Research Article

Estimation of Navigation Mark Floating Based on Fractional-Order Gradient Descent with Momentum for RBF Neural Network

Qionglin Fang

Navigation College, Jimei University, Xiamen 361021, Fujian, China

Correspondence should be addressed to Qionglin Fang; fql1437@163.com

Received 3 November 2020; Revised 2 March 2021; Accepted 26 March 2021; Published 19 April 2021

Academic Editor: Juan C. Jauregui-Correa

Copyright © 2021 Qionglin Fang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To address the difficulty of estimating the drift of the navigation marks, a fractional-order gradient with the momentum RBF neural network (FOGDM-RBF) is designed. The convergence is proved, and it is used to estimate the drifting trajectory of the navigation marks with different geographical locations. First, the weight of the neural network is set. The navigation mark's meteorological, hydrological, and initial position data are taken as the input of the neural network. The neural network is trained and used to estimate the mark's position. The navigation mark's position is taken at a later time as the output of the neural network. The difference between the later position and the estimated position obtained from the neural network is the error function of the neural network. The influence of sea conditions and months are analyzed. The experimental results and error analysis show that FOGDM-RBF is better than other algorithms at trajectory estimation and interpolation, has better accuracy and generalization, and does not easily fall into the local optimum. It is effective at accelerating convergence speed and improving the performance of a gradient descent method.

1. Introduction

A navigation mark is an artificial mark that warns of the boundary of a channel. It helps ships navigate, locate, and avoid obstacles [1]. Navigation marks placed at sea are extremely susceptible to drifting due to wind and waves. Drifting of the navigation marks poses threats to the safety of the ships. Cheng [2] used an optimized calculation method of anchor chain length and found that the maximum drift value can be reduced to below the drift alarm threshold. This reduces the number of navigation mark's false alarms.

In recent years, artificial neural networks have been widely used in pattern recognition, expert systems, robots, complex system controls, and so on. The weights of the neural networks are obtained by training the networks. Gradient descent (GD) is a basic method for updating and optimizing the weights of neural networks. Yao et al. proposed a three-term gradient descent method with subspace techniques [3]. Liu and Lan presented an adaptive neural

gradient descent control for a class of nonlinear dynamic systems with the chaotic phenomenon [4]. Standard GD, however, has two main shortcomings: slow training speed and ease of falling into the local optimal solution. It takes a long time to find a convergent solution, and at each step, the direction has to be calculated and adjusted. When applied to large datasets, every input sample needs to update its parameters, and every iteration needs to traverse all samples. Once falling into a saddle point, the gradient becomes zero, and the model parameters are not updated.

The gradient descent with momentum (GDM) method was proposed to solve this problem [5]. By accumulating past gradient values, fluctuation on the path to the optimal value is reduced and convergence is accelerated. GDM can accelerate learning when the direction of the current and past gradients is uniform, while restraining oscillation when the direction of the current and past gradients is inconsistent.

Fractional calculus exists since 1695 when Leibniz and L'Hospital initiated the discussion aimed at calculus based

on fractional derivatives and integrals. Fractional calculus has demonstrated more advantages than integral calculus in the field of neural networks. Wang et al. [6] proposed a fractional gradient descent method for the backpropagation (BP) training of neural networks. The Caputo derivative was employed to evaluate the fractional gradient of the error defined as the traditional quadratic energy function. However, the boundedness of the weight sequences was not rigorously proved. Khan et al. [7] proposed a fractional descent-based learning algorithm for training radial basis-function neural networks that is a convex combination of conventional and fractional gradients. However, its network architecture can be improved. Yang et al. [8] extended the steepest fractional descent approach to BP training of the feedforward neural networks (FNN). However, the results fluctuated with different fractional orders, requiring the optimal fractional order to be designated manually. Liu et al. proposed a quasi fractional-order gradient descent method with adaptive step size for system identification [9]. Wei et al. proposed a generalization of the gradient method with the fractional-order gradient direction [10]. Cheng et al. used the multi-innovation fractional-order stochastic gradient for identification of Hammerstein nonlinear ARMAX systems [11].

To improve the training ability of a neural network, this paper proposes a fractional-order gradient descent with the momentum method for training RBF neural networks. Compared with reference [12], the convergence of the proposed algorithm is proved in this paper. Compared with references [13, 14], this paper uses both the momentum and gradient descent method.

