Design of fractional-order variants of complex LMS and NLMS algorithms for adaptive channel equalization

Syed Muslim Shah · Raza Samar · Noor M. Khan · Muhammad Asif Zahoor Raja

Abstract Equalization filtering is an effective technique applied to minimize the inter-symbol interference (ISI) in multipath fading channels; the problem gets worse for higher-order constellations which are required for high data rates in today’s communication systems. The least mean square (LMS) filter is a computationally efficient and easily implementable algorithm but suffers from slow convergence; highly complex filters are required to nullify the effects of ISI. In this paper, we develop complex modified fractional-order (FO) nonlinear variants of the LMS and the NLMS algorithms and apply in adaptive channel equalization, in both feed-forward and decision feedback configurations. In addition to the standard first-order derivative, the update in the modified LMS also depends on the FO derivative of the mean square error, the final update is formed using a combination of conventional update term and a nonlinear term obtained through Riemann–Liouville fractional derivative. The step size of the FNLMS scheme in fractional part is not only a function of the input energy but also the FO. The differintegral operator working as differentiator helps improve the convergence rate because the algorithm becomes nonlinear; the fractional algorithms provide more parameters to control the rate of convergence and have simple implementation with almost similar complexity. The performances of the schemes are validated through extensive simulation results for block fading channels (frequency flat and selective) to evaluate the symbol error rate for higher-order quadrature amplitude modulation schemes, mean square error and combined channel and equalizer responses to show the improved inverse modeling of the channel. Simulation experiments confirm the superiority of the proposed algorithm over the traditional counterparts.

Keywords Adaptive equalization · Fading channels · Fractional derivatives · Least means square

1 Introduction

An adaptive equalizer is an essential part of every digital receiver whether wired or wireless and applied to minimize the effects of frequency-selective channels [1,2] as the latter cause inter-symbol interference (ISI). In the training mode, a known sequence of symbols is sent by the transmitter during which the equalizer filter adjusts its coefficients appropriately until the symbols in the sequence are received without ISI [3]. Usu-
ally, the training length and frame size are decided in such a way that the equalizer weights remain in the optimum setting during data transmission phase. The design objective is to use less number of training symbols while achieving acceptable steady state performance in decision-directed mode. Various adaptive algorithms such as least mean square (LMS), Normalized LMS, sparse least mean fourth [4,5], bias compensated [6] and variable tap length equalizers [7], recursive least squares (RLS), Kalman filtering [8,9], self-iterating soft equalizers based on maximum likelihood optimization [10] and hybrid approaches [11] have been applied and analyzed for the equalization problem. These algorithms either have convergence and tracking problems [1,12–14] or the computationally complexity is too high [13] to be implemented.

Recently, fractional calculus has been applied in control and signal processing applications. In a fractional-order (FO) system, inputs and outputs of the system are related through differential equations having non-integer orders, which can be positive, negative and even complex [15]. Recent applications in control include FO control of thermal systems [16], discrete and stochastic methods for FO optimal control problems [17,18], fraction dynamics [19], fractional flow dynamics characterized by Burger’s equation [20], embedded applications of FO control in [21] and operational matrices along with spectral techniques discussed in FC [22]. In fractional signal processing techniques such as digital Riesz and its adjustable FO differentiators [23,24] and the other recent works [25–28] are for discrete-time fixed filters. Similarly design of analog variable FO differentiator and integrator discussed in [29], transfer function inversion in [30] and discrete-time differential systems and identification [31–33]. The development of corresponding adaptive filters is still in its infancy; in [34] the performance of filtered-x LMS (FxLMS) and filtered-x fractional LMS (FLMS) has been compared for the application in active noise control, and a fractional normalized version of filtered-error LMS (FN-FeLMS) has been developed in [35]; reference [36] provides further extension four variants of the FxLMS from the FN-FeLMS in [35]. Effectiveness of the FLMS algorithm has been shown in parameter estimation problem for Box–Jenkins [37] and two-stage control autoregressive moving average systems [38]. In [39], an FO variant of the LMS algorithm has been developed, and it utilizes the Caputo definition of fractional derivative [15,40] in which the FO is an alternate option to step size; in [41], an adaptive mechanism has been developed for the FLMS and has been used in the prediction of chaotic time series. In communications signal processing, an FO variant has been developed in [42] for Gordad-based blind equalizer which exhibits better performance than its conventional counterparts. In these cited works which deals fractional adaptive strategies [34–41], FO signal processing (FOSP) techniques are limited to system identification problems while assuming the inputs and outputs of the filters as real quantities with filter weights mostly positive real numbers.

In this paper, we apply the FOSP technique in inverse system identification and all the signals including the filter input, output and weights are in complex form. We apply the FO derivatives in the equalization problem; the convergence can be improved with three adaptation parameters, namely an FO and the two different step sizes in which one is used for the conventional update and the other for the fractional update. The NLMS algorithm has faster convergence time [42–46]; the FO NLMS further improves the convergence rate due to the addition of another update term. The paper provides a comparative analysis of the new strategies and the traditional counterparts. Both frequency flat and selective fading channels have been simulated with different parameters of root mean square (RMS) delay spread and bandwidths. Five to thirty taps filters have been considered, and we found that the FLMS-based equalizer performs well with less number of taps and converge at a faster rate so that a small number of training symbols are required. Binary/quadrature phase shift keying (BPSK/QPSK) modulated symbols are used for training purpose while quadrature amplitude modulation (QAM) of orders 4, 16, 64 and 256 have been used for data transmission. The main contributions of the paper are as under:

- Design of FO nonlinear variants of the LMS and NLMS algorithms for channel equalization (both feed-forward (FF) and decision feedback equalization (DFE) configurations).
- Novel tapped delay lines filter structure for the DFE based on the newly designed fractional adaptive schemes.
- The algorithms are designed to achieve faster convergence; the step size in the fractional update is a function of the FO, resulting in improved symbol error rate (SER) at different signal-to-noise ratios.
Verification and validation of the proposed algorithms for different modulation schemes and a number of step sizes, FOs and training symbols

Simulations for different frequency-selective channels and different number of filter taps, SER performance of different modulation schemes

The presentation in the rest of the paper is as follows: Firstly, we define the notation to describe the system model; this is followed by the presentation of the designed methodology of the equalization problem based on the traditional and the proposed FO strategies. Next, we discuss the simulation results and performance analysis. Finally, the conclusions are drawn.

2 System model

We shall specify signals and parameters used in the model as vectors and matrices. Vectors will be represented by boldface small letters and matrices with boldface capital letters. Estimated parameters are represented by the same symbol but having a (cap) above it. Conjugation, transposition and Hermitian operations on vectors are represented by superscripts symbols *, T and H, respectively; n, k and v represent the noise, iteration number (time index) and the fractional order, respectively. A block diagram for the equalization process is shown in Fig. 1. The transmitted symbols set \( x(n) = [x(n), x(n - 1), ..., x(n - M + 1)]^T \) is passed through the channel modeled as a finite impulse response (FIR) \([43,45,46]\) filter \( h = [h_0, h_1, ..., h_{M-1}]^T \) where \( M \) is the number of taps of channel which depends on Doppler shift and sampling frequency \([44-46]\). Noise is added to the channel output; we assume zero-mean additive white Gaussian noise (AWGN). To cancel out the effects of the channel, the noisy observation is input to an \( N \)-taps \((N > M)\) equalizer filter with weight vector \( w = [w_0, w_1, ..., w_{N-1}]^T \); its output is passed to the detector to decide which symbol was sent. The weights of the equalizer are updated through an adaptive algorithm; we use least mean square (LMS) and normalized LMS (NLMS) as the standard adaptation algorithms, and the fractional LMS (FLMS) and the fractional normalized LMS (FNLMS) as their fractional variants, respectively. The input regression vector of the equalizer is formed from the channel output symbols, that is, \( y(k) = [y(k), y(k - 1), ..., y(k - N + 1)]^T \). We define error as the difference between the true or desired response and the estimated output of the equalizer filter. We use binary phase shift keying (BPSK) modulation for training symbols and quadrature phase shift keying (QPSK) modulation for the data transmission. The received noisy signal through the channel can be written as:

\[
y(k) = \sum_{i=0}^{M} h_i x(k - i) + n(k)
\]

\[
= h_0 x(k) + \sum_{i=1}^{M} h_i x(k - i) + n(k)
\]  (1)

The second term on right side of Eq. (1) is the intersymbol interference (ISI) which is contributed by the multipath signals arrived with certain delays.

