Performance Analysis of Fractional Learning Algorithms

Abdul Wahab∗†  Shujaat Khan‡  Imran Naseem§ ¶  Jong Chul Ye‡

January 11, 2022

Abstract

Fractional learning algorithms are trending in signal processing and adaptive filtering recently. However, it is unclear whether the proclaimed superiority over conventional algorithms is well-grounded or is a myth as their performance has never been extensively analyzed. In this article, a rigorous analysis of fractional variants of the least mean squares and steepest descent algorithms is performed. Some critical schematic kinks in fractional learning algorithms are identified. Their origins and consequences on the performance of the learning algorithms are discussed and swift ready-witted remedies are proposed. Apposite numerical experiments are conducted to discuss the convergence and efficiency of the fractional learning algorithms in stochastic environments.

Keywords. Least Mean Squares; Fractional Least Mean Squares; Fractional derivatives, Gradient descent.

1 Introduction

The least mean square (LMS) algorithms are of paramount importance in the field of signal processing since their emergence [61, 62, 60]. In particular, they are used profusely in adaptive filtering and signal analysis [27, 64, 24, 5]. The key aspects that make LMS algorithms attractive are their low complexity, stability, and an unbiased mean convergence to the so-called Wiener solution in stationary environments [48]. Unfortunately, its rate of convergence depends on the eigenvalue spread of the correlation matrix of the input signal in non-stationary environments [62, 27]. Accordingly, many variant algorithms were proposed to achieve better performance by curtailing the influence of the spectral properties of the input signal correlation matrix; see, for instance, the LMS-Newton algorithm [25], transform-domain algorithm [37], and affine projection algorithm [49]. On the other hand, a desire for computationally simpler algorithms has also led to the development of many variants such as quantized-error algorithms [6, 19, 30] and normalized LMS algorithms [65, 48, 36]. A decent list of these variant algorithms along with the details of their key features is provided in [24, Ch. 4]. We also refer to fairly recent survey articles [63, 26] on the history of adaptive filtering and the development of the LMS algorithms.

∗Corresponding Author. E-mail address: abdul.wahab@nu.edu.kz.
†Department of Mathematics, School of Sciences and Humanities, Nazarbayev University, 53, Kabanbay Batyr Ave., 010000, Nur-Sultan, Kazakhstan (abdul.wahab@nu.edu.kz).
‡Bio-Imaging, Signal Processing and Learning Lab., Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, 34141, Daejeon, South Korea (shujaat@kaist.ac.kr, jong.ye@kaist.ac.kr).
§School of Electrical, Electronic and Computer Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, Western Australia 6009, Australia (imran.naseem@uwa.edu.au).
¶College of Engineering, Karachi Institute of Economics and Technology, Korangi Creek, 75190, Pakistan.
Recently, so-called fractional LMS and relevant learning algorithms are trending and many variants have been proposed over the past decade. The central idea behind these algorithms is to replace the classical integer-order gradient with a fractional gradient to achieve better performance. The idea of using fractional derivatives in the LMS algorithms first appeared in [42] in the context of a system identification problem. Subsequently, the algorithm was applied for chaotic and nonstationary time series prediction [47], parameter estimation of CARMA systems [41, 2], active noise control systems [4, 45], Hammerstein nonlinear autoregressive systems [13, 10], and nonlinear Box-Jenkins systems [11, 3]. Many variant fractional LMS algorithms were also proposed such as fractional steepest descent approach [40, 53], fractional normalized LMS [16, 33], bias compensated fractional normalized LMS [66, 14, 55], fractional filtered-X normalized LMS [67], complex fractional LMS [43, 34], momentum fractional LMS [32, 69], and Volterra fractional LMS [9, 12].

Unfortunately, there were two kinks in the initial designs of the fractional LMS algorithms: (1) The emergence of the complex outputs and, (2) the long-memory characteristic of the fractional derivatives. The iterate-update rules involved a fractional power of the past iterates. Consequently, the fractional algorithms would render complex outputs whenever the intermediate iterates became negative. This led to the introduction of an absolute value in the iterate-update rule as a ready-witted remedy; see, for example, [10, 4, 8]. Secondly, the fractional derivatives were non-local in nature because they were defined in terms of convolution integrals, unlike their integer-order counterparts. Accordingly, the fractional LMS algorithms carried forward information of all the past states that created doubts about their ability to provide punctual geometric information, or simply about their convergence to the optimal solution even in stationary environments. On the other hand, due to long-memory requirements, the computational cost was also high. This led to the developments of variable initial value and variable fractional-order schemes; see, for instance, [20, 21, 22, 23]. The modified algorithms were used in many applied problems [31, 44, 68]. More recently, these algorithms were also used in neural network designs [18, 46, 57, 58].

Despite their widespread use, the performance analysis of the fractional LMS algorithms was mostly done heuristically. The first attempt can be traced back to Pu et al. [40] who performed the rate of convergence analysis of a fractional steepest descent algorithm (however, the algorithm suffered from the issue of complex outputs and their analysis relied on some detrimental approximations; see [53]). Chen et al. [17] and Wei et al. [59, 54] studied variable initial value and fractional-order gradient methods and showed that, under the assumption of convergence, the solution converges to the Wiener solution in stationary cases. Chaudary et al. [14] and [15] discussed the convergence of two variants of fractional LMS in non-stationary cases, however, both results had technical flaws (see, [56, 55]). Bershad, Wen and So [7] performed extensive numerical simulations and established that the performance of fractional LMS algorithms was no better than the conventional LMS algorithms in stochastic cases.

The main goal of this article is to rigorously analyze the performance of the fractional learning algorithms from both mathematical and numerical points of view. As a simple example, we consider a system identification problem for which we study two representative fractional algorithms. We aim to rigorously derive the update rules and discuss the underlying assumptions and schematic kinks of the fractional LMS algorithms. Moreover, we study the connection, if any, between the ordinary and fractional critical points and discuss the convergence of the algorithms.

The rest of this article is arranged as follows. In Section 2, we introduce some preliminaries, a model problem, and two representative fractional algorithms considered in this study. In Section 3, we perform rigorous mathematical analysis of the representative algorithms. We first
derive the iterate-update rules and discuss their underlying assumptions. The schematic kinks in the fractional algorithms are identified and discussed from the mathematical and geometrical points of view. The convergence of the algorithms in stationary environments is also discussed. In Section 4, some numerical experiments are conducted for performance analysis in stochastic environments. The article ends with a brief discussion and summary of the results in Section 5.

2 Problem Formulation

The primary concern of this article is the performance analysis of fractional learning algorithms. For simplicity, we entertain two representative fractional LMS algorithms for a system identification problem. Accordingly, we feel it important to provide some preliminaries from fractional Calculus (Section 2.1), give a brief introduction to the system identification problem (Section 2.2), and introduce the iterate-update rules for the prototype fractional algorithms (Section 2.3).

2.1 Elements of Fractional Calculus

In this article, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$ represent the sets of natural numbers, integers, real numbers, and complex numbers, respectively. Moreover, for any $z = x + \sqrt{-1} y \in \mathbb{C}$, we denote the real and imaginary components of $z$ by $\Re\{z\} := x$ and $\Im\{z\} := y$, respectively. Further, for any vector $u \in \mathbb{R}^N$, the quantity $(u)_i$ represents the $i$-th component of the vector $u$, for any $1 \leq i \leq N \in \mathbb{N}$. Similarly, for any matrix $M \in \mathbb{R}^{N \times M}$, the $ij$-th entry is represented by $(M)_{ij}$, for $1 \leq i \leq N \in \mathbb{N}$ and $1 \leq j \leq M \in \mathbb{N}$. Finally, throughout this article, all the vector quantities are represented by lower-case bold letters, and all the matrices are represented by upper-case bold letters.

Definition 2.1 (Gamma Function). For any $z \in \mathbb{C}$ such that $\Re\{z\} > 0$, the Euler’s gamma function, denoted by $\Gamma$, is defined by

$$
\Gamma(z) := \int_0^\infty t^{z-1} e^{-t} dt.
$$

Note that, $\Gamma(1+z) = z\Gamma(z)$ and therefore, $\Gamma(1+n) = n!$ for any $n \in \mathbb{Z}$ such that $n \geq 0$.

Definition 2.2 (Riemann-Liouville Fractional Integrals \cite[Eqs. (2.1.1)-(2.1.2)]{35}). Let $z \in \mathbb{C}$ such that $0 < \Re\{z\} < 1$. The left fractional integral of order $z$ over an interval $(a,b) \subset \mathbb{R}$ of a left integrable function $\varphi$, with $a < b$, is defined by

$$
I_L^z[\varphi](t) := \frac{1}{\Gamma(z)} \int_a^t \frac{\varphi(\tau)}{(t-\tau)^{1-z}} d\tau, \quad \forall t \in (a,b). \tag{2.1}
$$

Similarly, the right fractional integral of order $z$ over $(a,b) \subset \mathbb{R}$ of a right integrable function $\varphi$ is defined by

$$
I_R^z[\varphi](t) := \frac{1}{\Gamma(z)} \int_t^b \frac{\varphi(\tau)}{(\tau-t)^{1-z}} d\tau, \quad \forall t \in (a,b). \tag{2.2}
$$
Definition 2.3 (Riemann-Liouville Fractional Derivatives [35, Eqs. (2.1.5)-(2.1.6)]). Let \( z \in \mathbb{C} \) such that \( m - 1 < \Re(z) < m \) for some \( m \in \mathbb{N} \). The left Riemann-Liouville fractional derivative of order \( z \) over an interval \((a, b)\) with \( a < b \) of a sufficiently smooth function \( \varphi \) is defined by

\[
{L_a}^z \varphi(t) := \frac{d^m}{dt^m} \left( I^{m-z}_a \varphi(t) \right) = \frac{1}{\Gamma(m - z)} \frac{d^m}{dt^m} \int_a^t \frac{\varphi(\tau)}{(t - \tau)^{1-m+z}} d\tau, \quad \forall t \in (a, b). \tag{2.3}
\]

Similarly, the right Riemann-Liouville fractional derivative of \( \varphi \) of order \( z \) is defined by

\[
{R_a}^z \varphi(t) := (-1)^m \frac{d^m}{dt^m} \left( I^{m-z}_R \varphi(t) \right) = \frac{(-1)^m}{\Gamma(m - z)} \frac{d^m}{dt^m} \int_t^b \frac{\varphi(\tau)}{\tau - t)^{1-m+z}} d\tau, \quad \forall t \in (a, b). \tag{2.4}
\]

Remark 2.4. If \( z \in \mathbb{R} \) in Definition 2.3 such that \( z \rightarrow m \) then,

\[
{L_a}^z \varphi(t)(t) \rightarrow \frac{d^m \varphi(t)}{dt^m} \text{ and } {R_a}^z \varphi(t)(t) \rightarrow \frac{d^m \varphi(t)}{dt^m}, \quad \forall t \in (a, b).
\]

Refer, for instance, to the monographs [35, 38] for detailed expositions.

The following theorem holds.

