International Journal of Difference Equations
ISSN 0973-6069, Volume 15, Number 1, pp. 1–10 (2020)
http://campus.mst.edu/ijde

Function Approximation Using a Discrete Fractional Order Gradient Descent Law

Mohamed Aburakhis and **Youssef N. Raffoul**
University of Dayton
Department of Mathematics
Dayton, 45469, US
aburakhism1@gmail.com
yraffoul1@udayton.edu

Abstract

We use discrete fractional calculus (DFC) to generalize the discrete-time gradient descent law. A discrete fractional-order gradient descent law (DFOGDL) is designed based on Caputo fractional difference in the form of a backward difference. We use DFOGDL to estimate the parameters of a classical integer order discrete-time system. Lastly, we prove the stability of estimating the unknown parameters of parametric function using DFOGDL.

AMS Subject Classifications: 65D15, 39A06.
Keywords: Function approximation, linear difference equations.

1 Introduction

Generally, function approximation is a method of finding a function in a class that matches the behavior of the function's uncertainty. According to Weierstrass theorem [10], any sufficiently smooth function can be approximated by a polynomial with sufficiently large coefficients.

Function approximation is a learning technique that is used to estimate the uncertainty Y by providing approximated function $\hat{Y}$. The learning process needs a dynamic operator to be performed such as differentiation or difference operators. The derivative is used as a memory in continuous-time systems and the difference operator is used in discrete-time systems, however, the authors in [13] used the difference operator instead of derivatives in the continuous-time system.

Received June 21, 2020; Accepted September 5, 2020
Communicated by Martin Bohner

Gradient descent algorithm is the most common iterative process that is used to approximate the uncertainty on-line. The idea of the gradient descent law is to drive the value of the estimated parameters towards their actual value by minimizing the cost function.

In this paper, we employ the advantages of the DFC to approximate uncertain functions via generalization of the gradient descent law in discrete-time domain to include the fractional orders, and then use it to approximate the uncertainty.

A discrete systems can be achieved by discretizing the continuous counterparts through a sampling process, where the sampling time has to be carefully chosen. Or some systems are naturally in discrete form such as a model described by difference equations that represent the number of infected people per day, where the number of infected individuals changes at only certain time intervals. In general, the difference operator has two criteria, forward difference and backward difference, in this paper, we use the backward difference due its features over the forward difference, which is explained later in the paper.

Any discrete-time signal has a value for every discrete point of time. In this paper, we will symbolize these discrete points of time by the constant normalized integer index k.

Historically, inordinate attention has been paid to continuous fractional calculus, and only in the last few decades DFC has come into its own [1–3, 5, 7, 8, 12]. Atici and Eloe developed and applied a transformation method for fractional calculus problems that expanded upon the approach formulated by Miller and Ross's exploration of fractional difference equations [6, 7].

The stability of the delta fractional order Caputo difference equation has been studied in [8]. The stability theorem for a discrete fractional Lyapunov direct method has been proven. In [4], we generalized the order of gradient descent law in continuous-time to approximate the uncertainty of the controller's parameter. However, to our knowledge, there are no studies that use discrete fractional order techniques for estimating the parameter of classical discrete systems. In this paper, we utilize fractional order calculus, in the gradient descent law, to increase the degrees of freedom and thus provide extra flexibility to the designer.

More specifically, DFC is employed for generalizing the difference operator of the classical gradient descent law. The new DFOGDL is used to estimate the parameters of structured uncertainty based on backward difference technique. The stability analysis of using DFOGDL is performed based on the stability analysis of using classical integer order gradient descent law.

2 Preliminaries

This section seeks to define and explain the underlying concepts behind the DFC.