The estimated output of the equalizer depends on the properly adjusted weights of the filter and is given as:

\[
\hat{x}(k - \Delta) = w^H y(k).
\]  (2)

The design objective for the equalizer weights \( w \) to be adjusted such that to minimize the instantaneous error \( e(k) = x(k - \Delta) - \hat{x}(k - \Delta) \) during the data transmission phase, in which case the combined response of the channel and the equalizer becomes an impulse at the delay of \( z^{-\Delta} \). Practically this is difficult to achieve, especially in the cases where spectral nulls exist due to severe ISI or highly frequency-selective channels. In such situations, we need to use filters with fast convergence \([1,7,43]\), or explore new techniques such as...

---

**Fig. 1** Schematic diagram of the equalization process
fractional signal processing to precisely compensate the effects of the channel.

3 Design methodology

In this section, the design methodology for the adaptation of the equalizer weights is presented, and it consists of two subsections. The first subsection describes the standard LMS and the NLMS adaptive algorithms, and the second subsection provides the details of the fractional variants of these schemes.

3.1 LMS and NLMS adaptive strategies

To effectively cancel out the effects of the channel induced ISI, the objective of the equalizer is to minimize the mean squared error (MSE) which is the cost function. In the training phase, the objective function is written as:

$$\min_{w} E \left[ (e^*(k) (d(k) - w^H y(k))) \right].$$

(3)

During the training phase, \(d(k) = x(k - \Delta)\) is known to the receiver; in the decision-directed (normal) mode of the operation, \(d(k)\) is the output of the detector which is represented by \(\hat{x}(k - \Delta)\). The MSE \([35, 43, 45, 46]\) for the equalizer filter is given by expanding the quantity in (3) and then taking the expectation, that is,

$$E \left[ |e(k)|^2 \right] = \sigma_x^2 - w^H E \left[ y(k) x^* (k - \Delta) \right] - w^H E \left[ x(k - \Delta) y^* (k) \right] + w^H E \left[ y(k) y^H (k) \right] w.$$  

(4)

Here, the term \(\sigma_x^2 = E \left[ d^2 (k) \right]\) is the average power of the transmitted symbol set, the quantity \(E \left[ y(k) x^* (k - \Delta) \right] = E \left[ x(k - \Delta) y^* (k) \right]\) is the cross correlation vector \(p\) between the input and output of the channel, and \(E \left[ y(k) y^H (k) \right]\) is the channel output autocorrelation matrix \(R\) having size \(N \times N\). It can be seen that the MSE is a function of the equalizer weights; the simplified objective function is as follows:

$$J(w) = \sigma_x^2 - 2 w^H p + w^H R w.$$  

(5)

Minimum MSE (MMSE) is the optimal value of the MSE with respect to the equalizer coefficients, that is, \(\min J(w)\) over all the available weights \([43–46]\). Differentiating the cost function (5) with respect to \(w\) and equating to zero gives the optimal Wiener solution for the weight vector: \(w_{opt} = R^{-1} p\), the resulting MMSE is:

$$J_{\min} (w_{opt}) = \sigma_x^2 - p^T R^{-1} p.$$  

(6)

The computational complexity of computing the inverse \((R^{-1})\) is of the order of \(O \sim N^3\) \([8]\), which increases quickly with \(N\) and more so when the multiplications are in the complex domain. Statistical information about the channel is also required for the calculation of \(R\) which needs to be estimated. Since this information is not known \textit{a priori} and is time varying, the quantities \(R\) and \(p\) are seldom calculated.

Many techniques have been developed to optimize (5) such as in \([1–10]\). The steepest descent (SD) algorithm is widely used due to its low computational cost (order \(2N + 1\)); its update equation is:

$$w(k+1) = w(k) + \mu \left[ - \nabla J (w(k)) \right],$$  

(7)

where the step size \(\mu\) needs to be suitably chosen to control the speed of convergence. The step size is bounded as \(0 < \mu < 2 / \sum_{i=1}^{N} \lambda_i\) where \(\lambda_i\) denotes eigenvalues of the correlation matrix \(R\) \([43]\). Substituting for the gradient, the weight adaptation equation becomes:

$$w(k+1) = w(k) + \mu \left( p - R w(k) \right),$$  

(8)

which require prior knowledge of \(p\) and \(R\). Taking the first-order derivative of (3), the gradient can be written as \(2E \left[ e_n y_n^* \right]\); now replacing the expectation by its one point approximation, Eq. (7) can be written as the standard LMS update equation:

$$w(k+1) = w(k) + \mu e_n y(k).$$  

(9)

The LMS weight update equation is easy to implement and has a low computational cost. It has relatively poor convergence properties \([1, 8, 9]\); various strategies are applied to create variants at the expense of computational cost; however, faster convergence and stability remain the challenging propositions \([4–7]\). The step size is the only parameter that can be adjusted to improve the rate of convergence of the LMS algorithm; there are trade-offs with the rate of convergence, the average MSE and the tracking capability of the filter. A variable step size may be selected to increase the speed of convergence, minimize excess MSE and increase the tracking capability \([4, 5]\). Normalized LMS is used to adjust its weight automatically and try to achieve
the aforementioned objectives [35,45,46]; the variable step size in a given iteration is calculated as:

$$\mu_k = \beta \frac{1}{\| y (k) \|^2}.$$  \hfill (10)

Since, this step size is based on the instantaneous values, there is a chance of misadjustments; the new step-size parameter $\beta$ with limits $0 < \beta < 2$ is introduced. Substituting for $\mu_k$, we get the normalized LMS (NLMS) equation:

$$w (k + 1) = w (k) + \beta \frac{e^* (k) y (k)}{\| y (k) \|^2}.$$  \hfill (11)

In the next section, we exploit the strength of fractional derivatives in searching the minimum point on performance surface. As the gradient in the SD is based on the assumption that the weight vector follows a linear surface instead of quadratic performance surface; we, therefore, apply fractional derivatives in addition to the first order to obtain the fractional variants of the LMS and NLMS algorithms.

