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

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

***Abstract*—We introduce a novel family of adaptive filtering algorithms based on a relative logarithmic cost. The new family intrinsically combines the higher and lower order measures of the error into a single continuous update 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 that improve the convergence performance of the conventional 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 extends the stability bound on the step size. 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 analyzes and simulation results. We show the extended stability bound 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.**

EDICS Category: MLR-LEAR, ASP-ANAL, MLR-APPL

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]. In the literature, 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 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]. On the other hand, 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 [7]. However, note that the normalized least mean fourth (NLMF) algorithm does not solve the stability issues [6]. In [6], authors propose 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 heavy tailed impulsive interferences, e.g., in applications involving high power noise signals [8]. 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 [9]. Note that, usually, the algorithms using lower-order measures of the error in their cost function are 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 [10].

The mixed-norm algorithms minimize a combination of different error norms in order to achieve improved convergence performance [11], [12]. For example, [12] 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 other hand, the mixture of experts algorithms adaptively combine different algorithms and provide improved performance irrespective of the environment statistics [13]–[16]. However, note that such mixture approaches require to operate several different algorithms on parallel, which may be infeasible in different applications [17]. In [18], authors propose an adaptive combination of L_1 and L_2 norms of the error in parallel, however, the resulting algorithm demonstrates impulsive perturbations on the learning curves. This results since the impulsive interferences severely degrade the algorithmic updates. In general, the samples contaminated with impulses contain little useful information [9]. Hence, the robust algorithms need to be less sensitive only against

This work is in part supported by the Outstanding Researcher Programme of Turkish Academy of Sciences and TUBITAK project 112E161.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara 06800 Turkey, Tel: +90 (312) 290-2336, Fax: +90 (312) 290-1223 (e-mail: sayin@ee.bilkent.edu.tr, vanli@ee.bilkent.edu.tr, kozat@ee.bilkent.edu.tr).

[1]Time index appears as a subscript.

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 [19]. 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 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 in order to improve the convergence performances. We particularly choose the logarithm function as the normalizing diminishing return function [20] in our cost definitions since the logarithmic function is differentiable and results efficient and mathematically tractable adaptive algorithms. As shown in the paper, by using the logarithm function, we are able to use of the higher-order statistics of the error for small perturbations. On the other hand, 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 increasing error amount. In this sense, the new framework is akin to a continuous generalization of the switched norm algorithms, hence greatly improve 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 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, steady-state and 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.

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, steady-state and tracking performances of those members in Section IV. In Section V, we compare the stability bound on the step-sizes and the 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

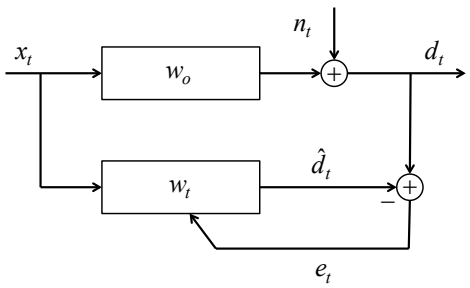


Fig. 1: General system identification configuration.

(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, $\mathrm{Tr}(\mathbf{A})$ 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 and we define the error signal as $e_t \triangleq d_t - \hat{d}_t = d_t - \mathbf{w}_t^T \mathbf{x}_t$. 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], [10] or a linear combination of different error powers [11], [12] are also widely used.

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

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

where $\alpha > 0$ is a design parameter and $F(e_t)$ is a conventional cost function of the error signal e_t, e.g., $F(e_t) = E[|e_t|]$. As an illustration, in Fig. 2, we compare $|e_t|$ and $|e_t| - \ln(1 + |e_t|)$. From this plot, we observe that the logarithm based cost function is less steep for small perturbations on the error while both logarithmic square and absolute difference cost functions exhibit comparable steepness for large error values. Indeed, 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 higher-order statistics for small perturbations of the error.

[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.

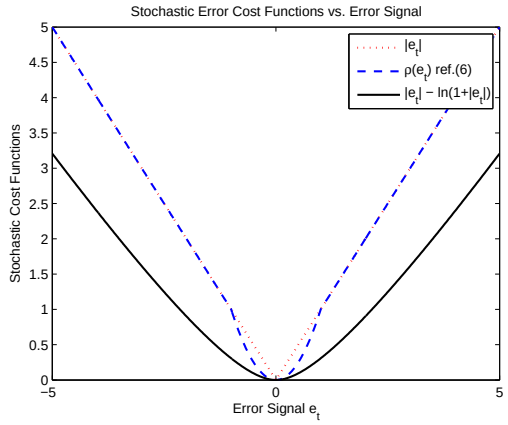


Fig. 2: Here, we plot stochastic cost functions to illustrate decreased steepness of the least squares algorithms in the logarithmic error cost framework for small error amounts.

Remark 2.1: In [21], the authors propose a stochastic cost function using the logarithm function as follows

$$J_{\,[21]}(e_t) \triangleq \frac{1}{2\gamma} \ln \left(1 + \gamma \left(\frac{e_t}{\|\mathbf{x}_t\|} \right)^2 \right).$$

Note that the cost function $J_{\,[21]}(e_t)$ is the subtracted term in (1) for $F(e_t) = \frac{e_t^2}{\|\mathbf{x}_t\|^2}$. The Hessian matrix of $J_{\,[21]}(e_t)$ is given by

$$\mathbf{H}\left(J_{\,[21]}(e_t)\right) = \frac{\mathbf{x}_t \mathbf{x}_t^T}{\|\mathbf{x}_t\|^2 \left(1 + \gamma \left(\frac{e_t}{\|\mathbf{x}_t\|} \right)^2 \right)} \times \left(1 - \frac{2\gamma e_t^2}{\|\mathbf{x}_t\|^2 \left(1 + \gamma \left(\frac{e_t}{\|\mathbf{x}_t\|} \right)^2 \right)} \right).$$

We emphasize that $\mathbf{H}\left(J_{\,[21]}(e_t)\right)$ is positive semi-definite provided that $\gamma \left(\frac{e_t}{\|\mathbf{x}_t\|} \right)^2 \leq 1$, thus, the parameter γ should be chosen carefully to be able to efficiently use the gradient descent algorithms. On the other hand, we show that the new cost function in (1) is a convex function enabling the use of the diminishing return property [20] of the logarithm function for stable and robust updates.

