Iranian Journal of Fuzzy Systems

Volume 21, Number 2, (2024), pp. 67-85

Original paper

Solving variable-order fractional delay differential algebraic equations via fuzzy systems with application in delay optimal control problems

M. Mortezaee [1], M. Ghovatmand [2] and A. R. Nazemi [3]

[1,2,3] *Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran*

mortezaee91@gmail.com, mghovat@shahroodut.ac.ir, nazemi20042003@gmail.com

Abstract

In this paper, a new approach based on fuzzy systems is used for solving variable-order fractional delay differential algebraic equations. The fractional derivatives are considered in the Atangana-Baleanu sense that is a new derivative with the non-singular and non-local kernel. By relying on the ability of fuzzy systems in function approximation, the fuzzy solutions of variables are substituted in variable-order fractional delay differential algebraic equations. The obtained algebraic equations system is then transformed into an error function minimization problem. A learning algorithm is used to achieve the adjustable parameters of fuzzy solutions. It is shown that the variable-order fractional delay optimal control problems can be reformulated as variable-order fractional delay differential algebraic equations and solved by the proposed method. The efficiency and accuracy of the presented approach are assessed through some illustrative examples of the variable-order fractional delay differential algebraic equations.

Keywords: Variable-order fractional delay differential algebraic equations, Atangana-Baleanu derivative, fuzzy system, optimization, variable-order fractional delay optimal control problems.

1 Introduction

Fractional calculus or the study of fractional order integral and derivative operators emerged in 1695 with a very deep question raised in a letter of L'Hospital to Leibniz. For a long time, due to the existence of multiple nonequivalent definitions of fractional derivatives and their nonlocal character, fractional calculus was not been considered by researchers. However, in recent decades, fractional calculus has been used as a powerful tool in many branches of science and engineering, such as physics, economics, chemistry, signal and image processing, biology, and control theory [34, 36]. Fractional differential equations (FDEs) provide an exact description of different nonlinear phenomena and inherent relation to various materials and processes with memory, hence, fractional order modeling of many real phenomena has more advantages and consistency rather than classical integer-order mathematical modeling. More details regarding the theory and applications of fractional calculus can find in [8].

Although the FDEs are capable of addressing some very relevant physical problems, it cannot exactly describe important classes of physical phenomena where the order itself is a function of either dependent or independent variables. So it is significant to develop the concept of variable-order calculus. Variable-order fractional operators are an extension of constant-order fractional operators the order can vary continuously as a function of time, space or an independent external variable. Since the kernel of the variable-order operators has a variable exponent, obtaining analytical solutions of variable-order fractional differential equations (V-OFDEs) is difficult. Therefore, numerical solutions have become the key to solving V-OFDEs. Recently, researchers have presented some numerical methods for solving V-OFDEs. For instance, one can refer to numerical methods based on Legendre wavelets functions [7, 11], finite difference schemes in [6, 33, 40], methods based on the fundamental theorem of fractional calculus and the two-step Lagrange polynomial in [32], method of approximate particular solutions in [12], the method based on cubic spline interpolation [42], simplified

Corresponding Author: A. R. Nazemi

Received: September 2022; Revised: October 2023; Accepted: April 2024.

https://doi.org/10.22111/IJFS.2024.43498.7646

reproducing kernel method in [17], reproducing kernel methods in [19, 20], optimization method based on generalized polynomials in [9] and characteristic finite difference method [30].

Numerical methods based on Artificial Intelligence have attracted the attention of researchers because of their unique merits in the ability to approximate complicated nonlinear functions with simple models. The primary advantage of these methods is that the obtained solutions are differentiable and in closed analytic form. Fuzzy systems are a category of artificial intelligence tools. Fuzzy systems can provide a more transparent representation of the studied system, compared with the others Artificial Intelligence methods such as artificial neural networks. The existence of this merit in fuzzy systems is due to the possible linguistic interpretation in the form of rules. These techniques have universal approximation capabilities, hence, they have been used for a wide variety of applications (see [5, 10, 23, 24, 25, 26, 27, 29, 39, 41, 43]).

Fractional differential-algebraic equations (FDAEs) are composed of fractional differential equations and algebraic equations. Many important mathematical models can be expressed in terms of a system of differential-algebraic equations with fractional order [4, 31, 37]. This kind of mathematical model has received much attention; nevertheless, the numerical methods in this field are still young; a few studies have been considered on the numerical methods for solving FDAEs. For example, Numerical Methods based on Said Ball Curve [15], Chebyshev Pseudo spectral [1], Sliding Mode Control [35], homotopy analysis [44], Generalized JacobiGalerkin [13] and Bezier curves [14].

Fractional delay differential-algebraic equations (FDDAEs) are a category of FDAEs that include one or multiple delays in the variable or in its derivative. FDDAEs are more accurate in describing some scientific and engineering problems with memory function and algebraic restrictions. Since the concept of FDDAEs is a new subject in mathematics, the number of existing approaches to solve them is limited [21]. As the generalized form of FDDAEs, the variable-order fractional delay differential algebraic equations (V-OFDDAEs) means the delay differential algebraic equations with variable-order fractional derivatives. To the best of our knowledge, there has been no research focusing on V-OFDDAEs, until now.

Motivated by the above mentioned discussions, the main objective of this paper is to present an efficient fuzzy system for solving V-OFDDAEs with Atangana-Baleanu (AB) derivative. by relying on the functional approximation capability of the fuzzy systems via fuzzy interpolation scheme, the fuzzy solutions of variables are substituted in V-OFDDAEs. This substituting lead to reducing the V-OFDADEs to a simpler problem that consists of solving a system of algebraic equations. Finally, the parameters of the fuzzy system are adjusted to minimize an appropriate error function via a learning algorithm. Also, it is shown that variable-order fractional delay optimal control problem (V-OFDOCP) can be reformulated as Hamiltonian V-OFDDAE. The derived V-OFDDAE can then be solved by the proposed fuzzy system. Here, the contributions of this paper are briefly mentioned as follows:

- A novel concept of FDAEs is introduced by using the concept of variable-order fractional derivatives in the Atangana-Baleanu type.
- There is not any report about solving this problem.
- A new formulation of V-OFDOCPs with AB derivative is proposed.
- To solve the new problem, a computational method based on the fuzzy system is proposed, which has not been investigated in the literature, to the best of our knowledge.
- An upper error bound between the exact solution and the proposed fuzzy solution with respect to the number of fuzzy rules and solution errors is obtained.
- The convergence of the proposed fuzzy system is proved.

The outline of this paper is as follows. Some fundamental concepts regarding fractional order calculus are described in Section 2. In Section 3, a brief introduction to the fuzzy system approach is provided. Next, a new method based on the fuzzy system is utilized to achieve the solution of V-OFDDAEs. A learning optimization algorithm to adjust the parameters of the fuzzy system is then presented in Section 4. In Section 5, it is shown that the V-OFDOCP can be rewritten as a V-OFDDAE. The usefulness of the proposed method is illustrated by providing several numerical examples in Section 6. Finally, a conclusion is given in Section 7.

2 Fractional calculus

In 1993 , Ross and Samko introduced operators that the order is not a constant during the process, but variable on time. By developing variable-order fractional calculus theory, many new definitions are proposed. Here, the Atangana-Baleanu type definition of the variable-order fractional derivative is given which is used in this paper.

Definition 2.1. [2] *Let f be a function such that $f \in H^1(0,T), T > 0$, $\alpha(t) \in (0,1)$ and $M(\alpha(t))$ be a normalization function with $M(0) = M(1) = 1$. The definition of the left Atangana-Baleanu (AB) fractional derivative in the Caputo sense is given as*

$$ {}_0^{ABC}D_t^{\alpha(t)} f(t) = \frac{M(\alpha(t))}{1-\alpha(t)} \int_0^t E_{\alpha(t)}(-\frac{\alpha(t)}{1-\alpha(t)}(t-\tau)^{\alpha(t)})\dot{f}(\tau)d\tau, \tag{1} $$

and in the Riemann-Liouville sense is defined by

$$ {}_0^{ABR}D_t^{\alpha(t)} f(t) = \frac{M(\alpha(t))}{1-\alpha(t)} \frac{d}{dt} \int_0^t E_{\alpha(t)}(-\frac{\alpha(t)}{1-\alpha(t)}(t-\tau)^{\alpha(t)}) f(\tau)d\tau, \tag{2} $$

where E_α denotes the generalized Mittag-Leffler function

$$ E_\alpha = \sum_{k=1}^{\infty} \frac{t^k}{\Gamma(\alpha k + 1)}. $$

Definition 2.2. [2] *The definition of the right AB fractional derivative in the Caputo sense is given as*

$$ {}_t^{ABC}D_T^{\alpha(t)} f(t) = -\frac{M(\alpha(t))}{1-\alpha(t)} \int_t^T E_{\alpha(t)}(-\frac{\alpha(t)}{1-\alpha(t)}(\tau-t)^{\alpha(t)})\dot{f}(\tau)d\tau, \tag{3} $$

and in the Riemann-Liouville sense is defined by

$$ {}_t^{ABR}D_T^{\alpha(t)} f(t) = -\frac{M(\alpha(t))}{1-\alpha(t)} \frac{d}{dt} \int_t^T E_{\alpha(t)}(-\frac{\alpha(t)}{1-\alpha(t)}(\tau-t)^{\alpha(t)}) f(\tau)d\tau. \tag{4} $$

The relations between the left and right AB fractional derivatives in the Riemann-Liouville sense and the Caputo sense are defined as follows

$$ {}_0^{ABC}D_t^{\alpha(t)} f(t) = {}_0^{ABR}D_t^{\alpha(t)} f(t) - \frac{M(\alpha(t))}{1-\alpha(t)} f(0) E_{\alpha(t)}(-\frac{\alpha(t)}{1-\alpha(t)} t^{\alpha(t)}), \tag{5} $$

$$ {}_t^{ABC}D_T^{\alpha(t)} f(t) = {}_t^{ABR}D_T^{\alpha(t)} f(t) - \frac{M(\alpha(t))}{1-\alpha(t)} f(T) E_{\alpha(t)}(-\frac{\alpha(t)}{1-\alpha(t)} (T-t)^{\alpha(t)}). \tag{6} $$

The following definition provides the formula of integration by parts for AB fractional derivatives that is essential for proving results concerning variational problems.