3.2 Fractional LMS/NLMS adaptive strategies

The basic concepts, underlying theories and strong mathematical foundations of fractional calculus are exploited in many fields of engineering; recently, the research community has shown greater interest in signal processing by designing new adaptive strategies based on fractional variants of the LMS algorithm [11,41]. In the LMS algorithm, the optimum weights are obtained with infinite number of iterations considering small step sizes at the cost of very poor convergence [4–7], although keeping the step size parameter small gives better steady state behavior. To improve the convergence performance and retain comparable complexity, the FLMS algorithms [37,38] were designed by employing fractional-order derivative of the cost function in addition to the first-order derivative. The basic weights updating relation for the FLMS similar to Eq. (7) is written as:

$$w (k + 1) = w (k) + \frac{\mu_1}{2} \left[ - \frac{\partial J (w (k))}{\partial w} \right] + \frac{\mu_2}{2} \left[ - \frac{\partial J^\nu (w (k))}{\partial w^\nu} \right],$$  \hfill (12)

where $\mu_1$ and $\mu_2$ are the step sizes for the standard and fractional update parts, respectively, while $\nu$ denotes the fractional order (FO) and its value is normally chosen between 0 and 1. The approach is almost similar to the augmented error methodology which is based on the MSE and minimization of the rate of change of the squared error or error penalty [47,48]. Different definitions have been provided for the FO integration and derivative operations, for instance, Grünwald–Letnikov, Riemann–Liouville, Erdélyi-Kober, Hadamard, Caputo, Riesz [15–17,39,40] and many others. Equivalence of these relations is well established on the basis of many functions and can be seen in [15,40]. The Grünwald–Letnikov (GL) definition of fractional derivative $a D^\nu_x$ of order $\nu$ with lower terminal at $a$ for function $f (x)$ is written as [15]:

$$a D^\nu_x f (x) = \lim_{h \to 0} h^\nu \sum_{m=0}^{\infty} (-1)^m \binom{\nu}{m} \times f (x + (\nu - m)h)$$  \hfill (13)

Similarly, the Riemann–Liouville (RL) definition of the FO derivative $a D^\nu_x$ with order $\nu$ is written as [15]:

$$a D^\nu_{a,x} f (x) = \frac{1}{\Gamma (n-\nu)} \frac{d^n}{dx^n} \int_a^x (x-\tau)^{n-\nu-1} f (\tau) \, d\tau,$$

$$x > a, \quad n - 1 < \nu \leq n, \quad (14)$$

A third definition, which is mostly applied in control applications [15,40], is the Caputo FO derivative of the function $f (t)$ is defined by [15]:

$$C_a D^\nu_t f (t) = \frac{1}{\Gamma (n-\nu)} \int_v^t (t-\tau)^{n-\nu-1} f (\tau) \, d\tau$$  \hfill (15)

Here $\Gamma$ is the gamma function, which for $z - \nu + 1 > 0$ is computed as [7]:

$$\Gamma (z - \nu + 1) = \int_0^\infty e^{-t} t^{z-\nu} \, dt$$  \hfill (16)

Also;

$$\left\{ \begin{array}{l}
\Gamma (z + 1) = z!
\Gamma (z + 1) = z \Gamma (z)
\end{array} \right\}$$  \hfill (17)

Another important function is the Beta function denoted by $\beta$ is defined as:

$$\beta (z_1, z_2) = \int_0^1 t^{z_1-1} (1-t)^{z_2-1} \, dt$$  \hfill (18)

The $\beta$ and $\Gamma$ functions are related by the following equation:

$$\beta (z_1, z_2) = \frac{\Gamma (z_1) \Gamma (z_2)}{\Gamma (z_1 + z_2)} (Re (z_1) > 0, \, Re (z_2) > 0)$$  \hfill (19)
One of the popular methods is to perform the FO differentiation is to use iterative integral method [15]. Consider, for example, the case of order $2$, i.e., $n = 2$, the fractional integration as below:

$$cD_x^{-2} f(x) = \int_c^x dx_1 \int_c^{x_1} f(t) dt$$

$$= \int_c^x (x-t) f(t) dt$$  \hspace{1cm} (20)

For $n$ number of repetitions Eq. (20) becomes:

$$cD_x^{-n} f(x) = \int_c^x (x-t)^{n-1} f(t) dt$$  \hspace{1cm} (21)

In terms of the gamma function, (21) can be written as:

$$cD_x^{-n} f(x) = \frac{1}{\Gamma(n)} \int_c^x (x-t)^{n-1} f(t) dt$$  \hspace{1cm} (22)

which for any FO becomes:

$$cD_x^{-v} f(x) = \frac{1}{\Gamma(v)} \int_c^x (x-t)^{v-1} f(t) dt$$  \hspace{1cm} (23)

which is the FO Riemann integral.

Next, we use the above method of fractional integrals to obtain the FO derivative such that a positive order corresponds to differentiation and a negative order as integration which is already described in Eq. (23). Consider for a real fractional order $v > 0$, $n = v + 1$ and $\rho = n - v$ such that, $0 \leq \rho \leq 1$, the fractional derivative $v$ for a function $f(x)$ is defined as:

$$cD_x^v f(x) = cD_x^{-\rho} \left[ cD_x^{-\rho} f(x) \right] \hspace{1cm} x > 0$$  \hspace{1cm} (24)

For the lower terminal as 0, that is, $c = 0$, $\rho > -1$, Eq. (24) is simplified to:

$$0D_x^v f(x) = 0D_x^{-\rho} \left[ 0D_x^{-\rho} f(x) \right]$$  \hspace{1cm} (25)

The inner part is simplified to be equal to the following:

$$cD_x^{-\rho} f(x) = \frac{1}{\Gamma(\rho)} \int_0^x (x-t)^{\rho-1} t^\rho dt$$  \hspace{1cm} (26)

This can be written in terms of mixed $\Gamma$ and $\beta$ functions as:

$$0D_t^- f(x) = \frac{\beta(p + 1, \rho)}{\Gamma(\rho)} t^{p+\rho}$$  \hspace{1cm} (27)

Or in terms of only gamma functions, it can be further simplified to:

$$0D_t^- f(x) = \frac{\Gamma(p + 1)}{\Gamma(p + \rho + 1)} t^{p+\rho}$$  \hspace{1cm} (28)

Putting the value in Eq. (25), we have:

$$0D_x^v f(x) = 0D_x^{-\rho} \left[ \frac{\Gamma(p + 1)}{\Gamma(p + \rho + 1)} t^{p+\rho} \right]$$  \hspace{1cm} (29)

Using the above method of repeating integrals, we have the FO derivative of the function $f(x)$ as:

$$0D_x^v f(x) = \frac{\Gamma(p + 1)}{\Gamma(p + \rho + n + 1)} t^{p+\rho-n}$$  \hspace{1cm} (30)

Putting $\rho = n - v$

$$0D_x^v f(x) = \frac{\Gamma(p + 1)}{\Gamma(p - v + 1)} t^{p-v}$$  \hspace{1cm} (31)

For $t > 0$, the RL-based FO differentiation for $v > -1$ of a polynomial function $g(t) = t^z$ is defined as [ch.7 in [15]]:

$$0D_t^v t^z = \frac{\Gamma(z + 1)}{\Gamma(z - v + 1)} t^{z-v}.$$  \hspace{1cm} (32)

It is worth mentioning to note that:

$$\frac{\Gamma(z + 1)}{\Gamma(z - v + 1)} = z(z - 1)(z - 2) \ldots (z - v + 1)$$  \hspace{1cm} (33)

For $v \geq 0$, the operation in (4) is differentiation; otherwise, it becomes integration. Equation (4) can also be written as:

$$0D_t^v t^z = \frac{\beta(z + 1, v)}{\Gamma(v)} t^{z-v}$$  \hspace{1cm} (34)

It is always better to use $\beta$ function in certain cases instead of the combination of $\Gamma$ functions, that is, to reduce complexity. The fractional update part is obtained by using the RL differintegral operator while differentiating the cost function with respect to the filter weights:

$$\Delta w_j(k) = -\mu_f \frac{\partial \eta^T J(k)}{\partial w^v} = -\mu_f \frac{\partial \left( e^T(k) \right)^2}{\partial w^v} \big|_{w=w(k)}$$  \hspace{1cm} (35)

where $\mu_f$ is step size for the fractional part adaptation. As described above, we take fractional derivative of order $v$ of the MSE; hence, the weight update term is:

$$\frac{\partial J^v(w(k))}{\partial w^v} = -\Gamma(2) y^T(k) e^v(k) \odot w^{1-v}(k) \frac{\Gamma(2 - v)}{\Gamma(2 - v)}$$  \hspace{1cm} (36)

where $\odot$ shows element-wise multiplication. Using Eq. (36) in Eq. (12), we get:

$$w(k+1) = w(k) + \mu_1 e^v(k) y^T(k) + \mu_2 e^v(k) y^T(k) \odot \left[ \frac{\Gamma(2)}{\Gamma(2 - v)} w^{1-v}(k) \right]$$  \hspace{1cm} (37)
where the symbol $\odot$ represents element-wise multiplication. The relation given in (37) is the weight updating formula for the FLMS for the equalizer (nonlinear), while the detail information for derivation of standard fractional LMS can be seen in [34–38] for real cases.