Theorem 2.5 ([35, Property 2.1, Page 71]). If \( \alpha, \beta \in \mathbb{C} \) such that \( \Re(\alpha) \geq 0 \) and \( \Re(\beta) > 0 \) then

\[
{L_a}^z \alpha^\beta(t - a)^\beta - 1(t) = \frac{\Gamma(\beta)}{\Gamma(\beta - \alpha)} (t - a)^{\beta - 1}, \tag{2.5}
\]

\[
{R_a}^z \alpha^\beta(b - t)^\beta - 1(t) = \frac{\Gamma(\beta)}{\Gamma(\beta - \alpha)} (b - t)^{\beta - 1}, \tag{2.6}
\]

for all \( t \in (a, b) \subset \mathbb{R} \). In particular,

\[
{L_a}^z \alpha^1(t) = \frac{(t - a)^{-\alpha}}{\Gamma(1 - \alpha)} \text{ and } {R_a}^z \alpha^1(t) = \frac{(b - t)^{-\alpha}}{\Gamma(1 - \alpha)} \tag{2.7}
\]

2.2 Adaptive Filters for System Identification

The system identification problem is well-known in adaptive filtering. However, it is the main building block of the analytic arguments in this study. Therefore, we elaborate on it anyway to facilitate the ensuing discussion. Consider a simple problem of identification of a real-valued discrete-time filter with the unknown finite impulse response,

\[
w_n := \begin{bmatrix} w_n(0) & w_n(1) & \cdots & w_n(N - 1) \end{bmatrix}^T \in \mathbb{R}^N,
\]

that describes the behavior of input,

\[
x_n := \begin{bmatrix} x_n & x_{n-1} & \cdots & x_{n-N+1} \end{bmatrix}^T \in \mathbb{R}^N,
\]

to desired output, \( d_n \in \mathbb{R} \). The subscript \( n \in \mathbb{Z} \) is the time index and the superposed \( ^T \) indicates the transpose operation. The target output \( d_n \) is furnished in order to supervise the filter weights \( w_n \) so that the system renders filter output, \( y_n = w_n^T x_n \in \mathbb{R} \), that resembles the target in the least mean square sense. The mean square error (MSE), denoted by \( \mathcal{E} \), is defined in terms of the output estimation error, \( e_n := d_n - y_n = d_n - w_n^T x_n \) (for a fixed \( w_n \)), as

\[
\mathcal{E}(w_n) := E[\epsilon_n^2] = E[(d_n - w_n^T x_n)^2] = \sigma_d^2 - 2w_n^T p + w_n^T R w_n. \tag{2.8}
\]
Here, $R := E [x_n x_n^\top] \in \mathbb{R}^{N \times N}$ is the auto-correlation of the input, $p := E [d_n x_n] \in \mathbb{R}^N$ is the cross-correlation between the input and the target output, and $\sigma_d^2 := E [d_n^2]$ is the variance of the target output.

To solve the system identification problem, the MSE is minimized for optimal weight filter, $w^* := [w^*(0) \ w^*(1) \ \cdots \ w^*(N-1)]^\top$.

Towards this end, the least mean squares (LMS) algorithm is usually invoked. The idea is to allow the weights to be time-varying so that they can be optimized in an iterative manner along the steepest descent of $\mathcal{E}$. Accordingly, utilizing the available data, the instantaneous gradient vector is derived as

$$\nabla_{w_n} \hat{\mathcal{E}}(w_n) = \nabla_{w_n} e_n^2 = \frac{\partial e_n^2}{\partial e_n} \frac{\partial e_n}{\partial y_n} \nabla_{w_n} y_n = -2 e_n x_n,$$ (2.9)

thanks to the chain rule for the ordinary derivatives of composite functions. Since the negative of the gradient vector always points towards the direction of the steepest descent of the hyper-paraboloid surface formed by $\mathcal{E}$, directional increments opposite to the gradient vector gradually move the successive weight iterates closer to the minimum of $\mathcal{E}$. Accordingly, the LMS weight-update rule is defined as

$$w_{n+1} := w_n - \mu \hat{\mathcal{E}}(w_n) = w_n + \mu f e_n x_n,$$ (2.10)

with an initial guess based on a priori information and a parameter $\mu_f > 0$ controlling the rate of learning. The LMS algorithm is very stable and efficient with a computational cost of $O(N)$ per iteration. However, its convergence highly depends on the condition number of the auto-correlation matrix $R$ and is relatively poor. Accordingly, various remedial interventions have been proposed in the conventional LMS algorithms and numerous modified algorithms are available achieving better convergence rates at the cost of increased computational complexity and reduced efficiency. A detailed account of these variants is out of the scope of the present investigation, however, it is emphasized that the ensuing discussion is also relevant to these variants and similar analyses hold with appropriate adjustments.

### 2.3 Fractional Learning Algorithms

In the recent past, a plethora of fractional variants of the LMS and steepest descent algorithms (SDA) have been proposed in adaptive signal processing wherein classical integer-order derivatives are fully or partially replaced by the fractional-order derivatives with order parameter (say) $\alpha \in (0, 1)$. The simplest form of the fractional LMS iterate-update rule appears to be

$$w_{n+1} := w_n - \frac{\mu_f}{2} \nabla_{w_n} \hat{\mathcal{E}}(w_n) = w_n + \mu_f e_n x_n,$$ (2.11)

in terms of the fractional gradient

$$\left(\nabla_{w_n}^{\alpha} \hat{\mathcal{E}}(w_n)\right)_{t} := \frac{D_{w_n(t)}^{\alpha}}{\alpha} \left[\hat{\mathcal{E}}(w_n)\right].$$

Here, $\mu_f$ is a control parameter that supervises the rate of learning due to fractional gradient. Some of the variant algorithms completely replace the integer-order gradient in conventional counterparts with that of fractional-order (in which case $\mu_f = 0$). On the other hand, some algorithms use both $\mu_f, \mu_f > 0$, i.e., both fractional and integer-order gradients are assumed to
play a role in the weight update through an iterative procedure. Moreover, when $\mu_f = 0$, the fractional LMS algorithm tends to the LMS algorithm.

There are several variants of the fractional LMS algorithms available in the literature. In this investigation, we discuss two representative iterate-update rules for brevity and clarity. Nevertheless, we emphasize that the analysis performed here is relevant to other variants also since weight iterates corresponding to many of them usually follow slightly modified versions of the representative iterate-update rules. The first representative iterate-update rule proposed in [42] suggests

$$w_{n+1}(l) = w_n(l) + \mu_\ell e_n x_{n-l} + \frac{\mu_f}{\Gamma(2 - \alpha)} e_n x_{n-l} w_n^{1-\alpha}(l), \quad l = 0, 1, \ldots, N - 1,$$

(2.12)

or in vector form

$$w_{n+1} = w_n + \mu_\ell e_n x_n + \frac{\mu_f}{\Gamma(2 - \alpha)} e_n x_n \odot w_n^{1-\alpha}.$$

(2.13)

Here, $\odot$ represents element-wise product and notation $u^{1-\alpha}$ is used for element-wise power of any $u \in \mathbb{R}^N$, i.e., $u^{1-\alpha} = [u_1^{1-\alpha}, u_2^{1-\alpha}, \ldots, u_N^{1-\alpha}]$. Alternatively, by letting

$$F_\alpha(u) := \frac{1}{\Gamma(2 - \alpha)} \text{diag}(u_1^{1-\alpha}, u_2^{1-\alpha}, \ldots, u_N^{1-\alpha}) \in \mathbb{R}^{N \times N},$$

(2.14)

the iterate-update rule (2.12) can be written in vector form as

$$w_{n+1} = w_n + \mu_\ell e_n x_n + \mu_f e_n F_\alpha(w_n)x_n.$$

(2.15)

The second representative iterate-update rule proposed, e.g., in [22, 66], is given as

$$w_{n+1}(l) = w_n(l) + \frac{\mu_f}{\Gamma(2 - \alpha)} e_n x_{n-l} (w_n(l) - w_{n-1}(l))^{1-\alpha}, \quad l = 0, 1, \cdots, N - 1,$$

(2.16)

or in vector form as

$$w_{n+1} = w_n(l) + \mu_f e_n F_\alpha(w_n - w_{n-1})x_n.$$

(2.17)

Remark that, there is a possibility of having negative values under fractional powers in (2.12) and (2.16) that could lead to complex outputs and could cause emergency exits. As a ready-witted remedy, absolute values are introduced in the iterate-update rules, i.e.,

$$w_{n+1}(l) = w_n(l) + \mu_\ell e_n x_{n-l} + \frac{\mu_f}{\Gamma(2 - \alpha)} e_n x_{n-l} |w_n(l)|^{1-\alpha},$$

(2.18)

$$w_{n+1}(l) = w_n(l) + \frac{\mu_f}{\Gamma(2 - \alpha)} e_n x_{n-l} |w_n(l) - w_{n-1}(l)|^{1-\alpha}.$$

(2.19)

The derivation of (2.12) and (2.16) from (2.11) will be discussed bit-by-bit in Section 3.1 and the underlying assumptions will be highlighted.

**Remark 2.6.** It is already evident that the fractional LMS algorithms are computationally more expensive than the conventional LMS algorithms as they require additional operations to compute the fractional terms in the iterate-update rules.
3 Schematic Kinks in Fractional Learning Algorithms

In this section, we perform a rigorous mathematical analysis of the fractional LMS algorithms. We begin by deriving the representative iterate-update rules (2.12) and (2.16) with an aim to understand their main assumptions (Section 3.1). Then, we investigate the origin and the remedies of complex outputs (Section 3.2). We also discuss the geometrical interpretation of the fractional derivatives and thereby constructed learning algorithms (Section 3.3). This will also help us understand the long memory and short memory characteristics. Finally, we discuss the convergence of these algorithms in stationary environments (Section 3.4).

3.1 Derivation of Fractional Iterate-Update Rules

We discuss the derivation of both representative iterate-update rules separately.

3.1.1 Derivation of (2.12)

Most common argument used while deriving the iterate-update rule (2.12) consists of using chain rule on objective functional \( \hat{\mathcal{E}}(w_n) \) for evaluating its fractional gradient, exactly in the same fashion as in (2.9); see, for instance, [42, 43, 4]. Precisely, the MSE in (2.8) is fractionally differentiated as

\[
L_0^\alpha D_w^\alpha (\hat{\mathcal{E}}(w_n)) = L_0^\alpha D_w^\alpha (\hat{\mathcal{E}}(w_n)) = (2e_n) (-1) (x_n - l)^{1-\alpha} \Gamma(2-\alpha)
\]

using the formula (2.5) with \( a = 0 \). Eq. (3.1) renders iterate-update rules (2.12), (2.13), and (2.15) on substitution in (2.11). Unfortunately, the conventional chain rule used to derive (3.1) is mathematically invalid for fractional derivatives (see, for instance, [50] for detailed discussion).

