algorithm.

the model (23) has not changed the definitions of *a priori* error. Hence, the weighted energy recursion is given by

$$E\left[\|\tilde{\mathbf{w}}_{t+1}\|^2\right] = E\left[\|\tilde{\mathbf{w}}_t\|^2\right] - 2\mu h_G(e_t) E\left[\|\tilde{\mathbf{w}}_t\|^2_{\mathbf{R}}\right] + \mu^2 E\left[\|\mathbf{x}_t\|^2\right] h_U(e_t) + \mathrm{Tr}(\mathbf{Q}).$$

Correspondingly, at the steady-state, we have

$$2\mu h_G(e_t) E\left[\|\tilde{\mathbf{w}}_t\|^2_{\mathbf{R}}\right] = \mu^2 E\left[\|\mathbf{x}_t\|^2\right] h_U(e_t) + \mathrm{Tr}(\mathbf{Q}).$$

Hence, assuming the adaptive filter is mean square stable, by the Assumptions 1–3 and following derivations in [1], at the steady-state, we get

$$\zeta' = \frac{\mu E\left[\|\mathbf{x}_t\|^2\right] h_U(e_t) + \mu^{-1}\mathrm{Tr}(\mathbf{Q})}{2h_G(e_t)},$$

where ζ' denotes the steady-state EMSE of the algorithm.

Then, we can calculate the tracking performance of the proposed algorithms. Under the Assumption 4, the tracking performance of the LMLS algorithm is roughly given by

$$\zeta'_{\mathrm{LMLS}} \cong \frac{3\mu\alpha\sigma_n^4 \mathrm{Tr}(\mathbf{R}) + \mu^{-1}\alpha^{-1}\mathrm{Tr}(\mathbf{Q})}{6\sigma_n^2}.$$

Correspondingly, through the Assumption 5, we obtain the tracking EMSE of the LLAD as

$$\zeta'_{\mathrm{LLAD}} = \frac{\mu\alpha \mathrm{Tr}(\mathbf{R})\sigma_n^2 + \mu^{-1}\alpha^{-1}\mathrm{Tr}(\mathbf{Q})}{2 - \mu\alpha\mathrm{Tr}(\mathbf{R})}.$$

In the next section, we compare the new algorithms with the conventional LMS and SA in terms of the stability bound and robustness.

V. Comparison With the Conventional Algorithms

We re-emphasize that the cost function $J(e_t)$ intrinsically combines the costs, mainly, $F(e_t)$ and $F^2(e_t)$ based on the relative error amount. Based on the stochastic gradient approach, i.e., removing the expectation in the gradient descent, $F^2(e_t)$ and $F(e_t^2)$ results in the same algorithm. Hence, in this section we compare the stability of the LMLS algorithm with the LMF and LMS algorithms and analyze the robustness of the LLAD algorithm in the impulsive noise environments.

A. Stability Bound for the LMLS Algorithm

We again refer to the stochastic gradient update (5), which we rewrite as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu' \mathbf{x}_t \frac{\partial f(e_t)}{\partial e_t},$$

where $\mu' \triangleq \mu \frac{\alpha f(e_t)}{1+\alpha f(e_t)}$. Note that $\mu' \leq \mu$ irrespective of the design parameter α or the error amount.

Analytically, for stable updates the step size μ should satisfy

$$E\left[\|\tilde{\mathbf{w}}_{t+1}\|^2\right] \leq E\left[\|\tilde{\mathbf{w}}_t\|^2\right].$$

By (9), the Assumptions 1–3, and $\mathbf{\Sigma} = \mathbf{I}$, the stability bound on the step size is given by

$$\mu \leq \frac{2}{E\left[\|\mathbf{x}_t\|^2\right]} \inf_{E[e_{a,t}^2] \in \Omega} \left\{ E[e_{a,t} e_t] \frac{h_G(e_t)}{h_U(e_t)} \right\},$$

where

$$\Omega \triangleq \left\{ E[e_{a,t}^2] : \lambda \leq E[e_{a,t}^2] \leq \frac{1}{4} \mathrm{Tr}(\mathbf{R}) E[\|\tilde{\mathbf{w}}_0\|^2] \right\},$$

with the Cramer-Rao lower bound λ for t observations [34]. As an example the step size bound for the LMLS yields

$$\mu \leq \frac{1}{E\left[\|\mathbf{x}_t\|^2\right]} \inf_{E[e_{a,t}^2] \in \Omega} \left\{ \frac{E[e_{a,t} e_t]}{E\left[e_t^2\right]} \beta \right\},$$

where

$$\beta \triangleq \frac{E\left[\frac{\alpha e_t^4}{1+\alpha e_t^2}\right]}{E\left[\frac{\alpha^2 e_t^6}{\left(1+\alpha e_t^2\right)^2}\right]} = \frac{E\left[\frac{\alpha e_t^4}{\left(1+\alpha e_t^2\right)^2}\right] + E\left[\frac{\alpha^2 e_t^6}{\left(1+\alpha e_t^2\right)^2}\right]}{E\left[\frac{\alpha^2 e_t^6}{\left(1+\alpha e_t^2\right)^2}\right]} \geq 1.$$