Additionally, in Fig. 2, a tapped delay line realization is structured for two-tap FLMS based on Eq. (37). The block containing the contents as $(\hat{1} - v)$ is used for power of the input weight. Note that the fractional variant incorporates the effects of previous weight vector with power $1 - v$ as well as the combined effect of the term $\mu_2 \Gamma(2 - v)$ as $\Gamma(2) = 1$. The designed algorithm is more flexible and has more adjustable parameters, that is, $\mu_2$ and $v$ which further scale the value of weight vector with various powers; hence better control on the rate of convergence and error performance. Since, each individual weight in the vector has different magnitude than the others; the simplified gradient search mechanism in the LMS cannot provide the required intelligence based on the prominence. The partial update, therefore, provides a sense of importance to each individual weight based on its magnitude and the FO. This helps capture the required parameters at a faster rate and keep the error small; which results in improved convergence rate.

Table 1 summarizes the above designed adaptive strategy for the equalization problem based on the above fractional variant of the LMS. For efficient computer implementation, we can introduce two extra variables $L = e(n) u^H(n)$ and $g = \mu_2 \frac{1}{\Gamma(2 - v)}$, and these

![Diagram of Two-tap fractional LMS with input delayed lines](image)

Table 1  Pseudocode of fractional LMS algorithm for equalization

| Filter Order $N$, Step Sizes $\mu_l$ & $\mu_f$ |
|-----------------------------------------------|
| Path Delay $\Delta$                          |
| Input Vector $u$                             |
| Filter Output $\hat{x}$                      |
| Initialize with zeros $u, w, \ldots$         |

```
Algorithm: for $k = 1, 2, 3, \ldots$

$u(k) = [u(k), u(k-1) \ldots u(k-N+1)]$

$d(k) = x(k - \Delta)$

$\hat{x}(k) = u w$

$e(k) = d(k) - \hat{x}(k)$

$w(k + 1) = w(k) + \mu_2 e(k) u^H(k) + \mu_f e(k) u^H(k) \odot \frac{1-v(k)}{\Gamma(2-v)}$
```

significantly reduce the complexity as the former quantity is calculated once per iteration, while the latter is constant. The vector $y(k)$ is generated through Eq. (1), and the vector $u$ is used to create input for the equalizer to incorporate the effects of ISI.

Similarly, we can obtain the final weight update equation for the fractional normalized LMS (FNLMS) using fractional derivatives in fractional update part to have automatic step-size adaptation, the latter helps to obtain a stable and fast convergence, and it is desirable to have input dependent step size. The difference in of the squares of the a posteriori and a priori errors is given by the following relation:
\[ \Delta \hat{e}^2 (k) = -4 \mu (k) e^2 (k) y^T (k) y (k) + 4 \mu^2 (k) e^2 (k) \left( y^T (k) y (k) \right)^2 \] (38)

Differentiating Eq. (38) with respect to \( \mu (k) \) and equating to zero for optimal value, the step size for the standard NLMS can be calculated as:

\[ \mu_l (k) = \frac{\mu}{\| y (k) \|^2} \] (39)

We calculate the a-posteriori error and take the FO derivative to obtain the FO update part. Taking fractional differentiation of Eq. (38) with respect to \( \mu_f (k) \), we get the following equation for optimum value at a given iteration and fractional order:

\[ -\frac{\Gamma (2)}{\Gamma (2 - v)} \mu_f^{1-v} (k) + \frac{\Gamma (3)}{\Gamma (3 - v)} \mu_f^{2-v} (k) \left( y^T (k) y (k) \right) = 0 \] (40)

Simplifying Eq. (40) results in the following relation for step-size adaptation in fractional part:

\[ \mu_f (k) = \frac{\Gamma (3 - v)}{\Gamma (2 - v) \Gamma (3)} \frac{\mu_f}{\| y (k) \|^2} \] (41)

which in mixed form of gamma and beta functions can be written as:

\[ \mu_f (k) = \frac{3 \Gamma (3 - v)}{\beta (4, 2 - v) \Gamma (6 - v)} \frac{\mu_f}{\| y (k) \|^2} \] (42)

The fractional orders are kept such that their ranges ensure the conditions mentioned in Eq. (14). For FNLMs, the fractional update becomes:

\[ \Delta w_f (k) = \mu_f \frac{\Gamma (3 - v)}{\Gamma (2 - v) \Gamma (3)} y^T (k) e (k) \]
\[ \circ \frac{w_l^{1-v} (k)}{\Gamma (2 - v)} \] (43)

The FNLMs final update is, therefore, as given below:

\[ w (k + 1) = w (k) + \mu_l (k) e (k) y^T (k) + \mu_f (k) e (k) y^T (k) \circ \left[ \frac{w_l^{1-v} (k)}{\Gamma (2 - v)} \right]. \] (44)

The computational complexities of the NLMS (11) as well as its fractional variant (16) increase due to the calculation of normalization \( \| y (k) \|^2 \) and then division of the update term by the norm; for fractional part it further increases. However, the norm can be calculated efficiently through recursive technique of adding the square of new term and subtracting the square of oldest term from the norm in previous iteration; also, the parameters for \( \Gamma \) function can be calculated before the adaptive algorithm starts and kept as constant as mentioned previously.

To keep the step size within bounds to avoid large values due to very small value of instantaneous norm of the channel output, it is standard practice to add another small term \( \varepsilon \) in the norm [34,45], that is, \( \mu_l (k) = \frac{1}{\| y (k) \|^2 + \varepsilon} \), and the final FNLMs weight update equation becomes:

\[ w (k + 1) = w (k) + \mu_l \frac{e (k) y^T (k)}{\| y (k) \|^2 + \varepsilon} + \mu_f \frac{e (k) y^T (k)}{\| y (k) \|^2 + \varepsilon} \circ \left[ \frac{w_l^{1-v} (k)}{\Gamma (2 - v)} \right]. \] (45)

Here, the term \( G = \frac{\Gamma (3 - v) \Gamma (2 - v) \Gamma (3)}{\Gamma (2 - v) \Gamma (3)} \) is a constant and its value depends on the FO. As we increase \( v \), the value of \( G \) decreases and vice versa. This help increase the stability by keeping the overall step size small in the steady state. Both the standard NLMS and FNLMs algorithms applied to the feed-forward (FF) equalization problem are summarized in Table 2.

