Article

A Numerical Method for a System of Fractional Differential-Algebraic Equations Based on Sliding Mode Control

Yongpeng Tai ¹ , Ning Chen ^{2,*}, Lijin Wang ², Zaiyong Feng ³ and Jun Xu ²

¹ College of Automobile and Traffic Engineering, Nanjing Forestry University, Nanjing 210037, China; tai@njfu.edu.cn

² College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China; wlj200869@163.com (L.W.); xj_nfu@163.com (J.X.)

³ Department of Mathematics Teaching, Nanjing Institute of Railway Technology, Nanjing 210031, China; zyfeng@njrts.edu.cn

* Correspondence: chenning@njfu.com.cn

Received: 29 June 2020; Accepted: 9 July 2020; Published: 11 July 2020

Abstract: Fractional calculus is widely used in engineering fields. In complex mechanical systems, multi-body dynamics can be modelled by fractional differential-algebraic equations when considering the fractional constitutive relations of some materials. In recent years, there have been a few works about the numerical method of the fractional differential-algebraic equations. However, most of the methods cannot be directly applied in the equations of dynamic systems. This paper presents a numerical algorithm of fractional differential-algebraic equations based on the theory of sliding mode control and the fractional calculus definition of Grünwald–Letnikov. The algebraic equation is considered as the sliding mode surface. The validity of the present method is verified by comparing with an example with exact solutions. The accuracy and efficiency of the present method are studied. It is found that the present method has very high accuracy and low time consumption. The effect of violation corrections on the accuracy is investigated for different time steps.

Keywords: numerical method; fractional order; differential-algebraic equations; sliding mode control

1. Introduction

Fractional calculus (FC) means non-integer order derivative or integral of the variable function, which as an extension of the classical calculus theory, has become an important branch of differential equations [1]. It has been confirmed that many physical phenomena can be described more accurately by using fractional calculus than classical integer-order derivative [2]. Recently, the research and application of fractional calculus theory have become more and more popular in several fields of mathematics, physics, mechanics, control, electrochemistry and engineering, etc. For example, fractional calculus has been successfully applied in viscoelastic material [3–5], heat conduction [6], electrical spectroscopy impedance [7] and control strategies [8–11].

Fractional differential-algebraic equations (FDAEs) are used to describe many physical and engineering problems, which consist of fractional differential equations and algebraic equations. In complex mechanical systems, multi-body dynamics are usually described by differential-algebraic equations (DAEs) [12–14]. However, if some physical characteristics of viscoelastic damping materials are modeled by the fractional constitutive relationship, a group of FDAEs are set up for the system. In control theory, if a fractional chaotic system is controlled by the sliding mode method, the control equations of motion are FDAEs and the sliding surface corresponds to the algebraic equations. FDAEs are also applied in other fields such as electrochemistry, fractional controller, biochemistry and so

on [15]. Among the FDAEs in various fields, the form of FDAEs in mechanical systems is different from others, which contains not only fractional order differential equations, but also integral order differential equations. This class of FDAEs is of significance in an engineering background and will be investigated in this paper.

Due to fractional calculus, the exact solutions of FDAEs are usually too difficult to find. Therefore, the numerical methods for solving FDAEs become important and worth studying. In the past, the numerical methods for fractional differential equations are studied extensively. Theoretically, these methods can be used for FDAEs by equivalent conversation, but at the same time, it will bring some shortcomings, such as unclear physical meaning, not satisfying the constraints, long calculation time and so on. In recent years, some numerical methods have been proposed for solving FDAEs. Researchers investigated the numerical solutions of FDAEs by using various methods [15–22]. However, most of the methods cannot be used directly for solving the FDAEs modelled for mechanical systems or established with an engineering background.