Definition 2.3. [2] *Suppose that $f \in H^1(0,T), T > 0$, and $\alpha(t) \in (0,1)$. Then*

$$ \int_0^T {}_0^{ABC}D_t^{\alpha(t)} f(t) g(t) dt = \int_0^T f(t) {}_t^{ABR}D_T^{\alpha(t)} g(t) dt + \frac{M(\alpha(t))}{1-\alpha(t)} f(t)(\mathbf{e}^1_{\alpha(t),1,\frac{-\alpha(t)}{1-\alpha(t)},T} - g)(t)|_0^T, $$

where $(\mathbf{e}^\gamma_{\rho,\mu,\omega,b} - \varphi)(x) = \int_x^b (t-x)^{\mu-1} E^\gamma_{\rho,\mu}[\omega(t-x)^\rho]\varphi(t)dt,\ x < b$, and $E^\gamma_{\rho,\mu}(z) = \sum_{k=0}^\infty \frac{(\gamma)_k z^k}{\Gamma(\rho k + \mu) k!}$.

In this paper, the FDDAE is generalized by replacing the classical derivative by the Atangana-Baleanu variable-order fractional derivative. These systems can be presented by the following formula

$$ E {}_0^{ABC}D_t^{\alpha(t)} x(t) = \mathcal{A}x(t) + \mathcal{B}x(t-\tau) + \mathcal{F}(t), 0 < t \le T, \tag{7} $$

$$ x(t) = \phi(t), \quad -\tau \le t \le 0, \tag{8} $$

where $E, \mathcal{A}, \mathcal{B} \in \mathbb{R}^{n,n}$, $\det(E) = 0$, $\mathcal{F} : [0,T] \to \mathbb{R}^n$ and $\tau \in {\mathbb{R}^+}^n$ is a vector of constant delay.

3 Fuzzy system approach

A fuzzy system consists of four principal parts whose basic configuration is depicted in Figure 1,

a) a fuzzzifier, which is a map from crisp points into fuzzy sets in the input space,

b) fuzzy rule base, a set of fuzzy rules that perform the fuzzy system. The fuzzy rules have the following form:

$$\begin{cases} R_j \text{ (}j\text{th rule)}: & \text{If } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and} \cdots \text{and } x_n \text{ is } A_n^j, \\ & \text{Then } z \text{ is } B^j, \end{cases} \tag{9}$$

where n is the number of input variables, $x_i, (i = 1, \cdots, n)$ and z are the input and output variables, respectively and A_i^j and B^j are linguistic terms characterized by fuzzy membership functions $\mu_{A_i^j}(x_i)$ and $\mu_{B^j}(z)$, respectively,

c) inference engine, which determines a mapping based on fuzzy logic operations on fuzzy rule base,

d) a defuzzifier, which is a map from fuzzy sets into crisp points in the output space.

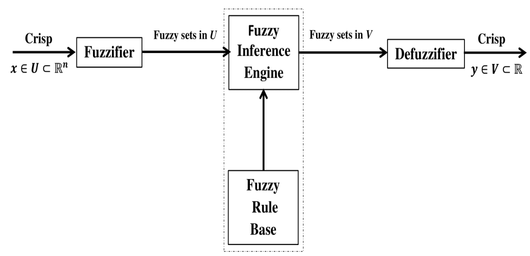


Figure 1: Basic configuration of fuzzy systems.

A subset of the fuzzy systems of Figure 1 with singleton fuzzifier, product inference, centroid defuzzifier, and Gaussian membership function consists of all functions of the form

$$f(x) = \frac{\sum_{j=1}^{m} c_j (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}{\sum_{j=1}^{m} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}, \tag{10}$$

where $f : U \subset \mathbb{R}^n \longrightarrow \mathbb{R}$, $x = (x_1, x_2, \ldots, x_n) \in U$, $\mu_{A_i^j}(x_i)$ is a Gaussian membership function and m is the number of fuzzy rules.

Wang and Mendel in [38] proved that fuzzy systems (10) can uniformly approximate any nonlinear continuous function under a compact set to any degree of accuracy. This property can be stated by means of the following theorem:

Theorem 3.1. (Universal Approximation Theorem) [38] *For any given real continuous $g(x)$ on the compact set $U \subset \mathbb{R}^n$ and arbitrary $\varepsilon > 0$, there exists $f(x) \in Y$ (Y is the set of all the functions* (10)*) such that*

$$\sup_{x \in U} |f(x) - g(x)| < \varepsilon. \tag{11}$$

In this paper we consider a fuzzy system with an input variable $t \in [0, T]$ and output variable $f(t) \in \mathbb{R}$ which can be implemented by the following function:

$$f(t) = \sum_{i=1}^{m} c_i \frac{\mu_i}{\sum_{i=1}^{m} \mu_i} = C^{\mathbb{T}} \Psi, \tag{12}$$

where $C = [c_1, c_2, \ldots, c_m]^{\mathbb{T}}$, $\Psi = [\psi_1, \psi_2, \ldots, \psi_m]^{\mathbb{T}}$, $\psi_i = \dfrac{\mu_i}{\sum_{i=1}^{m} \mu_i}$ and μ_i $(i = 1, 2, \ldots, m)$ are Gaussian membership functions in the following form:

$$\mu_i = \exp(-\frac{1}{2}(\frac{t - a_i}{\sigma_i})^2), \tag{13}$$

where a_i and σ_i are the mean values and the standard deviations of the membership distributions, respectively. These parameters are assigned as follows:

$$a_i = (i - 1)\frac{T}{m - 1}, \tag{14}$$

$$\sigma_i = \frac{T}{m - 1}, \tag{15}$$

where $m > 1$ is the numbers of fuzzy rules.

Remark 3.1. *The Gaussian function is a nonlinear function in C^∞. Hence this makes the approximate solution* (7)-(8) *smooth, continuous and differentiable in C^∞ in order to save the properties of the exact solution.*

Remark 3.2. *From* (15) *, it is easy to find out that the standard deviations σ_i will be forced to be zero with infinite numbers of membership functions, i.e., $\lim_{m\to\infty} \sigma_i = 0$. Therefore, when m approaches to ∞ membership function in* (13) *degenerate into the singleton case from the Gaussian case.*

Based on Theorem 3.1, it is proved that any continuous function $g(t) \in [0, T]$ can be approximated by the proposed fuzzy system $f(t)$ with a sufficient number of fuzzy rules. In the following, an upper bound error is derived for this approximation. Next the convergent property for the case $m \to \infty$, i.e., all points t in the domain $[0, T]$ are considered, is discussed.

Theorem 3.2. *Assume that $g(t)$ is a continuously differentiable function and there exists a set of parameters $c_i = g(t_i)$ as $m \to \infty$. Then*

I) $|g(t)-f(t)| \le 0.83452 g_s \Delta_t + \max_i |e_i|$, *where* $\Delta_t = \dfrac{T}{m-1}$, $e_i = g(t_i) - c_i$ *is error factor and* $g_s = \sup_{t\in[0,T]} |\dfrac{dg(t)}{dt}|$.

II) $\lim_{m\to\infty} f(t) = g(t)$.

Proof. I) We have

$$|g(t) - f(t)| = |g(t) - \sum_{i=1}^{m} c_i \frac{\mu_i}{\sum_{i=1}^m \mu_i}| = |\sum_{i=1}^{m} (g(t) - c_i) \frac{\mu_i}{\sum_{i=1}^m \mu_i}| \le \sum_{i=1}^{m} |(g(t) - c_i)| \frac{\mu_i}{\sum_{i=1}^m \mu_i}. \tag{16}$$

On the other hand

$$|g(t) - c_i| = |g(t) - g(t_i) + g(t_i) - c_i| = |g(t) - g(t_i) + e_i| \le |g(t) - g(t_i)| + |e_i|.$$

From the mean value theorem, we have the following result

$$|g(t) - c_i| \le \left| \frac{dg(t)}{dt}|_{\hat{t}_i} (t - t_i) \right| + |e_i| \le \left| \frac{dg(t)}{dt}|_{\hat{t}_i} \right| |(t - t_i)| + |e_i| \le g_s |(t - t_i)| + |e_i|, \tag{17}$$

where $\hat{t}_i$ are some values between t and t_i. By substituting (17) into (16), the following equation can be obtained:

$$\begin{aligned} |g(t) - f(t)| &\le \frac{\sum_{i=1}^m g_s |(t - t_i)| \mu_i}{\sum_{i=1}^m \mu_i} + \frac{\sum_{i=1}^m |e_i| \mu_i}{\sum_{i=1}^m \mu_i} \le \frac{\sum_{i=1}^m g_s |(t - t_i)| \mu_i}{\sum_{i=1}^m \mu_i} + \frac{\sum_{i=1}^m \max_i |e_i| \mu_i}{\sum_{i=1}^m \mu_i} \\ &\le \frac{\sum_{i=1}^m g_s |(t - t_i)| \mu_i}{\sum_{i=1}^m \mu_i} + \max_i |e_i|. \end{aligned} \tag{18}$$

The first term in the right side of (18) can be reduced to

$$\theta := g_s \frac{\sum_{i=1}^m |(t - t_i)| \mu_i}{\sum_{i=1}^m \mu_i},$$

where

$$\mu_i = \exp(-\frac{1}{2}(\frac{t - t_i}{\Delta_t})^2), \quad t_i = (i - 1)\Delta_t \quad i = 1, 2, \ldots, m.$$

Denote $t = (j - 1)\Delta_t, \quad j \in [1, m]$, then

$$\theta = g_s \Delta_t \frac{\sum_{i=1}^m |(j - i)| \exp(-\frac{1}{2}(j - i)^2)}{\sum_{i=1}^m \exp(-\frac{1}{2}(j - i)^2)}.$$

It can be verified numerically that for $j \in [1, m]$ and $m \ge 2$

$$\frac{\sum_{i=1}^m |(j - i)| \exp(-\frac{1}{2}(j - i)^2)}{\sum_{i=1}^m \exp(-\frac{1}{2}(j - i)^2)} \le 0.83452.$$

Hence

$$\theta \le 0.83452 g_s \Delta_t. \tag{19}$$

This completes the proof of I.