In recent years, many works have been achieved in the area of neural network. Zouari et al. proposed an adaptive backstepping control for a single-link flexible robot manipulator-driven DC motor [15]. Zouari et al. presented an adaptive backstepping control for a class of uncertain single-input single-output nonlinear systems [16]. Zouari et al. used a robust adaptive control for a class of nonlinear systems using the backstepping method [17]. Zouari et al. adopted a robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems [18]. Zouari et al. proposed a neuro-adaptive tracking control of noninteger order systems with input nonlinearities and time-varying output constraints [19]. Haddad et al. used a variable-structure backstepping controller for multivariable nonlinear systems with actuator nonlinearities based on the adaptive fuzzy system [20]. Zouari presented a neural network-based adaptive backstepping dynamic surface control of drug dosage regimens in cancer treatment [21].

The training of neural networks has met some challenges, such as slow training speed, difficulty in parameter selection, fall into local optimum, and oscillation of the convergence process. The proposed method is used for predicting the drift of the navigation marks. The main innovations of this paper are as follows:

- (1) Fractional calculus is applied to the gradient descent with momentum generalization of the gradient

method with the fractional-order gradient direction algorithm for training neural networks

- (2) The convergence of the proposed algorithm is proved
- (3) The new algorithm is used for predicting the drift of the navigation marks

The paper is organized as follows. Section 2 briefly describes the theoretical background and related works on the estimation of drifting of the navigation marks. Section 3 presents the proposed drifting estimation algorithm using fractional-order gradient descent with momentum for the RBF neural network. Experimental results are shown in Section 4, and Section 5 concludes the paper.

2. Related Work

2.1. Dynamic Equation of Navigation Mark Drifting. To fix a navigation mark in a designed position so it can endure a severe marine environment without shifting, the mooring equipment is generally composed of the main anchor chain, a sinking stone, an auxiliary anchor chain, and an anchor. Therefore, compared with its designed latitude and longitude, the drifting of a navigation mark is relatively small. The drifting motion of a navigation mark can be regarded as a linear model.

The delays in communication from the equipment influence the reporting intervals on the positioning of the navigation marks. If $T(k)$ is defined as the k^{th} time interval between the navigation mark's position reports, then $\varphi(k)$ denotes the latitude of the navigation mark, $\lambda(k)$ denotes the longitude of the navigation mark, $v(k)$ denotes the speed over the ground of the navigation mark, and $\theta(k)$ denotes the moving angle over the ground of the navigation mark. The latitude and longitude of the navigation mark at the $k + 1^{\text{th}}$ time instant can be calculated as follows:

$$\varphi(k + 1) = \varphi(k) + v(k)T(k)\cos(\theta(k)), \quad (1)$$

$$\lambda(k + 1) = \lambda(k) + v(k)T(k)\sin(\theta(k)). \quad (2)$$

Define x as the state vector, that is,

$$x(k) = \begin{bmatrix} \varphi(k) \\ \lambda(k) \end{bmatrix}. \quad (3)$$

Define $u(k)$ as the input vector, which can be calculated as follows:

$$u(k) = \begin{bmatrix} v(k)\cos(\theta(k)) \\ v(k)\sin(\theta(k)) \end{bmatrix}. \quad (4)$$

Consider the presence of noise in the actual system. Define w as system noise. The system state equation of equations (1) and (2) can be written as follows:

$$x(k + 1) = x(k) + T(k)u(k) + w(k). \quad (5)$$

Define z as the measurement vector. Define ε as the measurement noise. The system observation value can be written as follows:

$$z(k+1) = x(k+1) + \varepsilon(k+1). \quad (6)$$

2.2. RBF Neural Network. In 1985, Powell proposed a Radial Basis Function (RBF) method for multivariate interpolation. The most frequently used Radial Basis Function is the Gauss function:

$$\varphi_i(x) = e^{-\left(\|x - \mu_i\|^2 / 2\sigma_i^2\right)}, \quad (7)$$

$$\varphi = [\varphi_1, \varphi_2, \dots, \varphi_P], \quad (8)$$

where x is the input vector, $\|x\|$ denotes the Euclidean norm of x , φ_i denotes the Radial Basis Function, x_i denotes the central vector of the function, σ_j denotes the width of the Radial Basis Function, μ_i denotes the threshold vector, P denotes the number of hidden layer nodes, N denotes the number of input training samples, and y denotes the output of the neural network:

$$y = W(n)\phi(x(n)), \quad (9)$$

$$W(n) = [w_1(n), w_2(n), \dots, w_P(n)]. \quad (10)$$

2.3. Fractional-Order Calculus. Fractional calculus is the operation of derivatives and integrals extended to fractional orders. It provides a more precise tool for describing physical systems. The Riemann–Liouville (RL) fractional differential integral is one of its most common definitions. For a function x that is defined in $[t_0, t]$, the RL fractional integrator is defined as follows:

$${}_{t_0}D_t^\alpha x(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t - \tau)^{\alpha-1} x(\tau) d\tau, \quad (11)$$

where α is the fractional order, τ is the integral variable, and $\Gamma(x)$ is the gamma function that is defined as follows:

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt. \quad (12)$$

The RL fractional derivative is defined as follows:

$${}_{t_0}D_t^\alpha x(t) = \frac{d^m}{dt^m} \left[\frac{1}{\Gamma(m-\alpha)} \int_{t_0}^t (t - \tau)^{m-\alpha-1} x(\tau) d\tau \right], \quad (13)$$

where $\alpha \in [m-1, m)$, and m is a positive integer near α . When $\alpha=1$, the algorithm will degenerate to the integral order case. When α is greater than 1, the performance will be worse.