Definition 2.1 (See [7, 9])**.** For a function $x(k)$, the forward and backward difference operators Delta (Δ) and Nabla (∇) are defined by

$$\Delta^m x(k) = \Delta(\Delta^{m-1}x(k)), \tag{2.1}$$

$$\nabla^m x(k) = \nabla(\nabla^{m-1}x(k)), \tag{2.2}$$

where $\Delta x(k) = x(k+1) - x(k)$, $\nabla x(k) = x(k) - x(k-1)$, $m = 1, 2, 3, \dots$, and $k \in \mathbb{N}_a = \{a, a+1, \dots\}$.

For the sake of simplicity, we choose the initial index to be $a = 0$. Next definition will introduce the generalization of the falling and raising factorial power.

Definition 2.2 (See [7, 9])**.** The generalization of the raising factorial for non-integer real power α is

$$k^{\overline{\alpha}} = \frac{\Gamma(k+\alpha)}{\Gamma(k)}, \tag{2.3}$$

and the generalization of the falling factorial for non-integer real power is

$$k^{\underline{\alpha}} = \frac{\Gamma(k+1)}{\Gamma(k-\alpha+1)}, \tag{2.4}$$

where $\Gamma(\cdot)$ is the Gamma function, $\alpha \in \mathbb{R}$.

The fractional order sum introduced next is a generalization of the integer order one.

Definition 2.3 (See [7, 9])**.** For $f : \mathbb{N}_a \to \mathbb{R}$, the backward fractional sum ($\nabla^{-\alpha}$) of order v is defined by

$$\nabla_a^{-\alpha} f(k) = \frac{1}{\Gamma(\alpha)} \sum_{s=a}^{k} (k-s+1)^{\overline{\alpha-1}} f(j), \tag{2.5}$$

where $k \in \mathbb{N}_a$.

Definition 2.4 (See [7, 9])**.** For $f : \mathbb{N}_a \to \mathbb{R}$, the forward fractional sum ($\Delta^{-\alpha}$) of order α is defined by

$$\Delta_a^{-\alpha} f(k) = \frac{1}{\Gamma(\alpha)} \sum_{s=a}^{k-\alpha} (k-s-1)^{\underline{\alpha-1}} f(j), \tag{2.6}$$

where a is the initial value and $k \in \mathbb{N}_{a+\alpha}$.

The DFOGDL will be designed using the backward operator (∇^{-v}) because it maps functions defined on $\mathbb{N}_a$ to functions defined on $\mathbb{N}_{a+1}$, which is the same domain. On the contrary, the operator Δ^{-v} maps functions from $\mathbb{N}_a$ to $\mathbb{N}_{a+v}$. The backward Caputo fractional difference for order α is

$${}^{C}\nabla_a^{\alpha} f(k) = \frac{1}{\Gamma(m-\alpha)} \sum_{s=a}^{k} (k-s+1)^{\overline{m-\alpha-1}} \nabla^m f(j). \tag{2.7}$$

3 Gradient Descent Law and Problem Statement

In this section, we will introduce the parametric discrete function that will be approximated by the gradient descent algorithm. The classical gradient descent law will be introduced including its stability analysis. The stability will be proven using the matrix convergence stability. Equation (3.1) represents the linear parametric model, which will be considered. The unknown parameter vector $W \in \mathbb{R}^n$ will be estimated on-line using $\hat{W}(k)$ vector.

$$Y(k) = W^\top \Phi(x(k)), \tag{3.1}$$

where $Y \in \mathbb{R}$ and $\Phi(x(k)) \in \mathbb{R}^n$ is a known regressor.

We define the parameter error vector as a difference between the estimated parameter vector and the actual unknown vector $\tilde{W}(k) = \hat{W}(k) - W$. Then, we can represent the approximation error as a function of the parameter error as follows

$$\begin{aligned}\Psi(k) &= \hat{y}(k) - y(k),\\ &= \hat{W}^\top(k)\Phi(x(k)) - W^\top\Phi(x(k)),\\ &= \tilde{W}^\top(k)\Phi(x(k)).\end{aligned} \tag{3.2}$$