Note that the LMS algorithm has a similar bound where $\beta = 1$. Hence, we point out that the LMLS achieves comparable stability performance with the LMS algorithm while performing analogous performance with the LMF algorithm, which has several stability issues [3]–[5].

B. Robustness Analysis for the LLAD Algorithm

Although the performance analysis of the adaptive filters assumes white Gaussian noise signals, in practical applications the impulsive noise is a common problem [9]. In order to analyze the performance in the impulsive noise environments, we use the following model.

Impulsive Noise Model: We model the noise as a summation of two independent random terms [35], [36] as

$$n_t = n_{o,t} + b_t n_{i,t},$$

where $n_{o,t}$ is the ordinary noise signal that is zero-mean Gaussian with variance $\sigma_{n_o}^2$ and $n_{i,t}$ is the impulse-noise that is also zero-mean Gaussian with significantly large variance $\sigma_{n_i}^2$. Here, b_t is generated through a Bernoulli random process and determines the occurrence of the impulses in the noise signal with $p_B(b_t = 1) = \nu_i$ and $p_B(b_t = 0) = 1 - \nu_i$ where ν_i is the frequency of the impulses in the noise signal. The corresponding probability density function is given by

$$p_n(n_t) = \frac{1-\nu_i}{\sqrt{2\pi}\sigma_{n_o}} \exp\left(-\frac{n_t^2}{2\sigma_{n_o}^2}\right) + \frac{\nu_i}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{n_t^2}{2\sigma_n^2}\right),$$

where $\sigma_n^2 = \sigma_{n_o}^2 + \sigma_{n_i}^2$.

We particularly analyze the steady-state performance of the LLAD algorithm (for which $f(e_t) = |e_t|$) in the impulsive noise environments since we motivate the LLAD algorithm as improving the steady-state convergence performance of the SA. At the steady-state, for $\mathbf{\Sigma} = \mathbf{I}$, (9) yields

$$E\left[\|\mathbf{x}_t\|^2\right] = \frac{2E\left[\frac{\alpha e_{a,t} e_t}{1+\alpha|e_t|}\right]}{\mu E\left[\frac{\alpha^2 e_t^2}{(1+\alpha|e_t|)^2}\right]}. \tag{24}$$

Since the noise is not a Gaussian random variable in the impulsive noise environment, the estimation error e_t is not a Gaussian process and the Price's Theorem is not applicable. However, we can use the Price's Theorem conditioning the expectations on the noise n_t as in [37]. Instead of this, we assume that in the impulse-free environment, $\frac{\alpha e_{a,t} e_t}{1+\alpha|e_t|} \approx \alpha e_{a,t} e_t$ since at the steady-state, the error is assumed to take relatively small values whereas if the impulse-noise occurs, $\frac{\alpha e_{a,t} e_t}{1+\alpha|e_t|} \approx e_{a,t} \mathrm{sign}(e_t)$ due to the large perturbation on the error. We now evaluate each term in (24) separately. We first consider the numerator of the RHS of (24) and write

$$\begin{aligned} E\left[\frac{\alpha e_{a,t} e_t}{1+\alpha|e_t|}\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\alpha e_{a,t}(e_{a,t}+n_t)}{1+\alpha|e_{a,t}+n_t|} \frac{\exp\left(-\frac{e_{a,t}^2}{2\sigma_{e_a}^2}\right)}{\sqrt{2\pi}\sigma_{e_a}} p_n(n_t) de_{a,t} dn_t \\ &= \alpha \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e_{a,t} e_t \frac{\exp\left(-\frac{e_{a,t}^2}{2\sigma_{e_a}^2} - \frac{n_t^2}{2\sigma_{n_o}^2}\right)}{2\pi\sigma_{e_a}\sigma_{n_o}} (1-\nu_i) de_{a,t} dn_t \\ &\quad + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e_{a,t} \mathrm{sign}(e_{a,t}+n_t) \\ &\quad \times \frac{\exp\left(-\frac{e_{a,t}^2}{2\sigma_{e_a}^2} - \frac{n_t^2}{2\sigma_n^2}\right)}{2\pi\sigma_{e_a}\sigma_n} \nu_i de_{a,t} dn_t, \end{aligned}$$