Lemma 1. For the function $f(x) = x^v$ when $0 \leq m \leq p < m+1$, the following formula holds:

$${}_{x_0}D_x^p f(x) = \frac{\Gamma(v+1)}{\Gamma(v-p+1)} x^{v-p}. \quad (14)$$

3. Main Results

3.1. FOGDM-RBF Algorithm. Define d^n as the expected output response of the n times of iterations of the neural network. Then, the error signal can be defined as follows:

$$e^n = d^n - y^n = d^n - W^n \varphi(x^n). \quad (15)$$

In 1986, American cognitive psychologists D. E. Rumelhart and J. L. McClelland introduced the generalized delta rule (GDR) in neural network learning. The delta rule and gradient descent minimize the square of the difference between the actual and the desired output of the neural network. The loss function generally uses the least square error function. After defining the error criterion function, the next step is to adjust the weight to minimize the criterion function. The loss function can be minimized by improving the training mode. Therefore, the objective function is often defined as follows:

$$J^n = \frac{1}{2}(e^n)^2 = \frac{1}{2}[d^n - y^n]^2 = \frac{1}{2}[d^n - W^n \varphi(x^n)]^2. \quad (16)$$

3.2. Denote

$$\Delta w^{n+1} = w^{n+1} - \Delta w^n. \quad (17)$$

Based on the gradient descent with momentum algorithm, the following can be obtained:

$$\Delta w^{n+1} = \gamma_n \Delta w^n - \eta_c D_{w^n}^\alpha J^n, \quad (18)$$

$$0 < \alpha < 1, \quad (19)$$

where $\eta > 0$ is the learning rate and γ_n is the momentum coefficient designed as follows:

$$\gamma_n = \begin{cases} \frac{\gamma \|c D_{w^n}^\alpha J^n\|}{\|\Delta w^n\|}, & \|\Delta w^n\| \neq 0, \\ 0, & \text{else,} \end{cases} \quad (20)$$

$$0 < \gamma < \eta, \quad (21)$$

where $\gamma \in (0, \eta)$ is the momentum factor and $\|\cdot\|$ is the Euclidean norm. It used a modified Riemann–Liouville fractional definition [7]. According to the definition of the fractional derivative, the following can be obtained:

$${}_c D_{w_i}^\alpha J^n = \sum_{j=1}^N J_j' (w \phi(x_j^n)) \phi(x_j^n) \frac{(w_i^n - c)^{1-\alpha}}{(1-\alpha)\Gamma(1-\alpha)}. \quad (22)$$

3.3. Convergence Analysis of FOGAM-RBF. The relevant lemma is introduced first, and then, the convergence of the FOGDM-RBF algorithm is proved.

The following assumptions are given:

$|\phi|, |\phi'|, |\phi''|, |y|, |y'|$, and $|y''|$ are uniformly bounded, (A1)

$$\lambda_1 = \max\{|\phi|, |\phi'|, |\phi''|, |y|, |y'|, |y''|\}, \quad (23)$$

w is uniformly bounded, (A2),

$$\lambda_2 = \left\| \max_{n>0} w^n \right\|. \quad (24)$$

This condition can be easily satisfied since the most common Gauss function is uniformly bounded and differentiable.

Lemma 2 (see [22]). *Every bounded monotonic sequence of real numbers converges.*

Theorem 1. *Assume (A1), (A2), and (A3) are valid, and*

$$(\eta - \gamma)(1 - \alpha)\Gamma(1 - \alpha)(\lambda_2 - c)^{\alpha-1} \geq NP\lambda_1^3(\eta^2 + \gamma^2). \quad (25)$$