The normalized gradient descent algorithm [11] will be applied to update the estimated parameter vector on-line. It can be designed based on the cost function. Here, the cost function that we need to minimize is the approximation error Ψ. We defined it as

$$J(k) = \frac{\Psi^2(k)}{2m(k)}, \tag{3.3}$$

where $m = \nu + \|\Phi(k)\|^2$. The cost function J is minimized with respect to the estimated parameter vector $\hat{W}$ in a steepest descent direction. Then, the gradient descent law is

$$\hat{W}(k) = \hat{W}(k-1) - \gamma\frac{\partial J(k-1)}{\partial \hat{W}(k-1)}, \tag{3.4}$$

$$= \hat{W}(k-1) - \gamma\frac{\Phi(x(k-1))\Psi(k-1)}{\nu + \|\Phi(x(k-1))\|^2}, \tag{3.5}$$

where $\nu > 0$ and $\gamma > 0$ is the learning gain. The difference equation (3.5) can be rewritten as

$$\nabla\hat{W}(k) = -\gamma\frac{\Phi(x(k-1))\Psi(k-1)}{m(k-1)}. \tag{3.6}$$

The parameter error vector can be represented as

$$\tilde{W}(k) = \tilde{W}(k-1) - \gamma\frac{\Phi(x(k-1))\Psi(k-1)}{m(k-1)}, \tag{3.7}$$

$$= \tilde{W}(k-1) - \gamma\frac{\Phi(x(k-1))\Phi^\top(k-1)\tilde{W}(k-1)}{m(k-1)}, \tag{3.8}$$

$$= \tilde{W}(k-1) - \gamma A_m(k-1)\tilde{W}(k-1), \tag{3.9}$$

where $A_m(\cdot) = \dfrac{\Phi(x(\cdot))\Phi(x(\cdot))^\top}{m(\cdot)}$. At the step time $k+1$

$$\tilde{W}(k+1) = [I - \gamma A_m(k)]\tilde{W}(k), \tag{3.10}$$

where I is the n-by-n identity matrix. Now, let us consider the matrix $\Phi\Phi^\top$.

Theorem 3.1. *For the matrices A_1 and A_2,*

$$\mathrm{Rank}(A_1A_2) \leq \min\{\mathrm{Rank}(A_1), \mathrm{Rank}(A_2)\}. \tag{3.11}$$

Proof. Since the rows of the matrix A_1A_2 are linear combinations of the rows of the matrix A_2, the number of linearly independent rows of the matrix A_1A_2 is less than the number of linearly independent rows of the matrix A_2. Hence,

$$\text{Rank}(A_1A_2) \leq \text{Rank}(A_2). \tag{3.12}$$

Apply the same argument to the columns of A_1

$$\text{Rank}(A_1A_2) \leq \text{Rank}(A_1), \tag{3.13}$$

where the row and column ranks of (A_1) are equal since $\text{Rank}(A_1) = \text{Rank}(A_1^\top)$. □

Lemma 3.2. *The positive semi-definite matrix $A = vv^\top$, where v be an n-by-1 non-zero vector, has only one non-zero eigenvalue:*

$$\lambda(A) = \|v\|^2. \tag{3.14}$$

Proof. By using the rank-nullity theorem that states that if A is an n-by-n matrix, the rank of A plus the nullity of A is equal to n. The kernel of A is

$$\text{kernel}(A) = \{u : Au = 0\}, \tag{3.15}$$

where the dimension of the kernel of the matrix A is the nullity of A. From Theorem 3.1, the rank of A is 1, thus, it has $n-1$ linearly independent solutions. Moreover, the range of A is

$$\text{Range}(A) = \{b : \exists\, u \text{ with } Au = b\}, \tag{3.16}$$

where the rank of A is the dimension of the range of A which is equal 1, so A has $n-1$ linearly independent eigenvectors corresponding to $n-1$ zero eigenvalues. □