The fractional chain rule for left Riemann-Liouville derivative is derived using Faà di Bruno formula and is given by (see, for instance, [38, Eq. (2.209)])

\[
L^a D_t^p F(h(t)) = (t-a)^{-p} \Gamma(1-p) F(h(t)) + \sum_{k=1}^{\infty} \left( \frac{p}{k} \right) k! (t-a)^{k-p} \Gamma(k-p+1) \sum_{m=1}^{k} F^{(m)}(h(t)) \sum_{r=1}^{k} \frac{1}{a_r!} \left( \frac{h^{(r)}(t)}{r!} \right)^{a_r},
\]

where \( F \) and \( h \) are sufficiently smooth functions, the sum \( \sum \) extends over all combinations of non-negative integer values of \( a_1, \ldots, a_k \) such that

\[
\sum_{r=1}^{k} r a_r = k \quad \text{and} \quad \sum_{r} a_r = m.
\]

A mathematically valid procedure to derive the iterate-update rule (2.12) must avoid using fractional chain rule. Towards this end, the MSE in (2.8) is expanded as

\[
\hat{\mathcal{E}}(w_n) = \sigma^2 - 2 \sum_{i=0}^{N-1} w_n(i)p_i(n) + \sum_{i,j=0}^{N-1} w_n(i)w_n(j)R_{ij}(n),
\]

(3.4)
where \( p_i := (p)_i \) and \( R_{ij} := (R)_{ij} \) are the components of the cross-correlation vector \( p \) and the auto-correlation matrix \( R \). To find the component fractional derivative, \( ^0 \mathcal{D}_w^\alpha (t)[\hat{\mathcal{E}}] \), we re-arrange (3.4) further as
\[
\hat{\mathcal{E}}(w_n) = \sigma_n^2 - 2 \sum_{i=0}^{N-1} \sum_{i \neq l}^{N-1} w_n(i)p_i(n) - 2w_n(l)p_l(n) + \sum_{i,j=0}^{N-1} w_n(i)w_n(j)R_{ij}(n) \\
+ \sum_{n=0}^{N-1} w_n(i)w_n(l)R_{il}(n) + \sum_{j=0}^{N-1} w_n(l)w_n(j)R_{lj}(n) + w_n^2(l)R_{ll}(n) \\
= \Psi_n(l) + 2w_n(l) \left[ \sum_{i=1, i \neq l}^{N-1} w_n(i)R_{il}(n) - p_l(n) \right] + w_n^2(l)R_{ll}(n),
\]
where the fact that \( R \) is symmetric (i.e., \( R_{ij} = R_{ji} \)) is used. Here
\[
\Psi_n(l) := \sigma_n^2 - 2 \sum_{i=0, i \neq l}^{N-1} w_n(i)p_i(n) + \sum_{i,j=0}^{N-1} w_n(i)w_n(j)R_{ij}(n),
\] (3.5)
is a constant with respect to \( w_n(l) \). Therefore, by the definition of the Left-Riemann-Liouville derivative and invoking the rule (2.5), one arrives at
\[
^0 \mathcal{D}_w^\alpha (t)[\hat{\mathcal{E}}(w_n)] = \frac{w_n^{-\alpha}(l)}{\Gamma(1-\alpha)} \Psi_n(l) + \frac{2w_n^{1-\alpha}(l)}{\Gamma(2-\alpha)} \left[ \sum_{i=0}^{N-1} w_n(i)R_{il}(n) - p_l(n) \right] \\
+ \frac{2w_n^{2-\alpha}(l)}{\Gamma(3-\alpha)} R_{ll}(n).
\] (3.6)

Let us now try to put (3.6) in the form (3.1). Towards this end, we express the derivative as
\[
^0 \mathcal{D}_w^\alpha (t)[\hat{\mathcal{E}}(w_n)] = \frac{w_n^{-\alpha}(l)}{\Gamma(1-\alpha)} \Psi_n(l) - \frac{2w_n^{1-\alpha}(l)}{\Gamma(2-\alpha)} \left[ p_l(n) - \sum_{i=0}^{N-1} w_n(i)R_{il}(n) + w_n(l)R_{ll}(n) \right] \\
+ \frac{2w_n^{2-\alpha}(l)}{\Gamma(3-\alpha)} R_{ll}(n)
\]
\[
= \frac{w_n^{-\alpha}(l)}{\Gamma(1-\alpha)} \Psi_n(l) - \frac{2w_n^{1-\alpha}(l)}{\Gamma(2-\alpha)} \left[ p_l(n) - \sum_{i=0}^{N-1} w_n(i)R_{il}(n) \right] \\
+ \frac{2w_n^{2-\alpha}(l)}{\Gamma(3-\alpha)} R_{ll}(n) \left[ \frac{1}{\Gamma(3-\alpha)} - \frac{1}{\Gamma(2-\alpha)} \right]
\]
\[
= \frac{w_n^{-\alpha}(l)}{\Gamma(1-\alpha)} \Psi_n(l) - \frac{2w_n^{1-\alpha}(l)}{\Gamma(2-\alpha)} \left[ d_n - w_n x_{n-1} \right] x_{n-l} - \frac{2(1-\alpha)w_n^{2-\alpha}(l)}{\Gamma(3-\alpha)} R_{ll}(n),
\] (3.7)
since \( \Gamma(x+1) = x\Gamma(x) \), \( p_l(n) = d_n x_{n-1} \), and \( R_{il} = x_{n-i} x_{n-l} \). Therefore, one can conclude that
\[
^0 \mathcal{D}_w^\alpha (t)[\hat{\mathcal{E}}(w_n)] \approx -2\varepsilon_n x_{n-1} \frac{w_n^{1-\alpha}(l)}{\Gamma(2-\alpha)},
\] (3.8)
as in (3.1), subject to following assumptions.

Assumptions.

A1 For all values of \( l \) and \( n \), we have \( w_n(l) > 0 \).

This assumption was tacitly made when the Definition 2.3 of the left Riemann-Liouville derivative was used setting \( a = 0 \) as the lower limit of the integral in (2.3). Note that, \( w_n(l) \) corresponds to the upper limit. This assumption may not be valid, especially, in the stochastic case. We will discuss this point in detail in Section 3.2.1.

A2 Additive constants in a function do not affect its extreme points.

This assertion stems from the hypothesis that the fractional derivative of the constant term, \( \Psi_n(l) \), can be neglected without affecting the extrema of the function \( \hat{E}(w_n) \). This hypothesis is used in the literature (see, e.g., [22, Remark 1]). The assertion is true when integer-order derivatives are used. However, it is invalid when fractional-order derivatives are used as they are non-zero for non-zero constants; see Theorem 2.5. We will elaborate on this point further in Section 3.3.2.

A3 The first term dominates the expression

\[
\frac{2w_n^{1-\alpha}(l)}{\Gamma(2-\alpha)} \left[ d_n - w_n x_n \right] x_{n-l} + \frac{2(1-\alpha)w_n^{2-\alpha}(l)}{\Gamma(3-\alpha)} R_{ll}(n),
\]

and therefore, the second term,

\[
\frac{2(1-\alpha)w_n^{2-\alpha}(l)}{\Gamma(3-\alpha)} R_{ll}(n),
\]

can be suppressed.

This assumption may be reasonable when \( \alpha \to 1 \).

3.1.2 Derivation of (2.16)

The iterate-update rule (2.16) has been derived, for instance, in [66, Sec. 3.2] and [22, Sec. 3.2]. However, we briefly give the idea of the derivation for completeness. Towards this end, the MSE (2.8) is expanded as

\[
\hat{E}(w_n) = \left( d_n - \sum_{i=0}^{N-1} w_n(i)x_{n-i} \right)^2
\]

\[
= \left( d_n - \sum_{i=0, i \neq l}^{N-1} w_n(i)x_{n-i} - w_n(l)x_{n-l} - [w_n(l) - w_n-1(l)]x_{n-l} \right)^2
\]

\[
= \left( \Phi_n(l) - [w_n(l) - w_n-1(l)]x_{n-l} \right)^2
\]

\[
= \Phi_n^2(l) - 2\Phi_n(l)[w_n(l) - w_n-1(l)]x_{n-l} + \left( [w_n(l) - w_n-1(l)]x_{n-l} \right)^2,
\]

(3.11)
where
\[ \Phi_n(l) := d_n - \sum_{i=0, i \neq l}^{N-1} w_n(i)x_{n-i} - w_{n-1}(l)x_{n-l}. \tag{3.12} \]

Differentiating (3.11) using the left Riemann-Liouville derivative \( w_{n-1}(l) L^\alpha D^\alpha_{w_n(l)} \) defined through (2.3), invoking the power rule (2.5), and neglecting the constant term \( \Phi^2_n(l) \) as in the previous case for the derivation of (2.12), one arrives at
\[
\begin{align*}
& w_{n-1}(l) L^\alpha D^\alpha_{w_n(l)} \left[ \hat{E}(w_n) \right] \\
& \approx -2 \frac{\Phi_n(l)}{\Gamma(2-\alpha)} x_{n-l} \left[ w_n(l) - w_{n-1}(l) \right]^{1-\alpha} + \frac{2}{\Gamma(3-\alpha)} x_{n-l}^2 \left[ w_n(l) - w_{n-1}(l) \right]^{2-\alpha}, \\
& \approx -2 \frac{\Phi_n(l)}{\Gamma(2-\alpha)} e_n x_{n-l} \left[ w_n(l) - w_{n-1}(l) \right]^{1-\alpha}. \tag{3.13}
\end{align*}
\]

This furnishes (2.16) on substitution in (2.11) together with \( \mu_l = 0 \). Note that, to arrive at equation (3.13), following assumptions were made.

**Assumptions.**

**B1** For all values of \( l \) and \( n \), we have \( w_{n-1}(l) < w_n(l) \).

As mentioned in the previous case, the first assumption tacitly made while using the Definition 2.3 of the left Riemann-Liouville derivative was \( w_{n-1}(l) < w_n(l) \). Here, \( w_{n-1}(l) \) and \( w_n(l) \) respectively correspond to the lower and the upper limits of the integral in (2.3). This assumption may not be valid, e.g., when any one of the optimal weights \( w^*(l) \) is negative. In that case, the sequence \( (w_n(l))_{n\in\mathbb{N}} \) is expected to be decreasing so that \( \lim_{n\to\infty} w_n(l) \) converge to the negative value. We will discuss this point in detail in Section 3.2.1.

**B2** Additive constants in a function do not affect its extreme points

This assertion stems from the hypothesis that the fractional derivative of the additive constant term, \( \Phi^2_n(l) \), can be neglected without affecting the extrema of the function \( \hat{E}(w_n) \). We will elaborate on this point further in Sections 3.3.2 and 3.3.3.

**B3** There exists a step size \( \mu_f \in \mathbb{R} \) such that \( |w_n(l) - w_{n-1}(l)| < 1 \).

Under this assumption, in the expression
\[
-2 \frac{\Phi_n(l)}{\Gamma(2-\alpha)} x_{n-l} \left[ w_n(l) - w_{n-1}(l) \right]^{1-\alpha} + \frac{2}{\Gamma(3-\alpha)} x_{n-l}^2 \left[ w_n(l) - w_{n-1}(l) \right]^{2-\alpha},
\]
the first term is dominant and thus, the second term can be neglected.
Remark 3.1. Under the assumption B3, if the step size $\mu_f$ is set properly, $w_n(l)$ changes slowly, i.e., $w_n(l) \approx w_{n-1}(l)$ and consequently, $\Phi_n(l) \approx e_n$. Indeed, we have

$$\Phi_n(l) = d_n - \sum_{i=0,i \neq l}^{N-1} w_n(i)x_{n-i} - w_{n-1}(l)x_{n-l}$$

$$\approx d_n - \sum_{i=0,i \neq l}^{N-1} w_n(i)x_{n-i} - w_n(l)x_{n-l}$$

$$= d_n - \sum_{i=0}^{N-1} w_n(i)x_{n-i} = e_n.$$  

The downside of Assumption (B3) is a slow convergence rate for the fractional LMS algorithms defined by the iterate-update rule of the type (2.16) (i.e., the algorithms with variable initial terms).