II) From (I), the approximation upper bound between the exact solution and the proposed fuzzy solution can be described by the following equation

$$|g(t) - f(t)| \leq 0.83452 g_s \Delta_t + \max_i |e_i|. \tag{20}$$

Take the limit for both sides of (20) with m approaching infinity and note the fact that g_s is bound and $\Delta_t \to 0$. We have

$$|g(t) - \lim_{m\to\infty} f(t)| \leq \lim_{m\to\infty} \max_i |e_i|. \tag{21}$$

From Remark 3.2 and (21)

$$|g(t) - \lim_{m\to\infty} f(t)| = 0, \tag{22}$$

i.e., $g(t) = \lim_{m\to\infty} f(t)$. The proof is complete.

□

Due to the universal approximation property of fuzzy system, the defined fuzzy system (12) can be used for approximating the unknown solutions of the initial value problem (7) and (8). Hence with the help of fuzzy system (12), the following fuzzy solution models are introduced to approximate the solutions of the V-OFDDAEs (7) as follows:

$$FS_{x_d} = \sum_{i=1}^{m} c_i^{x_d} \frac{\mu_i^{x_d}}{\sum_{i=1}^{m} \mu_i^{x_d}} = C_{x_d}^{\mathbb{T}} \Psi, \quad d = 1, 2, \ldots, n. \tag{23}$$

According to the initial conditions, the fuzzy solutions can be selected as:

$$x_{FS_d}(C_{x_d}, t) = \phi(0) + t FS_{x_d}, \quad d = 1, 2, \ldots, n. \tag{24}$$

Therefore each fuzzy solution is obtained from summation of two terms. The first term contains no adjustable parameters and satisfies the initial conditions. The second term employs a fuzzy solution with adjustable parameters. The number of fuzzy rules can be different for each fuzzy solution. By replacing the fuzzy solutions (24) into V-OFDDAEs (7), we have

$$E_0^{ABC} D_t^{\alpha(t)} x_{FS}(t) = \mathcal{A} x_{FS}(t) + \mathcal{B} x_{FS}(t-\tau) + \mathcal{F}(t), 0 < t \leq T, \tag{25}$$

where $x_{FS} = [x_{FS_1}, x_{FS_2}, \ldots, x_{FS_n}]$ and $\Upsilon = [C_{x_1}, C_{x_2}, \ldots, C_{x_n}]$ is the vector of all adjustable parameters in fuzzy solutions (24). We now define the following approximated error functions with the fuzzy solutions as

$$\mathcal{E}(\Upsilon, t_i) = \Big(\mathcal{A} x_{FS}(t) + \mathcal{B} x_{FS}(\Upsilon, t_i - \tau) + \mathcal{F}(t_i) - E_0^{ABC} D_t^{\alpha(t)} x_{FS}(\Upsilon, t_i)\Big)^2, i = 2, 3, ..., m,$$

where $t_i = (i-1)\Delta_t$, $i = 1, 2, ..., m$, are collocation points in interval $[0, T]$. In order to find the parameters of fuzzy solutions (24), we consider the following unconstrained optimization problem

$$\min_{\Upsilon} \mathcal{E}(\Upsilon) = \frac{1}{2} \parallel \eta(\Upsilon) \parallel_2^2, \tag{26}$$

where $\eta(\Upsilon) = [\mathcal{E}(\Upsilon, t_1), \mathcal{E}(\Upsilon, t_2), \ldots, \mathcal{E}(\Upsilon, t_m)]^{\mathbb{T}}$ is the vector of the collection of error functions in t_i. In this way, the initial value problem (7)-(8) is transformed into an unconstrained minimization problem. Several optimization algorithms have been proposed to solve an unconstrained optimization problem [3]-[18]. However, in the next section, we state a learning algorithm to solve unconstrained optimization problem (26).

4 Learning algorithm of fuzzy system

learning the fuzzy system means to find the correct parameters of fuzzy solutions so that minimize error function $\mathcal{E}(\Upsilon)$ in (26). In unsupervised learning, it is impossible to use the backpropagation algorithm because the error at each output is not available to the learning fuzzy system. So, standard optimization techniques must be used. One of the simplest is the gradient descent method, the weights are initialized randomly and then, the following change rule is applied:

$$\Upsilon_i(j+1) = \Upsilon_i(j) - \kappa \frac{\partial \mathcal{E}(\Upsilon)}{\partial \Upsilon_i}, \tag{27}$$

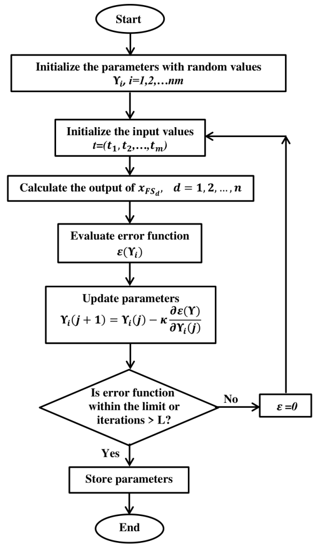


Figure 2: Flowchart of the structure learning algorithm for fuzzy system.

where κ is the learning rate, j is the iteration step, $\Upsilon_i, i = 1, 2, \ldots, nm$ are all adjustable parameters in fuzzy solutions (24) and m is the number of fuzzy rules. For instance, unsupervised back propagation algorithm of the x_{FS_1} is as

$$C_{x_1}(j+1) = C_{x_1}(j) - \kappa\frac{\partial\mathcal{E}(\Upsilon)}{\partial C_{x_1}}, \tag{28}$$

Similarly, other parameters used to develop the x_{FS_d}, $d = 2, 3, \ldots, n$ are optimized using (28). The flowchart of the structure learning algorithm for fuzzy system is shown in Figure 2. In the following steps, the learning algorithm is described.

Step 1: Randomly select the initial values of fuzzy solutions parameters $\Upsilon_i, i = 1, 2, \ldots, nm$ and select an error tolerance parameter $\varepsilon > 0$ and an iteration L.

Step 2: Initialize the input vector $t = (t_1, t_2, \ldots, t_m)$.

Step 3: Compute the output values of x_{FS_d}, $d = 1, 2, \ldots, n$.

Step 4: Calculate the error function $\mathcal{E}(\Upsilon)$ using (26).

Step 5: Update the parameters using unsupervised back propagation algorithm (27).

Step 6: If the error function $\mathcal{E}(\Upsilon) \leq \varepsilon$ or iterations $> L$, then go to step 7 otherwise go to step 2.

Step 7: After completing the learning algorithm, the final parameters are stored and then the converged fuzzy solutions can be used for testing.

Theorem 4.1. *Suppose that* $\gamma^j(\Upsilon_0) = \{\Upsilon(j), j = 1, 2, \ldots\}$ *is a sequence of* (27) *in which the initial point is* $\Upsilon(0)$ *and the level set* $L(\Upsilon_0) = \{\Upsilon(j) \mid \mathcal{E}(\Upsilon(j)) \leq \mathcal{E}(\Upsilon(0))\}$ *is bounded. Then*

(a) $\gamma^j(\Upsilon_0)$ *is bounded.*

(b) *There exists* $\bar{\Upsilon}$ *such that* $\lim_{j\to\infty}\Upsilon(j) = \bar{\Upsilon}$.

Proof. (a) Since in the proposed fuzzy system the adjustable parameters are optimized by the gradient descent algorithm, thus $\mathcal{E}(\Upsilon(j))$ in (26) along $\{\Upsilon(j), j = 1, 2, \ldots\}$ is monotone nonincreasing. Therefore $\gamma^j(\Upsilon_0) \subseteq L(\Upsilon(0))$, that is to say $\gamma^j(\Upsilon_0) = \{\Upsilon(j), j = 1, 2, \ldots\}$ is bounded.

(b) By (a), $\gamma^j(\Upsilon_0) = \{\Upsilon(j), j = 1, 2, \ldots\}$ is a bounded set of points. Thus there exists limiting point $\bar{\Upsilon}$, and there exists a sequence $\{j_k\} \to \infty$ such that $\gamma^{j_k}(\Upsilon_0) = \{\Upsilon(j_k)\} \to \bar{\Upsilon}$, as $k \to \infty$, which indicates that $\bar{\Upsilon}$ is ω-limit point of $\gamma^j(\Upsilon_0)$. Using the LaSalle invariant set theorem for discrete time dynamical systems (see [22]), one has that $\{\Upsilon(j)\} \to \bar{\Upsilon} \in M$ as $j \to \infty$, where M is the largest invariant set in $\mathcal{M} = \{\Upsilon(j) \mid \mathcal{E}(\Upsilon(j+1)) - \mathcal{E}(\Upsilon(j)) = 0\}$. □

5 Formulation of the V-OFDOCPs

V-OFDOCPs are a category of Variable-order fractional optimal control problems (V-OFOCPs) that involve time-delay systems. V-OFDOCPs have been used in modeling many real-life phenomena, hence, these problems have been investigated by researchers. Generally, the numerical methods to solve V-OFDOCPs are divided into indirect and direct methods. The direct methods solve the problems by approximating the state and control functions, whereas indirect methods derive a class of V-OFDDAEs by the Hamilton's principle. In this section, we investigate some of the control problems that are well suitable for the V-OFDDAE system framework. In particular, we show that the V-OFDOCPs can be reformulated as a V-OFDDAE. The derived V-OFDDAE can then be solved by proposed method.