In the case of decision feedback equalization (DFE), there is a feedback filter which further helps reduce the post-decision ISI. Both the NLMS and FNLMs algor-
Table 3  DFE algorithms based on NLMS and FNLMS

| Normalized LMS Algorithm | Normalized Fractional LMS Algorithm |
|--------------------------|-------------------------------------|
| Filter Orders N_f&N_b    | Filter Orders N_f&N_b               |
| Step Size μ_l            | Step Sizes μ_l, μ_f                 |
| Path Delay Δ             | Define G, v                         |
| Input Vector y           | Path Delay Δ                        |
| Filter Output ̂x          | Input Vector y                      |
| Initialize u_f, u_b, w_f, w_b | Filter Output ̂x            |
| N = N_f + N_b            | Initialize u_f, u_b, w_f, w_b       |

Algorithm: for k = 1, 2, 3… up to Training + Δ

\[ u_f(k) = \left[ u(k) u_f \left( 1 : N_f - 1 \right) \right] \]
\[ \hat{x}(k) = w_f^H u_f + w_b^H u_b \]
\[ d(k) = x(k) \]
\[ e(k) = d(k) - \hat{x}(k) \]

\[ u = \left[ u_f u_b \right] \]
\[ w(k + 1) = w(k) + \mu f \frac{n^H(k)}{|n(k)|^2} e(k) \]
\[ w_f = w_N \left( 1 \rightarrow N_f \right) \text{ update FF taps} \]
\[ w_b = w_N \left( N_f + 1 \rightarrow N \right) \text{ update FB taps} \]

4 Simulation results

In this section, the performance of the integer and fractional-order (FO) LMS and NLMS (FLMS and FNLMS) algorithms will be illustrated through simulations. In the first subsection, simulations results show learning performance based on the mean error, mean square deviation (MSD) with different values of μ_l, μ_f, β, γ; (here, β, γ are used as step sizes for the NLMS and FNLMS algorithms to avoid confusion with the LMS and FLMS) these are the step sizes which control the rate of convergence of the standard LMS, FLMS, NLMS and FNLMS schemes, respectively. In the second part, we simulate for the adaptive equalization problem under different parameter settings to evaluate the performance through symbol error rate (SER), combined channel and equalizer response and scatter plots for the equalized symbols. The response of the equalizer combined with the channel show its inverse behavior.

4.1 Learning performance

To evaluate the comparative performance of the LMS, NLMS algorithms and their fractional variants as described in Sect. 3, we choose the optimum weights vector \( w_o \) and adaptively adjust the weights using the adaptation equations and observe the Mean Deviation (MD) and MSD as the figure of merits for different numbers filter weights and step sizes. This is essentially a system identification approach, and the figures are illustrated through convergence performance. The metrics MSD and MD are defined as:

\[ MSD = E \left[ \| w_o - w(k) \|^2 \right]. \]  
\[ MD = E \left[ w_o - w(k) \right]. \]
The MD is the square root of MSD and is approximated by \( \Delta w = \sqrt{\sum (w_o - w(k))^2} / N \). Before the algorithm is run, the weights are initialized to zero. The algorithms are simulated with 12, 16 and 30 taps. In the first simulation scenario, the reference vector \( w_o \) is a constant while in the other cases, \( w_o \) is generated from independent and identically distribution. We perform simulations for different values of \( \mu_l, \mu_f, \beta, \gamma \), to show their effect on the convergence rate. The order of fractional derivative is fixed to 0.5; however, in the next subsection of adaptive equalization, we simulate the performance for different fractional orders.

Figures 3, 4, 5 and 6 show the error performance curves (Eq. 19) for the LMS, FLMS, NLMS and FNLMS algorithms. Simulations are performed for 1000 iterations for each algorithm. Consider Fig. 3 (left), the filters have 30 taps each, and we take the desired reference weight vector as:

\[ w_o = [1, 2, 2, 2, 1, 1, 2, 2, 3, 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 3, 2, 2]. \]

In this case, we keep equal step sizes for the standard and fractional algorithms, that is, \( \mu_l = \mu_f = 0.005 \), and \( \beta = \gamma = 0.5 \). It is seen that the convergence of the FLMS algorithm is faster than the standard LMS. Also the FNLMS algorithm converges faster than the NLMS counterpart. The normalized algorithms con-
verge faster than the other two algorithms. In the right part of the figure, we simulate for a 16 tap case with \( \mathbf{w}_o \) taken as a random vector, \( \mu_1 = \mu_f = 0.005 \), \( \beta = 0.4 \) and \( \gamma = 0.8 \). Again the convergences of the proposed fractional variants are faster than their standard counterparts.

In Fig. 4 (left), we have filters having 30 taps. We have reduced the step sizes for the LMS and FLMS algorithms, i.e., \( \mu_1 = \mu_f = 0.003 \) and \( \beta = \gamma = 0.5 \). The convergence of the FLMS is faster than the standard LMS. In the right part of the figure, we again consider a 30 taps case with \( \mathbf{w}_o \) as random and increased step sizes such that \( \mu_1 = \mu_f = 0.01 \) and \( \beta = 0.5 \), \( \gamma = 0.5 \). The observation is the same; the convergences of the fractional variants are faster than the standard.

Figure 5 shows the learning performance for the four algorithms. We have the case with equal step sizes for the standard and fractional algorithms, i.e., \( \mu_1 = \mu_f = 0.002 \) but \( \beta = 0.2 \), \( \gamma = 0.8 \) (left) and \( \beta = 0.1 \), \( \gamma = 0.9 \) (right). The convergence of the FLMS is faster than the LMS algorithm. In the normalized case, the fractional scheme converges much faster than the NLMS algorithm as the former depends on both \( \beta \) and \( \gamma \) at the same time. The convergence of the NLMS is slower such that it equal to FLMS for small value of \( \beta \).
Table 4 SER of LMS and fractional LMS algorithm

| SNR (dB) | Training symbols | Errors LMS, $\mu_l = 0.005$ | Errors FLMS $\mu_f, = 0.005, \nu = 0.5$ | SER LMS | SER FLMS |
|----------|-----------------|-----------------------------|---------------------------------|--------|--------|
| 20       | 100             | 621,587                     | 180,042                         | 0.621587 | 0.180042 |
| 150      | 49,8306         | 115,111                      | 0.498306                         | 0.115111 |        |
| 200      | 192,945         | 1108                         | 0.192945                         | 0.001108 |        |
| 300      | 12,192          | 484                          | 0.012192                         | 0.000484 |        |
| 400      | 2261            | 22                           | 0.002261                         | 0.000022 |        |
| 500      | 724             | 17                           | 0.000724                         | 0.000017 |        |
| 30       | 734,020         | 67,782                       | 0.734020                         | 0.067782 |        |
| 200      | 543,595         | 4                            | 0.543595                         | 0.000004 |        |
| 400      | 3018            | 0                            | 0.003018                         | 0.00        |        |

In Fig. 6 (left), we have filters with 12 taps. We have the step sizes for the four algorithms as: $\mu_l = \mu_f = 0.002$ and $\beta = 0.1, \gamma = 0.9$. The convergence of the FLMS technique is faster than the standard LMS algorithm. In the right part of the figure, we again plot the MSD (Eq. 18) for the same parameters. Again the observation is the same that the fractional variants have speedy convergences than the standard counterparts and we move from the LMS to FLMS to NLMS to FNLMS in terms of convergence rate.

From the learning curves, we have the conclusion that the fractional variants converge faster than their integer order counterparts. The fractional versions provide more freedom to increase the rate of convergence as the step size of the standard algorithms is embedded in the step size for fractional part.