3.2 Emergence of Complex Outputs

It is interesting to note that both representative iterate-update rules (2.12) and (2.16) (and, in fact, all the fractional iterate-update rules available in the literature) contain fractional powers of the quantities involving iterates $w_n(l)$. Therefore, whenever a fractional iterate $w_n(l)$ under the fractional power is negative, the resultant becomes complex. It stymies the applicability of the fractional variants of the LMS algorithm for not only negative sought values but also for positive sought values. A simple justification is that the LMS iterate does not move in a straight path towards the optimal solution, it rather takes a zigzag path (see, e.g., Fig. 1). That motivated a heuristic introduction of the absolute value in the update rules as given in (2.18) and (2.19) without any retrospective or prospective analysis. In this section, we identify the origin of the complex outputs and try to make sense of an absolute value in the iterate-update rules.

![Figure 1: Illustration of the LMS learning path.](image)

3.2.1 The Genesis of Complex Outputs

Let us start our discussion with the iterate-update rules of the form (2.12). We recall that the left Riemann-Liouville fractional derivative $^{\text{L}}D_{\alpha}^\omega w_n(l)$ was used with $\alpha = 0$ (the lower terminal
It is important to clarify here that there is no restriction on the choice of \( a = 0 \) in the derivation of the iterate-update rule. The choice of \( a \) correlates with the underlying (but unspecified) Assumption (A1) that
\[
w_n(l) > 0, \quad \text{for all} \quad 0 \leq l \leq N - 1, \quad n \in \mathbb{N}.
\]

Unfortunately, it stymies the applicability of the iterate-update rule (2.12) for negative sought values when the path to the optimal solution of the LMS algorithm goes through negative iterates \( w_n(l) \), i.e., when \( w_n(l) < 0 \) for some \( l \) and \( n \). Indeed, the fractional derivative (2.3) with \( a = 0 \) and \( t = w_n(l) \) will produce a complex denominator as \( w_n(l) - \tau < 0 \) for fractional exponent \( \alpha \in (0, 1) \). At first, it seems strange when the fractional derivative of a real-valued function of a real variable defined over a real domain turns out to be complex. However, a closer look demystifies that the definition of the left Riemann-Liouville derivative (2.3) is used out of its domain of definition \( \mathbb{R}^+ := (0, +\infty) \).

It is important to clarify here that there is no restriction \( \textit{a priori} \) on the choice of \( a \); it could assume any real value (positive or negative) with appropriate consideration in the update rule. However, whatever the choice of \( a \) is made, one must conform to the domain of definition of the derivative, i.e., \( w_n(l) > a \) to avoid getting complex outputs. Thus, for \( a = 0 \), the value \( w_n(l) < 0 \) does not conform to the domain of definition and consequently, the algorithm furnishes absurd complex outputs.

A very similar observation can be made for the update rules of the type (2.16). As we have specified in Section 3.1.2, the left Riemann-Liouville derivative \( w_{n-1}(l)^{\mathcal{L}} D_{w_n(l)}^\alpha \), defined through (2.3), was used to derive the iterate-update rule (2.16). Therefore, the tacitly made (but unannounced) Assumption (B1) was that
\[
w_{n-1}(l) < w_n(l), \quad \text{for all} \quad 0 \leq l \leq N - 1, \quad n \in \mathbb{N},
\]
that is, the sequence \((w_n(l))_{n \in \mathbb{N}}\) is pointwise monotone strictly increasing for every \( l \). However, in case if the optimal weight \( w^*(l) \) was negative, and the algorithm was initialized by \( 0 \leq w_0(l) < w_1(l) = 1 \), the sequence \((w_n(l))_{n \in \mathbb{N}}\) from (2.16) should be decreasing so that it could converge to \( w^*(l) < 0 \). On the other hand, if one assumes for the sake of argument that the algorithm is convergent, even then, a subsequence will be decreasing most likely since the LMS algorithm does not take a straight path towards the optimal solution, i.e., we may find ourselves in a situation where \( w_{n-1}(l) > w_n(l) \) for some \( l \) and \( n \). This observation is not restricted only to negative optimal weights. Indeed, if the positive weights are sought and we initialize the iterates with a positive value greater than the sought weight for some \( 0 \leq l \leq N - 1 \), we face the same issue. In all these situations, the algorithm will provide complex outputs. Towards this end, whenever \( w_n(l) < w_{n-1}(l) \),
\[
w_{n-1}(l)^{\mathcal{L}} D_{w_n(l)}^\alpha [w_n(l) - w_{n-1}(l)] = \frac{1}{\Gamma(1 - \alpha)} \frac{d}{dw_n(l)} \int_{w_{n-1}(l)}^{w_n(l)} \frac{\tau - w_{n-1}(l)}{(w_n(l) - \tau)^\alpha} d\tau
\[
= \frac{1}{\Gamma(1 - \alpha)} \frac{d}{dw_n(l)} \int_{w_{n-1}(l)}^{w_n(l)} \frac{w_{n-1}(l) - \tau}{(w_n(l) - \tau)^\alpha} d\tau.
\]
This indicates that \( \tau \in (w_n(l), w_{n-1}(l)) \) and, therefore, the denominator is complex because \( w_n(l) - \tau < 0 \). This can also be verified by the expression (2.5) in Theorem 2.5. Indeed, we have
\[
w_{n-1}(l)^{\mathcal{L}} D_{w_n(l)}^\alpha [w_n(l) - w_{n-1}(l)] = \frac{\Gamma(2)}{\Gamma(2 - \alpha)} (w_n(l) - w_{n-1}(l))^{1-\alpha} \in \mathbb{C}.
\]
We conclude once again that the left Riemann-Liouville derivative (2.3) is used outside of its domain of definition. In fact, the definition (2.3) was valid for $w_n(l) - w_{n-1}(l) \in \mathbb{R}^+$ but it was used for $w_n(l) - w_{n-1}(l) \in \mathbb{R}^-$. 

### 3.2.2 Remedies and Justification of the Absolute Value

We discuss the simple case of (2.12). The ensuing discussion is also valid for the case of (2.16) with $w_n(l)$ replaced by $w_n(l) - w_{n-1}(l)$ and $\frac{L}{0}D^\alpha_{w_n(l)}$ replaced by $w_{n-1}(l)L D^\alpha_{w_n(l)}$.

There are two possible ways to handle the problem of complex outputs. The first one is to identify a suitable lower limit $a$ such that

$$\min_l w^*(l) > a, \quad 0 \leq l \leq N - 1,$$

using à priori information about the system. Unfortunately, it is not practical to identify such a suitable lower limit $a$ in the broader context of adaptive signal processing and general applications. The other way is to use the emergence of the complex outputs in the existing setup (with $a = 0$) as an indication that one is going out of the domain of definition of the left-Riemann Liouville derivative, i.e.,

$$w_n(l) < 0 \quad \text{whenever} \quad \frac{L}{0}D^\alpha_{w_n(l)}[e_n^2] \in \mathbb{C} \quad \text{or equivalently} \quad \Im \left\{ \frac{L}{0}D^\alpha_{w_n(l)}[e_n^2] \right\} \neq 0. \quad (3.18)$$

Based on this criterion, one could take once again two possible routes discussed below.

1. We first find an appropriate lower limit $a < 0$ so that

$$\min_{l,n} w_n(l) > a, \quad 0 \leq l \leq N - 1, \quad n \in \mathbb{N}. \quad (3.19)$$

Once such a number ‘$a$’ is chosen, completely relaunch the entire algorithm by using the derivatives over the domain of definition interval $(a,w_n(l))$ in (2.3). Towards this end, one can set $a(n) := \min_l w_n(l)$. However, this choice may not work for all $n$ because $w_n$ does not take a straight path from a time instance $n$ to another time instance $n + 1$ to reach the optimal solution $w^*$. Moreover, it may necessitate relaunching of the algorithm multiple times that will significantly increase computation cost as well as the efficiency of the algorithm. So, it is not practically suitable to adopt this remedy.

2. The second way is to make use of the right Riemann-Liouville derivative (2.4) that has a variable lower limit and a fixed upper limit at $b$, i.e., it has domain of definition $(t,b)$. When the criterion (3.18) indicates that $w_n(l) < 0$, it will be suitable to apply right Riemann-Liouville derivative over the interval $(w_n(l),0)$ because $w_n(l)$ will conform well to its domain of definition. The idea is to use the appropriate definition of the fractional derivative according to each input $w_n(l)$. The issue here is that the criterion (3.18) needs to be verified on term-to-term bases, i.e., a decision has to be made for each and every $l$ and $n$, which is of course hectic. A quick fix to this is to identify what change will it bring to the update rule when we use right derivative (2.4) instead of the left one (2.3). Towards this end, it can be easily seen that when (2.4) is used, the update rule (2.12) will read as

$$w_{n+1}(l) = w_n(l) + \mu_x e_n x_{n-1} + \frac{\mu_x}{1(2-\alpha)} e_n x_{n-1} \left(-w_n(l)\right)^{1-\alpha}. \quad (3.20)$$
Therefore, the fixed algorithm that switches the definition of the derivative term-by-term according to the criterion (3.18) can be written by combining (2.12) and (3.20) as

\[
    w_{n+1}(l) = w_n(l) + \mu l e_n x_{n-l} + \frac{\mu f}{\Gamma(2-\alpha)} e_n x_{n-l} \begin{cases} 
        (w_n(l))^{1-\alpha}, & w_n(l) \geq 0, \\
        (-w_n(l))^{1-\alpha}, & w_n(l) < 0,
    \end{cases}
\]

or equivalently

\[
    w_{n+1}(l) = w_n(l) + \mu l e_n x_{n-l} + \frac{\mu f}{\Gamma(2-\alpha)} e_n x_{n-l} |w_n(l)|^{1-\alpha}.
\]

This justifies the use of an absolute value in the update rules (2.18) and (2.19). Hence, Assumptions (A1) and (B1) are no more restrictive if the absolute values are used in the iterate-update rules.

### 3.3 Geometric Evaluation of the Fractional Learning Algorithms

Let us now discuss the idea of using fractional derivatives in learning algorithms from a geometrical point of view. It will also help us understand the relationship, if there is any, between the so-called fractional critical points and the ordinary critical points of a function.