In this paper, we consider the following class of V-OFDOCPs

$$\text{minimize} \quad J(u) = \int_0^T F(t, x(t), u(t))dt, \tag{29}$$

subject to

$$\mathcal{M}\dot{x}(t) + \mathcal{N}{}_0^{ABC}D_t^{\alpha(t)}x(t) = G(t, x(t), u(t), x(t-\tau_x), u(t-\tau_u)), 0 < t \leq T, \tag{30}$$

$$x(t) = \phi(t), \quad -\tau_x \leq t \leq 0, \tag{31}$$

$$u(t) = \varphi(t), \quad -\tau_u \leq t \leq 0, \tag{32}$$

where $(\mathcal{M}, \mathcal{N}) \neq (0, 0), x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control variable, $F \in \mathbb{R}$ has continuous first and second partial derivatives with respect to all its arguments, and $G \in \mathbb{R}^n$ is Lipschitz continuous. Also we assume that T is fixed.

In order to reformulate the problem (29)-(32) as a V-OFDDAE , we obtain the necessary optimality conditions corresponding to this V-OFDOCP. To start, according to the problem (29)-(32), we define the Hamiltonian function H as following:

$$H(t, x(t), u(t), x(t-\tau_x), u(t-\tau_u), \lambda(t)) = F(t, x(t), u(t)) + \lambda^{\mathbb{T}}G(t, x(t), x(t-\tau_x), u(t), u(t-\tau_u)), \tag{33}$$

where $\lambda \in \mathbb{R}^n$ is a vector of Lagrange multipliers. The following theorem expresses the necessary optimality conditions of the problem (29)-(32).

Theorem 5.1. *If $x(t)$, $\lambda(t)$ and $u(t)$ are the optimal values of the state, co-state and control respectively, they must satisfy the following conditions*

$$\begin{cases} \mathcal{M}\dot{x}(t) + \mathcal{N}{}_0^{ABC}D_t^{\alpha(t)}x(t) = \dfrac{\partial H}{\partial \lambda(t)}, \qquad 0 < t \leq T, \\ \mathcal{M}\dot{\lambda}(t) - \mathcal{N}{}_t^{ABC}D_T^{\alpha(t)}\lambda(t) = -\dfrac{\partial H}{\partial x(t)} - \chi_{[0,T-\tau_x]}\mathcal{H}_x(t+\tau_x), \qquad 0 \leq t < T, \\ \dfrac{\partial H}{\partial u(t)} + \chi_{[0,T-\tau_u]}\mathcal{H}_u(t+\tau_u) = 0, \qquad 0 \leq t \leq T, \\ x(t) = \phi(t), \quad -\tau_x \leq t \leq 0, \\ u(t) = \varphi(t), \quad -\tau_u \leq t \leq 0, \\ \lambda(T) = 0, \end{cases} \tag{34}$$

where $\mathcal{H}_x(t) = \dfrac{\partial H}{\partial x(t-\tau_x)}$ and $\mathcal{H}_u(t) = \dfrac{\partial H}{\partial u(t-\tau_u)}$.

Proof. We follow the traditional approach for finding the necessary condition of delay fractional optimal control, as

$$\mathcal{J}(u) = \int_0^T \left(H(t, x(t), u(t), x(t-\tau_x), u(t-\tau_u), \lambda(t)) - \lambda \left(\mathcal{M}\dot{x}(t) + \mathcal{N}{}_0^{ABC}D_t^{\alpha(t)}x(t) \right) \right) dt. \tag{35}$$

We consider variations of the form

$$x(t) + \delta x(t), \quad u(t) + \delta u(t), \quad \lambda(t) + \delta\lambda(t), \quad x(t-\tau_x) + \delta x(t-\tau_x), \quad u(t-\tau_u) + \delta u(t-\tau_u).$$

Minimization of $\mathcal{J}$ and hence minimization of $\mathcal{J}$ requires that the first variation of $\mathcal{J}$ must vanish when evaluated along a minimizer, we get

$$0 = \int_0^T \{ \frac{\partial H}{\partial x(t)}\delta x(t) + \frac{\partial H}{\partial x(t-\tau_x)}\delta x(t-\tau_x) + \frac{\partial H}{\partial u(t)}\delta u(t) + \frac{\partial H}{\partial u(t-\tau_u)}\delta u(t-\tau_u) + \frac{\partial H}{\partial \lambda(t)}\delta\lambda(t)$$
$$- \delta\lambda(t) \left(\mathcal{M}\dot{x}(t) + \mathcal{N}{}_0^{ABC}D_t^{\alpha(t)}x(t) \right) - \lambda(t) \left(\mathcal{M}\dot{\delta x}(t) + \mathcal{N}{}_0^{ABC}D_t^{\alpha(t)}\delta x(t) \right) \}dt.$$

Integration by parts gives the relations

$$\int_0^T \lambda\delta\dot{x}(t)dt = -\int_0^T \delta x(t)\dot{\lambda}(t)dt + \lambda(T)\delta x(T),$$

and

$$\int_0^T \lambda_0^{ABC}D_t^{\alpha(t)}\delta x(t) = \int_0^T \delta x(t)_t^{ABR}D_T^{\alpha(t)}\lambda(t)dt,$$

because $x(0)$ is specified, we have $\delta x(0) = 0$. By using (6)

$$\int_0^T \delta x(t)_t^{ABR}D_T^{\alpha(t)}\lambda(t)dt = \int_0^T \delta x(t)_t^{ABC}D_T^{\alpha(t)}\lambda(t)dt + \lambda(T)\frac{M(\alpha(t))}{1-\alpha(t)}\int_0^T \delta x(t)E_\alpha(t)(\frac{-\alpha(t)}{1-\alpha(t)}(T-t)^{\alpha(t)})dt.$$

Also

$$\begin{aligned}\int_0^T \frac{\partial H}{\partial x(t-\tau_x)}\delta x(t-\tau_x)dt &= \int_0^T \mathcal{H}_x(t)\delta x(t-\tau_x)dt\\ &= \int_{\tau_x}^T \mathcal{H}_x(t)\delta x(t-\tau_x)dt\\ &= \int_0^{T-\tau_x} \mathcal{H}_x(t+\tau_x)\delta x(t)dt\\ &= \int_0^T \mathcal{H}_x(t+\tau_x)\delta x(t)dt,\end{aligned}$$

and

$$\begin{aligned}\int_0^T \frac{\partial H}{\partial u(t-\tau_u)}\delta x(t-\tau_u)dt &= \int_0^T \mathcal{H}_u(t)\delta u(t-\tau_u)dt\\ &= \int_{\tau_u}^T \mathcal{H}_u(t)\delta u(t-\tau_u)dt\\ &= \int_0^{T-\tau_u} \mathcal{H}_u(t+\tau_u)\delta u(t)dt\\ &= \int_0^T \mathcal{H}_u(t+\tau_u)\delta u(t)dt,\end{aligned}$$

since $x(t), -\tau_x \le t \le 0$, and $u(t), -\tau_u \le t \le 0$, are specified and $\mathcal{H}_x = \mathcal{H}_u = 0$ for $t \ge T$. So, we deduce the following formula

$$\begin{aligned}&-\lambda(T)\left(\mathcal{N}\frac{M(\alpha(t))}{1-\alpha(t)}\int_0^T \delta x(t)E_{\alpha(t)}(\frac{-\alpha(t)}{1-\alpha(t)}(T-t)^{\alpha(t)})dt + \mathcal{M}\delta x(T)\right)\\ &+\int_0^T \{\delta x(t)\left(\frac{\partial H}{\partial x} + \mathcal{H}_x + \mathcal{M}\dot{\lambda} - \mathcal{N}_t^{ABC}D_T^{\alpha(t)}\lambda(t)\right)\\ &+\delta u(t)(\frac{\partial H}{\partial u} + \mathcal{H}_u) + \delta\lambda(t)\left(\frac{\partial H}{\partial \lambda} - \mathcal{M}\dot{x}(t) - \mathcal{N}_0^{ABC}D_t^{\alpha(t)}x(t)\right)\}dt = 0.\end{aligned}$$

Assuming $\mathcal{N}\frac{M(\alpha(t))}{1-\alpha(t)}\int_0^T \delta x(t)E_{\alpha(t)}(\frac{-\alpha(t)}{1-\alpha(t)}(T-t)^{\alpha(t)})dt \neq -\mathcal{M}\delta x(T)$ and since the variation functions were chosen arbitrarily, then the proof of theorem is complete. □

The first two equations in (34) are the canonical Hamilton equations. The third equation can be viewed as an algebraic constraint. Therefore we shall call (34) a Hamiltonian V-OFDDAE.

6 Numerical examples

In this section, several examples are provided to illustrate the efficiency and validity of the proposed numerical approach. For all examples, the values of the error function $E(\Upsilon)$ and the elapsed central processing unit (CPU) time (in seconds) for different values of $\alpha(t)$ are listed. By providing an error table for different values of $\alpha(t)$, we confirm that the solutions for $0 < \alpha(t) \leq 1$ are accurate. All the numerical results are carried out on a personal computer with a 2.60 GHz Intel Core i7 processor and 6 GB of RAM running Windows.