Hence, since $\sigma_n^2 \gg \sigma_{e_a}^2$, the expectation leads to

$$E\left[\frac{\alpha e_{a,t} e_t}{1+\alpha|e_t|}\right] = \alpha(1-\nu_i)\sigma_{e_a}^2 + \sqrt{\frac{2}{\pi}} \nu_i \frac{\sigma_{e_a}^2}{\sigma_n}. \tag{25}$$

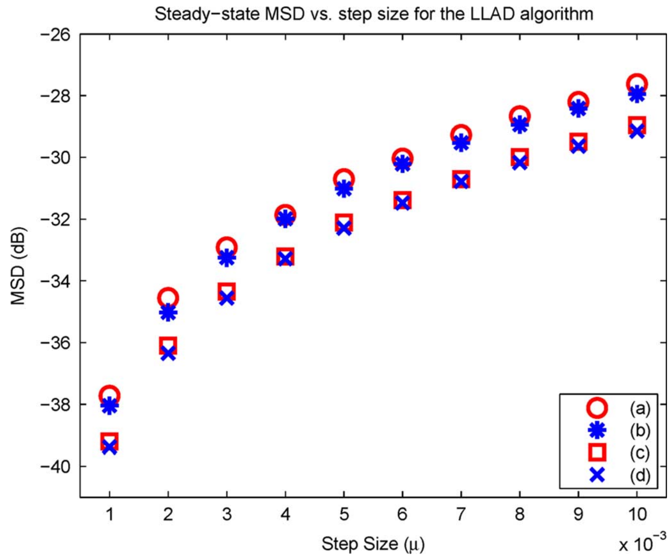


Fig. 5. Dependence of the steady-state MSD on the step size μ for the LLAD algorithm in the 5% impulsive noise environment where (a), (b), (c), and (d) denote $\alpha = 1$ (simulation), $\alpha = 1$ (theory), $\alpha_{\rm opt} = 2.2942$ (simulation), and $\alpha_{\rm opt} = 2.2942$ (theory) cases, respectively.

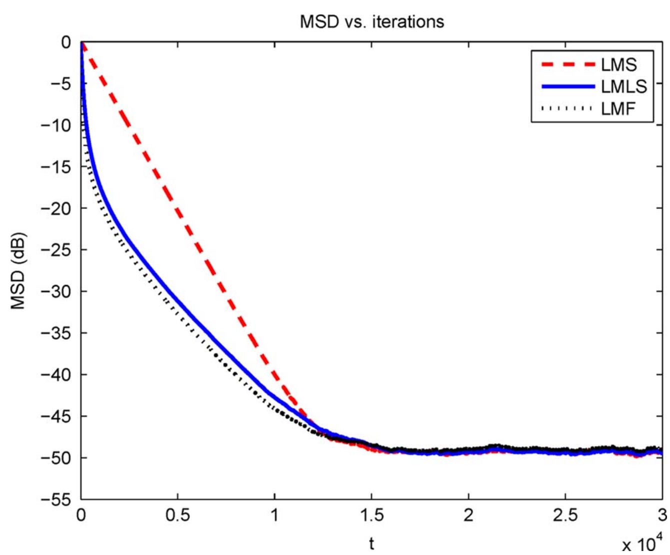


Fig. 6. Comparison of the MSD of the LMLS, LMS and LMF algorithms for the same steady-state MSD where $\mu_{\rm LMLS} = \mu_{\rm LMF} = 0.01$ and $\mu_{\rm LMS} = 0.00047$.

Following similar steps for the denominator of the RHS of (24), we obtain

$$E\left[\frac{\alpha^2 e_t^2}{(1+\alpha|e_t|)^2}\right] = \alpha^2(1-\nu_i)\left(\sigma_{e_a}^2 + \sigma_{n_o}^2\right) + \nu_i. \tag{26}$$

By (24), (25) and (26), the EMSE of the LLAD algorithm in the impulsive noise environment is given by

$$\zeta^*_{\rm LLAD} = \frac{\mu {\rm Tr}(\mathbf{R})\left(\nu_i + \alpha^2(1-\nu_i)\sigma_{n_o}^2\right)}{\alpha(1-\nu_i)(2-\alpha\mu{\rm Tr}(\mathbf{R})) + \sqrt{\frac{8}{\pi}\frac{\nu_i}{\sigma_n}}}. \tag{27}$$

Note that for $\nu_i = 0$ (impulse-free) (27) yields (22).