The relative logarithmic error cost we introduce in (1) can also be expressed as

$$J(e_t) = \frac{1}{\alpha} \ln \left(\frac{\exp\left(\alpha F(e_t)\right)}{1 + \alpha F(e_t)} \right). \tag{2}$$

Since $\exp(\alpha F(e_t)) = \sum_{m=0}^{\infty} \frac{1}{m!} \alpha^m F^m(e_t)$, we obtain

$$J(e_t) = \frac{1}{\alpha} \ln \left(1 + \frac{\frac{\alpha^2}{2!} F^2(e_t) + \frac{\alpha^3}{3!} F^3(e_t) + \cdots}{1 + \alpha F(e_t)} \right). \tag{3}$$

Since $F(e_t)$ is a non-negative function, $J(e_t)$ is also a non-negative function by (3).

Remark 2.2: 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 $\mathbf{H}\left(F(e_t)\right)$ is a positive semi-definite matrix.

We obtain the first gradient of (1) as follows

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

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

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

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 as shown in Remark 2.2 the Hessian matrix of the logarithmic cost function is positive semi-definite. For example, the mean-square error cost function $F(e_t) = E[e_t^2]$ yields to the Wiener solution $\mathbf{w}_o = E[\mathbf{x}_t \mathbf{x}_t^T]^{-1} E[\mathbf{x}_t d_t]$.

Remark 2.3: By Maclaurin series of the natural logarithm for $\alpha F(e_t) \leq 1$, (1) yields

$$\begin{aligned} J(e_t) &= F(e_t) - \frac{1}{\alpha} \left(\alpha F(e_t) - \frac{\alpha^2}{2} F^2(e_t) + \cdots \right) \\ &= \frac{\alpha}{2} F^2(e_t) - \frac{\alpha^2}{3} F^3(e_t) + \cdots, \end{aligned} \tag{4}$$

which is an infinite combination of the conventional cost function for small values of $F(e_t)$. We emphasize that the cost function (4) yields to the second power of the cost function $F(e_t)$ for small values of the error while for large error values, the cost function $J(e_t)$ resembles $F(e_t)$ as follows:

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

Hence, the new methods are the combinations of the algorithms with mainly $F^2(e_t)$ or $F(e_t)$ cost functions based on the error amount. It is important to 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. 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 [19]

$$\rho(e_t) \triangleq \begin{cases} \frac{1}{2} e_t^2 & \text{for } |e_t| \leq \gamma, \\ \gamma |e_t| - \frac{1}{2}\gamma^2 & \text{for } |e_t| > \gamma, \end{cases} \tag{5}$$

where $\gamma > 0$ denotes the cut-off value. In Fig. 2, we also compare the Huber objective function (for $\gamma = 1$) and the introduced cost (1) with $F(e_t) = E[|e_t|]$ (for $\alpha = 1$). Note that (5) uses a piecewise-function combining two different algorithms based on the comparison of the error with the cut-off value γ. On the other hand, 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.

Remark 2.4: 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

$$J_{\arctan}(e_t) \triangleq F(e_t) - \frac{1}{\alpha}\arctan\left(\alpha F(e_t)\right) \tag{6}$$

and the Taylor series expansion of the second term in (6) 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 (6) 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. Novel 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.

Remark 3.1: In the previous section, we motivate the logarithm based error cost framework as a continuous generalization of the switched norm algorithms. The switched norm update involves a cut-off γ in the comparison of the error amount. Similarly, we utilize a design parameter α in (1) in order to determine the asymptotic cut-off value. For example, a larger α decreases the weight of the logarithmic term in the cost (1) and the resulting algorithm behaves more like minimizing the cost $F(e_t)$. In the performance analyzes, we show that a sufficiently small design parameter, i.e., $\alpha = 1$, does not have determinative influence on the steady-state convergence performance under the Gaussian noise signal assumption. Hence, in the following algorithms we choose $\alpha = 1$. On the other hand, we resort to the usage of α in order to facilitate the performance analyzes of the algorithms. Additionally, in the impulsive noise environments, we show that the optimization of α improves the steady-state convergence performance of the introduced algorithms.

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

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t - \mu\,\nabla_{\mathbf{w}} e_t \nabla_{e_t} f(e_t)\frac{f(e_t)}{1+f(e_t)},\\ &= \mathbf{w}_t + \mu\,\mathbf{x}_t \nabla_{e_t} f(e_t)\frac{f(e_t)}{1+f(e_t)}.\end{aligned} \tag{7}$$

In the following subsections, we introduce algorithms improving the performance of the conventional algorithms such as the LMS (i.e. $f(e_t) = e_t^2$), sign algorithm (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

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t + \mu \mathbf{x}_t e_t \frac{e_t^2}{1+e_t^2}\\ &= \mathbf{w}_t + \mu \frac{\mathbf{x}_t e_t^3}{1+e_t^2}.\end{aligned} \tag{8}$$

Note that we include the multiplier '2' coming from the gradient $\nabla_{e_t} e_t^2 = 2e_t$ into the step-size μ. The algorithm (8) 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 [11] 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 $F(e_t) = E[|e_t|]$, (7) yields

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t + \mu \mathbf{x}_t \mathrm{sign}(e_t)\frac{|e_t|}{1+|e_t|}\\ &= \mathbf{w}_t + \mu \frac{\mathbf{x}_t e_t}{1+|e_t|}.\end{aligned} \tag{9}$$

The algorithm (9) 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 $F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right) = E\left[\frac{e_t^2}{\|\mathbf{x}_t\|^2}\right]$. The Hessian matrix of the new cost function $J_{\rm new}(e_t)$ is also positive semi-definite provided that the Hessian matrix of $F\left(\frac{e_t}{\|\mathbf{x}_t\|}\right)$ is positive semi-definite as shown in Remark 2.2.