Example 6.1. *Consider the following multi-delay V-OFDDAE as*

$$\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} {}_0^{ABC}D_t^{\alpha(t)} x(t) = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} x(t) + \begin{bmatrix}x_1(t-1)\\ x_2(t-0.6)\end{bmatrix} + \begin{bmatrix}0\\ f(t)\end{bmatrix}, \quad t \geq 0, \tag{36}$$

$$x(t) = \begin{bmatrix}\exp(0.5t)\\ \sin(\pi t)\end{bmatrix}, \quad t \leq 0, \tag{37}$$

The function $f(t)$ is chosen such that the exact solution of the problem is $x^*(t) = \begin{bmatrix}\exp(0.5t)\\ \sin(\pi t)\end{bmatrix}$. Considering the conditions $x_1(0) = 1$ and $x_2(0) = 0$, we can choose the fuzzy solutions as

$$\begin{aligned} x_{1_{FS}} &= 1 + tFS_{x_1}, \\ x_{2_{FS}} &= tFS_{x_2}. \end{aligned}$$

Figures 3 and 4 present the exact and approximate values of $x_1(t)$ and $x_2(t)$ with $m = 11$ for some different variable orders $\alpha(t)$. The absolute errors of $x_1(t)$ and $x_2(t)$ with $m = 11$ at various values of $\alpha(t)$ are shown in Figure 5. The values of the error function $E(\Upsilon)$ and CPU time for different values of $\alpha(t)$ are listed in Table 1. From Figures 3 - 5 and Table 1, it is clear that using the proposed method leads to good approximations of the exact solutions ($m = 11$).

Table 1: The values of the error function $E(\Upsilon)$ and CPU time for different values of $\alpha(t)$ in Example 6.1.

$\alpha(t)$	$E(\Upsilon)$	CPU time(s)
$\alpha_1(t) = \frac{1}{1+\exp(-t)}$	4.4750×10^{-20}	0.344
$\alpha_2(t) = 0.7 + 0.05\sin(\frac{t}{10})$	3.0586×10^{-20}	0.437
$\alpha_3(t) = \tanh(t+1)$	1.5952×10^{-16}	0.594
$\alpha_4(t) = 0.6$	1.2874×10^{-19}	0.360

Example 6.2. *Consider the following V-OFDDAE*

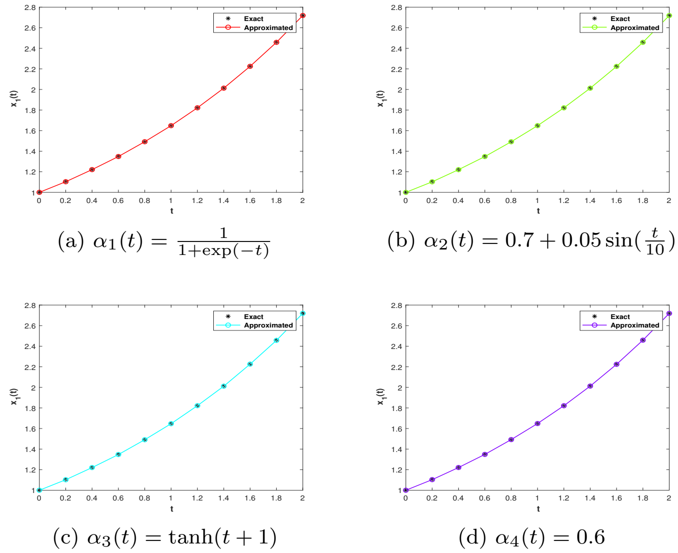
$$\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} {}_0^{ABC}D_t^{\alpha(t)} x(t) = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} x(t) + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} x(t-1) + \begin{bmatrix}1 - \exp(t) - \exp(t-1)\\ 1 - 2t\end{bmatrix}, \quad t \geq 0, \tag{38}$$

$$x(t) = \begin{bmatrix}\exp(t)\\ t\end{bmatrix}, \quad t \leq 0, \tag{39}$$

For $\alpha(t) = 1$ the exact solution of the mentioned problem is $x(t) = \begin{bmatrix}\exp(t)\\ t\end{bmatrix}$. Here, the proposed method is employed to solve problem (38)-(39). Considering the conditions $x_1(0) = 1$ and $x_2(0) = 0$, we can choose the fuzzy solutions with 11 fuzzy rules (m=11) as

$$\begin{aligned} x_{1_{FS}} &= 1 + tFS_{x_1}, \\ x_{2_{FS}} &= tFS_{x_2}. \end{aligned}$$

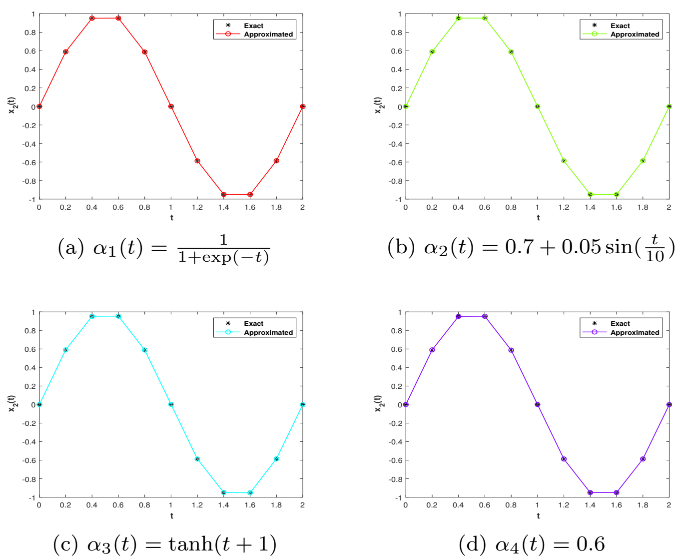
The exact solution at $\alpha(t) = 1$ and the behavior of the numerical solutions $x_1(t)$ and $x_2(t)$ in different values of $\alpha(t)$ is presented in Figure 6. In Table 2, the absolute errors of $x(t)$ for $\alpha(t) = 1$ are shown. Also, Table 3 indicates the values of the error function $E(\Upsilon)$ and the elapsed CPU time for different values of $\alpha(t)$. From Figure 6 and Tables 2 and 3, it is clear that good approximation results are achieved by the present method, with a small number of fuzzy rules.



(a) $\alpha_1(t) = \frac{1}{1+\exp(-t)}$ (b) $\alpha_2(t) = 0.7 + 0.05\sin(\frac{t}{10})$

(c) $\alpha_3(t) = \tanh(t+1)$ (d) $\alpha_4(t) = 0.6$

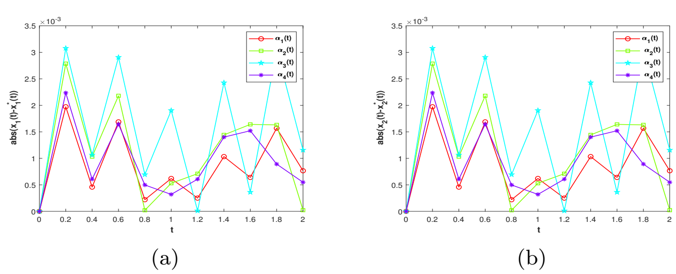
Figure 3: Approximate solutions of $x_1(t)$ at different values of $\alpha(t)$ for Example 6.1.



(a) $\alpha_1(t) = \frac{1}{1+\exp(-t)}$ (b) $\alpha_2(t) = 0.7 + 0.05\sin(\frac{t}{10})$

(c) $\alpha_3(t) = \tanh(t+1)$ (d) $\alpha_4(t) = 0.6$

Figure 4: Approximate solutions of $x_2(t)$ at different values of $\alpha(t)$ for Example 6.1.



(a) (b)

Figure 5: Absolute errors of $x_1(t)$ and $x_2(t)$ at different values of $\alpha(t)$ for Example 6.1.

Table 2: The absolute errors of $x(t)$ at $\alpha(t) = 1$ in Example 6.2.

t	Error of $x_1(t)$	Error of $x_2(t)$
0.2	1.765×10^{-4}	4.706×10^{-11}
0.4	1.280×10^{-4}	1.057×10^{-10}
0.6	1.003×10^{-4}	1.649×10^{-10}
0.8	7.714×10^{-5}	1.285×10^{-10}
1	5.757×10^{-5}	1.866×10^{-11}
1.2	2.184×10^{-4}	4.790×10^{-11}
1.4	1.582×10^{-4}	4.078×10^{-11}
1.6	1.224×10^{-4}	1.010×10^{-10}
1.8	9.553×10^{-5}	1.481×10^{-10}
2	8.394×10^{-5}	3.119×10^{-10}

Table 3: The values of the error function $E(\Upsilon)$ and CPU time for different values of $\alpha(t)$ in Example 6.2.

$\alpha(t)$	$E(\Upsilon)$	CPU time(s)
$\alpha_1(t) = 1$	2.5410×10^{-21}	0.125
$\alpha_2(t) = 0.8 + 0.03\sin(\frac{t}{10})$	4.3343×10^{-22}	0.422
$\alpha_3(t) = 0.6$	8.7169×10^{-22}	0.390
$\alpha_4(t) = 0.1 + 0.4t$	1.0537×10^{-22}	0.485
$\alpha_5(t) = 0.9 - 0.4t$	3.5363×10^{-22}	0.391

Example 6.3. *Consider the following V-OFDDAE as*

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&0\end{bmatrix} {}_0^{ABC}D_t^{\alpha(t)}x(t) = \begin{bmatrix}0&0&1&0\\0&0&0&1\\0&-4&0&0\\1&-1&0&0\end{bmatrix} x(t) + \begin{bmatrix}0&0&0&0\\0&0&0&1\\5&0&0&0\\0&5&0&0\end{bmatrix} x(t-\pi) + \begin{bmatrix}0\\0\\0\\\cos(t)\sin(2t)-\sin(t)\cos(2t)\end{bmatrix}, \quad t \geq 0, \tag{40}$$

$$x(t) = \begin{bmatrix}\sin(t)\cos(2t)\\\cos(t)\sin(2t)\\\cos(t)\cos(2t)-2\sin(t)\sin(2t)\\2\cos(t)\cos(2t)-\sin(t)\sin(2t)\end{bmatrix}, \quad t \leq 0. \tag{41}$$

The exact solution of this V-OFDDAE for $\alpha(t) = 1$ *is* $x(t) = \begin{bmatrix}\sin(t)\cos(2t)\\\cos(t)\sin(2t)\\\cos(t)\cos(2t)-2\sin(t)\sin(2t)\\2\cos(t)\cos(2t)-\sin(t)\sin(2t)\end{bmatrix}$. *The exact solution at* $\alpha(t) = 1$ *and the approximate solutions of* $x_1(t), x_2(t), x_3(t)$ *and* $x_4(t)$ *with* $m = 11$ *for different values of* $\alpha(t)$ *are presented in Figure 7. In Table 4, the absolute errors of* $x(t)$ *for* $\alpha(t) = 1$ *are shown. Moreover, Table 5 indicates the values of the error function* $E(\Upsilon)$ *and CPU time for different values of* $\alpha(t)$.