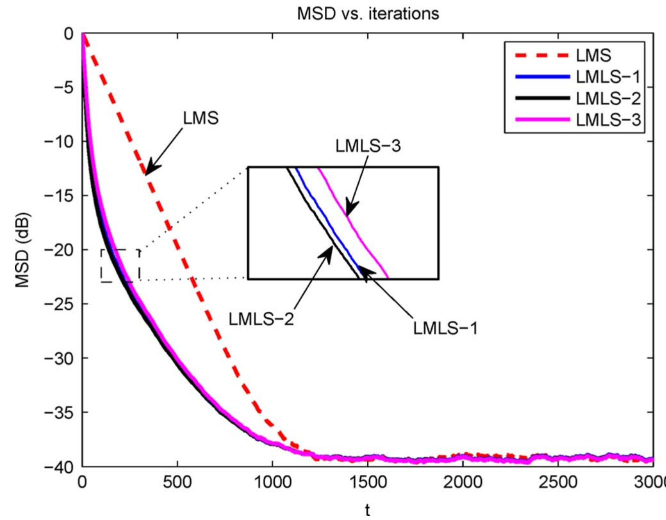


Fig. 7. Comparison of the MSD of the LMLS algorithms with different α's and the LMS algorithm for the same steady-state MSD where $\mu_{\rm LMLS} = 0.1$ and $\mu_{\rm LMS} = 0.0047$. LMLS-1, LMLS-2 and LMLS-3 correspond to $\alpha = 1$, $\alpha = 2$ and $\alpha = 0.5$, respectively.

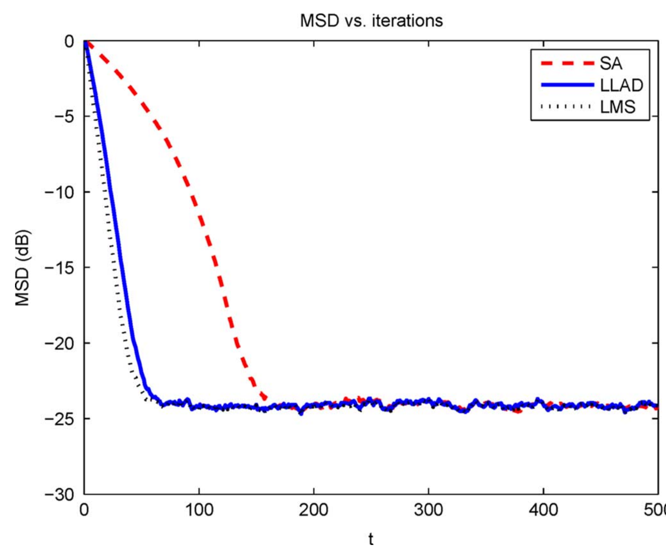


Fig. 8. Comparison of the MSD of the LLAD, SA and LMS algorithms in impulse-free noise environment with $\mu_{\rm LLAD} = 0.12$, $\mu_{\rm SA} = 0.01$ and $\mu_{\rm LMS} = 0.1$.

Remark 5.1: Increasing ν_i, i.e., more frequent impulses, causes larger steady-state EMSE. However, through the optimization of α, we can minimize the steady-state EMSE. After some algebra, the optimum design parameter in the impulsive noise environment is roughly given by

$$\alpha_{\rm opt} \approx \sqrt{\frac{\nu_i}{1-\nu_i}\frac{1}{\sigma_{n_o}}}.$$

In Fig. 5, we plot the dependence of the steady-state MSD with the step size in 5%, i.e., $\nu_i = 0.05$, impulsive noise environment where $\mathbf{C_x} = \mathbf{I}_5$, $\sigma_{n_o}^2 = 0.01$, and $\sigma_{n_i}^2 = 10^4$ after 200 independent trials. We observe that $\alpha_{\rm opt}$ improves the convergence performance and the theoretical analyses through the impulsive noise model matches with the simulation results. We next demonstrate the performance of the introduced algorithms in different applications.