Since $A_m = \dfrac{\Phi\Phi^\top}{\nu + \Phi^\top\Phi}$ and $\nu > 0$, A_m is an n-by-n bounded positive semi-definite symmetric matrix with $n-1$ zero eigenvalues and only one positive less than one eigenvalue. The stability will be studied by using a matrix convergence analysis. It is well known that a matrix is convergent if and only if the modulus of that matrix is less than one. Thus, matrix $[I - \gamma A_m(k)]$, whose eigenvalues are $1 - \gamma\lambda(A_m(k))$, where $\lambda(A_m(k))$ is the eigenvalues of $A_m(k)$, is a convergent matrix if $|1 - \gamma\lambda(A_m(k))| < 1$, which means

$$\begin{gathered} -1 < 1 - \gamma\lambda(A_m(k)) < 1, \\ 0 < \gamma\lambda(A_m(k)) < 2. \end{gathered}$$

Hence, the learning rate could be chosen as follows

$$0 < \gamma < \frac{2}{\lambda_{\max}(A_m(k))} \tag{3.17}$$

to guarantee the convergence of the matrix $[I - \gamma A_m(k)]$. Therefore, the parameter error vector is stable if the condition (3.17) is satisfied. For a more robust but expensive calculation, the learning rate γ could be updated every iteration since the regressor $\Phi(x(k))$ is accessible and $\lambda_{\max}(A_m(k))$ can be calculated. Thus,

$$0 < \gamma(j) < \frac{2}{\lambda(A_m(j))} \tag{3.18}$$

as requirement to verify for $\gamma(j)$, with (3.10) can be rewritten as

$$\tilde{W}(k) = \tilde{W}(0) - \sum_{j=0}^{k-1} \gamma(j) A_m(j) \tilde{W}(j). \tag{3.19}$$

Note that the order of the difference operator in (3.6) is 1, which can be generalized to cover fractional orders. In the next section, we will generalize the classical integer order of the discrete gradient descent law to discrete fractional order gradient descent law.

4 Parameter Estimation Using DFOGDL Law

Consider the initial value theorem and its corollary.

Theorem 4.1 (See [9]). *Let $f : \mathbb{N}_{a+1} \to \mathbb{R}$ and $m - 1 < \alpha < m$, $m = 1, 2, \ldots$. Then, for $0 \leq N \leq m - 1$, the Caputo-based nabla fractional order initial value problem*

$$\begin{aligned} {}^{C}\nabla_a^{\alpha} z(k) &= f(k),\ k \in \mathbb{N}_{a+1} \\ \nabla^N z(a) &= c_N \end{aligned}$$

has the solution

$$z(k) = \nabla_a^{-\alpha} f(k) + \sum_{N=0}^{m-1} \frac{(k-a)^{\overline{N}}}{(N+1)!} z(a). \tag{4.1}$$

Noticeably, the initial conditions of difference equations based on the Caputo definition take on the same form as for an integer order difference equations.

Corollary 4.2. *For $f : \mathbb{N}_1 \to \mathbb{R}$ and $0 < \alpha < 1$, then, the Caputo-based nabla fractional order initial value problem*

$$\begin{aligned} {}^{C}\nabla_0^{\alpha} z(k) &= f(k),\ k \in \mathbb{N}_1, \\ z(0) &= z_0 \end{aligned}$$

has the solution

$$z(k) = z_0 + \nabla_a^{-\alpha} f(k),\ k \in \mathbb{N}_1. \tag{4.2}$$

Corollary 4.2 can be used to generalize the order of the difference operator in (3.6) to cover the fractional order difference operator of order $0 < \alpha \leq 1$, and then we will come up with DFOGDL,

$$\hat{W}(k) = \hat{W}(0) - \sum_{j=0}^{k-1} \gamma(j) C(k,j) \frac{\Phi(x(j))}{m(j)} \Psi(j), \tag{4.3}$$

where

$$C(k,j) = \frac{1}{\Gamma(\alpha)} \frac{\Gamma(k-(j+1)+\alpha)}{\Gamma(k-(j+1)+1)}. \tag{4.4}$$