#### 3.3.1 Geometrical Interpretation of the Fractional Derivatives

The geometrical interpretation of a fractional derivative is still unsettled and debatable despite being a centuries-old concept. No generally acceptable geometric explanation has been provided yet since the appearance of the idea. Only a few vague interpretations have appeared so far that are far from being universally acceptable and practically functional. No solid connection is established in the literature between the fractional derivatives and the extreme or critical points of a sufficiently smooth function, unlike classical integer-order derivatives. It is well-known that the integer-order derivatives of a function \( \varphi : \mathbb{R} \rightarrow \mathbb{R} \) are, specifically, local (pointwise defined) and are linked to the geometry of \( \varphi \) and thus, have a clear geometrical meaning. Precisely, they provide suitable information about the behavior of the graph of \( \varphi \), e.g., the regions where \( \varphi \) is increasing, decreasing, concave, or the points where \( \varphi \) has extreme values, inflections, cusps, vertical tangent, and so on. On the other hand, the fractional derivatives are non-local being defined in terms of an improper integral and have so-called memory characteristics. Therefore, they provide very little punctual geometrical insight, at least regarding the behavior of the geometry of \( \varphi \). We invite the interested readers to go through the articles [28, 29, 39, 51, 52] for detailed discussions regarding the geometrical and physical interpretations of the fractional derivatives.

The conventional integer-order gradient vector has a geometrical and physical significance that has been vital in the success of the SDA. The gradient, \( \nabla \varphi \) of a smooth function \( \varphi \), defined by

\[
    \nabla u \varphi := \left( \frac{\partial \varphi}{\partial u_1}, \frac{\partial \varphi}{\partial u_2}, \ldots, \frac{\partial \varphi}{\partial u_N} \right)^\top \in \mathbb{R}^N, \quad \text{for any } u \in \mathbb{R}^N,
\]

points towards the direction in which \( \varphi \) assumes its most pronounced increase in the slope and its length effectively renders the value of that slope (see, e.g., Fig. 2). At each given point,
the SDA learns the direction of the steepest descent of the function by means of the gradient vector. In contrast, the fractional gradient, defined by

$$\nabla^\alpha \Phi := \begin{pmatrix} L_{a} D_{u_1} \Phi & L_{a} D_{u_2} \Phi & \cdots & L_{a} D_{u_N} \Phi \end{pmatrix}^\top \in \mathbb{R}^N,$$

for any $u \in \mathbb{R}^N$, (3.24)

points towards a direction other than that of the integer-order gradient. Therefore, it is impossible for it to point towards the direction of the most pronounced increase in the slope of $\varphi$, unless exponent $\alpha \to 1$, when $\nabla^\alpha \varphi \to \nabla \varphi$ consequently (see, Remark 2.4). Hence, the fractional SDA, learning the direction of the steepest descent through fractional gradients, cannot theoretically converge to an extreme point faster than the conventional SDA.

Figure 2: Illustration of the gradient vector and the tangent plane at point $(1, 2, 5)$ of the function $z = x^2 + y^2$.

### 3.3.2 Fractional Extreme Points

Let us now explore the connection (if there is any) between the extreme points of a function with the so-called fractional extreme points. We will also discuss the validity of Assumptions (A2) and (B2) about the constant terms and their influence on the fractional extreme points.

First of all, we note that the LMS iterate-update rule (2.10) renders an optimal solution $w^*$ minimizing the sample MSE (2.8) when the instantaneous gradient vector $\nabla_{w_n} \hat{E}(w_n) \to 0$. All such vectors $w_n \in \mathbb{R}^N$ for which $\nabla_{w_n} \hat{E}(w_n) = 0$ are the critical points of the quadratic objective function $\hat{E}(w_n) = \epsilon_n^2$, where $\epsilon_n = d_n - w_n^\top x_n$ is the transmission error. Since, $\hat{E}(w_n) = \epsilon_n^2$ is a hyper-paraboloid, it has a unique critical point for which it achieves a minimum value. On contrary, the fractional rule (2.11) renders an optimal solution $w_f^*$ that minimizes the sample MSE (2.8) when

$$\mu_f \nabla_{w_n} \hat{E}(w_n) + \mu_f \nabla_{w_n}^\alpha \hat{E}(w_n) = 0.$$

(3.25)

We introduced the subscript $f$ in $w_f^*$ in order to distinguish it from the classical optimal solution $w^*$, where $f$ stands for fractional. We will show in a while using a simple example (see, Example 3.3) that (3.25) is not generally satisfied by the optimal solution $w^*$. In other words, the fractional and conventional gradients have different critical points and thus, $w^*$ and $w_f^*$ are different.
For simplicity and clarity of our arguments in ensuing discussion, we fix $\mu_l = 0$ (i.e., only fractional gradient is used in fractional algorithms) and investigate the points where $\nabla_{w_n}^\alpha \mathcal{E}(w_n) = 0$. First, we consider the update rule (2.12) with $a = 0$. Evidently, it implies that

$$0 = D_{w_n(l)}^\alpha [c_n^2(w_n)] = D_{w_n(l)}^\alpha [(d_n - w_n^\top x_n)^2], \quad \text{for all } 0 \leq l \leq N - 1. \quad (3.26)$$

Consider the one-dimensional case (i.e., $N = 1$) for simplicity, and set $w_n = (w_n), x_n = (x_n) \in \mathbb{R}^1$. Then, from Theorem 2.5,

$$D_{w_n}^\alpha [(d_n - w_n x_n)^2] = d_n^2 \left( \frac{D_{w_n}^\alpha [1]}{\Gamma(1 - \alpha)} - 2d_n x_n \left( \frac{D_{w_n}^\alpha [w_n]}{\Gamma(2 - \alpha)} \right) + x_n^2 \left( \frac{D_{w_n}^\alpha [w_n^2]}{\Gamma(3 - \alpha)} \right) \right) \quad (3.28)$$

This furnishes a quadratic equation for critical points by setting $D_{w_n}^\alpha [(d_n - w_n x_n)^2] = 0$ as

$$2x_n^2 (w^*_n)^2 - 2(2 - \alpha) d_n x_n (w^*_n) + (2 - \alpha)(1 - \alpha) d_n^2 = 0. \quad (3.29)$$

Hence, there are two critical points for the fractional gradient given by

$$w^*_n = \frac{(2 - \alpha) d_n x_n \pm \sqrt{(2 - \alpha)^2 d_n^2 x_n^2 - 2(1 - \alpha)(2 - \alpha) d_n^2 x_n^2}}{2x_n^2}$$

$$= \frac{d_n}{2x_n} \left( 2 - \alpha \pm \sqrt{(2 - \alpha)^2 - 2(1 - \alpha)(2 - \alpha)} \right)$$

$$= \frac{d_n}{2x_n} \left( 2 - \alpha \pm \sqrt{\alpha(2 - \alpha)} \right). \quad (3.29)$$

It is worthwhile mentioning that the conventional integer-order gradient of the quadratic objective function $c_n^2$ has only one critical point $w^* = d_n / x_n$.

**Remark 3.2.** Following remarks are in order.

1. *The fractional gradient always has two distinct critical points $w_j^*$ for $\alpha \in (0, 1)$ and both of them are different from the critical point $w^*$ of the conventional gradient. Moreover, one of $w_j^*$ approaches to $w^* = d_n / x_n$ while the other approaches to 0 as $\alpha \to 1$.

2. *It is not clear which one of $w_j^*$ given in (3.29) would the fractional LMS algorithm converge to in practice. It appears that the fractional algorithm is highly sensitive to the choice of the initial guess, and that will dictate the convergence of the algorithm to any one of those critical points.*

3. *The roots $w_j^*$ given in (3.29) are not only dependent on the objective function but also on the choice of the exponents, i.e., if we vary $\alpha$ then the fractional critical points of the objective function also vary accordingly. Hence, the fractional exponent $\alpha$ induces significant deviations in the output of the fractional derivative by contributing to the steady-state error variability; instead of serving as an additional control parameter in the LMS algorithm for tailored convergence rate and enhancement of its performance.*
Therefore, the fractional critical points satisfy the equation
\[ x_n^2 w_n^2 - (2 - \alpha) d_n x_n w_n = 0. \] (3.30)

This, in turn, would have furnished a set of two critical points \( w^*_f = 0 \) and \( w^*_f = (2 - \alpha) d_n / x_n \). At one hand, it is clear that the constant terms have influence on the fractional critical point \( w^*_f \), unlike \( w^* \) (see, Example 3.3). At the other hand, removing the derivatives of the additive constant terms can be a blessing in disguise as one of the roots of (3.30) is always \( w^*_f = 0 \) while the other one is \( w^*_f = (2 - \alpha) d_n / x_n = (2 - \alpha) w^* \) that is much similar to the true critical point of the conventional gradient, especially, when \( \alpha \to 1 \).

We substantiate Remark 3.2 by a simple example below.

Example 3.3. We choose a quadratic function \( \varphi : [0, 1] \to \mathbb{R} \), defined by
\[ \varphi(t) := 2t^2 - t + c, \quad c \in \mathbb{R}. \] (3.31)

It is trivially known that the only critical point of \( \varphi \) is \( t^* = 1/4 \) irrespective of the choice of \( c \in \mathbb{R} \). Moreover, \( \varphi(t) \) has a minimum at \( t^* \) since \( \varphi''(1/4) > 0 \).

We know that the true minimum point is \( t^* = 1/4 > 0 \), so we can safely choose the lower limit of the fractional integral to be \( 0 \leq a < 1/4 \) (avoiding any breach of the domain of definition of fractional derivative) and look for fractional critical points \( t^*_f \) in the interval \( (a, t^*) \). Our aim is to elaborate on the role of \( a, c, \) and \( \alpha \) on \( t^*_f \).

We first express \( \varphi(t) \) as
\[ \varphi(t) = 2((t - a) + a)^2 - ((t - a) + a) + c = 2(t - a)^2 + (4a - 1)(t - a) + (2a^2 - a + c), \]
so that its left Riemann-Liouville derivative of order \( \alpha \in (0, 1) \) over \( (a, t) \) is
\[ L^\alpha D^\alpha_a (\varphi(t)) := \frac{4}{\Gamma(3 - \alpha)} (t - a)^{2-\alpha} + \frac{4a - 1}{\Gamma(2 - \alpha)} (t - a)^{1-\alpha} + \frac{2a^2 - a + c}{\Gamma(1 - \alpha)} (t - a)^{-\alpha} \]
\[ = \frac{(t - a)^{-\alpha}}{\Gamma(3 - \alpha)} \left( 4(t - a)^2 + (4a - 1)(t - a) + (2a^2 - a + c)(1 - \alpha)(2 - \alpha) \right). \] (3.32)

Therefore, the fractional critical points satisfy the equation
\[ 4(t^*_f - a)^2 + (4a - 1)(2 - \alpha)(t^*_f - a) + (2a^2 - a + c)(1 - \alpha)(2 - \alpha) = 0, \] (3.33)
and are given by
\[ t^*_f = a + \frac{(4a - 1)(2 - \alpha) \pm \sqrt{(4a - 1)^2(2 - \alpha)^2 - 4(2a^2 - a + c)(1 - \alpha)(2 - \alpha)}}{8}. \] (3.34)
After fairly easy manipulations, we get

\[
t_f^* = a + \frac{-(4a - 1)(2 - \alpha) \pm \sqrt{(2 - \alpha)[\alpha(4a - 1)^2 - 2(8c - 1)(1 - \alpha)]}}{8}.
\] (3.35)

One can draw following conclusions.