Then, the following results hold:

$$J(w^{n+1}) \leq J(w^n), \quad (26)$$

$$\lim_{n \rightarrow +\infty} J(w^n) = J^*, \quad (27)$$

$$\lim_{n \rightarrow +\infty} \|{}_c D_{w^n}^\alpha J(w^n)\| = 0. \quad (28)$$

Proof: by using the Taylor mean value theorem with the Lagrange remainder, the following can be obtained:

$$\begin{aligned} J(w^{n+1}) - J(w^n) &= \sum_{i=1}^N [J_i(w^{n+1}\phi(x_i)) - J_i(w^n\phi(x_i))], \\ &= \sum_{i=1}^N J'_i(w^n\phi(x_i)) [w^{n+1}\phi(x_i) - w^n\phi(x_i)] + \frac{1}{2} \sum_{i=1}^N J''_i(w^n\phi(x_i)) [w^{n+1}\phi(x_i) - w^n\phi(x_i)]^2, \\ &= \sum_{i=1}^N J'_i(w^n\phi(x_i)) \Delta w^{n+1}\phi(x_i) + \frac{1}{2} \sum_{i=1}^N J''_i(w^n\phi(x_i)) (\Delta w^{n+1})^2 \phi^2(x_i). \end{aligned} \quad (29)$$

From equation (22), it can be obtained that if $\forall 1 \leq j \leq P$, $(w_j(n) - c)^{1-\alpha} \neq 0$, the following equation holds:

$$\sum_{j=1}^N J'_j(w\phi(x_j^n)) \phi(x_j^n) = (1 - \alpha)\Gamma(1 - \alpha)(w^n - c)^{\alpha-1} {}_c D_{w^n}^\alpha J^n. \quad (30)$$

Substituting equation (30) into the first item of equation (29), the following is yielded:

$$\begin{aligned} \sum_{j=1}^N J'_j(w\phi(x_j^n)) \Delta w^{n+1}\phi(x_j^n) &= (1 - \alpha)\Gamma(1 - \alpha)(w^n - c)^{\alpha-1} \\ &\quad \Delta w^{n+1} {}_c D_{w^n}^\alpha J^n. \end{aligned} \quad (31)$$

From equation (18), the following can be obtained:

$$\begin{aligned} \Delta w_c^{n+1} {}_c D_{w^n}^\alpha J^n &= [\gamma_n \Delta w^n - \eta {}_c D_{w^n}^\alpha J^n] {}_c D_{w^n}^\alpha J^n, \\ &= \gamma_n \Delta w_c^n {}_c D_{w^n}^\alpha J^n - \eta \|{}_c D_{w^n}^\alpha J^n\|^2 \leq \|\gamma_n\| \cdot \|\Delta w^n\| \cdot \|{}_c D_{w^n}^\alpha J^n\| - \eta \|{}_c D_{w^n}^\alpha J^n\|^2. \end{aligned} \quad (32)$$

Substituting equation (26) into equation (32), the following is yielded:

$$\begin{aligned} \Delta w_c^{n+1} {}_c D_{w^n}^\alpha J^n &\leq \frac{\gamma \|{}_c D_{w^n}^\alpha J^n\|}{\|\Delta w^n\|} \cdot \|\Delta w^n\| \cdot \|{}_c D_{w^n}^\alpha J^n\| - \eta \|{}_c D_{w^n}^\alpha J^n\|^2 \\ &= \gamma \|{}_c D_{w^n}^\alpha J^n\|^2 - \eta \|{}_c D_{w^n}^\alpha J^n\|^2, \\ &= -(\eta - \gamma) \|{}_c D_{w^n}^\alpha J^n\|^2. \end{aligned} \quad (33)$$

Substituting equation (33) into equation (31) yields the following:

$$\begin{aligned} \sum_{j=1}^N J'_j(w\phi(x_j^n)) \Delta w^{n+1}\phi(x_j^n) &\leq -(\eta - \gamma)(1 - \alpha)\Gamma(1 - \alpha) \\ &\quad (w^n - c)^{\alpha-1} \|{}_c D_{w^n}^\alpha J^n\|^2. \end{aligned} \quad (34)$$

From equation (24), the following can be obtained:

$$\begin{aligned}
-(w-c)^{\alpha-1} &= -\frac{1}{(w-c)^{1-\alpha}} \leq -\frac{1}{(\lambda_2-c)^{1-\alpha}}, \\
&= -(\lambda_2-c)^{\alpha-1}.
\end{aligned} \tag{35}$$