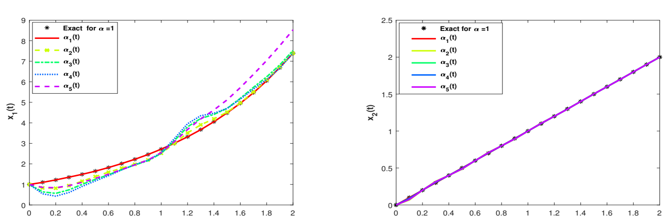


Figure 6: Approximate solutions of $x_1(t)$ and $x_2(t)$ at different values of $\alpha(t)$ for Example 6.2.

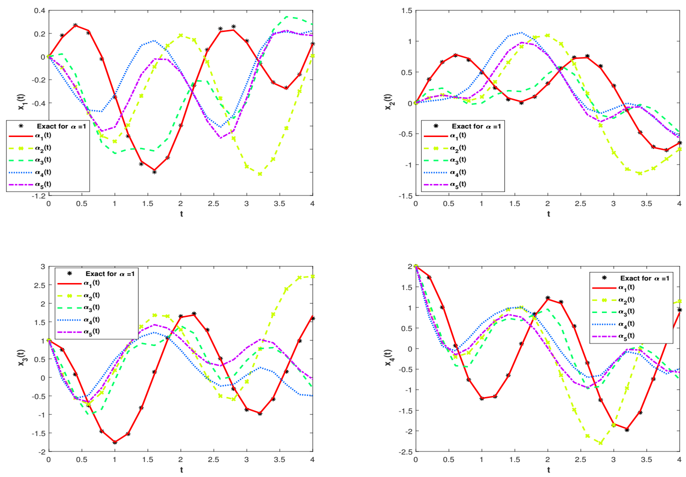


Figure 7: Approximate solutions of $x_1(t), x_2(t), x_3(t)$ and $x_4(t)$ at different values of $\alpha(t)$ for Example 6.3.

Table 4: The absolute errors of $x(t)$ at $\alpha(t) = 1$ in Example 6.3.

t	Error of $x_1(t)$	Error of $x_2(t)$	Error of $x_3(t)$	Error of $x_4(t)$
0.4	3.26×10^{-3}	3.26×10^{-3}	6.60×10^{-2}	6.51×10^{-2}
0.8	2.27×10^{-2}	2.27×10^{-2}	9.91×10^{-3}	1.28×10^{-2}
1.2	1.86×10^{-2}	1.86×10^{-2}	7.63×10^{-3}	3.75×10^{-3}
1.6	1.71×10^{-2}	1.71×10^{-2}	5.45×10^{-2}	4.45×10^{-2}
2	9.73×10^{-3}	9.73×10^{-3}	2.59×10^{-2}	3.30×10^{-2}
2.4	1.23×10^{-2}	1.23×10^{-2}	4.64×10^{-2}	3.71×10^{-2}
2.8	3.07×10^{-2}	3.07×10^{-2}	3.37×10^{-2}	2.27×10^{-2}
3.2	1.36×10^{-3}	1.36×10^{-3}	1.31×10^{-2}	2.86×10^{-2}
3.6	1.01×10^{-2}	1.01×10^{-2}	7.00×10^{-2}	1.37×10^{-2}
4	6.25×10^{-4}	6.25×10^{-4}	7.82×10^{-2}	6.22×10^{-2}

Table 5: The values of the error function $E(\Upsilon)$ and CPU time for different values of $\alpha(t)$ in Example 6.3.

$\alpha(t)$	$E(\Upsilon)$	CPU time(s)
$\alpha_1(t) = 1$	2.4961×10^{-14}	0.187
$\alpha_2(t) = \frac{1}{1+\exp(-t)}$	1.7088×10^{-16}	0.953
$\alpha_3(t) = 1 - \frac{(\cos(t))^2}{3}$	1.1053×10^{-16}	0.969
$\alpha_4(t) = 0.2 + (\frac{t}{5})^2$	1.0023×10^{-16}	0.906
$\alpha_5(t) = 0.6$	2.0192×10^{-17}	0.985

Example 6.4. *Consider the following V-OFDOCP with delay in state as*

$$\text{minimize} \quad J(u) = \frac{1}{2}\int_0^2 (x^2(t) + u^2(t))dt, \tag{42}$$

subject to

$$ {}_0^{ABC}D_t^{\alpha(t)} x(t) = tx(t-1) + u(t), \qquad 0 < t \leq 2, \tag{43}$$

$$x(t) = 1, \qquad -1 \leq t \leq 0. \tag{44}$$

Here, the proposed method is employed to solve problem (42)-(44). *The Hamiltonian function in this problem is*

$$H(x(t), x(t-1), u(t), \lambda(t), t) = \frac{1}{2}(x^2(t) + u^2(t)) + \lambda(t)(tx(t-1) + u(t)).$$

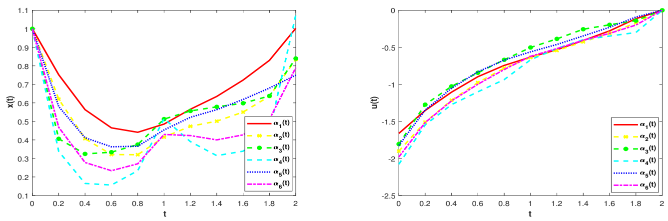


Figure 8: Approximate solutions of $x(t)$ and $u(t)$ at different values of $\alpha(t)$ for Example 6.4.

Therefore the optimality conditions given by equation (34) *becomes*

$$\begin{cases} {}_{t}^{ABC}D_2^{\alpha(t)}\lambda(t) = x(t) + \chi_{[0,1]}(t+1)\lambda(t+1), \qquad 0 \le t < 2, \\ {}_{0}^{ABC}D_t^{\alpha(t)}x(t) = tx(t-1) + u(t), \qquad 0 < t \le 2, \\ u(t) + \lambda(t) = 0, \\ x(t) = 1, \qquad -1 \le t \le 0. \\ \lambda(2) = 0. \end{cases} \tag{45}$$

Considering the conditions $x(0) = 1$ *and* $\lambda(2) = 0$*, we can choose the fuzzy solutions as*

$$x_{FS} = 1 + tFS_x,$$
$$\lambda_{FS} = (2-t)FS_\lambda,$$
$$u_{FS} = FS_u.$$

In Table 6, a comparison is made between the values of performance index obtained by Grunwald-Letnikov approximation [16], together with the presented method for different values of $\alpha(t)$*. The behavior of the numerical solutions of the state variable* $x(t)$ *and the control variable* $u(t)$ *in different values of* $\alpha(t)$ *is presented in Figure 8. Also, Table 7 indicates the values of the error function* $E(\Upsilon)$ *and the elapsed CPU time for different values of* $\alpha(t)$*. Values of the obtained results of the considered problem are provided in Table 8.*

Table 6: Results of J at different values of $\alpha(t)$ for Example 6.4.

$\alpha(t)$	1	0.9	0.8	$1-\frac{\cos(t)^2}{3}$	$1-\frac{0.5}{1+\exp(-t)}$	$\tanh(t+1)$
The proposed method $m = 11$	1.1033	1.0836	1.0211	0.8659	1.0985	0.9356
The method of [16]	1.2018	1.2232	1.2546	–	–	–

Table 7: The values of the error function $E(\Upsilon)$ and CPU time for Example 6.4.

$\alpha(t)$	$E(\Upsilon)$	CPU time(s)
$\alpha_1(t) = 1$	1.53319×10^{-23}	0.256
$\alpha_2(t) = 0.9$	1.71997×10^{-20}	0.468
$\alpha_3(t) = 1 - \frac{\cos(t)^2}{3}$	1.4041×10^{-22}	0.672
$\alpha_4(t) = 1 - \frac{0.5}{1+\exp(-t)}$	4.6986×10^{-23}	0.656
$\alpha_5(t) = \tanh(t+1)$	1.7422×10^{-23}	0.594
$\alpha_6(t) = 0.8$	6.0986×10^{-20}	0.469

Table 8: The approximate solutions at different values of $\alpha(t)$ in Example 6.4.