1. The fractional critical points depend on \( \alpha, a, \) and \( c \).

2. The points \( t_f^* \) are imaginary when

\[
c > \frac{2(1 - \alpha) + \alpha(4a - 1)^2}{16(1 - \alpha)},
\] (3.36)

i.e., there are no real points such that \( L_0^\alpha D_1^\alpha \varphi(t) = 0 \). We remind here that additive constant \( c \) had no role in the extreme values of the integer-order derivative. It substantiates our point that the additive constants have a significant influence on the optimal solution in the fractional LMS algorithm. Moreover, quadratic equation (3.30) is a particular case always having two real distinct roots for all \( \alpha \in (0, 1) \) thanks to a specific choice of \( c \) corresponding to the MSE objective functional.

3. Both the roots \( t_f^* \) are different from \( t^* = 1/4 \).

4. Finally, it is interesting that, for \( a = 0 \),

\[
L_0^\alpha D_1^\alpha[\varphi(t^*)] = \frac{(t^*)^{-\alpha}}{\Gamma(3 - \alpha)} \left[ 4(t^*)^2 - (2 - \alpha)t^* + c(1 - \alpha)(2 - \alpha) \right]
\]

\[
- \frac{4^\alpha}{\Gamma(1 - \alpha)} \left[ c - \frac{1}{4(1 - \alpha)} \right],
\] (3.37)

which is strictly negative at the true critical point \( t^* = 1/4 \) for all \( c < 1/4(1 - \alpha) \). Note that, there are two real values of \( t_f^* \) for such \( c \) since it avoids the condition (3.36). This justifies our claim that (3.25) does not hold in general at the true critical point. More precisely, the ordinary derivative at \( t^* \) is zero but the fractional derivative at \( t^* \) in (3.37) is non-zero (except for \( c = 1/4(1 - \alpha) \)), and therefore, the sum in (3.25) is also non-zero at \( t^* \).

We end this subsection with the following general result regarding the behavior of fractional derivatives at the true minimum that justifies the observation made in (3.37).

**Theorem 3.4** ([1, Theorem 2.4]). If \( \varphi : [0, 1] \to \mathbb{R} \) is a twice continuously differentiable function that attains its minimum at \( t^* \in (0, 1) \) then

\[
L_0^\alpha D_1^\alpha[\varphi(t^*)] \leq \frac{(t^*)^{-\alpha}}{\Gamma(1 - \alpha)} \varphi(t^*), \quad \alpha \in (0, 1).
\] (3.38)

### 3.3.3 Short-Memory Characteristic

It has been mentioned earlier that the fractional derivatives are non-local because they are defined in terms of integrals. Therefore, they have a so-called *long-memory* characteristic, i.e., they carry forward information of all the past states. As a way to deal with this non-locality
issue, iterate-update rules of the form (2.16) were designed by iterating the lower limit of the fractional derivative (2.3) for each time instance \( n \) so that only a short memory of the fractional derivative could be retained (the so-called short-memory characteristic). A natural choice of the variable limit was \( \alpha = a_{n,l} = w_{n-1}(l) \). Therefore, \( w_{n-1}(l)LD_w^\alpha \left[ \hat{\xi}(w_n) \right] \) was used in fractional LMS iterate-update rule (2.16). Below, we discuss the critical values corresponding to this fractional derivative to elaborate on the role of this short-term memory effect on the performance of the fractional LMS algorithms.

As in the Section 3.3.2, we consider the one-dimensional case for simplicity. Using the form (3.11) of the sample MSE, we evaluate the fractional derivative as

\[
\begin{align*}
\frac{d}{dn} L D_w^\alpha \left[ (d_n - w_n x_n)^2 \right] & = \Phi_n^2(0) \left( w_{n-1} L D_w^\alpha [1] \right) - 2 \Phi_n(0) x_n \left( w_{n-1} L D_w^\alpha [\Delta w_n] \right) + x_n^2 \left( w_{n-1} L D_w^\alpha [(\Delta w_n)^2] \right) \\
& = \frac{\Phi_n^2(0)}{\Gamma(1 - \alpha)} [\Delta w_n]^{-\alpha} - \frac{2 \Phi_n(0) x_n}{\Gamma(2 - \alpha)} [\Delta w_n]^{1 - \alpha} + \frac{2x_n^2}{\Gamma(3 - \alpha)} [\Delta w_n]^{2 - \alpha} \\
& = \frac{[\Delta w_n]^{-\alpha}}{\Gamma(3 - \alpha)} \left( (2 - \alpha)(1 - \alpha) \Phi_n^2(0) - 2(2 - \alpha) \Phi_n(0) x_n [\Delta w_n] + 2x_n^2 [\Delta w_n]^2 \right),
\end{align*}
\]

(3.39)

where \( \Phi_n(l) \) is given in (3.12) and \( \Delta w_n := (w_n - w_{n-1}) \). This furnishes a quadratic equation for critical points \( w_f^* \) by setting the fractional derivative in (3.39) to zero, i.e.,

\[
2x_n^2 (w_f^* - w_{n-1})^2 - 2(2 - \alpha) \Phi_n(0) x_n (w_f^* - w_{n-1}) + (2 - \alpha)(1 - \alpha) \Phi_n^2(0) = 0.
\]

(3.40)

Remark that the quadratic equation (3.40) is essentially equivalent to (3.28) with \( w_f^* \) and \( d_n \) replaced with \( w_f^* - w_{n-1} \) and \( \Phi_n^2(0) \), respectively. Hence, there are two critical points for the fractional gradient given by

\[
w_{f,\pm,n} = w_{n-1} + \frac{d_n - w_{n-1} x_n}{2x_n} \left( (2 - \alpha) \pm \sqrt{(2 - \alpha)^2} \right),
\]

(3.41)

and the conclusions drawn in Remark 3.2 in Section 3.3.2 are also relevant to the algorithms designed with short memory characteristics. However, now there are two sequences of critical points \( (w_{f,\pm,n}) \) (as the fractional derivative changes each time due to the variable lower limit \( a_{n,l} = w_{n-1}(l) \)).

There is no apparent big difference between the algorithm derived using short memory characteristic due to variable initial terms and the one without it. However, this is not true. The difference lies with the convergence guarantee and the steady state error produced by the two variants. As will be discussed in Section 3.4, the fractional part of the algorithm (2.18) approaches zero as \( n \to \infty \) and the convergence of the algorithm is achieved thanks to the integer-order gradient part. The fractional part merely contributes to the steady state error as it seldom becomes zero at the true extreme point in finite time (see Observation 4 in Example 3.3 and Theorem 3.4). On contrary, the short memory algorithms of the form (2.19) usually do not have integer gradient part (i.e., \( \mu_l = 0 \)), but they have guaranteed convergence to the Wiener solution in stationary cases under the assumption of the convergence of the LMS solution thanks to the short memory characteristic; see, Proposition 3.5. However, their rate of convergence is slower than the standard LMS algorithm due to their construction under Assumption (B2); see Remark 3.1 and Example 3.6. Moreover, the critical points corresponding to the rule (2.19), \( w_{f,\pm,n} \), both approach to \( w^* \) as \( n \to \infty \) for any \( \alpha \in (0, 1) \), unlike those corresponding to (2.18).
3.4 Convergence Analysis

Let us first discuss the convergence of the algorithm (2.19).

Proposition 3.5. Under the assumption that the sequence \((w_n)_{n \in \mathbb{N}}\) given by the update rule (2.10) converges to the Wiener solution \(w^*\), the sequence \((w_{f, \pm, n})_{n \in \mathbb{N}}\) also converges to \(w^*\).

Proof. We prove the result for one-dimensional case. We refer to [17] for more detailed discussions on the general case. Let \(\lim_{n \to \infty} w_n = w^*\). Then, it can be seen that the sequence \((w_{f, \pm, n})_{n \in \mathbb{N}}\) also converges to \(w^*\). Indeed, from (3.41),

\[
\lim_{n \to \infty} w_{f, \pm, n} = \lim_{n \to \infty} w_{n-1} + \lim_{n \to \infty} \frac{d_n - w_{n-1}x_n}{2x_n} \left((2 - \alpha) \pm \sqrt{\alpha(2 - \alpha)}\right) = w^*. \tag{3.42}
\]

As we mentioned in the previous section, we substantiate our claim that the convergence of the algorithm (2.19) is slower than the standard LMS algorithm through a simple example below.

Example 3.6. We choose a quadratic function \(\varphi_2 : [0, 1] \to \mathbb{R}\), defined by

\[
\varphi_2(t) := (2t - 3)^2, \tag{3.43}
\]

that has the only critical point \(t^* = 3/2\) where it is minimum. In Fig. 3, we show the performance of the fractional algorithm

\[
t_{n+1} = t_n - \frac{\mu f}{2L} x_{n-1} D_{t_n}^\alpha [\varphi_2(t_n)]. \tag{3.44}
\]

It can be observed that the solution converges to the true extreme point \(t^* = 3/2\). However, the convergence is very slow as anticipated in Remark 3.1.

Let us now discuss the convergence of the algorithm (2.18). Towards this end, we have the following lemma.
Lemma 3.7. If the sequence \((w_n)_{n \in \mathbb{N}}\) given by the update rule (2.10) converges to the Wiener solution \(w^*\) then the sequence \((u_n)_{n \in \mathbb{N}} \subset \mathbb{R}^N\), defined by
\[
u_n(l) := \mu f \epsilon_n x_{n-\ell} |w_n(l)|^{1-\alpha}, \quad 0 \leq l \leq N - 1, \tag{3.45}
\]
is also convergent and \(u_n \to 0 \in \mathbb{R}^N\).

Proof. If \(w_n\) converge to \(w^*\) as \(n \to +\infty\), then \(w_n(l) \to w^*(l)\) for each \(l\). Recall that every convergent sequence is bounded, therefore, there exist a number \(M \in \mathbb{R}^+\) such that
\[|w_n(l)| \leq M\]
for all \(0 \leq l \leq N - 1\) and \(n \in \mathbb{N}\). Let \(\epsilon > 0\) be arbitrary. By convergence of \(w_n(l) \to w^*(l)\), there exists a natural number \(N_\epsilon\) such that
\[
|w_k(l) - w^*(l)| < \epsilon \left(\frac{\mu \ell}{2\mu f M^{1-\alpha}}\right), \quad \text{for all } k > N_\epsilon + 1. \tag{3.46}
\]
Note also that, by definition of the general term given in (2.10), we have
\[
|\mu \ell e_k x_{k-l}| = |w_k(l) - w_{k-1}(l)| \leq |w_k(l) - w^*(l)| + |w_{k-1}(l) - w^*(l)| < \epsilon \left(\frac{\mu \ell}{\mu f M^{1-\alpha}}\right), \quad \text{for all } k > N_\epsilon. \tag{3.47}
\]
Thus, from (3.46) and (3.47),
\[
|\nu_k(l)| = |\mu \ell e_k x_{k-l}| |w_k|^{1-\alpha} \leq \frac{\mu f}{\mu \ell} |\mu \ell e_k x_{k-l}| M^{1-\alpha} < \epsilon, \quad \text{for all } k > N_\epsilon. \tag{3.48}
\]
This completes the proof. \(\Box\)

Theorem 3.8. Under the assumption of Lemma 3.7, the sequence \((v_n)_{n \in \mathbb{N}}\) given by \(v_n := w_n + u_n\), generated by algorithm (2.18), converges to \(w^*\).