Substituting equation (35) into equation (34), the following is yielded:

$$\begin{aligned}
\sum_{j=1}^N J'_j(w\phi(x_j^n)) \Delta w^{n+1} \phi(x_j^n) &\leq -(\eta-\gamma)(1-\alpha)\Gamma(1-\alpha) \\
&\cdot (\lambda_2-c)^{\alpha-1} \|_c D_{w_i^n}^\alpha J^n\|^2.
\end{aligned} \tag{36}$$

$$\begin{aligned}
\sum_{j=1}^N J'_j(w\phi(x_j^n)) \Delta w^{n+1} \phi(x_j^n) &\leq 0 - (\eta-\gamma)(1-\alpha)\Gamma(1-\alpha)(w_2^n-c)^{\alpha-1} \|_c D_{w_2^n}^\alpha J^n\|^2 - \dots - (\eta-\gamma)(1-\alpha)\Gamma \\
&\cdot (1-\alpha)(w_p^n-c)^{\alpha-1} \|_c D_{w_p^n}^\alpha J^n\|^2 \leq -(\eta-\gamma)(1-\alpha)\Gamma(1-\alpha)(\lambda_2-c)^{\alpha-1} \|_c D_{w^n}^\alpha J^n\|^2.
\end{aligned} \tag{38}$$

From equations (36) and (38), the following is found:

$$\begin{aligned}
\sum_{j=1}^N J'_j(w\phi(x_j^n)) \Delta w^{n+1} \phi(x_j^n) &\leq -(\eta-\gamma)(1-\alpha)\Gamma(1-\alpha) \\
&\cdot (\lambda_2-c)^{\alpha-1} \|_c D_{w_i^n}^\alpha J^n\|^2.
\end{aligned} \tag{39}$$

From equation (18), the following can be obtained:

$$\|\Delta w^{n+1}\|^2 = \|\gamma_n \Delta w^n - \eta_c D_{w^n}^\alpha J^n\|^2 \leq 2\gamma_n^2 \|\Delta w^n\|^2 + 2\eta^2 \|_c D_{w^n}^\alpha J^n\|^2. \tag{40}$$

Substituting equation (20) into equation (40), the following is yielded:

$$\begin{aligned}
\|\Delta w^{n+1}\|^2 &\leq 2 \frac{\gamma_c^2 \|_c D_{w^n}^\alpha J^n\|^2}{\|\Delta w^n\|^2} \|\Delta w^n\|^2 + 2\eta^2 \|_c D_{w^n}^\alpha J^n\|^2 \\
&= 2(\gamma^2 + \eta^2) \|_c D_{w^n}^\alpha J^n\|^2.
\end{aligned} \tag{41}$$

Substituting equation (41) into the second item of equation (29), the following is yielded:

$$\begin{aligned}
\frac{1}{2} \sum_{i=1}^N J''_i(w^n \phi(x_i)) (\Delta w^{n+1})^2 \phi^2(x_i) &\leq \sum_{i=1}^N J''_i(w^n \phi(x_i)) \\
&\cdot (\gamma^2 + \eta^2) \|_c D_{w^n}^\alpha J^n\|^2 \phi^2(x_i).
\end{aligned} \tag{42}$$

From equation (23), the following can be obtained:

$$\|\phi(x_i)\| \leq \sqrt{P} \lambda_1. \tag{43}$$

Substituting equation (43) into equation (42), the following is yielded:

If $\exists 1 \leq j \leq P$, $(w_j^n - c)^{1-\alpha} = 0$ and there is no loss of generality, it can be assumed that $(w_1^n - c)^{1-\alpha} = 0$.

From equation (22), the following can be obtained:

$${}_c D_{w_1^n}^\alpha J^n = 0. \tag{37}$$

Substituting equation (37) into equation (34), the following is yielded:

$$\begin{aligned}
\frac{1}{2} \sum_{i=1}^N J''_i(w^n \phi(x_i)) (\Delta w^{n+1})^2 \phi^2(x_i) &\leq P \lambda_1^2 \sum_{i=1}^N J''_i(w^n \phi(x_i)) \\
&\cdot (\gamma^2 + \eta^2) \|_c D_{w^n}^\alpha J^n\|^2.
\end{aligned} \tag{44}$$

Substituting equation (23) into equation (44), the following is yielded:

$$\frac{1}{2} \sum_{i=1}^N J''_i(w^n \phi(x_i)) (\Delta w^{n+1})^2 \phi^2(x_i) \leq NP \lambda_1^3 (\gamma^2 + \eta^2) \|_c D_{w^n}^\alpha J^n\|^2. \tag{45}$$

Substituting equation (45) into equation (29), the following is yielded:

$$\begin{aligned}
J(w^{n+1}) - J(w^n) &\leq \sum_{i=1}^N J'_i(w^n \phi(x_i)) \Delta w^{n+1} \phi(x_i) + NP \lambda_1^3 \\
&\cdot (\gamma^2 + \eta^2) \|_c D_{w^n}^\alpha J^n\|^2.
\end{aligned} \tag{46}$$