	$\alpha(t)=1$		$\alpha(t)=\tanh(t+1)$	
t	$x(t)$	$u(t)$	$x(t)$	$u(t)$
0	1	−1.6623	1	−1.8238
0.2	0.7519	−1.3530	0.5801	−1.3472
0.4	0.5634	−1.1055	0.4117	−1.0524
0.6	0.4651	−0.8970	0.3628	−0.8309
0.8	0.4412	−0.7445	0.3659	−0.6725
1	0.4856	−0.6325	0.4540	−0.5613
1.2	0.5657	−0.5325	0.5223	−0.4616
1.4	0.6348	−0.4080	0.5633	−0.3460
1.6	0.7231	−0.2781	0.6184	−0.2300
1.8	0.8297	−0.1133	0.6800	−0.0912
2	1.0026	0	0.7497	0

Example 6.5. *Consider the following V-OFDOCP with different delays in states as*

$$\text{minimize} \quad J(u)=\frac{1}{2}\int_0^1((x_1(t)+x_2(t))^2+u^2(t))dt, \tag{46}$$

subject to

$${}_0^{ABC}D_t^{\alpha(t)}x_1(t)=tx_1(t)+x_2(t-\frac{1}{4}), \qquad 0<t\leq 1, \tag{47}$$

$${}_0^{ABC}D_t^{\alpha(t)}x_2(t)=t^2x_2(t)-5x_1(t-\frac{1}{4})-x_2(t-\frac{1}{4})+u(t), \qquad 0<t\leq 1, \tag{48}$$

$$x_1(t)=x_2(t)=1, \qquad \frac{-1}{4}\leq t\leq 0. \tag{49}$$

The necessary conditions of optimality are described by

$$\begin{cases} {}_t^{ABC}D_1^{\alpha(t)}\lambda_1(t)=x_1(t)+x_2(t)+t\lambda_1(t)-\chi_{[0,\frac{3}{4}]}5\lambda_2(t+\frac{1}{4}), \qquad 0\leq t<1,\\ {}_t^{ABC}D_1^{\alpha(t)}\lambda_2(t)=x_1(t)+x_2(t)+t^2\lambda_2(t)+\chi_{[0,\frac{3}{4}]}\lambda_1(t+\frac{1}{4})-\chi_{[0,\frac{3}{4}]}\lambda_2(t+\frac{1}{4}), \qquad 0\leq t<1,\\ {}_0^{ABC}D_t^{\alpha(t)}x_1(t)=tx_1(t)+x_2(t-\frac{1}{4}), \qquad 0<t\leq 1,\\ {}_0^{ABC}D_t^{\alpha(t)}x_2(t)=t^2x_2(t)-5x_1(t-\frac{1}{4})-x_2(t-\frac{1}{4})+u(t), \qquad 0<t\leq 1,\\ u(t)+\lambda_2(t)=0, \qquad 0\leq t\leq 1,\\ x_1(t)=x_2(t)=1, \qquad \frac{-1}{4}\leq t\leq 0,\\ \lambda_1(1)=\lambda_2(1)=0. \end{cases} \tag{50}$$

The approximate solutions of $x_1(t), x_2(t)$ and $u(t)$ for different values of $\alpha(t)$ are presented in Figure 9. In Table 9, the approximate values of J obtained by Grunwald-Letnikov approximation [16] and the numerical results of the suggested method with different values of $\alpha(t)$ are listed. Moreover, Table 10 indicates the values of the error function $E(\Upsilon)$ and CPU time for different values of $\alpha(t)$. Values of the obtained results of the considered problem are provided in Table 11.

Table 9: Results of J at different values of $\alpha(t)$ for Example 6.5.

$\alpha(t)$	1	0.9	0.8	$0.9+0.01\exp(t)$	$\tanh(\frac{3}{2}(t+1))$	$\cos(\frac{(t+1)\pi}{18})$
The proposed method $m=21$	1.5002	3.2710	2.5612	3.1082	2.2178	2.1222
The method of [16]	1.7548	2.2392	2.7998	-	-	-

Table 10: The values of the error function $E(\Upsilon)$ and CPU time for Example 6.5.

$\alpha(t)$	$E(\Upsilon)$	CPU time(s)
$\alpha_1(t)=1$	1.14086×10^{-23}	0.359
$\alpha_2(t)=0.9+0.01\exp(t)$	1.5120×10^{-23}	1.181
$\alpha_3(t)=\tanh(\frac{3}{2}(t+1))$	1.2678×10^{-18}	1.25
$\alpha_4(t)=0.9$	5.68724×10^{-23}	0.875
$\alpha_5(t)=\cos(\frac{(t+1)\pi}{18})$	4.0750×10^{-22}	1.141
$\alpha_6(t)=0.8$	5.81165×10^{-23}	0.938

To end this section, we present some advantages of the proposed method.

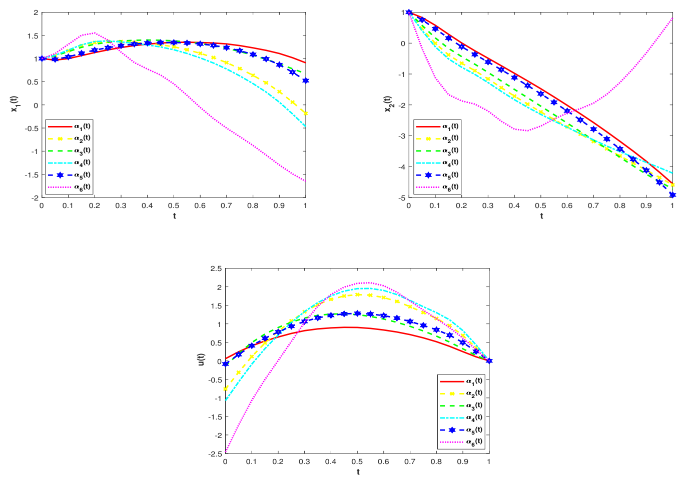


Figure 9: Approximate solutions of $x_1(t), x_2(t)$ and $u(t)$ at different values of $\alpha(t)$ for Example 6.5.

Table 11: The approximate solutions at different values of $\alpha(t)$ in Example 6.5.

	$\alpha(t) = 1$			$\alpha(t) = 0.9 + 0.01\exp(t)$		
t	$x_1(t)$	$x_2(t)$	$u(t)$	$x_1(t)$	$x_2(t)$	$u(t)$
0	1	1	0.0591	1	1	-0.7598
0.1	0.9997	0.5778	0.4070	1.1686	-0.0070	0.1160
0.2	1.1303	0.0061	0.6445	1.3462	-0.6509	0.7970
0.3	1.2391	-0.4899	0.8156	1.3665	-1.1579	1.3255
0.4	1.3119	-0.9774	0.8900	1.3298	-1.7166	1.6609
0.5	1.3504	-1.4784	0.9017	1.2536	-2.2291	1.7885
0.6	1.3469	-2.0141	0.8335	1.1139	-2.7102	1.7102
0.7	1.3167	-2.5744	0.7090	0.9127	-3.1870	1.4623
0.8	1.2316	-3.1817	0.5169	0.6385	-3.6552	1.1439
0.9	1.1097	-3.8292	0.2485	0.2850	-4.1174	0.6856
1	0.9086	-4.5577	0	-0.1839	-4.5936	0

- The main advantage of the fuzzy system is its representative power; i.e. it is capable to describe a highly nonlinear system by using a small number of rules.
- We can use more number of rules or more collocation points over the interval $[0, T]$ to obtain more accurate approximations.
- The fuzzy solutions of variables are differentiable functions of time t, thus we can calculate the solution at each arbitrary point over the interval $[0, T]$.
- The employment of the fuzzy system provides a solution of V-OFDDAEs with superior interpolation properties.
- The computational burden can be greatly reduced using the proposed approach compared with existing methods.
- In all examples, by providing an error table for different values of $\alpha(t)$, we confirm that the solutions for $0 < \alpha(t) \leq 1$ are accurate.
- The convergence of the proposed scheme is also provided.

7 Conclusion

In this paper, an artificial intelligence method based on fuzzy systems has been introduced for the numerical solution of V-OFDDAEs. For this aim, by relying on the ability of fuzzy systems in function approximation, the fuzzy solutions of variables are substituted in V-OFDDAEs. Then, the parameters of fuzzy solutions are adjusted via a learning algorithm. It is shown that the V-OFDOCPs can be reformulated as a V-OFDDAE and are then solved by the proposed method. The simulation results confirm the effectiveness and capability of the suggested technique to solve V-OFDDAEs. As a future work, fuzzy systems can be used for solving fractional differential algebraic equations with time-varying delays.

Acknowledgement

The authors wish to express their appreciation for several excellent suggestions for improvements in this paper made by the referees.

References

[1] H. F. Ahmed, *Numerical study on factional differential-algebraic systems by means of Chebyshev pseudo spectral method*, Journal of Taibah University for Science, **14**(1) (2020), 1023-1032. https://doi.org/10.1080/16583655.2020.1798071

[2] G. M. Bahaa, *Optimal control problem for variable-order fractional differential systems with time delay involving*, Atangana Baleanu Derivatives, Chaos, Solitons and Fractals, **122** (2019), 129-142. https://doi.org/10.1080/16583655.2020.1798071

[3] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear programming-theory and algorithms*, Third Edition, John Wiley and Sons, Hoboken, New Jersey, 2006. https://doi.org/10.1002/0471787779

[4] C. Bendtsen, P. G. Thomsen, *Numerical solution of differential algebraic equations, Technical report*, Department of Mathematical Modelling, Denmark, 1999. https://doi.org/10.1016/S0377-0427(97)00178-7

[5] I. Boulkaibet, K. Belarbi, S. Bououden, T. Marwala, M. Chadli, *A new T-S fuzzy model predictive control for nonlinear processes*, Expert Systems With Applications, **88** (2017), 132-151. https://doi.org/10.1016/j.eswa.2017.06.039

[6] J. Cao, Y. Qiu, G. Song, *A compact finite difference scheme for variable order subdiffusion equation*, Communications in Nonlinear Science and Numerical Simulation, **48** (2017), 140-149. https://doi.org/10.1016/j.cnsns.2016.12.022

[7] Y. Chen, Y. Wei, D. Liu, H. Yu, *Numerical solution for a class of nonlinear variable order fractional differential equations with Legendre wavelets*, Applied Mathematics Letters, **46** (2015), 83-88. https://doi.org/10.1016/j.aml.2015.02.010