4.2 Adaptive equalization

In this part, we present the simulation results for the adaptive equalizer with the newly developed algorithms along with the standard LMS. We simulate the problem with different channels including flat fading and frequency-selective channels with different root mean square (RMS) delay spread (DS) and bandwidth. We
generated channels with RMS DS values of the order of 50 nanoseconds (ns) to 1.2 microseconds (μs) which covers wireless indoor to urban macrocell environments. The number of taps of the channel depends on the DS, and larger values of the RMS DS have more multipaths (power delay profile). As already discussed, the NLMS algorithm has a fast convergence time due to automatic adjustment of the step size in the algorithm; the FNLMS scheme further decreases the convergence time due to the addition of another controlling term. Here we provide a comparative simulation analysis of the four techniques for the performance evaluation of the adaptive linear equalizer. We consider different FO and step sizes to compare the SER performance of both the algorithms for 200 independent Monte Carlo runs, (DFE). We have used various numbers of training symbols (TS) for the learning phase. Five to thirty taps filters have been considered with binary phase shift keying (BPSK) modulated symbols used for training purpose while quadrature phase shift keying (QPSK) modulation has been used for data symbols. In the case of DFE, we use higher-order quadrature amplitude modulation (QAM) in the decision-directed mode. Both TS and data symbols are drawn from a finite symbol set which are equally likely independent and identically distributed. The QPSK and square QAM have been considered; the former has fixed amplitude for all the symbols but different phases, and in the latter, each symbol is characterized by an amplitude and phase. We consider different FO and step sizes to compare the SER performance of both the algorithms for 200 independent Monte Carlo runs,
In Table 4, the SERs are shown for the LMS and FLMS for different numbers of training symbols. The sequence in these simulations corresponds to a slow fading channel with five taps, and there are 10000 QPSK symbols used in decision-directed mode. Note that, at 20 dB SNR, the FLMS has small SER with 300 training symbols (TS) as compared to the 500 TS used by standard LMS. Similarly, at 30 dB SNR, the SER of the FLMS algorithm at 100 symbols is not only smaller
than standard LMS with 200 symbols but even comparable to that obtained at 400 TS of the LMS. It thus helps in reducing the number of TSs, which can be utilized for the data symbols; resulting in increasing the packet efficiency. Figures 7 and 8 show two different fading channels (almost flat) with scatter plots of the equalized symbols for various numbers of TSs. It can be seen that the FLMS is able to separate the symbols more than the LMS algorithm. Effectively, the FLMS has superior performance in decreasing the variance of the error from the desired symbols. Large values of variance result in high uncertainty and result in more erroneous decisions by the demodulator.

In Table 5, the SERs are shown for the NLMS and FNLMS schemes for different numbers of TS. Since, these algorithms exhibit faster convergence, we consider frequency-selective fading channels with up to 28 taps and there are 10,000 QPSK symbols used in decision-directed mode. The SER is averaged over 200 independent runs and is obtained for SNR values of 5, 10, 15, 20, 25 and 30 dB. We have used 100, 150 and 200 TS. In the high SNR regime, and given TS, the
FNLMS algorithm has superior error performance as compared to the standard NLMS scheme.

In the next simulations, we illustrate the performance for different fading channels using scatter plots for various numbers of TS. Figure 8 (left) shows the scatter plots for the 27 tap channel and its equalization with 28 taps filters. The \( \Delta \) is set to 14. It can be seen that the FNLMS scheme is able to separate the symbols more than the NLMS, especially at 100 TS. We observe that with FNLMS, there were no errors in the case of 150 and 200 training symbols, while the NLMS has 10 and 5 erroneous symbols, respectively.

Figure 8 (right) shows the frequency response of selective channel and the corresponding time domain impulse response (first row) for a 200 TS case. In the case of 200 TS, we observe no errors for both the algorithms, and the plots correspond to correct decisions. The simulation parameters set were as: \( \beta = 0.4, \gamma = 0.4 \) and \( \text{SNR} = 20 \, \text{dB} \). The \( \Delta \) is set to 14. Again; it can be seen that the FNLMS is able to separate the symbols more than the NLMS equalizer, especially at 100 training symbols. We observed 65, 35 and 39 errors corresponding to 100, 150 and 200 symbols for the NLMS while 8, 2 and 1 errors for the FNLMS equalizer. Figure 9 (bottom) show the scatter plots for fading channel with more selectivity. We observed 9, 7 and 5 errors for the FNLMS and 65, 3 and 8 errors for NLMS equalizer corresponding to 100, 150 and 200 TS, respectively. Simulations were performed for \( \beta = 0.4, \gamma = 0.3 \) and \( \text{SNR} = 30 \, \text{dB} \).

Figure 9 (top) shows scatter plots for various numbers of TS for the given fading channels with 27 taps channel and its equalization with 28 taps filters. The simulation parameters are set as: \( \beta = 0.4, \gamma = 0.4 \) and \( \text{SNR} = 20 \, \text{dB} \). The \( \Delta \) is set to 14. Again; it can be seen that the FNLMS is able to separate the symbols more than the NLMS equalizer, especially at 100 training symbols. We observed 65, 35 and 39 errors corresponding to 100, 150 and 200 symbols for the NLMS while 8, 2 and 1 errors for the FNLMS equalizer.
Design of fractional-order variants of complex LMS and NLMS algorithms

Figure 11 shows the corresponding MSE plot for both the schemes, and it can be seen that the MSE of the FLMS is much smaller than the LMS for the case 300 training symbols.

Figure 12 (left) shows another frequency-selective fading channel having 17 taps with delay $\Delta = 15$. We evaluate the performance of the channel for different step size of the FNLMS algorithm. The constellation diagrams for equalized signal are shown. In Fig. 13, QAM constellations are shown to have a comparative analysis. The feed-forward filter has $N_f = 51$ taps; the feedback filter has $N_b = 1$ tap. We simulate for 50 iterations for 500 data symbols used in the decision-directed mode. We use the same number of training symbols, i.e., 100, 300, 400 and 600 for 4QAM, 16QAM, 64QAM and 256QAM, respectively. It can be seen in the scatter plots that the FNLMS is able to isolate the symbols more than the normal LMS. Seemingly, the FNLMS has better performance in decreasing the variance of the error from the preferred symbols. Large variations from desired symbol result in high uncertainty, especially in higher-order modulation schemes and more erroneous decisions by the demodulator.

In Table 6, we fix the FO to 0.4 and simulate the performance for different values of step sizes as shown. The step size for the NLMS is also fixed at 0.4. It can be observed that the proposed fractional scheme outperforms the standard NLMS algorithm. In most cases, the fractional variant is error free.

In Table 7, SER values are given for QAM symbols using the 17 tap channel; we fix $\mu_l = \mu_f = 0.4$ and check the performance under different values of the fractional orders. SER in column two with numbers corresponds to NLMS while the rest values are for FNLMS. We select different SNR values for different QAM constellations. For all the fractional orders, the FNLMS performs better than its conventional counterpart.