Substituting equation (39) into equation (46), the following is yielded:

$$\begin{aligned}
J(w^{n+1}) - J(w^n) &\leq -(\eta-\gamma)(1-\alpha)\Gamma(1-\alpha)(\lambda_2-c)^{\alpha-1} \\
&\cdot \|_c D_{w^n}^\alpha J^n\|^2 + NP \lambda_1^3 (\gamma^2 + \eta^2) \|_c D_{w^n}^\alpha J^n\|^2 \\
&= -[(\eta-\gamma)(1-\alpha)\Gamma(1-\alpha)(\lambda_2-c)^{\alpha-1} \\
&\quad - NP \lambda_1^3 (\gamma^2 + \eta^2)] \|_c D_{w^n}^\alpha J^n\|^2.
\end{aligned} \tag{47}$$

When equation (25) holds, the following can be obtained:

$$J(w^{n+1}) - J(w^n) \leq 0. \quad (48)$$

This completes the proof of statement (26) of Theorem 1.

From equation (48), the monotone decreasing sequence $J(w^n)$ is bound below $J(w^n) \geq 0$. Hence, from Lemma 2, $J(w^n)$ is convergent. Therefore, there exists $J^* \geq 0$, such that

$$\lim_{n \rightarrow +\infty} J(w^n) = J^*. \quad (49)$$

This completes the proof of statement (27) of Theorem 1. $\square$

3.4. Denote

$$\xi = (\eta - \gamma)(1 - \alpha)\Gamma(1 - \alpha)(\lambda_2 - c)^{\alpha-1} - NP\lambda_1^3(\gamma^2 + \eta^2). \quad (50)$$

Substituting equation (50) into equation (47), the following is yielded:

$$J(w^{n+1}) - J(w^n) \leq -\xi \|D_{w^n}^\alpha J^n\|^2. \quad (51)$$

Thus, the following is found:

$$J(w^{n+1}) \leq J(w^1) - \xi \sum_{i=1}^n \|D_{w^i}^\alpha J^i\|^2. \quad (52)$$

Since $J(w^{n+1}) \geq 0$, the following can be obtained:

$$\xi \sum_{i=1}^n \|D_{w^i}^\alpha J^i\|^2 \leq J(w^1) - J(w^{n+1}) \leq J(w^1). \quad (53)$$

When $n \rightarrow +\infty$, the following is true:

$$\sum_{i=1}^{+\infty} \|D_{w^i}^\alpha J^i\|^2 \leq \frac{1}{\xi} J(w^1) < +\infty. \quad (54)$$

Thus, the following is obtained:

$$\lim_{n \rightarrow +\infty} \|D_{w^n}^\alpha J(w^n)\| = 0. \quad (55)$$

This completes the proof of statement (28) of Theorem 1.

3.5. Navigation Mark Prediction Based on FOGDM-RBF. Prediction of the navigation mark is conducted in the following manner. Firstly, the weight of the neural network is set. The meteorological, hydrological, and position data of the navigation mark are used as the input for the network. Future position data is used as the output of the neural network. The difference between the mark's future position and the estimated position obtained from the network is taken as the error function and used to train the network. The weights of the trained neural network are used to estimate future positions, and the difference between the estimated position and the actual position is used to verify the neural network. The proposed algorithm is shown in Table 1.

4. Experiment and Analysis

4.1. Introduction of the Experimental Environment. Four navigation marks at the Xiamen city's port were used to

evaluate the performance of the improved Kalman algorithm. Table 1 lists position samples of the navigation mark #1 on January 1, 2019. Its designative position was 117.50708330 E and 23.75405560 N. The experiment used the mark's telemetry and telecontrol system. A GPS intelligent-integrated solar beacon lamp was used. Table 1 lists the position samples of the navigation mark #4 on January 1, 2019.

Table 2 lists the wind speed, maximum wind speed, wind direction, and wind scale collected by the big buoy #4.

4.2. Probability Density of Drifting. The probability density of the navigation mark's drifting is shown in Figure 1.

Most of the hot water areas for most of the light buoy offset are near the geometric center position of the active water area, which shows that, in the process of migration, these light buoys have certain regularity and show the characteristics of divergence from the center.

The offset hot water area of the light buoy deviates from the geometric center position of its active water area, which shows that the light buoys are asymmetric in the migration process, and their active hot water area tends to a certain area under the action of wind and current.

4.3. Analysis of the Influence of Sea Conditions. The estimated trajectory of the navigation mark in the Xiamen Port in 2019 shows an ENE direction of the normal wind and SE and SW directions of the strong wind. The east-northeast monsoon with high wind speeds prevails between September and March, and small-speed southeast winds prevail between April and August. Therefore, in winter and spring, navigation marks generally tend to shift southwest, while in summer and fall, the navigation marks tend to shift northeast.