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

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu\frac{\mathbf{x}_t e_t^3}{\|\mathbf{x}_t\|^2\left(\|\mathbf{x}_t\|^2 + e_t^2\right)}. \tag{10}$$

We point out that (10) is also proposed as the stable normalized least mean-fourth algorithm in [6].

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

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

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

IV. Performance Analysis

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. Different choice of $\mathbf{\Sigma}$ leads to the different performance measures of the algorithm [1]. In the analyzes, we include the design parameter α in order to facilitate the theoretical analyzes. After some algebra, we obtain the weighted-energy recursion [1], [22], [23] as

$$\begin{aligned} E\left[\|\tilde{\mathbf{w}}_{t+1}\|_{\mathbf{\Sigma}}^2\right] &= E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}}^2\right] \\ &\quad -\mu 2E\left[e_{a,t}^{\mathbf{\Sigma}}\nabla_{e_t} f(e_t)\frac{\alpha f(e_t)}{1+\alpha f(e_t)}\right] \\ &\quad +\mu^2 E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2\left(\nabla_{e_t} f(e_t)\frac{\alpha f(e_t)}{1+\alpha f(e_t)}\right)^2\right]. \end{aligned} \tag{11}$$

For notational simplicity, we define

$$g(e_t) \triangleq \nabla_{e_t} f(e_t)\frac{\alpha f(e_t)}{1+\alpha f(e_t)}. \tag{12}$$

Then, (11) yields the general weighted-energy recursion [23] as follows

$$\begin{aligned} E\left[\|\tilde{\mathbf{w}}_{t+1}\|_{\mathbf{\Sigma}}^2\right] &= E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}}^2\right] - \mu 2E\left[e_{a,t}^{\mathbf{\Sigma}} g(e_t)\right] \\ &\quad + \mu^2 E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2 g^2(e_t)\right]. \end{aligned} \tag{13}$$

In the subsequent analysis of (11), 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 auto-correlation matrix $\mathbf{R} \triangleq E\left[\mathbf{x}_t\mathbf{x}_t^T\right]$.

Assumption 2:
The estimation error e_t and the noise n_t are jointly Gaussian. The Gaussian estimation error assumption is acceptable for sufficiently small step size μ and through the Assumption 1 [1].

Assumption 3:
The estimation error e_t 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 [23], and by Assumption 2.

Assumption 4:
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 through the energy recursion (11).

A. Transient Analysis

In the following we evaluate (11) term by term. We first consider the second term in the right hand side (RHS) of (13) and introduce the following lemma

Lemma 1: *Under Assumptions 1-4, we have*

$$E[e_{a,t}^{\mathbf{\Sigma}} g(e_t)] = E[e_{a,t}^{\mathbf{\Sigma}} e_t]\frac{E[e_t g(e_t)]}{E[e_t^2]}. \tag{14}$$

Proof: The proof of Lemma 1 follows from the Price's result [24], [25]. 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 [26]. Hence by Assumptions 2 and 3, we obtain (14) and the proof is concluded. □

Since $e_t = e_{a,t} + n_t$, we obtain

$$E\left[e_{a,t}^{\mathbf{\Sigma}} e_t\right] = E\left[e_{a,t}^{\mathbf{\Sigma}} e_{a,t}\right] = E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}\mathbf{x}_t\mathbf{x}_t^T}^2\right], \tag{15}$$

by Assumption 1. Additionally, by the independence assumption for the regressor $\mathbf{x}_t$ (i.e., Assumptions 1 and 4), we can simplify the third term in the RHS of (13). Hence, the weighted-error recursion (13) could be written as follows [23]

$$\begin{aligned} E\left[\|\tilde{\mathbf{w}}_{t+1}\|_{\mathbf{\Sigma}}^2\right] &= E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}}^2\right] - \mu 2h_G(e_t)\,E\left[\|\tilde{\mathbf{w}}_t\|_{\mathbf{\Sigma}\mathbf{R}}^2\right] \\ &\quad + \mu^2 E\left[\|\mathbf{x}_t\|_{\mathbf{\Sigma}}^2\right]h_U(e_t), \end{aligned} \tag{16}$$

where

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

Remark 4.1: In the Appendices we evaluate the functions $h_G(e_t)$ and $h_U(e_t)$ for the LMLS and LLAD algorithms and tabulate the evaluated results with the results for the LMS algorithm, LMF algorithm and SA in Table I.

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

Using (16), 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

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

This completes the transient analysis of the MSD for the white regressor data since $h_U(e_t)$ and $h_G(e_t)$ are given in Table I, and the right hand side only depends on $E[\|\tilde{\mathbf{w}}_t\|^2]$.

ii) For the correlated regression data, by the Cayley-Hamilton theorem after some algebra we get the state-space recursion

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

where the vectors are defined as

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

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 $i \in \{0,1,...,p-1\}$ 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 $E\left[\|\tilde{\mathbf{w}}_t\|^2\right]$ and the second entry gives the learning curves for the excess mean square error $E\left[e_{a,t}^2\right]$.