Differential-algebraic systems mainly come from engineering applications. As mentioned before, DAEs is usually used to describe dynamic problems. In recent decades, with the application of fractional order theory in engineering, it is more and more common to use FDAEs to describe dynamic problems in mechanical systems or the fields with engineering background. For example, viscoelastic materials are widely used in complex mechanical systems due to their special damping characteristics, and have achieved an excellent vibration isolation effect. Although the constitutive relation of viscoelastic materials can be described by higher-order integer equation, the fractional-order equation can describe its characteristics more accurately. Therefore, when the fractional-order constitutive relation of viscoelastic damping material is considered, the multi-body dynamic equation becomes an FDAE. This kind of FDAE contains both fractional derivative terms caused by viscoelastic materials and integer derivative terms caused by conventional materials. At present, there are few studies on numerical algorithms for this kind of FDAE. With the combination of fractional calculus and engineering technology, the numerical algorithm of FDAEs will play a crucial role as that of DAEs in aerospace, robot, structural mechanics and other engineering fields. This defines the motivation of the study in this paper.

Nowadays, control theory has mature applications in nonlinear dynamic systems of engineering. Some nonlinear control technologies also have been successfully used for solving FDAEs. Wang et al. [20] proposed a numerical method of FDAEs based on the sliding mode control theory, and they derived the numerical algorithm by the predictor-corrector method [23]. Based on Wang's sliding mode method, this paper proposes a new numerical algorithm using the Grünwald–Letnikov definition, which has higher precision and more efficient calculation.

This paper is organized as follows. Section 2 introduces the form of FDAEs and the Grünwald–Letnikov definition of fractional calculus. Section 3 gives the equivalent fractional differential equation converted from the FDAEs by using sliding mode control theory. Section 4 presents the framework of the numerical algorithm proposed in this paper. Section 5 derives the violation corrections for the present method. Section 6 verifies the validity of the present method, compares the present method with other methods, and discusses the effect of violation corrections. The final section contains the concluding remarks.

2. Fractional Differential-Algebraic Equations

Fractional calculus can more accurately describe the physical properties of some viscoelastic components in complex mechanical systems than classical integer-order theory. The application of fractional calculus in these systems leads to the generation of a class of FDAEs, which has an important

engineering background and is formally an integer-order differential-algebraic equation with fractional derivatives added. Consider an affine nonlinear fractional differential-algebraic equation, given by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1\mathbf{x}^{(\alpha)}(t) + \mathbf{f}(\mathbf{x}, t) + \mathbf{B}_2\mathbf{w}(t) \\ \mathbf{s}(\mathbf{x}) = 0 \end{cases} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is the state vector, $\mathbf{x}^{(\alpha)}(t)$ is the fractional derivatives of order α , $\mathbf{f}(\mathbf{x}, t)$ is the nonlinear part, $\mathbf{w}(t)$ is the external input, $\mathbf{s}(\mathbf{x})$ is the algebraic equation (constraint equation), and $\mathbf{A}$, $\mathbf{B}_1$ and $\mathbf{B}_2$ are the coefficient matrices. Equation (1) is the state equation, which is widely representative in dynamic systems. Because many dynamic systems are second-order differential equations with multiple degrees of freedom, they need to be converted into the form of state equations for analysis and numerical calculation, while the independence of state variables can still be ensured when they are converted into integer type state equations. Here, we set the range of α from 0 to 1. In fact, in the fractional-order dynamic equation or control equation, the range of fractional order is usually related to the physical characteristics of the research object. For example, the order of viscoelastic material is between 0 and 1; when the order tends towards 0, it is linear stiffness; when the order tends towards 1, it is linear damping; and the order of the control parameter may be greater than 1. However, when α is greater than 1, it is easy to reduce the order of the fractional-order equation by using the state equation, and limit α to 0 to 1.

Among various definitions of fractional derivatives, the Riemann–Liouville definition is more reasonable in the dynamic description of natural science problems. However, there is a certain correlation between the discrete Grünwald–Letnikov definition and the continuous Riemann–Liouville definition. The Grünwald–Letnikov definition will degenerate to the Riemann–Liouville definition when time step tends towards 0. In this paper, we choose the Grünwald–Letnikov definition to solve the above equation, which is the most straightforward definition and convenient for numerical calculation [24]. According to the Grünwald–Letnikov definition [24], $\mathbf{x}^{(\alpha)}(t)$ can be expressed as