Proof. Simply consider \((v_n)_{n \in \mathbb{N}}\) by \(v_n(l) := w_n(l) + u_n(l)\) where \(w_n(l)\) and \(u_n(l)\) are as in Lemma 3.7. Then
\[
\lim_{n \to \infty} v_n = \lim_{n \to \infty} w_n + \lim_{n \to \infty} u_n = w^* + 0 = w^*. \tag{3.49}
\]
\(\Box\)

Theorem 3.8 substantiates that the convergence of the algorithm (2.18) is achieved due to the integer-order gradient factor. For finite number of iterations, the fractional term is not going to vanish no matter how small it may be and, therefore, contribute to the steady state error. In the next section, we conduct a few numerical experiments to support our findings.

4 Numerical Simulations and Discussions

To compare the performance of the fractional and integer-order gradient-based LMS algorithms, we consider three evaluation protocols:

(i) a system with negative desired weights,
\[
w = \begin{bmatrix} -15, & -14, & \cdots, & -2, & -1 \end{bmatrix},
\]
derived under both noise-free and noisy environments;
(ii) a system with positive desired weights,

\[ \mathbf{w} = [1, 2, \cdots, 14, 15] \]

under both noise-free and noisy environments;

(iii) a system with random desired weights under noisy environment using modified weight-update rules (2.18) and (2.19).

4.1 Experimental Setup

In the rest of Section 4, we consider the noisy environments with signal-to-noise ratio (SNR) of 10dB. The LMS and its fractional-order variants are configured to equal performance at \( \alpha = 1 \). The performance of the fractional LMS algorithms is observed for fractional exponents \( \alpha = 0.9, 0.8, 0.7, 0.6, 0.5, \) and \( 0.4 \). For all experiments, the step-size of LMS was fixed as \( \mu_l = 1 \times 10^{-2} \), the step-sizes of fractional LMS algorithm (2.18) were fixed as \( \mu_l = 5 \times 10^{-3} \) and \( \mu_f = 5 \times 10^{-3} \), and the step-size of fractional LMS algorithm (2.19) was fixed as \( \mu_f = 1 \times 10^{-2} \).

Protocols (i) and (ii) are used to substantiate that the iterate-update rule (2.12) (without modulus) is affected by complex outputs whether positive or negative weights are sought and, consequently, the fractional algorithm is divergent. Similar experimental trends can be delineated for the second iterate-update rule (2.16). Protocol (iii) is used for modified iterate-update rules (2.18) and (2.19) (with modulus). For system input, we have considered a random signal of length 1000 obtained from a zero-mean Gaussian distribution with unit variance. The experiments are repeated for 1000 independent rounds and mean results are reported. For each independent round, the weights were initialized with zeros except in algorithm (2.19) where weights were initialized randomly from a Gaussian distribution with unit variance. The performance of all the algorithms is evaluated on mean deviation (MD) which is the \( \ell_1 \)-norm of the difference between the sought and the approximated weights, i.e.,

\[ \Delta \mathbf{w}_n = \frac{\| \mathbf{w}_n - \hat{\mathbf{w}}_n \|_{\ell_1}}{N}, \]

where \( \mathbf{w}_n \) and \( \hat{\mathbf{w}}_n \) are the sought and approximated weight vectors at \( n \)th iteration, respectively. Here, \( \| \cdot \|_{\ell_1} = \| \cdot \| \) is the \( \ell_1 \)-norm and \( N \) is the length of the filter vector.

4.2 Complex Outputs and Divergence

Figures 4 and 5 show the learning curves for the LMS algorithm and the fractional LMS algorithm (2.12) for evaluation protocols (i) and (ii), respectively. For both protocols, we set up both algorithms on an equal convergence rate and compare their steady-state performance. It can be observed that the fractional LMS algorithm (2.12) failed to identify the system with negative (Fig. 4) as well as positive weights (Fig. 5) for all the listed values of \( \alpha \). This conforms to our theoretical findings in Section 3.2. Similar results hold for the fractional LMS algorithm with iterate-update rule (2.16).

4.3 Performance of the Modified Algorithms

For evaluating the performance of the iterate-update rules (2.18) and (2.19), we choose the random desired weights. The desired weight vector is a random signal of length 30 obtained from a zero-mean Gaussian distribution with a variance of 1. For each run, new desired weights
Figure 4: Protocol (i): Learning curves for different values of fractional power for noise-free (left) and noisy (right) environments.

Figure 5: Protocol (ii): Learning curves for different values of fractional power for noise-free (left) and noisy (right) environments.

were selected. We set up both algorithms at an equal convergence rate and compared the steady-state performance. Figure 6 shows the learning curves for the LMS and fractional LMS algorithms (2.18) and (2.19).

It can be seen from Fig. 6(a) that the fractional algorithm (2.18) shows comparable results with the LMS algorithm but its convergence rate is relatively low for all the listed values of $\alpha$. Based on the experimental result, we can conclude that there is no noticeable gain in using fractional LMS algorithm (2.18) while its computational complexity is higher than the LMS algorithm.

Figure 6(b) indicates that the learning rate of the fractional algorithm (2.19) is extremely low as compared to the LMS due to the fractional term $|w_n(l) - w_{n-1}(l)|^{1-\alpha}$. This phenomenon is correlated to the Assumption (B3) as well as the stochastic nature of the problem in the protocol (iii). For the stationary case discussed in Example 3.6, the algorithm was performing reasonably better than the stochastic problem in the protocol (iii). On the other hand, the difference $|w_n(l) - w_{n-1}(l)|$ is decreasing as the algorithm is progressing, therefore, the learning
rate is decreasing significantly. To elaborate further on this, we perform the same experiment with $|w_n(l) - w_{n-1}(l) + \epsilon|^{1-\alpha}$ instead of $|w_n(l) - w_{n-1}(l)|$ in the iterate-update rule with bias compensation parameter $\epsilon = 1 \times 10^{-10}$. We plot the learning curves again in Fig. 6(c) where we can see that the learning rate of the algorithm is improved. However, the algorithm is still not converging at a rate better than the LMS. We can conclude based on these experiments that there is no gain in using the fractional algorithm (2.19) instead of the conventional LMS algorithm.

![Graphs](image)

Figure 6: Protocol (iii): Learning curves for different values of fractional exponent for random weights under noisy environment.

5 Conclusion

In this article, we have rigorously analyzed the performance of the fractional learning algorithms. We have discussed the schematic kinks of the fractional learning algorithms and proposed their remedies. The following are our key observations.

1. The fractional gradients are commonly used in literature out of their domain of definition. This is the reason for complex outputs and divergence of the fractional learning algorithms both for negative and positive sought weights. The use of absolute values in the iterate-update rules to avoid complex outputs can be justified and is a ready-witted remedy.

2. The fractional gradients in stationary cases do not point opposite to the steepest descent and, therefore, cannot lead to the steepest path towards the minimizer of the mean square error. Therefore, their convergence rate cannot be faster than the steepest gradient descent algorithm. Numerical experiments suggest an identical trend in the stochastic case with a possibility of fractional learning algorithms having comparable convergence performance as of the LMS algorithm due to the stochastic nature of the instantaneous gradient.

3. Since, the geometric interpretation of the fractional derivatives is almost unknown, it is difficult to associate fractional derivatives to the extreme values of a function. In particular, the fractional derivatives are usually nonzero at the true critical point of the mean square error. The fractional critical point is non-unique, dependent on fractional exponents, additive constants, and chosen domain of definition for the fractional derivatives (or roughly the interval in which the solution is sought).
4. The analysis suggests that the performance of the fractional learning algorithms is dependent on the initial guess. In the stationary case, we have proved that the representative fractional iterate-update rules with modulus do converge but not necessarily to the Wiener solution. Their steady-state error, convergence rate, and computational cost cannot be better than the conventional steepest gradient descent. In the stochastic case, numerical simulations indicate that the performance of the fractional algorithms is comparable to that of the LMS algorithm at the best and that at the price of higher computational cost and higher steady-state error.

5. A rigorous stochastic study of the fractional algorithms requires a complete understanding of the statistical distribution of the fractional iterate-updates, which is still an open question.

Based on the rigorous mathematical analysis and numerical experiments performed in this article and the observations above, we conclude that the fractional learning algorithms cannot outperform the conventional integer-order learning algorithms even if their schematic kinks are removed. The fractional learning algorithms have higher computational costs, higher steady-state error, and relatively lower (or comparable at best) convergence rates than their conventional counterparts. Our conclusions conform to those drawn by Bershad, Wen, and So [7] using a comprehensive numerical study.

References

[1] M. Al-Refai, On the fractional derivatives at extreme points, Electron. J. Qual. Theory Differ. Equ., 55:(2012), pp. 1–5.

[2] M. S. Aslam, Comments on “Two-stage fractional least mean square identification algorithm for parameter estimation of CARMA systems”, Signal Process., 117: (2015), pp. 279–280.

[3] M. S. Aslam, Comments on “Design of fractional adaptive strategy for input nonlinear Box-Jenkins systems”, Signal Process., 119:(2016), pp. 169–173.

[4] M. S. Aslam and M. A. Z. Raja, A new adaptive strategy to improve online secondary path modeling in active noise control systems using fractional signal processing approach, Signal Process., 107: (2015), pp. 433–443.

[5] M. G. Bellanger, Adaptive Digital Filters and Signal Analysis, 2nd edn., Marcel Dekker Inc., New York, 2001.

[6] J. C. M. Bermudez and N. J. Bershad, A nonlinear analytical model for the quantized LMS algorithm: The arbitrary step size case, IEEE Trans. Signal Process., 44: (1996), pp. 1175–1183.

[7] N. J. Bershad, F. Wen, and H. C. So, Comments on “Fractional LMS algorithm”, Signal Process., 133:(2017), pp. 219–226.

[8] N. I. Chaudhary, R. Latif, M. A.Z. Raja, and J. A. T. Machado, An innovative fractional order LMS algorithm for power signal parameter estimation, Appl. Math. Model., 83: (2020), pp. 703–718.
[9] N. I. Chaudhary, M. A. Manzar, and M. A. Z. Raja, Fractional Volterra LMS algorithm with application to Hammerstein control autoregressive model identification, *Neural. Comput. Appl.*, 31(9): (2019), pp. 5227–5240.

[10] N. I. Chaudhary and M. A. Z. Raja, Identification of Hammerstein nonlinear ARMAX systems using nonlinear adaptive algorithms, *Nonlinear Dyn.*, 79(2): (2015), pp. 1385–1397.

[11] N. I. Chaudhary and M. A. Z. Raja, Design of fractional adaptive strategy for input nonlinear Box-Jenkins systems, *Signal Process.*, 116: (2015), pp. 141–151.

[12] N. I. Chaudhary, M. A. Z. Raja, M. S. Aslam, and N. Ahmed, Novel generalization of Volterra LMS algorithm to fractional order with application to system identification, *Neural. Comput. Appl.*, 29(6): (2018), pp. 41–58.

[13] N. I. Chaudhary, M. A. Z. Raja, and A. U. R. Khan, Design of modified fractional adaptive strategies for Hammerstein nonlinear control autoregressive systems, *Nonlinear Dyn.*, 82(4): (2015), pp. 1811–1830.