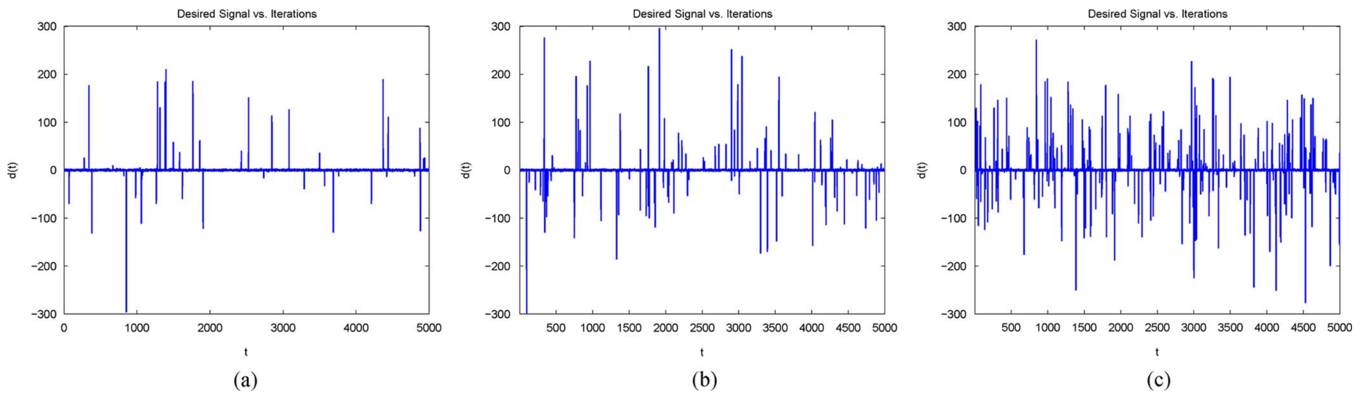


Fig. 9. Desired signal in 1%, 2% and 5% impulsive noise environments. (a) 1%, (b) 2%, (c) 5%.

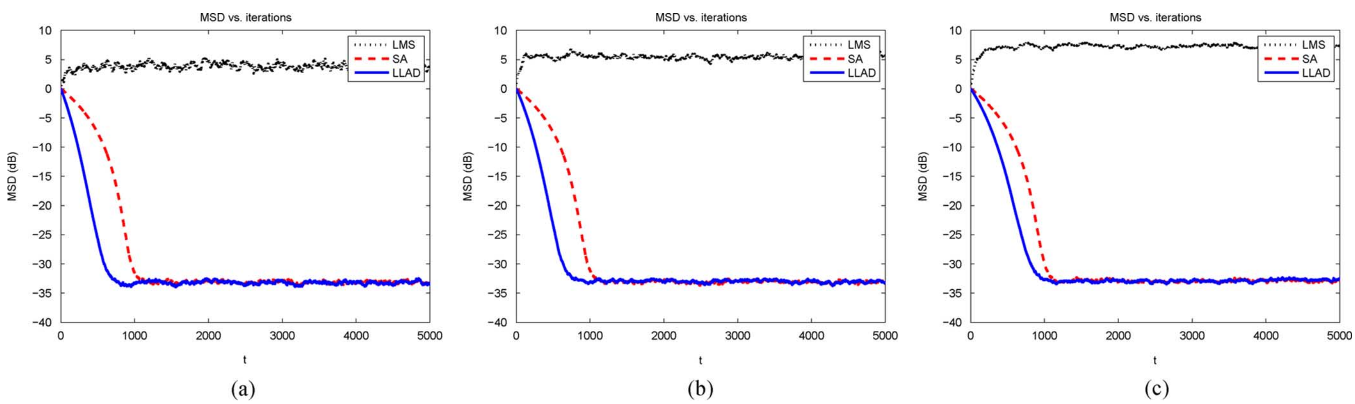


Fig. 10. Comparison of the MSD of the LLAD, SA and LMS algorithms in 1%, 2% and 5% impulsive noise environments. (a) 1% ($\alpha_{\rm opt} = 1.005$), (b) 2% ($\alpha_{\rm opt} = 1.4286$), (c) 5% ($\alpha_{\rm opt} = 2.2942$).

VI. Numerical Examples

In this section, we particularly compare the convergence rate of the algorithms for the same steady-state MSD through the specific choice of the step sizes for fair comparisons. Here, we have a stationary data $d_t = \mathbf{w}_o^T \mathbf{x}_t + n_t$ where $\mathbf{x}_t \in \mathbb{R}^5$ is zero-mean Gaussian i.i.d. regression signal with an auto-covariance matrix $\mathbf{C}_\mathbf{x} = \mathbf{I}_5$, n_t represents zero-mean i.i.d. noise signal and the parameter of interest $\mathbf{w}_o \in \mathbb{R}^5$ is randomly chosen. In following scenarios, we compare the algorithms under Gaussian noise and impulsive noise models subsequently.

Scenario 1 (Impulse-Free Environment): In this scenario, we use a zero-mean Gaussian i.i.d. noise signal with the variance $\sigma_n^2 = 0.01$ and the design parameter $\alpha = 1$. In Fig. 6, we compare the convergence rate of the LMLS, LMF and LMS algorithms for relatively small step sizes. We observe that LMLS and LMF algorithms achieve comparable performance and LMLS achieves better convergence performance than the LMS algorithm. In Fig. 7, we compare the LMLS and LMS algorithms for relatively large step sizes, i.e., $\mu_{\rm LMLS} = 0.1$ and $\mu_{\rm LMS} = 0.0047$. We only compare the LMLS and LMS algorithms since the LMF algorithm is not stable for such a step-size. Hence, the LMLS algorithm demonstrates comparable convergence performance as the LMF algorithm with improved stability performance. In addition, the Fig. 7 also demonstrates that α does not have significant impact on the convergence performance for sufficiently small μ.