The tide in the Xiamen Bay is mainly caused by the introduction of the tidal waves from the open sea, which belongs to the regular semidiurnal tide area. The average duration of the flood tides and ebb tides is the same. The direction of the fluctuation and the tidal current in the waters near the main channel is consistent with that in the channel. Therefore, in each ebb and flow period, the navigation mark is affected by the current, and its offset is reciprocated along the channel direction.

Affected by the wind and the current, the navigation mark's deviation direction in Xiamen Port's main channel changes from season to season, and its deviation law mainly depends on the impacting force of wind and current. If the wind force is dominant, the navigation mark deviates mainly from the downwind. If the flow force is dominant, the navigation mark deviates mainly along the channel axis.

4.4. Analysis of the Influence of Months of the Year. Figure 2 presents a month-to-month change in the drifting variance of the navigation mark #4. Its unit is 10^{-8} degrees. The blue cuboids represent the latitude variance from the designative position. The yellow cuboids represent the longitude variance from the designative position.

TABLE 1: Pseudocode of FGM-RBF for navigation mark drifting estimation.

Algorithm 1: FOGDM-RBF for AIS interpolation

Input: Navigation mark position data, meteorological and hydrological data as training samples and testing samples

Output: the estimated drifting position of testing samples

1: function FOGDM ()

2: Initialization of the neural network model;

3: Initialization of neural networks with weights;

4: $it = 0$;5: while $it < \max IT$ and the training error does not meet the learning requirements

6: Calculate the activation function of the neural network according to equation (7);

7: Calculate the output of neural network according to equation (10);

8: Calculate the output error of the neural network according to equation (15);

9: Calculate the objective function according to (16);

10: Calculate the caputo fractional derivative of the objective function according to (22);

11: Update the momentum coefficient according to equation (20);

12: Update the weight of the neural network according to equation (18);

13: $it = it + 1$;

14: end while

15: Use the obtained neural network weights and the input testing data set, calculate the estimated drifting position of testing samples.

16: End function

TABLE 2: Wind speed, maximum wind speed, wind direction, and wind scale collected by the buoy #4.

Time	Wind speed (m/s)	Maximum wind speed (m/s)	Wind direction	Wind scale
01-DEC-19 03.20.00.000000 AM	1.00	1.70	South	1.00
01-DEC-19 03.30.00.000000 AM	0.80	1.70	Southwest	1.00
01-DEC-19 03.40.00.000000 AM	1.00	2.10	West	1.00
01-DEC-19 03.50.00.000000 AM	1.60	2.80	Northwest	2.00
01-DEC-19 04.00.00.000000 AM	2.10	3.20	North-northwest	2.00
01-DEC-19 04.10.00.000000 AM	3.50	4.80	North-northwest	3.00

FIGURE 1: Probability density of the navigation mark's drifting. (a) Entrance buoy. (b) Buoy #3.

It can be seen that the drifting variance is smallest in the summer, especially in July. This is because the flow of ship traffic in the summer is dense. The wave wake pushes the navigation mark southwest. When the southeast wind prevails in Xiamen between April and August, it pushes the navigation mark southeast. The two forces counteract each other to a certain extent.

4.5. Comparison with Different Fractional Orders. The test accuracy with different fractional orders and batch sizes are shown in Figure 3.

The train accuracy with different fractional orders and batch sizes are shown in Figure 4.

Figure 5 shows the loss curve comparison with integral calculus.

FIGURE 2: Drifting variance from the designated position of the navigation mark #4 in 2019.

FIGURE 3: Test accuracy.

FIGURE 4: Train accuracy.

FIGURE 5: Comparison with integral calculus.

Figure 5 shows that the proposed algorithm has demonstrated more advantages than integral calculus in the field of neural networks with faster convergence speed.

5. Conclusions

This paper studied a fractional-order gradient descent with momentum for the RBF neural network to estimate the drifting trajectory of a navigation mark. In the study, fractional-order calculus was applied to a gradient descent with the momentum algorithm to train the neural network. The convergence of the proposed algorithm was then

proved, and the new algorithm was used to predict the drift of the navigation mark.

The limitation of the current algorithm is that it cannot solve the severe and irregular drifting, such as those produced by a typhoon. Further research is needed to improve the algorithm through artificial intelligence forecasting based on historical data.