From the simulation analysis, we can observe that the adaptive equalizers based on the FLMS and FNLMS strategies outperform the standard LMS and NLMS algorithms, respectively. We can conclude that the two fractional LMS algorithms perform well with less number of taps and converges faster so small number of training symbols are required. This helps in significant improvement in the SNR for the same SER and very less training symbols are needed, resulting in increasing the efficiency of the communication link. The inclusion of fractional term increases the complex-
### Table 6  SER of NLMS and fractional NLMS algorithm with different $\mu_2$ values

| QAM  | SNR (dB) | BER (SD) $\mu_1 = 0.4$ | BER (F) $\mu_2 = 0.1$ | BER (F) $\mu_2 = 0.2$ | BER (F) $\mu_2 = 0.3$ | BER (F) $\mu_2 = 0.4$ | BER (F) $\mu_2 = 0.5$ |
|------|----------|--------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| 4-QAM | 10       | 0.0008                   | 0.0001                 | 0.0001                 | 0                      | 0                      | 0                      |
|       | 12       | 0.0006                   | 0.0001                 | 0                      | 0                      | 0.0001                |
|       | 14       | 0.0002                   | 0                      | 0                      | 0                      | 0                      |
|       | 16       | 0                       | 0                      | 0                      | 0                      | 0                      |
|       | 18       | 0.0003                   | 0                      | 0                      | 0                      | 0                      |
|       | 20       | 0.0002                   | 0                      | 0                      | 0                      | 0                      |
| 16-QAM | 15       | 0.0009                   | 0.0002                 | 0.0002                 | 0.0001                 | 0.0001                | 0.0002                |
|       | 17       | 0.0002                   | 0.0001                 | 0.0001                 | 0.0001                 | 0.0001                |
|       | 19       | 0.0002                   | 0                      | 0                      | 0                      | 0                      |
|       | 21       | 0                       | 0                      | 0                      | 0                      | 0                      |
|       | 23       | 0.0001                   | 0                      | 0                      | 0                      | 0                      |
|       | 25       | 0.0000                   | 0                      | 0                      | 0                      | 0                      |
| 64-QAM | 20       | 0.0046                   | 0.0005                 | 0.0006                 | 0.0004                 | 0.0002                | 0.0003                |
|       | 22       | 0.0016                   | 0.0002                 | 0.0006                 | 0.0001                 | 0.0002                |
|       | 24       | 0.0026                   | 0.0001                 | 0.0001                 | 0                      | 0                      |
|       | 26       | 0.0023                   | 0                      | 0                      | 0                      | 0                      |
|       | 28       | 0.0020                   | 0                      | 0                      | 0                      | 0                      |
|       | 30       | 0                       | 0                      | 0                      | 0                      | 0                      |
| 256-QAM | 25       | 0.0081                   | 0.0009                 | 0.0008                 | 0.0008                 | 0.0007                |
|       | 27       | 0.0056                   | 0.0005                 | 0.0005                 | 0.0006                 |
|       | 29       | 0.0015                   | 0.0001                 | 0.0001                 | 0                      |
|       | 31       | 0.0006                   | 0                      | 0                      | 0                      |
|       | 33       | 0.0016                   | 0                      | 0                      | 0                      |
|       | 35       | 0.0003                   | 0                      | 0                      |

### Table 7  SER of NLMS and FNLMs for different fractional orders

| QAM  | SNR (dB) | NLMS $v = 0.01$ | NLMS $v = 0.1$ | NLMS $v = 0.2$ | NLMS $v = 0.3$ | NLMS $v = 0.4$ |
|------|----------|----------------|----------------|----------------|----------------|----------------|
| 4    | 10       | 0.0008         | 2.01e$-$5      | 4.01e$-$5      | 2.03e$-$5      | 2.02e$-$5      | 6.01e$-$5      |
|      | 20       | 0.0002         | 0              | 0              | 0              | 0              |
| 16   | 15       | 0.0009         | 1.61e$-$4      | 6.03e$-$5      | 1.02e$-$4      | 2.01e$-$4      | 1.20e$-$4      |
|      | 17       | 0.0002         | 0              | 2.04e$-$5      | 4.04e$-$5      | 2.03e$-$5      | 2.01e$-$5      |
|      | 25       | 0.0000         | 0              | 0              | 0              | 0              |
| 64   | 20       | 0.0046         | 6.22e$-$4      | 8.40e$-$4      | 6.43e$-$4      | 7.60e$-$4      | 4.80e$-$4      |
|      | 24       | 0.0026         | 6.01e$-$5      | 4.06e$-$5      | 0              | 2.03e$-$5      | 2.01e$-$5      |
|      | 30       | 0              | 1.02e$-$5      | 0              | 0              | 0              |
| 256  | 25       | 0.0081         | 7.40e$-$4      | 1.42e$-$3      | 1.31e$-$3      | 9.03e$-$4      | 6.80e$-$4      |
|      | 29       | 0.0015         | 3.81e$-$4      | 2.04e$-$5      | 0              | 4.02e$-$5      | 8.02e$-$4      |
|      | 35       | 0.0003         | 0              | 0              | 0              | 0              |
ity by a small amount such as one extra addition and calculation of the fractional power.

5 Conclusions

This paper proposed the application of fractional derivatives for improving the performance of both feedforward and decision feedback equalizers by introducing fractional updates in both the LMS and NLMS algorithms. We developed a filter structure for the equalizer filter for the fractional variants of the LMS; the performances in terms of mean square error, mean square deviation and symbol error rates of these fractional-order techniques were compared with the standard LMS and NLMS algorithms. It is worth mention to note that we have three adaptation parameters in the proposed techniques, that is, two step sizes and a fractional order. The extra two parameters facilitate to improve the convergence speed and better steady state performance. The former helps in reducing the number of training symbols required for better weight adaptation, and the latter helps in reducing the symbol error rate performance. Based on the simulation results for the performance metrics of symbol error rate, mean square error and combined channel and equalizer response, we can conclude that the fractional LMS and NLMS filters outperform the standard LMS and normalized LMS algorithms in both frequency flat as well as frequency-selective fading channels. Applying the new algorithms helps in significant improvement in the SNR and increases the link efficiency by using less number of training symbols and minimizing the effects of channel in a better way. We evaluated the error rate performance for various parameters of the fractional versions and found that the novel schemes outperform the traditional approaches in fading channels. From the simulation results, it can be inferred that the newer techniques help in reducing the number of training symbols and improve symbol error rate performance at low SNR values. It helps increasing the link efficiency not only by using higher-order modulation schemes but less number of training symbols. Owing to its strengths, the algorithms can also be promising in other adaptive signal processing applications like system identification, beamforming, active noise control, diffusion strategies in multi-agent systems, just to name a few. The algorithms are most favorable for adaptive signal processing applications owing to their reduced complexity, ease of implementation on signal processing systems, since these offer faster convergence than traditional counterparts. As a future research direction, the proof of convergence is required which include a complete analysis of the algorithm including the average mean square error behavior, average tap weight response to find out bounds on the step sizes as well as the fraction order. Further, the analysis requires the derivation of weight error correlation matrix involving nonlinear terms from the fractional weights update which can be approximated with binomial expansion, excess MSE and misadjustment and finally, to find the stability.