[14] N. I. Chaudhary, S. Zubair, M. S. Aslam, M. A. Z. Raja, and J. A. T. Machado, Design of momentum fractional LMS for Hammerstein nonlinear system identification with application to electrically stimulated muscle model, *Eur. Phys. J. Plus*, 134(8): (2019), 407.

[15] N. I. Chaudary, S. Zubair, and M. A. Z. Raja, A new computing approach for power signal modeling using fractional adaptive algorithms, *ISA Trans.*, 68: (2017), pp. 189–202.

[16] N. I. Chaudhary, S. Zubair, M. A. Z. Raja, and N. Dedovic, Normalized fractional adaptive methods for nonlinear control autoregressive systems, *Appl. Math. Model.*, 66: (2019), pp. 457–471.

[17] Y. Chen, Q. Gao, Y. Wei, and Y. Wang, Study on fractional order gradient method, *Appl. Math. Comput.*, 314: (2017), pp. 310–321.

[18] L. Chen, T. Huang, J. A. T. Machado, A. M. Lopes, Y. Chai, and R. Wu, Delay-dependent criterion for asymptotic stability of a class of fractional-order memristive neural networks with time-varying delays, *Neural Netw.*, 118: (2019), pp. 289–299.

[19] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, Quantized kernel least mean square algorithm, *IEEE Trans. Neural Netw. Learn. Syst.*, 23 (1): (2012), pp. 22–32.

[20] S. Cheng, Y. Wei, Y. Chen, S. Liang, and Y. Wang, A universal modified LMS algorithm with iteration order hybrid switching, *ISA Trans.*, 67: (2017), pp. 67–75.

[21] S. Cheng, Y. Wei, D. Sheng, Y. Chen, and Y. Wang, Identification for Hammerstein nonlinear ARMAX systems based on multi-innovation fractional order stochastic gradient, *Signal Process.*, 142: (2018), pp. 1–10.

[22] S. Cheng, Y. Wei, Y. Chen, Y. Li, and Y. Wang, An innovative fractional order LMS based on variable initial value and gradient order, *Signal Process.*, 133:(2017), pp. 260–269.

[23] R. Cui, Y. Wei, Y. Chen, S. Cheng, and Y. Wang, An innovative parameter estimation for fractional-order systems in the presence of outliers, *Nonlinear Dyn.*, 89(1): (2017), pp. 453–463.
[24] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Springer, New York, 2013.

[25] P. S. R. Diniz and L. W. Biscainho, Optimal variable step size for the LMS/Newton algorithm with application to subband adaptive filtering, *IEEE Trans. Signal Process.*, 40: (1992), pp. 2825–2829.

[26] P. S. R. Diniz and B. Widrow, History of adaptive filters, in *A Short History of Circuits and Systems*, eds. F. Maloberti and A. C. Davies, River Publishers, Delft, 2016.

[27] S. Haykin, *Adaptive Filter Theory*, 5th ed., Pearson, 2014.

[28] R. Herrmann, Towards a geometric interpretation of generalized fractional integrals - Erdélyi-Kober type integrals on $\mathbb{R}^N$, as an example, *Fract. Calc. Appl. Anal.*, 17(2): (2014), pp. 361–370.

[29] R. Hilfer, Mathematical and physical interpretations of fractional derivatives and integrals, in *Handbook of Fractional Calculus: Basic Theory*, vol. 1, ch. 3, pp. 47–86, de Gruyter, Berlin (2019), ISBN: 978-3-11-057162-2.

[30] A. Khalili, A. Rastegarnia, and S. Sanei, Quantized augmented complex least-mean square algorithm: Derivation and performance analysis, *Signal Process.*, 121:(2016), pp. 54–59.

[31] Z. A. Khan, N. I. Chaudhary, and S. Zubair, Fractional stochastic gradient descent for recommender systems, *Electron. Mark.*, 29(2): (2019), pp. 275–285.

[32] Z. A. Khan, S. Zubair, H. Alquhayz, M. Azeem, and A. Ditta, Design of momentum fractional stochastic gradient descent for recommender systems, *IEEE Access*, 7: (2019), pp. 179575–179590.

[33] Z. A. Khan, S. Zubair, N. I. Chaudhary, M. A. Z. Raja, F. A. Khan, and N. Dedovic, Design of normalized fractional SGD computing paradigm for recommender systems, *Neural Comput. Appl.*, 32: (2020), pp. 10245–10262.

[34] S. Khan, A. Wahab, I. Naseem, and M. Moinuddin, Comments on “Design of fractional-order variants of complex LMS and NLMS algorithms for adaptive channel equalization”, *Nonlinear Dyn.* 101:(2020), 1053–1060.

[35] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier, Amsterdam, 2006.

[36] J. I. Nagumo and A. Noda, A learning method for system identification, *IEEE Trans. Autom. Control*, 12 (3): (1967), 282–287.

[37] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, Transform domain LMS algorithm, *IEEE Trans. Acoust. Speech Signal Process.*, 31: (1983), pp. 609–615.

[38] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, CA, 1999.

[39] I. Podlubny, Geometric and physical interpretation of fractional integration and fractional differentiation, *Fract. Calc. Appl. Anal.*, 5(4): (2002), pp. 367–386.

[40] Y. F. Pu, J. L. Zhou, Y. Zhang, N. Zhang, G. Huang, and P. Siarry, Fractional extreme value adaptive training method: fractional steepest descent approach, *IEEE Trans. Neural Netw. Learn. Syst.*, 26(4): (2013), pp. 653–662.
M. A. Z. Raja and N. I. Chaudhary, Two-stage fractional least mean square identification algorithm for parameter estimation of CARMA systems, *Signal Process.*, 107: (2015), pp. 327–339.

M. A. Z. Raja and I. M. Qureshi, A modified least mean square algorithm using fractional derivative and its application to system identification, *Eur. J. Sci. Res.* 35(1): (2009), pp. 14–21.

S. M. Shah, R. Samar, N. M. Khan, and M. A. Z. Raja, Design of fractional-order variants of complex LMS and NLMS algorithms for adaptive channel equalization, *Nonlinear Dyn.*, 88(2): (2017), pp. 839–858.

S. M. Shah, R. Samar, and M. A. Z. Raja, Fractional-order algorithms for tracking Rayleigh fading channels, *Nonlinear Dyn.*, 92(3): (2018), pp. 1243–1259.

S. M. Shah, R. Samar, M. A. Z. Raja, and J. A. Chambers, Fractional normalized filtered error least mean square algorithm for application in active noise control systems, *Electron. Lett.*, 50(14): (2014), pp. 973–975.

D. Sheng, Y. Wei, Y. Chen, and Y. Wang, Convolutional neural networks with fractional order gradient method, *Neurocomputing*, 408: (2020), pp. 42–50.

B. Shoaib and I. M. Qureshi, A modified fractional least mean square algorithm for chaotic and nonstationary time series prediction, *Chin. Phys. B*, 23(3): (2014), 030502.

D. T. Slock, On the convergence behavior of the LMS and normalized LMS algorithms, *IEEE Trans. Signal Process.*, 40: (1993), pp. 2811–2825.

R. A. Soni, K. A. Gallivan, and W. K. Jenkins, Low-complexity data-reusing methods in adaptive filtering, *IEEE Trans. Signal Process.*, 52: (2004), pp. 394–405.

V. E. Tarasov, On chain rule for fractional derivatives, *Commun. Nonlinear Sci. Numer. Simulat.*, 30:(2016), pp. 1–4.

V. E. Tarasov, Geometric interpretation of fractional-order derivative, *Fract. Calc. Appl. Anal.*, 19(5): (2016), pp. 1200–1221.

M. H. Tavassoli, A. Tavassoli, and M. R. Ostad Rahimi, The geometric and physical interpretation of fractional order derivatives of polynomial functions, *Differ. Geom.-Dyn. Syst.*, 15: (2013), pp. 93–104.

A. Wahab and S. Khan, Comments on “Fractional extreme value adaptive training method: Fractional steepest descent approach”, *IEEE Trans. Neural Netw. Learn. Syst.*, 31(3): (2020), pp. 1066–1068.

A. Wahab and S. Khan, Comments on “Generalization of the gradient method with fractional order gradient direction”, arXiv:2009.05221v1, 2020.

A. Wahab, S. Khan, and F. Z. Khan, Comments on “Design of momentum fractional LMS for Hammerstein nonlinear system identification with application to electrically stimulated muscle model”, *Eur. Phys. J. Plus*, 136: (2021), 1004.

A. Wahab, S. Khan, and F. Z. Khan, Comments on “A new computing approach for power signal modeling using fractional adaptive algorithms”, arXiv:2003.09597, 2020.
[57] J. Wang, Y. Wen, Y. Gou, Z. Ye, and H. Chen, Fractional-order gradient descent learning of BP neural networks with Caputo derivative Neural Netw., 89: (2017), 19–30.

[58] J. Wang, G. Yang, B. Zhang, Z. Sun, Y. Liu, and J. Wang, Convergence analysis of Caputo-type fractional order complex-valued neural networks, IEEE Access, 5: (2017), pp. 14560–14571.

[59] Y. Wei, Y. Kang, W. Yin, and Y. Wang, Generalization of the gradient method with fractional order gradient direction, J. Franklin Inst., 357(4):(2020), pp. 2514–2532.

[60] B. Widrow, J. McCool, and M. Ball, The complex LMS algorithm, Proc. IEEE, 63: (1975), pp. 719-720.

[61] B. Widrow and M. E. Hoff, Adaptive switching circuits, WESCOM Conv. Rec. 4, pp. 96–140, (1960).

[62] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson Jr., Stationary and non-stationary learning characteristics of the LMS adaptive filters, Proc. IEEE, 64:(1976), pp 1151–1162.

[63] B. Widrow and D. Park, History of adaptive signal processing: Widrow’s group, in A Short History of Circuits and Systems, eds. F. Maloberti and A. C. Davies, River Publishers, Delft, 2016.

[64] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice Hall, Englewood Cliff, 1985.

[65] F. F. Yassa, Optimality in the choice of convergence factor for gradient based adaptive algorithms, IEEE Trans. Acoust. Speech Signal Process., 35: (1987), pp. 48–59.

[66] W. Yin, S. Cheng, Y. Wei, J. Shuai, and Y. Wang, A bias-compensated fractional order normalized least mean square algorithm with noisy inputs, Numer. Algorithms, 82(1): (2019), pp. 201–222.

[67] W. Yin, Y. Wei, T. Liu, and Y. Wang, A novel orthogonalized fractional order filtered-x normalized least mean squares algorithm for feed-forward vibration rejection, Mech. Syst. Signal Process., 119: (2019), pp. 138–154.

[68] H. Zhu, Z. Wu, C. Yang, T. Peng, Z. Chen, and X. Yang, Fractional steepest ascent method for TCU fault detection, IFAC-PapersOnLine, 51(24): (2018), pp. 1336–1342.

[69] S. Zubair, N. I. Chaudhary, Z. A. Khan, and W. Wang, Momentum fractional LMS for power signal parameter estimation, Signal Process., 142: (2018), pp. 441–449.