In Fig. 8, we compare the LLAD, SA and LMS algorithms in an impulse-free noise environment. We observe that the LLAD algorithm shows comparable convergence performance with the LMS algorithm, particularly, the logarithmic error cost framework improves the convergence performance of the SA.

Scenario 2 (Impulsive Noise Environment): Here, we use the impulsive noise model with $\sigma_{n_i}^2 = 10^4$. In this configuration, we optimize α so that the LLAD algorithm could achieve smaller steady-state MSD. In Fig. 9, we plot sample desired signals in 1%, 2% and 5% impulsive noise environments and Fig. 10 shows the corresponding time evolution of the MSD of the LLAD, SA and LMS algorithms. The step sizes are chosen as $\mu_{\rm LLAD} = \mu_{\rm LMS} = 0.0097, 0.007, 0.0043$ for 1%, 2% and 5% impulsive noise environments, respectively, and $\mu_{\rm SA} = 0.0015$. The figures show that in the impulsive noise environments, the LMS algorithm does not converge while the LLAD algorithm, which achieves comparable convergence performance with the LMS algorithm in the impulse free environment, performs still better than the SA.

Scenario 3 (Comparison With the Robust Huber Filter): For this example, we have the auto-covariance matrix of the regression signal $\mathbf{C}_\mathbf{x} = 10\,\mathbf{I}_5$ and the variance of the observation noise is $\sigma_n^2 = 0.1$. In Fig. 11, we set the step sizes of the algorithms as $\mu_{\rm SA} = \mu_{\rm Huber} = \mu_{\rm LLAD} = 0.03$, and $\mu_{\rm LMS} = 0.01$. We observe that the LLAD algorithm outperforms the robust Huber filter since the Huber filter in general updates similar to the SA while the LLAD algorithm takes more steeper steps similar to

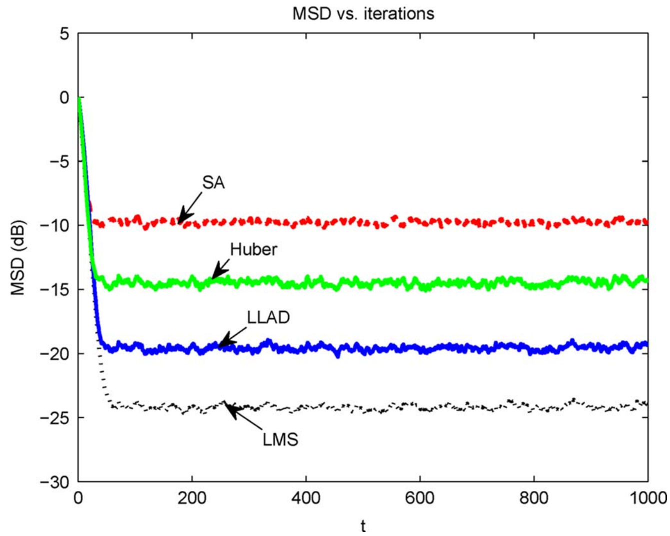


Fig. 11. Comparison of the time evolution of the MSD of the SA, robust Huber, LLAD and LMS algorithms in the impulse-free noise environment.

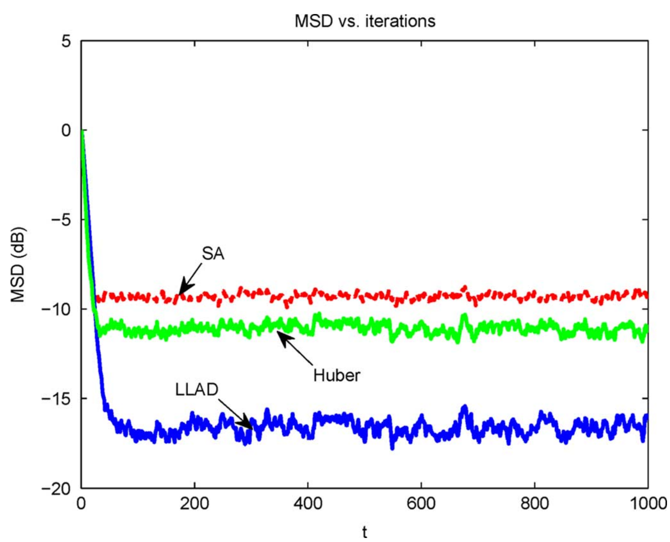


Fig. 12. Comparison of the time evolution of the MSD of the SA, robust Huber and LLAD algorithms in the 5% impulsive noise environment.