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

At the steady state, (11) and (15) yields

$$\mu E\left[\|\mathbf{x}_t\|^2_{\mathbf{\Sigma}}\right] h_U(e_t) = 2\, h_G(e_t) E\left[e_{a,t}^{\mathbf{\Sigma}} e_{a,t}\right]. \quad (17)$$

Without loss of generality, we set the weight matrix $\mathbf{\Sigma} = \mathbf{I}$, then (17) leads the steady state EMSE

$$\begin{aligned}\zeta &\triangleq E[e_{a,t}^2] \\ &= \frac{\mu}{2}E\left[\|\mathbf{x}_t\|^2\right]\frac{h_U(e_t)}{h_G(e_t)} \\ &= \frac{\mu}{2}\mathrm{Tr}(\mathbf{R})\frac{h_U(e_t)}{h_G(e_t)}. \end{aligned} \quad (18)$$

By Assumption 1, the steady state MSD is given by [23]

$$\begin{aligned}\eta &\triangleq E\left[\|\tilde{\mathbf{w}}_t\|^2\right] \\ &= \frac{p}{\mathrm{Tr}(\mathbf{R})}\zeta,\end{aligned}$$

where p denotes the filter length.

At the steady state, we additionally use the following assumptions, which directly follow from the property of a learning algorithm that as t goes to infinity, e_t goes to zero.

Assumption 5:

For sufficiently small μ, $h_G(e_t)$ and $h_U(e_t)$ functions of the LMLS algorithm as $t \to \infty$ is given by

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

Assumption 6:

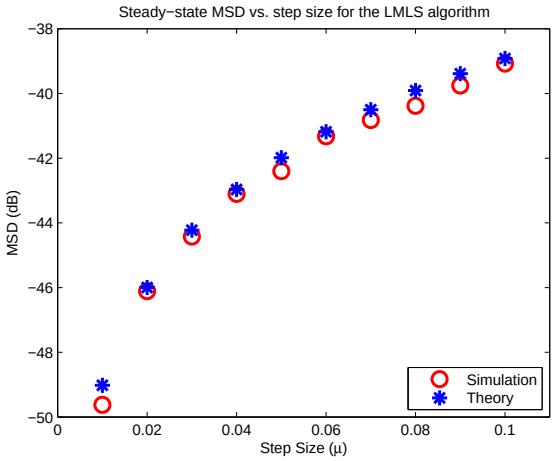
For sufficiently small μ, $h_G(e_t)$ and $h_U(e_t)$ functions of the LLAD algorithm as $t \to \infty$ is given by

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

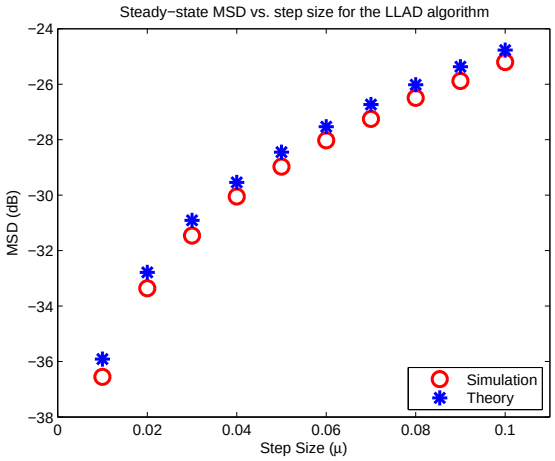
Now, we explicitly derive the steady state analysis of the LMLS and LLAD algorithms, respectively.

The LMLS Algorithm: For the LMLS algorithm, by Assumption 5, (18) leads

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

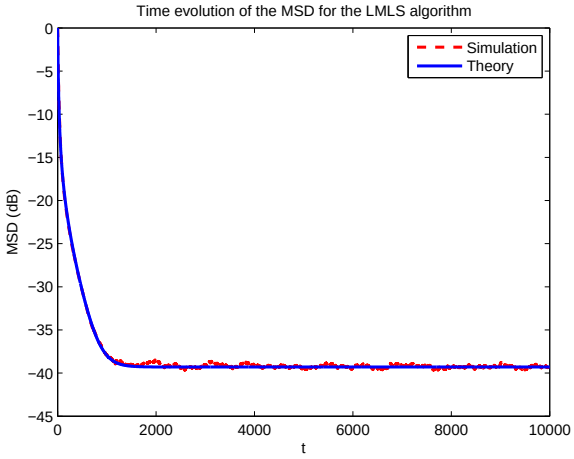


(a) The LMLS Algorithm

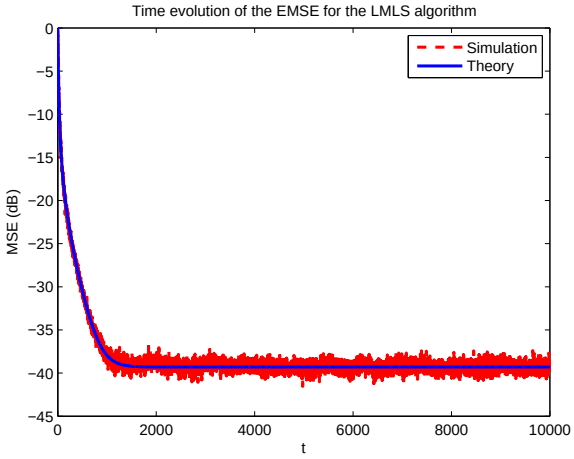


(b) The LLAD Algorithm

Fig. 3: Dependence of the steady-state MSD on the step size μ for the LMLS and LLAD algorithms.