References

1. Rupp, M.: Convergence properties of adaptive equalizer algorithms. IEEE Trans. Signal Process. 59(6), 2562–2574 (2011)
2. Rupp, M., García-Naya, J.A.: Equalizers in mobile communications: tutorial 38. IEEE Instrum. Meas. Mag. 15(3), 32–42 (2012)
3. Wang, P., Fan, P., Yuan, W., Darnell, M.: Data detection and coding for data-dependent superimposed training. IET Signal Process. 8(2), 138–145 (2013)
4. Gui, G., Adachi, F.: Sparse least mean fourth order algorithm for adaptive channel estimation in low signal-to-noise ratio region. Int. J. Commun. Syst. 27(11), 3147–3157 (2014)
5. Gui, G., Peng, W., Adachi, F.: Adaptive system identification using robust LMS/F algorithm. Int. J. Commun. Syst. 27, 2956–2963 (2014). doi: 10.1002/dac.2517
6. Kang, B., Yoo, J., Park, P.: Bias-compensated normalised LMS algorithm with noisy input. Electron. Lett. 49(8), 538–539 (2013)
7. Liu, Z.: Variable tap-length linear equaliser with variable tap-length adaptation step-size. Electron. Lett. 50(8), 587–589 (2014)
8. Eweda, E.: Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels. IEEE Trans. Signal Process. 42(11), 2937–2944 (1994)
9. Lindbom, L., Sternad, M., Ahlén, A.: Tracking of time-varying mobile radio channels 1. The Wiener LMS algorithm. IEEE Trans. Commun. 49(12), 2207–2217 (2001)
10. Jeong, S., Moon, J.: Self-iterating soft equalizer. IEEE Trans. Commun. 61(9), 3697–3709 (2013)
11. Wang, K., Sha, X., Mei, L., Qiu, X.: Performance analysis of hybrid carrier system with MMSE equalization over doubly-dispersive channels. IEEE Commun. Lett. 16(7), 1048–1051 (2012)
12. Rupp, M.: Robust design of adaptive equalizers. IEEE Trans. Signal Process. 60(4), 1612–1626 (2012)
13. Ammari, M.L., Zaouali, K., Fortier, P.: Adaptive modulation and decision feedback equalization for frequency-selective MIMO channels. Int. J. Commun. Syst. 27(11), 3323–3338 (2014)
14. Wang, C., Tang, T.: Several gradient-based iterative estimation algorithms for a class of nonlinear systems using the filtering technique. Nonlinear Dyn. 77(3), 769–780 (2014)
15. Podlubny, I.: Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, vol. 198. Academic press, London (1998)
16. Badri, V., Tavazoei, M.S.: Fractional order control of thermal systems: achievability of frequency-domain requirements. Nonlinear Dyn. 80(4), 1773–1783 (2014)
17. Almeida, R., Torres, D.F.: A discrete method to solve fractional optimal control problems. Nonlinear Dyn. 80(4), 1811–1816 (2014)
18. Hu, F., Zhu, W.Q., Chen, L.C.: Stochastic fractional optimal control of quasi-integrable Hamiltonian system with fractional derivative damping. Nonlinear Dyn. 70(2), 1459–1472 (2012)
19. Zhou, Yong, Ionescu, Clara, Tenreiro Machado, J.A.: Fractional dynamics and its applications. Nonlinear Dyn. 80(4), 1661–1664 (2015)
20. Xiao-Jun, Y., Machado, J.A.T., Hristov, J.: Nonlinear dynamics for local fractional Burgers’ equation arising in fractal flow. Nonlinear Dyn. 84(1), 3–7 (2017)
21. Duma, R., Dobra, P., Trusca, M.: Embedded application of fractional order control. IET Electron. Lett. 48(24), 1526–1528 (2012)
22. Bhrawy, A.H., Taha, T.M., Machado, J.A.T.: Review of operational matrices and spectral techniques for fractional calculus. Nonlinear Dyn. 81(3), 1023–1052 (2015)
23. Tseng, C.C., Lee, S.L.: Design of digital Riesz fractional order differentiator. Signal Process. 102, 32–45 (2014)
24. Tseng, C.C., Lee, S.L.: Design of adjustable fractional order differentiator using expansion of ideal frequency response. Signal Process. 92(2), 498–508 (2012)
25. Ortigueira, M.D., Coito, F.J., Trujillo, J.J.: Discrete-time fractional systems. Signal Process. 107, 198–217 (2015)
26. Krishna, B.T.: Studies on fractional order differentiators and integrators: a survey. Signal Process. 91(3), 386–426 (2011)
27. Malti, R., Victor, S., Oustaloup, A.: Advances in system identification using fractional models. J. Comput. Nonlinear Dyn. 3(2), 021401 (2008)
28. Sheng, H., Chen, Y.Q., Qiu, T.: On the robustness of Hurst estimators. IET Signal Process. 5(2), 209–225 (2011)
29. Charef, A., Idiou, D.: Design of analog variable fractional order differentiator and integrator. Nonlinear Dyn. 69(4), 1577–1588 (2012)
30. Ortigueira, M.D., Trujillo, J.J., Martynyuk, V.I., Coito, F.J.: A generalized power series and its application in the inversion of transfer functions. Signal Process. 107, 238–245 (2014)
31. Ortigueira, M.D., Coito, F.J., Trujillo, J.J.: Discrete-time fractional systems. Signal Process. 107, 198–217 (2014)
32. Valério, D., Ortigueira, M.D., da Costa, J.S.: Identifying a transfer function from a frequency response. J. Comput. Nonlinear Dyn. 3(2), 021207 (2008)
33. Ortigueira, M.D., Machado, J.A.T.: Fractional signal processing and applications. Signal Process. 83(11), 2285–2286 (2003)
34. Aslam, M.S., Raja, M.A.Z.: A new adaptive strategy to improve online secondary path modeling in active noise control systems using fractional signal processing approach. Signal Process. 107, 433–443 (2015)
35. Shah, S.M., Samar, R., Raja, M.A.Z., Chambers, J.A.: Fractional normalised filtered-error least mean squares algorithm for application in active noise control systems. Electron. Lett. 50(14), 973–975 (2014)
36. Shah, S.M., Samar, R., Khan, N.M., Raja, M.A.Z.: Fractional-order adaptive signal processing strategies for active noise control systems. Nonlinear Dyn. 85(3), 1363–1376 (2016)
37. Raja, M.A.Z., Chaudhary, N.I.: Adaptive strategies for parameter estimation of Box–Jenkins systems. IET Signal Process. 8(9), 968–980 (2014)
38. Raja, M.A.Z., Chaudhary, N.I.: Two-stage fractional least mean square identification algorithm for parameter estimation of CARMA systems. Signal Process. 107, 327–339 (2015)
39. Tan, Y., He, Z., Tian, B.: Generalization of modified LMS algorithm to fractional order. IEEE Signal Process. Lett. 22(9), 1244–1248 (2015)
40. Ortigueira, M., Coito, F.: On the usefulness of Riemann–Liouville and Caputo derivatives in describing fractional shift-invariant linear systems. J. Appl. Nonlinear Dyn. 1, 113–124 (2012)
41. Shoaib, B., Qureshi, I.M.: Adaptive step-size modified fractional least mean square algorithm for chaotic time series prediction. Chin. Phys. B 23(5), 050503 (2014)
42. Shah, S.M., Samar, R., Naqvi, S.M.R., Chambers, J.A.: Fractional order constant modulus blind algorithms with application to channel equalisation. Electron. Lett. 50(23), 1702–1704 (2014)
43. Martin, K.R.: Adaptive equalization for wireless channels. In: Ibndkahl, M. (ed.) Adaptive Signal Processing in Wireless Communications, Adaptation and Cross Layer Design in Wireless Networks, pp. 235–268. CRC Press, Boca Raton (2009)
44. Shah, S.M.: Riemann–Liouville operator-based fractional normalised least mean square algorithm with application to decision feedback equalisation of multipath channels. IET Signal Process. 10(6), 575–582 (2016)
45. Sayed, A.H.: Adaptive Filters. Wiley, New York (2008)
46. Diniz, S.R.: Adaptive Filtering, Algorithms and Practical Implementations. Springer, Berlin (2008)
47. Principe, J.C., Rao, Y.N., Erdogan, D.: Error whitening wiener filters: theory and algorithms. Chapter-10. In: S. Haykin, B. Widrow, (eds.) Least-Mean-Square Adaptive Filters. Wiley, New York (2003)
48. Rao, Yadunandana N., Erdogan, Deniz, Principe, Jose C.: Error whitening criterion for adaptive filtering: theory and algorithms. IEEE Trans. Signal Process. 53(3), 1057–1069 (2005)