the LMS algorithm thanks to the smooth transition. In the impulsive noise environment, we can optimize the breaking point of the robust Huber filter and we found that $\gamma_{\rm opt} = \alpha_{\rm opt}^{-1}$. The Fig. 12 compares the learning curves of the algorithm in the 5% impulsive noise environment where $\alpha_{\rm opt} = 0.72$ and $\gamma_{\rm opt} = 1.37$. The LLAD algorithm outperforms the robust Huber filter also in the impulsive noise environment.

VII. Concluding Remarks

In this paper, we present a novel family of adaptive filtering algorithms based on the logarithmic error cost framework. We mitigate the stability or convergence issues of the well known adaptive algorithms by introducing a relative cost measure. Through the relative logarithmic cost, we intrinsically combine the higher and the lower order measures of the error into a single continuous update based on the error amount. We propose important members of the new family, i.e., the LMLS and LLAD algorithms. The LMLS algorithm achieves comparable convergence performance with the LMF algorithm with significantly improved stability performance. In the impulse-free environment, the LLAD algorithm has analogous convergence performance with the LMS algorithm. Furthermore, the LLAD algorithm is robust against impulsive interferences and outperforms the SA. We also provide comprehensive performance analyses of the introduced algorithms, which match with the simulation results. Finally, we show the improved convergence performance of the new algorithms in several different system identification scenarios.

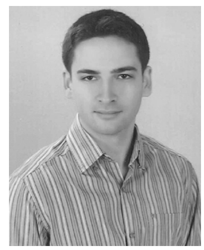
References

[1] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York, NY, USA: Wiley, 2003.
[2] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. Inf. Theory*, vol. 30, no. 2, pp. 275–283, Feb. 1984.
[3] V. H. Nascimento and J. C. M. Bermudez, "When is the least-mean fourth algorithm mean-square stable?," in *Proc. 2005 IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2005, vol. 4, pp. iv/341–iv/344.
[4] V. H. Nascimento and J. C. M. Bermudez, "Probability of divergence for the least-mean fourth algorithm," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1376–1385, Apr. 2006.
[5] P. I. Hubscher, J. C. M. Bermudez, and V. H. Nascimento, "A mean-square stability analysis of the least mean fourth adaptive algorithm," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4018–4028, Aug. 2007.
[6] E. Eweda, "Global stabilization of the least mean fourth algorithm," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1473–1477, Mar. 2012.
[7] E. Eweda and N. J. Bershad, "Stochastic analysis of a stable normalized least mean fourth algorithm for adaptive noise canceling with a white Gaussian reference," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6235–6244, Dec. 2012.
[8] V. H. Nascimento, "A simple model for the effect of normalization on the convergence rate of adaptive filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2004, vol. 2, p. ii-453-6.
[9] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, no. 7, pp. 986–1010, Jul. 1993.
[10] S. R. Kim and A. Efron, "Adaptive robust impulse noise filtering," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 1855–1866, Aug. 1995.
[11] V. J. Mathews and S. H. Cho, "Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 4, pp. 450–454, Apr. 1987.
[12] J. A. Chambers, O. Tanrikulu, and A. G. Constantinides, "Least mean mixed-norm adaptive filtering," *Electron. Lett.*, vol. 30, no. 19, pp. 1574–1575, 1994.
[13] J. Chambers and A. Avlonitis, "A robust mixed-norm adaptive filter algorithm," *IEEE Signal Process. Lett.*, vol. 4, no. 2, pp. 46–48, Feb. 1997.
[14] 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. 2003 IEEE 13th Workshop Neural Netw. Signal Process.*, 2003, pp. 239–248.
[15] J. Arenas-Garcia, V. Gomez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filters," *IEEE Trans. Instrumen. Measur.*, vol. 54, no. 6, pp. 2239–2249, Jun. 2005.
[16] 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.
[17] 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.
[18] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Steady-state MSE performance analysis of mixture approaches to adaptive filtering," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4050–4063, Aug. 2010.