Data Availability

The data used to support the findings of the study are included within this paper.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Q. L. Chen and Y. J. Lu, "Pattern recognition and numerical prediction of AIDS to navigation," *China Water Transport*, vol. 5, pp. 69–73, 2020.
- [2] X. Cheng, "Optimization study of anchor chain length of navigation navigation marks," *Journal of Waterway and Harbor*, vol. 40, no. 2, pp. 226–230, 2019.
- [3] S. Yao, Y. Wu, J. Yang, and J. Xu, "A three-term gradient descent method with subspace techniques," *Mathematical Problems in Engineering*, vol. 2021, 7 pages, 2021.
- [4] X. Liu and T. Lan, "Adaptive neural gradient descent control for a class of nonlinear dynamic systems with chaotic phenomenon," *Mathematical Problems in Engineering*, vol. 2015, 6 pages, 2015.
- [5] D. Utomo, "Stock price prediction using back propagation neural network based on gradient descent with momentum and adaptive learning rate," *Journal of Internet Banking and Commerce*, vol. 22, no. 3, pp. 1–16, 2017.
- [6] J. Wang, Y. Wen, Y. Gou, Z. Ye, and H. Chen, "Fractional-order gradient descent learning of BP neural networks with Caputo derivative," *Neural Networks*, vol. 89, pp. 19–30, 2017.
- [7] S. Khan, I. Naseem, M. A. Malik, R. Togneri, and M. Bennamoun, "A fractional gradient descent-based RBF neural network," *Circuits, Systems, and Signal Processing*, vol. 37, no. 12, pp. 5311–5332, 2018.
- [8] G. L. Yang, B. J. Zhang, Z. Y. Sang, J. Wang, and H. Chen, "A Caputo-type fractional-order gradient descent learning of BP neural networks," in *Proceedings of the 14th International Symposium, ISNN, Sapporo, Hakodate, and Muroran*, Hokkaido, Japan, June 2017.
- [9] J. J. Liu, R. Zhai, Y. H. Liu, W. L. Li, B. Z. Wang, and L. Y. Huang, "A quasi fractional order gradient descent method with adaptive stepsize and its application in system identification," *Applied Mathematics and Computation*, vol. 393, 2021.
- [10] Y. Wei, Y. Kang, W. Yin, and Y. Wang, "Generalization of the gradient method with fractional order gradient direction," *Journal of the Franklin Institute*, vol. 357, no. 4, pp. 2514–2532, 2020.
- [11] 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 Processing*, vol. 142, pp. 1–10, 2018.
- [12] Z. A. Khan, S. Zubair, H. Alquhayz, M. Azeem, and A. Ditta, "Design of momentum fractional stochastic gradient descent

- for recommender systems,” *IEEE Access*, vol. 7, pp. 179575–179590, 2019.
- [13] 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,” *The European Physical Journal Plus*, vol. 134, 2019.
 - [14] S. Zubair, N. I. Chaudhary, Z. A. Khan, and W. Wang, “Momentum fractional LMS for power signal parameter estimation,” *Signal Processing*, vol. 142, pp. 441–449, 2018.
 - [15] F. Zouari, K. Ben Saad, and M. Benrejeb, “Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor,” in *Proceedings of the 2013 International Conference on Control, Decision and Information Technologies, CoDIT 2013*, pp. 864–871, Hammamet, Tunisia, May 2013.
 - [16] F. Zouari, K. B. Saad, and M. Benrejeb, “Adaptive backstepping control for a class of uncertain single input single output nonlinear systems,” in *Proceedings of the 2013 10th International Multi-Conference on Systems, Signals and Devices*, Hammamet, Tunisia, March 2013.
 - [17] F. Zouari, K. B. Saad, and M. Benrejeb, “Robust adaptive control for a class of nonlinear systems using the backstepping method,” *International Journal of Advanced Robotic Systems*, vol. 10, p. 166, 2013.
 - [18] F. Zouari, K. B. Saad, and M. Benrejeb, “Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems,” *International Review on Modelling and Simulations*, vol. 5, no. 5, pp. 2075–2103, 2012.
 - [19] F. Zouari, A. Ibeas, A. Boulkroune, J. Cao, and M. M. Arefi, “Neuro-adaptive tracking control of non-integer order systems with input nonlinearities and time-varying output constraints,” *Information Sciences*, vol. 485, pp. 170–199, 2019.
 - [20] M. Haddad, F. Zouari, A. Boulkroune, and S. Hamel, “Variable-structure backstepping controller for multivariable nonlinear systems with actuator nonlinearities based on adaptive fuzzy system,” *Soft Computing*, vol. 23, no. 23, pp. 12277–12293, 2019.
 - [21] F. Zouari, “Neural Network Based adaptive backstepping dynamic surface control of drug dosage regimens in cancer treatment,” *Neurocomputing*, vol. 366, pp. 248–263, 2019.
 - [22] W. A. Sutherland, *Introduction to Metric and Topological Spaces*, p. 23, Oxford University Press, Oxford, UK, 1975.