[8] V. Daftardar-Gejji, *Fractional calculus and fractional differential equations*, Singapore, 2019. https://doi.org/10.1007/978-981-13-9227-6

[9] M. S. Dahaghin, H Hassani, *An optimization method based on the generalized polynomials for nonlinear variable-order time fractional diffusion-wave equation*, Nonlinear Dynamics, **88** (2017), 1587-1598. https://doi.org/10.1007/s11071-017-3330-7

[10] J. Dong, Y. Fu, *A design method for T-S fuzzy systems with partly immeasurable premise variables subject to actuator saturation*, Neurocomputing, **225** (2017), 164-173. https://doi.org/10.1016/j.neucom.2016.11.025

[11] A. El-Sayed, P. Agarwal, *Numerical solution of multiterm variable-order fractional differential equations via shifted Legendre polynomials*, Mathematical Methods in the Applied Sciences, **42**(11) (2019), 3978-3991. https://doi.org/10.1002/mma.5627

[12] Z. h. Fu, W. Chen, L. Ling, *Method of approximate particular solutions for constant - and variable-order fractional diffusion models*, Engineering Analysis with Boundary Elements, **57** (2015), 37-46. https://doi.org/10.1016/j.enganabound.2014.09.003

[13] F. Ghanbari, K. Ghanbari, P. Mokhtary, *Generalized Jacobi-Galerkin method for nonlinear fractional differential algebraic equations*, Computational and Applied Mathematics, **37**(4) (2018), 5456-5475. https://doi.org/10.1007/s40314-018-0645-z

[14] F. Ghomanjani, *A new approach for solving fractional differential-algebraicequations*, Journal of Taibah University for Science, **11** (2017), 1158-1164. https://doi.org/10.1016/j.jtusci.2017.03.006

[15] F. Ghomanjani, S. Noeiaghdam, *Application of said ball curve for solving fractional differential-algebraic equations*, Mathematics, **9** (2021), 1926. https://doi.org/10.3390/math9161926

[16] A. Jajarmi, D. Baleanu, *Suboptimal control of fractional-order dynamic systems with delay argument*, Journal of Vibration and Control, **24** (2018), 2430-2446. https://doi.org/10.1177/1077546316687936

[17] Y. Jia, M. Xu, Y. Lin, *A numerical solution for variable order fractional functional differential equation*, Applied Mathematics Letters, **64** (2017), 125-130. https://doi.org/10.1016/j.aml.2016.08.018

[18] K. Y. Lee, M. A. El-Sharkawi, *Modern heuristic optimization techniques: Theory and applications to power systems*, IEEE Press Series on Power Engineering, New Jersey, 2008. https://doi.org/10.1002/9780470225868

[19] X. Li, H. Li, B. Wu, *A new numerical method for variable order fractional functional differential equations*, Applied Mathematics Letters, **68** (2017), 80-86. https://doi.org/10.1016/j.aml.2017.01.001

[20] X. Li, B. Wu, *A new reproducing kernel method for variable order fractional boundary value problems for functional differential equations*, Journal of Computational and Applied Mathematics, **311** (2017), 387-393. https://doi.org/10.1016/j.cam.2016.08.010

[21] H. Liu, Y. Fu, B. Li, *Discrete waveform relaxation method for linear fractional delay differential-algebraic equations*, Discrete Dynamics in Nature and Society, **2017** (2017), Article ID 6306570, 9 pages. https://doi.org/10.1155/2017/6306570

[22] W. Mei, F. Bullo, *LaSalle invariance principle for discrete-time dynamical systems: A concise and self-contained tutorial*, (2017). https://doi.org/10.48550/arXiv.1710.03710

[23] S. Mirzajani, M. Pourmahmood Aghababa, A. Heydari, *Adaptive T-S fuzzy control design for fractional-order systems with parametric uncertainty and input constraint*, Fuzzy Sets and Systems, **365**(15) (2019), 22-39. https://doi.org/10.1016/j.fss.2018.03.018

[24] M. Mortezaee, M. Ghovatmand, A. Nazemi, *Solving variable-order fractional differential algebraic equations via generalized fuzzy hyperbolic model with application in electric circuit modeling*, Soft Computing, **24** (2020), 16745-16758. https://doi.org/10.1007/s00500-020-04969-7

[25] M. Mortezaee, M. Ghovatmand, A. Nazemi, *An application of generalized fuzzy hyperbolic model for solving fractional optimal control problems with Caputo-Fabrizio derivative*, Neural Processing Letters, **52**(3) (2020), 1997-2020. https://doi.org/10.1007/s11063-020-10334-4

[26] M. Mortezaee, M. Ghovatmand, A. Nazemi, *An application of a fuzzy system for solving time delay fractional optimal control problems with Atangana-Baleanu derivative*, Optimal Control Applications and Methods, **43**(6) (2022), 1753-1777. https://doi.org/10.1002/oca.2924

[27] S. Natarajan, A. K. Loganathan, *Fuzzy logic inherited machine learning based maximum power point tracker for cost-optimized grid connected hybrid renewable systems*, Iranian Journal of Fuzzy Systems, **21**(1) (2024), 103-128. https://doi.org/10.22111/IJFS.2023.44709.7874

[28] J. Nocedal, S. Wright, *Numerical optimization*, Second Edition, Berlin, 2006. https://doi.org/10.1007/978-0-387-40065-5

[29] M. Shariatzadeh, M. J. Rostami, M. Eftekhari, S. Saryazdi, *An adaptive image encryption scheme guided by fuzzy models*, Iranian Journal of Fuzzy Systems, **21**(1) (2024), 1-18. https://doi.org/10.22111/IJFS.2023.44875.7915

[30] S. Shen, F. Liu, V. Anh, I. Turner, J. Chen, *A characteristic difference method for the variable-order fractional advection-diffusion equation*, Journal of Applied Mathematics and Computing, **42** (2013), 371-386. https://doi.org/10.1007/s12190-012-0642-0

[31] B. Shiri, D. Baleanu, *System of fractional differential algebraic equations with applications*, Chaos, Solitons and Fractals, **120** (2019), 203-212. https://doi.org/10.1016/j.chaos.2019.01.028

[32] J. E. Solís-Pérez, J. F. Gómez-Aguilar, A. Atangana, *Novel numerical method for solving variable-order fractional differential equations with power, exponential and Mittag-Leffler laws*, Chaos, Solitons and Fractals, **114** (2018), 175-185. https://doi.org/10.1016/j.chaos.2018.06.032

[33] H. Sun, W. Chen, C. H. Li, Y. Chen, *Finite difference schemes for variable-order time fractional diffusion equation*, International Journal of Bifurcation and Chaos, **22** (2012), 1250085. https://doi.org/10.1142/S021812741250085X

[34] H. Sun, Y. Zhang, D. Baleanu, W. Chen , Y. Chen, *A new collection of real world applications of fractional calculus in science and engineering*, Communications in Nonlinear Science and Numerical Simulation, **64** (2018), 213-231. https://doi.org/10.1016/j.cnsns.2018.04.019

[35] Y. Tai, N. Chen, L. Wang, Z. Feng, J. Xu, *A numerical method for a system of fractional differential-algebraic equations based on sliding mode control*, Mathematics, **8**(7) (2020), 1134. https://doi.org/10.3390/math8071134

[36] V. E. Tarasov, *Handbook of fractional calculus with applications*, Berlin, 2019. https://doi.org/10.1515/9783110571622

[37] S. H. Wang, W. L. Jhu, C. F. Yung, P. F. Wang, *Numerical solutions of differential algebraic equations and its applications in solving TPPC problems*, The Journal of Marine Science and Technology, **19** (2011), 76-88. https://doi.org/10.51400/2709-6998.2139

[38] L. X. Wang, J. M. Mendel, *Fuzzy basis functions, universal approximation, and orthogonal least-squares learning*, IEEE Transactions on Neural Networks, **3**(5) (1992), 807-814. https://doi.org/10.1109/72.159070

[39] Z. G. Wu, S. H. Dong, P. Shi, H. Su, T. Huang, R. Lu, *Fuzzy-model-based nonfragile guaranteed cost control of nonlinear markov jump systems*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, **47**(8) (2017), 1-10. https://doi.org/10.1109/TSMC.2017.2675943

[40] T. Xu, S. H. Lu, W. Chen, H. Chen, *Finite difference scheme for multi-term variable-order fractional diffusion equation*, Advances in Difference Equations, **103** (2018), 1-13. https://doi.org/10.1186/s13662-018-1544-8

[41] M. Xue, H. Yan, H. Zhang, J. Sun, H. K. Lam, *Hidden-Markov-model-based asynchronous H_∞ tracking control of fuzzy Markov jump systems*, IEEE Transactions on Fuzzy Systems, **29**(5) (2020), 1081-1092. https://doi.org/10.1109/TFUZZ.2020.2968878

[42] S. H. Yaghoobi, B. Parsa Moghaddam, K. Ivaz, *An efficient cubic spline approximation for variable-order fractional differential equations with time delay*, Nonlinear Dynamics, **87** (2017), 815-826. https://doi.org/10.1007/s11071-016-3079-4

[43] Z. You, H. Yan, H. Zhang, S. Chen, M. Wang, *Fuzzy-dependent-switching control of nonlinear systems with aperiodic sampling*, IEEE Transactions on Fuzzy Systems, **29**(11) (2020), 3349-3359. https://doi.org/10.1109/TFUZZ.2020.3018552

[44] M. Zurigat, S. H. Momani, A. Alawneh, *Analytical approximate solutions of systems of fractional algebraic differential equations by homotopy analysis method*, Computers and Mathematics with Applications, **59** (2010), 1227-1235. https://doi.org/10.1016/j.camwa.2009.07.002