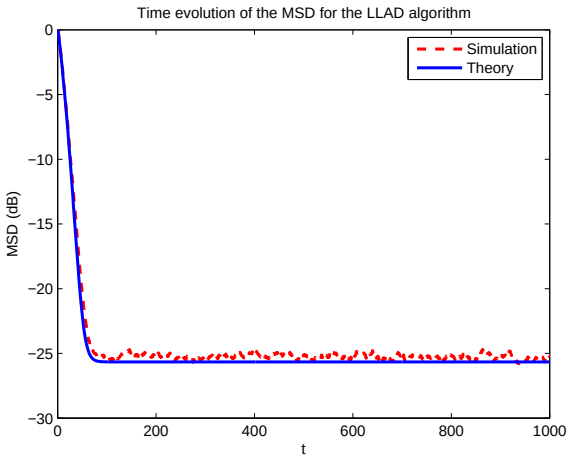


(a) MSD

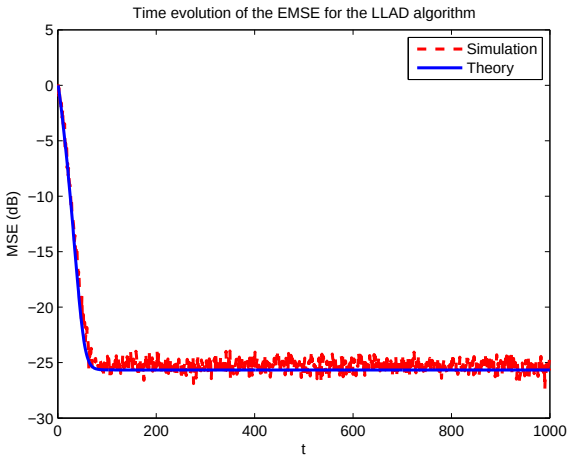


(b) EMSE

Fig. 4: Theoretical and simulated MSD and EMSE for the LMLS algorithm.



(a) MSD



(b) EMSE

Fig. 5: Theoretical and simulated MSD and EMSE for the LLAD algorithm.

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

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

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

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

where the smaller roots match with the simulations. Note that (20) for $\alpha = 1$ is the same with the EMSE of the LMF algorithm [23].

Remark 4.2: In (20), let $\tilde{\mu} \triangleq \mu\alpha$, then

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

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, smaller α results in a slower convergence rate. Hence, without loss of generality, we propose the algorithms with $\alpha = 1$ under the Gaussianity assumption.

The LLAD Algorithm: Similarly, for the LLAD algorithm, by Assumption 6, (18) yields

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

By Assumption 2, the EMSE and MSD for the LLAD algorithm is given by

$$\zeta_{\text{LLAD}} = \frac{\mu\alpha\text{Tr}(\mathbf{R})\sigma_n^2}{2 - \mu\alpha\text{Tr}(\mathbf{R})}. \quad (22)$$
$$\eta_{\text{LLAD}} = \frac{\mu\alpha p\sigma_n^2}{2 - \mu\alpha\text{Tr}(\mathbf{R})}$$

Note that (22) is the same with the EMSE of the LMS algorithm [23]. Hence, for sufficiently small α, the LLAD algorithm achieves similar 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 variances $\sigma_x^2 = 1$ 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 and Fig. 5, under the same configurations, we compare the simulated MSD and EMSE 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 analyzes match with our simulation 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

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

where $\mathbf{q}_t \in \mathbb{R}^p$ is a zero-mean vector process with covariance matrix $E[\mathbf{q}_t\mathbf{q}_t^T] = \mathbf{Q}$. We note that the model (23) has not changed the definitions of *a priori* error. Hence, by the Assumption 5, the tracking EMSE of the LMLS is the same with the tracking EMSE of the LMF and is approximately given by [1]

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

Similarly, through the Assumption 6, we obtain the tracking EMSE of the LLAD as

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

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 since for small perturbations on the error, the updates are mainly using the cost $F^2(e_t)$. Based on our 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 (7), which we rewrite as

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

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. Hence, intuitively we can state that for the introduced algorithms the step-size bound is at least as large as the step-size bound for the corresponding conventional algorithm.

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 (11), the Assumption 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 [27]. For 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

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

We re-emphasize that the LMLS extends the stability bound of the LMS algorithm (the same bound with $\beta = 1$) while performing comparable 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 the white Gaussian noise signals, in practical applications the impulsive noise is a common problem [8]. 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 [28], [29] 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. Since the noise is not a Gaussian random variable in impulsive noise environment, the Gaussianity assumption of the estimation error e_t and the Price's Theorem are not applicable. At the steady-state, for $\mathbf{\Sigma} = \mathbf{I}$, (11) yields

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

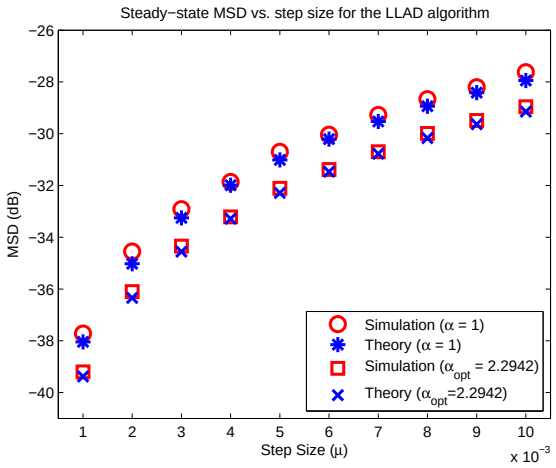


Fig. 6: Dependence of the steady-state MSD on the step size μ for the LLAD algorithm in the 5% impulsive noise environment.

We now evaluate the each term in (24) separately. We first consider the nominator of the RHS of (24), and write

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

where in the last step of the equation 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 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. Hence, since $\sigma_n^2 \gg \sigma_{e_a}^2$, the expectation leads

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

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

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

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

Remark 5.1: Increasing ν_i or in other words more frequent impulses cause 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 impulsive noise environment is roughly given by

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

In Fig. 6, 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 $\sigma_x^2 = 1$, $\sigma_{n_o}^2 = 0.01$ and $\sigma_{n_i}^2 = 10^4$ after 200 independent trials. We observe that α_{opt} improves the convergence performance and the theoretical analyzes through the impulsive noise model matches with the simulation results. We next demonstrate the performance of the introduced algorithms in different applications.