Note that if $\alpha = 1$, the kernel $C(k,j) = 1$. That makes the classical discrete gradient descent law a special case of FO one. This gives us an additional design parameter and the design flexibility is increased. The fractional order parameter error vector is

$$\tilde{W}(k) = \tilde{W}(0) - \sum_{j=0}^{k-1} \gamma(j) C(k,j) A_m(j) \tilde{W}(j), \tag{4.5}$$

since $0 < \alpha \leq 1$ and $s \leq k-1$, $0 < C(k,j) \leq 1$. The boundedness condition of (4.5) can be concluded by comparing (3.19) with (4.5),

$$0 < \gamma(j) < \frac{2}{C(k,j)\lambda(A_m(j))}. \tag{4.6}$$

Since $C(k,j) \leq 1$,

$$0 < \gamma(j) < \frac{2}{\lambda(A_m(j))} \leq \frac{2}{C(k,j)\lambda(A_m(j))}. \tag{4.7}$$

It is clear that the stability condition when using DFGODL has a broader range than the classical gradient degree law.

5 Example

Consider the discrete-time system

$$Y(k) = \sin(4x) - 2\cos^2(4x), \tag{5.1}$$

$$= [1 \;\; -2]\begin{bmatrix}\sin(4x)\\ \cos^2(4x)\end{bmatrix}, \tag{5.2}$$

where $x(k) \in [0, 2\pi]$ and $k = 1, 2, \ldots, 200$.

For the sake of comparison, we will fix the learning rate to be $\gamma = 1.8$, and we will simulate for $\alpha = 0.8,\ 0.9$ and the classical integer order case $\alpha = 1$. Figure 5 plots the true function and its estimate. Figure 5 shows the behavior of the parameters convergence. The plot shows that the use of the DFOGDL improves the convergence behavior. By comparing the DFOGDL (4.3) and the classical gradient descent law (3.5), we note that the DFOGDL saves all the past values of the updated parameters while the classical one is merely updating the instantaneous value.

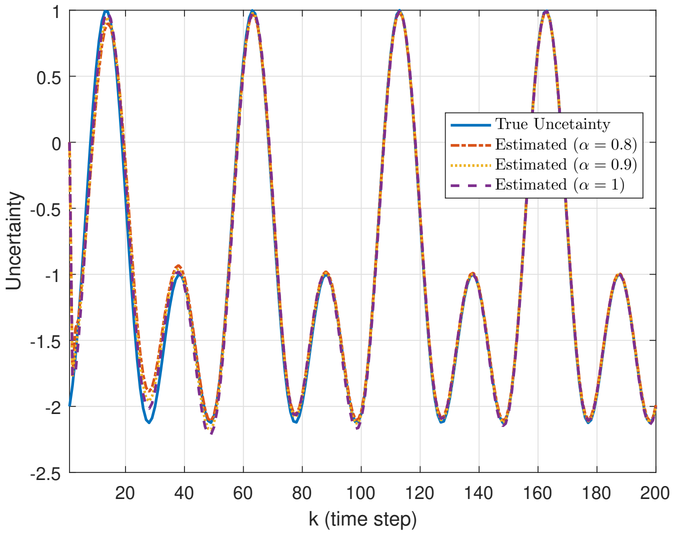

Figure 5.1: Uncertainty estimation using classical gradient descent law and DFOGDL laws ($\alpha = 0.8$, 0.9, and 1).