[19] J. Arenas-Garcia and A. R. Figueiras-Vidal, "Adaptive combination of normalised filters for robust system identification," *Electron. Lett.*, vol. 41, no. 15, pp. 874–875, 2005.
[20] P. Petrus, "Robust Huber adaptive filter," *IEEE Trans. Signal Process.*, vol. 47, no. 4, pp. 1129–1133, Apr. 1999.
[21] A. C. Singer, S. S. Kozat, and M. Feder, "Universal linear least squares prediction: Upper and lower bounds," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2354–2362, Aug. 2002.
[22] 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.
[23] S. S. Kozat, A. C. Singer, and G. C. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3730–3745, Jul. 2007.
[24] R. G. Bartle and D. R. Scherbert, *Introduction to Real Analysis*. New York, NY, USA: Wiley, 2011.
[25] T. Y. Al-Naffouri and A. H. Sayed, "Adaptive filters with error nonlinearities: Mean-square analysis and optimum design," *EURASIP J. Appl. Signal Process.*, no. 4, pp. 192–205, 2001.
[26] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of adaptive filters with error nonlinearities," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 653–663, Mar. 2003.
[27] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2–3, pp. 169–192, 2007.
[28] I. Song, P. Park, and R. W. Newcomb, "A normalized least mean squares algorithm with a step-size scaler against impulsive measurement noise," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 60, no. 7, pp. 442–445, 2013.
[29] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 639–652, Mar., 2003.
[30] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IEEE Trans. Inf. Theory*, vol. 4, no. 2, pp. 69–72, Feb. 1958.
[31] E. McMahon, "An extension of Price's theorem (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 2, pp. 168–168, Feb. 1964.
[32] T. Koh and E. J. Powers, "Efficient methods of estimate correlation functions of Gaussian processes and their performance analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 1032–1035, Aug. 1985.
[33] W. Grobner and N. Hofreiter, *Bestimmte Integrale*. Berlin, Germany: Springer-Verlag, 1966.
[34] H. L. V. Trees, *Detection, Estimation, and Modulation Theory*, ser. ser. Detection, Estimation, and Modulation Theory.. New York, NY, USA: Wiley, 2004, pt. pt. 1, no..
[35] X. Wang and H. V. Poor, "Joint channel estimation and symbol detection in Rayleigh flat-fading channels with impulsive noise," *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 19–21, Jan. 1997.
[36] S. C. Chan and Y. X. Zou, "A recursive least m-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 975–991, Apr. 2004.
[37] Y. R. Zheng and V. H. Nascimento, "Two variable step-size adaptive algorithms for non-Gaussian interference environment using fractionally lower-order moment minimization," *Digital Signal Process.*, vol. 23, pp. 831–844, 2013.

Muhammed O. Sayin was born in Erzincan, Turkey, in 1990. He received the B.S. degree with high honors in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2013.

He is currently working toward the M.S. degree in the Department of Electrical and Electronics Engineering at Bilkent University. His research interests include distributed signal processing, adaptive filtering theory, machine learning, and statistical signal processing.



N. Denizcan Vanli was born in Nigde, Turkey, in 1990. He received the B.S. degree with high honors in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2013.

He is currently working toward the M.S. degree in the Department of Electrical and Electronics Engineering at Bilkent University. His research interests include sequential learning, adaptive filtering, machine learning, and statistical signal processing.

Suleyman Serdar Kozat (A'10–M'11–SM'11) received the B.S. degree with full scholarship and high honors from Bilkent University, Turkey. He received the M.S. and Ph.D. degrees in electrical and computer engineering from University of Illinois at Urbana Champaign, Urbana, IL. Dr. Kozat is a graduate of Ankara Fen Lisesi.

After graduation, Dr. Kozat joined IBM Research, T. J. Watson Research Lab, Yorktown, New York, US as a Research Staff Member in the Pervasive Speech Technologies Group. While doing his Ph.D., he was also working as a Research Associate at Microsoft Research, Redmond, Washington, US in the Cryptography and Anti-Piracy Group. He holds several patent inventions due to his research accomplishments at IBM Research and Microsoft Research. After serving as an Assistant Professor at Koc University, Dr. Kozat is currently an Assistant Professor (with the Associate Professor degree) at the electrical and electronics department of Bilkent University.

Dr. Kozat is a Senior Member of the IEEE and the President of the IEEE Signal Processing Society, Turkey Chapter. He has been elected to the IEEE Signal Processing Theory and Methods Technical Committee and IEEE Machine Learning for Signal Processing Technical Committee, 2013. He has been awarded IBM Faculty Award by IBM Research in 2011, Outstanding Faculty Award by Koc University in 2011 (granted the first time in 16 years), Outstanding Young Researcher Award by the Turkish National Academy of Sciences in 2010, ODTU Prof. Dr. Mustafa N. Parlar Research Encouragement Award in 2011, Outstanding Faculty Award by Bilim Kahramanlari, 2013 and holds Career Award by the Scientific Research Council of Turkey, 2009.