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 a fair comparison. Here, we have a stationary data $d_t = \mathbf{w}_o^T\mathbf{x}_t + n_t$ where $\mathbf{x}_t$ is zero-mean Gaussian i.i.d. regression signal with variance $\sigma_x^2 = 1$, n_t represent 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 that 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. 7, 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. 8, we compare the LMLS and LMS algorithms for relatively large step sizes, i.e., $\mu_{\text{LMLS}} = 0.1$ and $\mu_{\text{LMS}} = 0.0047$. We only compare the LMLS and LMS algorithms since the LMF algorithm is not stable for such a step-size. Hence, the LMLS algorithm demonstrate comparable convergence performance with the LMF algorithm with extended stability bound.

In Fig. 9, we compare the LLAD, SA and LMS algorithms in impulse-free noise environment. We observe that the LLAD algorithm shows comparable convergence performance with the LMS algorithm, in other words, 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 that configuration, we resort to the design parameter since through the optimization of α, the LLAD algorithm could achieve smaller steady-state MSD. In Fig. 10, we plot sample desired signals in 1%, 2% and 5% impulsive noise environments and Fig. 11 shows the corresponding time evolution of the MSD of the LLAD, SA and LMS algorithms. The step sizes are chosen as $\mu_{\text{LLAD}} = \mu_{\text{LMS}} = 0.0097, 0.007, 0.0043$ for 1%, 2% and 5% impulsive noise environments, respectively, and $\mu_{\text{SA}} = 0.0015$. 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.

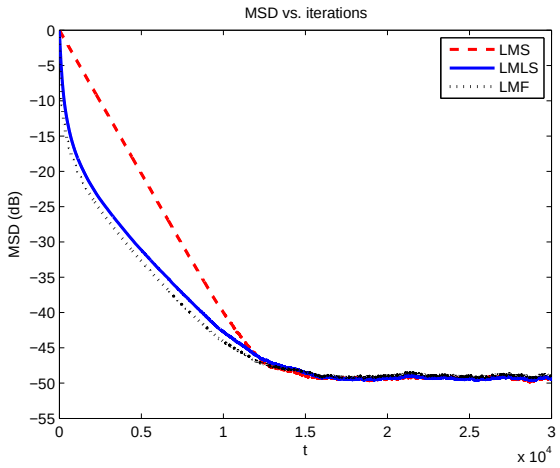


Fig. 7: Comparison of the MSD of the LMLS, LMS and LMF algorithms for the same steady state MSD where $\mu_{\text{LMLS}} = \mu_{\text{LMF}} = 0.01$ and $\mu_{\text{LMS}} = 0.00047$.

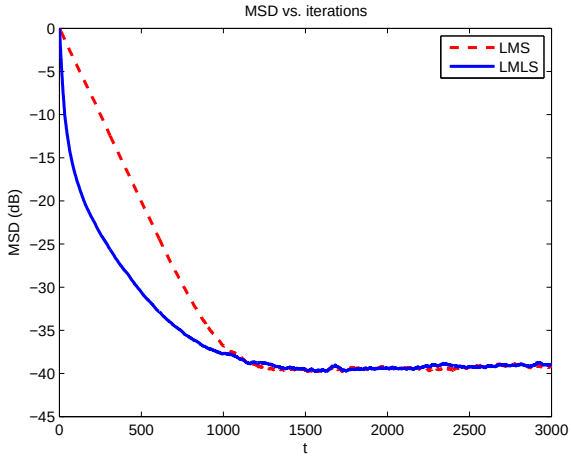
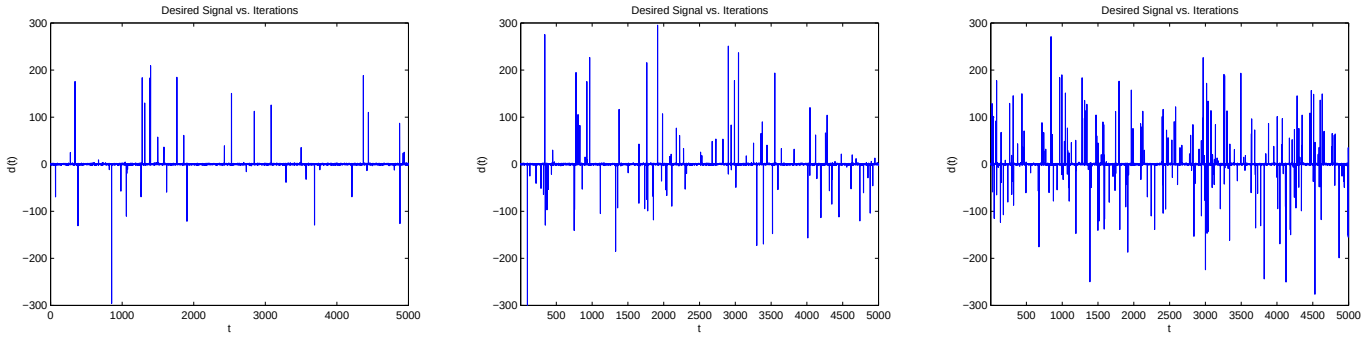


Fig. 8: Comparison of the MSD of the LMLS and LMS algorithms for the same steady state MSD where $\mu_{\text{LMLS}} = 0.1$ and $\mu_{\text{LMS}} = 0.0047$.