6 Conclusion

We generalized the order of the gradient descent law, used the new law (DFOGDL) for approximating the function and estimating the parameters on-line, and also proved the stability of using the DFOGDL for function approximation. The use of DFOGDL to

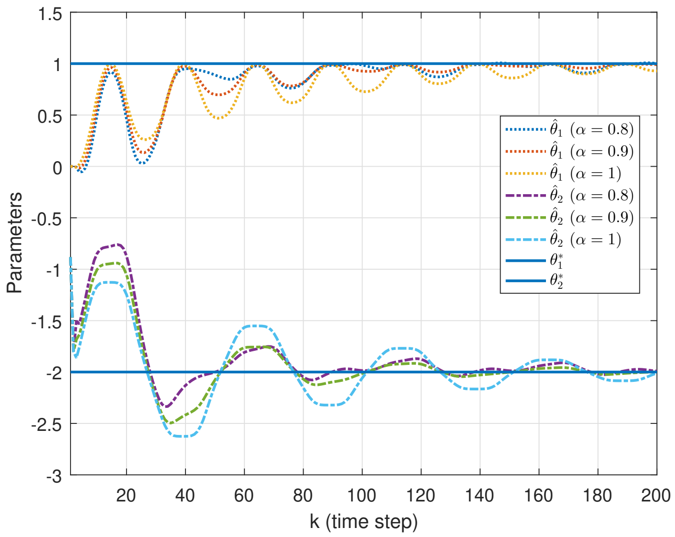

Figure 5.2: Parameter estimation using classical gradient descent law and DFOGDL laws ($\alpha = 0.8$, 0.9, and 1).

estimate an IO system parameter exhibits good behavior and faster parameter convergence. Moreover, it increases the degree of freedom and offers superior performance and design flexibility over the classical model. However, even with the generalization of the difference equation, the new fractional order law may lose its ability to identify the true parameters when the persistence of excitation of the regressor is absent.

References

[1] Thabet Abdeljawad. On Riemann and Caputo fractional differences. *Computers & Mathematics with Applications*, 62(3):1602–1611, 2011.

[2] Thabet Abdeljawad. On delta and nabla caputo fractional differences and dual identities. *Discrete Dynamics in Nature and Society*, 2013, 2013.

[3] Thabet Abdeljawad and Dumitru Baleanu. Fractional differences and integration by parts. *Journal of Computational Analysis & Applications*, 13(3), 2011.

[4] Mohamed Aburakhis and Raúl Ordóñez. Interaction of fractional order adaptive law and fractional order pid controller for the ball and beam control system. In *NAECON 2018-IEEE National Aerospace and Electronics Conference*, pages 451–456. IEEE, 2018.

[5] Ferhan Atici and Paul Eloe. Initial value problems in discrete fractional calculus. *Proceedings of the American Mathematical Society*, 137(3):981–989, 2009.

[6] Ferhan M Atici and Paul W Eloe. A transform method in discrete fractional calculus. *International Journal of Difference Equations*, 2(2), 2007.

[7] Ferhan M Atıcı and Paul W Eloe. Discrete fractional calculus with the nabla operator. *Electronic Journal of Qualitative Theory of Differential Equations*, 2009(3):1–12, 2009.

[8] Dumitru Baleanu, Guo-Cheng Wu, Yun-Ru Bai, and Fu-Lai Chen. Stability analysis of caputo–like discrete fractional systems. *Communications in Nonlinear Science and Numerical Simulation*, 48:520–530, 2017.

[9] Christopher Goodrich and Allan C Peterson. *Discrete fractional calculus*. Springer, 2015.

[10] Jeffrey T. Spooner, Manfredi Maggiore, Raúl Ordóñez, and Kevin M. Passino. *Stable adaptive control and estimation for nonlinear systems: neural and fuzzy approximation techniques*, volume 15. New York: Wiley, 2002.

[11] Gang Tao. *Adaptive control design and analysis*, volume 37. John Wiley & Sons, 2003.

[12] Guo-Cheng Wu, Dumitru Baleanu, and Wei-Hua Luo. Lyapunov functions for Riemann–Liouville-like fractional difference equations. *Applied Mathematics and Computation*, 314:228–236, 2017.

[13] Tansel Yucelen and Anthony J Calise. Derivative-free model reference adaptive control. *Journal of Guidance, Control, and Dynamics*, 34(4):933–950, 2011.