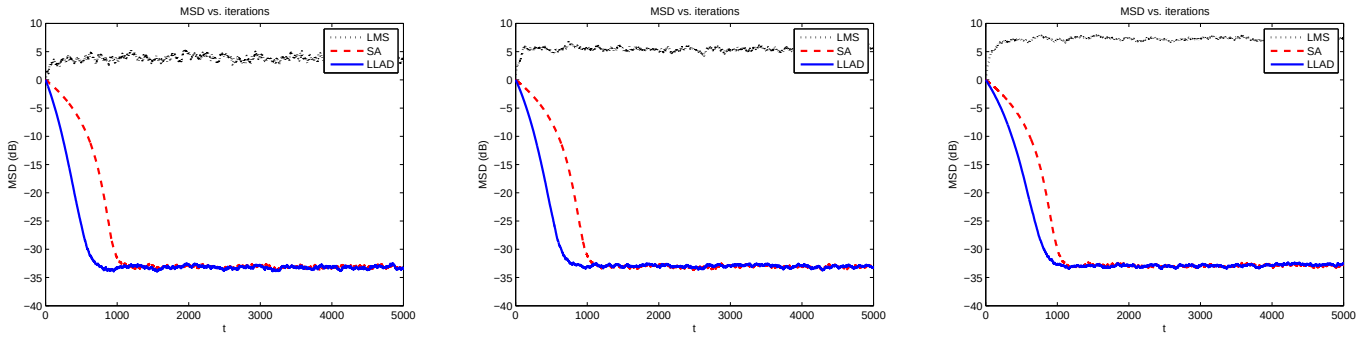
VII. Concluding Remarks

In this paper, we present a novel family of adaptive filtering algorithms based on the logarithmic error cost framework. 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 far larger stability bound on the step size. In the impulse-free environment, the LLAD algorithm has a similar convergence performance with the LMS algorithm. Furthermore, the LLAD algorithm is robust against impulsive interferences and outperforms the SA. We also provide comprehensive performance analyzes of the introduced algorithms, which match with our simulation results. For example, the steady-state analyzes in the impulse-free and impulsive noise environments. Finally, we show the improved convergence performance of the new algorithms in several different system identification scenarios.



(a) 1% (b) 2% (c) 5%

Fig. 10: Desired signal in 1%, 2% and 5% impulsive noise environments.



(a) 1% ($\alpha_{opt} = 1.005$) (b) 2% ($\alpha_{opt} = 1.4286$) (c) 5% ($\alpha_{opt} = 2.2942$)

Fig. 11: Comparison of the MSD of the LLAD, SA and LMS algorithms in 1%, 2% and 5% impulsive noise environments.

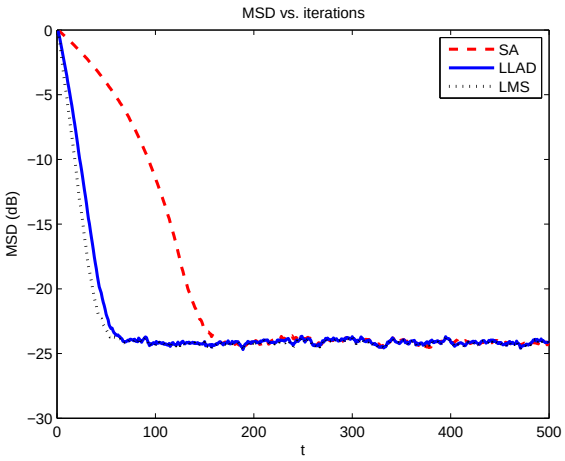


Fig. 9: Comparison of the MSD of the LLAD, SA and LMS algorithms in impulse-free noise environment with $\mu_{\text{LLAD}} = 0.12$, $\mu_{\text{SA}} = 0.01$ and $\mu_{\text{LMS}} = 0.1$.

Appendix A
Evaluation of $h_G(e_t)$

The LMLS algorithm: We have

$$h_G(e_t) = \frac{1}{\sigma_e^2} E\left[\frac{\alpha e_t^4}{1+\alpha e_t^2}\right], \\ = \frac{1}{\sigma_e^2}\left(\sigma_e^2 - \alpha^{-1} + \alpha^{-1} E\left[\frac{1}{1+\alpha e_t^2}\right]\right), \quad (28)$$

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 (12). According to Assumption 2, we obtain the last term in (28) as follows

$$E\left[\frac{1}{1+\alpha e_t^2}\right] = \frac{1}{\sqrt{2\pi}\sigma_e}\int_{-\infty}^{\infty}\frac{1}{1+\alpha e_t^2}\exp\left(-\frac{e_t^2}{2\sigma_e^2}\right)de_t \\ = \frac{1}{\sqrt{2\alpha\pi}\sigma_e}\int_{-\infty}^{\infty}\frac{\exp\left(-\lambda u^2\right)}{1+u^2}du \\ = \frac{1}{\sqrt{2\alpha\pi}\sigma_e}\pi\exp(\lambda)\text{erfc}(\sqrt{\lambda}), \quad (29)$$

where $u \triangleq \sqrt{\alpha}e_t$, $\lambda \triangleq \frac{1}{2\alpha\sigma_e^2}$, and the third line follows from [30] with $\text{erfc}(\cdot)$ denoting the complementary error function. Hence, putting (29) in (28), we obtain $h_G(e_t)$ for the LMLS update

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

The LLAD algorithm: We have

$$h_G(e_t) = \frac{1}{\sigma_e^2} E\left[\frac{\alpha e_t^2}{1+\alpha|e_t|}\right], \\ = \frac{1}{\sigma_e^2}\left(E[|e_t|] - \alpha^{-1} + \alpha^{-1} E\left[\frac{1}{1+\alpha|e_t|}\right]\right), \quad (30)$$

where the first line follows according to the definition of $g(e_t)$ in (12). According to Assumption 2, we obtain the last term

in (30) as follows

$$E\left[\frac{1}{1+\alpha|e_t|}\right] = \frac{1}{\sqrt{2\pi}\sigma_e}\int_{-\infty}^{\infty}\frac{1}{1+\alpha|e_t|}\exp\left(-\frac{e_t^2}{2\sigma_e^2}\right)de_t$$
$$= \frac{1}{\sqrt{2\pi}\alpha\sigma_e}\int_{-\infty}^{\infty}\frac{1}{1+|u|}\exp\left(-\kappa u^2\right)du$$
$$= \frac{1}{\sqrt{2\pi}\alpha\sigma_e}\frac{\pi\text{erfi}(\sqrt{\kappa})-\text{Ei}(\kappa)}{\exp(\kappa)}, \quad (31)$$

where $u \triangleq \alpha e_t$, and $\kappa \triangleq \frac{1}{2\alpha^2\sigma_e^2}$, and the third line follows from [30] with $\text{erfi}(z) = -j\text{erf}(jz)$ denoting the imaginary error function and $\text{Ei}(x)$ denoting the exponential integral, i.e.,

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

As a result, putting (31) in (30), we obtain $h_G(e_t)$ for the LLAD update

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

APPENDIX B
EVALUATION OF $h_U(e_t)$

The LMLS Algorithm: We have

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

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 $\mathbb{R}^2$. From Appendix A, we obtain

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

The LLAD Algorithm: Following similar lines to LMLS algorithm, we have

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

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 $\mathbb{R}^2$. From Appendix A, we obtain

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

where the third line follows from (31).

REFERENCES

[1] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley and Sons, 2003.

[2] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. Inform. Theory*, vol. 30, no. 2, pp. 275–283, 1984.

[3] V. Nascimento and J. Bermudez, "When is the least-mean fourth algorithm mean-square stable?" in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 4, 2005, pp. iv/341–iv/344 Vol. 4.

[4] V. Nascimento and J. C. M. Bermudez, "Probability of divergence for the least-mean fourth algorithm," *IEEE Trans. Signal Processing*, vol. 54, no. 4, pp. 1376–1385, 2006.

[5] P. Hubscher, J. Bermudez, and V. Nascimento, "A mean-square stability analysis of the least mean fourth adaptive algorithm," *IEEE Trans. on Signal Processing*, vol. 55, no. 8, pp. 4018–4028, 2007.

[6] E. Eweda and N. Bershad, "Stochastic analysis of a stable normalized least mean fourth algorithm for adaptive noise canceling with a white gaussian reference," *IEEE Trans. Signal Processing*, vol. 60, no. 12, pp. 6235–6244, 2012.

[7] V. Nascimento, "A simple model for the effect of normalization on the convergence rate of adaptive filters," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, vol. 2, 2004, pp. ii–453–6 vol.2.

[8] M. Shao and C. Nikias, "Signal processing with fractional lower order moments: stable processes and their applications," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 986–1010, 1993.

[9] S. R. Kim and A. Efron, "Adaptive robust impulse noise filtering," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1855–1866, 1995.

[10] V. J. Mathews and S.-H. Cho, "Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 4, pp. 450–454, 1987.

[11] J. Chambers, O. Tanrikulu, and A. Constantinides, "Least mean mixed-norm adaptive filtering," *Electron. Lett.*, vol. 30, no. 19, pp. 1574–1575, 1994.

[12] J. Chambers and A. Avlonitis, "A robust mixed-norm adaptive filter algorithm," *IEEE Signal Processing Lett.*, vol. 4, no. 2, pp. 46–48, 1997.

[13] J. Arenas-Garcia, V. Gomez-Verdejo, M. Martinez-Ramon, and A. Figueiras-Vidal, "Separate-variable adaptive combination of lms adaptive filters for plant identification," in *2003 IEEE 13th Workshop on Neural Networks for Signal Processing, 2003. NNSP'03.*, 2003, pp. 239–248.

[14] J. Arenas-Garcia, V. Gomez-Verdejo, and A. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of lms transversal filters," *IEEE Trans. Instrumentation and Measurement*, vol. 54, no. 6, pp. 2239–2249, 2005.

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

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

[17] S. Kozat, A. Erdogan, A. Singer, and A. Sayed, "Steady-state mse performance analysis of mixture approaches to adaptive filtering," *IEEE Trans. Signal Processing*, vol. 58, no. 8, pp. 4050–4063, 2010.

[18] J. Arenas-Garcia and A. Figueiras-Vidal, "Adaptive combination of normalised filters for robust system identification," *Electron. Lett.*, vol. 41, no. 15, pp. 874–875, 2005.

[19] P. Petrus, "Robust huber adaptive filter," *IEEE Trans. Signal Processing*, vol. 47, no. 4, pp. 1129–1133, 1999.
[20] R. G. Bartle and D. R. Scherbert, *Introduction to Real Analysis*. John Wiley and Sons, 2011.
[21] I. Song, P. Park, and R. Newcomb, "A normalized least mean squares algorithm with a step-size scaler against impulsive measurement noise," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 60, no. 7, pp. 442–445, 2013.
[22] T. Y. Al-Naffouri and A. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 639–652, 2003.
[23] ——, "Transient analysis of adaptive filters with error nonlinearities," *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 653–663, 2003.
[24] R. Price, "A useful theorem for nonlinear devices having gaussian inputs," *IEEE Trans. Inform. Theory*, vol. 4, no. 2, pp. 69–72, 1958.
[25] E. McMahon, "An extension of price's theorem (corresp.)," *IEEE Trans. Inform. Theory*, vol. 10, no. 2, pp. 168–168, 1964.
[26] T. Koh and E. Powers, "Efficient methods of estimate correlation functions of gaussian processes and their performance analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 4, pp. 1032–1035, 1985.
[27] H. Van Trees, *Detection, Estimation, and Modulation Theory*, ser. Detection, Estimation, and Modulation Theory. Wiley, 2004, no. pt. 1.
[28] X. Wang and H. Poor, "Joint channel estimation and symbol detection in rayleigh flat-fading channels with impulsive noise," *IEEE Comm. Lett.*, vol. 1, no. 1, pp. 19–21, 1997.
[29] 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 Processing*, vol. 52, no. 4, pp. 975–991, 2004.
[30] W. Grobner and N. Hofreiter, *Bestimmte Integrale*. Springer-Verlag, 1966.