$$\begin{aligned} \mathbf{x}^{(\alpha)}(t) &= \frac{1}{h^\alpha} \sum_{j=0}^N (-1)^j \frac{\Gamma(\alpha+1)}{j! \Gamma(\alpha-j+1)} \mathbf{x}(t-jh) \\ &= \frac{1}{h^\alpha} \mathbf{x}(t) + \mathbf{C}(\mathbf{x}, t) \end{aligned} \quad (2)$$

where h is the time step, $\Gamma()$ is the Gamma function and $\mathbf{C}(\mathbf{x}, t)$ is given by

$$\mathbf{C}(\mathbf{x}, t) = \frac{1}{h^\alpha} \sum_{j=1}^N (-1)^j \frac{\Gamma(\alpha+1)}{j! \Gamma(\alpha-j+1)} \mathbf{x}(t-jh) \quad (3)$$

3. Transformation Based on Sliding Mode Control Theory

Sliding mode control is a variable structure control method, which can switch the structure of the nonlinear system from one structure to another along the sliding mode surface and consequently make the system asymptotically stable. To solve Equation (1), an equivalent conversion is performed for FDAEs according to the sliding mode control theory. Consider the algebraic equation $\mathbf{s}(\mathbf{x}) = 0$ as the sliding mode surface, and then convert Equation (1) into fractional differential equation, given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1\mathbf{x}^{(\alpha)}(t) + \mathbf{B}_2\mathbf{w}(t) + \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (4)$$

where $\mathbf{u}$ is the control vector added to Equation (1) and $\mathbf{g}(\mathbf{x})$ is the control matrix. $\mathbf{g}(\mathbf{x})\mathbf{u}$ is the control term, which ensures that the state is on or close to the sliding mode surface.

Typically, Equation (4) can be integrated directly by numerical methods. Meanwhile, the control matrix $\mathbf{g}(\mathbf{x})$ must satisfy the following two conditions.

1. Based on nonlinear control theory, $\mathbf{g}(\mathbf{x})$ should satisfy the necessary and sufficient condition of the controllability for the nonlinear system Equation (4), namely

$$\text{rank} \begin{bmatrix} \mathbf{g} & \text{ad}_f \mathbf{g} & \text{ad}_f^2 \mathbf{g} & \cdots & \text{ad}_f^{n-2} \mathbf{g} & \text{ad}_f^{n-1} \mathbf{g} \end{bmatrix} = n \quad (5)$$

where, $\text{ad}_f \mathbf{g}$ is Lie bracket operation of vectors $f = \dot{\mathbf{x}}$ and $\mathbf{g}$.

2. In order to obtain control vector $\mathbf{u}$, $\mathbf{J}_s \mathbf{g}(\mathbf{x})$ should be invertible, where $\mathbf{J}_s$ is the Jacobi matrix of $\mathbf{s}(\mathbf{x})$.

According to the theory of sliding mode control, we need to design the reaching law and the control law. The reaching law approach is adopted to make the state move towards and reach the surface $\mathbf{s}(\mathbf{x}) = 0$, which is given by [25]

$$\dot{\mathbf{s}} = -\varepsilon \mathbf{s}^2 \text{sgn}(\mathbf{s}) - \lambda \mathbf{s} \quad (6)$$

where ε and λ are diagonal matrices with positive elements and sgn is the sign function. The control law can be directly solved from the reaching law. Using Equation (6), the derivative of $\mathbf{s}$ with respect to time can be expressed as

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \dot{\mathbf{x}}(t) = -\varepsilon \mathbf{s}^2 \text{sgn}(\mathbf{s}) - \lambda \mathbf{s} \quad (7)$$

Substituting Equation (4) into Equation (7), the control vector $\mathbf{u}$ can be obtained as

$$\mathbf{u} = [\mathbf{J}_s \mathbf{g}(\mathbf{x})]^{-1} \left[-\varepsilon \mathbf{s}^2 \text{sgn}(\mathbf{s}) - \lambda \mathbf{s} - \mathbf{J}_s (\mathbf{A}\mathbf{x}(t) + \mathbf{B}_1 \mathbf{x}^{(\alpha)}(t) + \mathbf{B}_2 \mathbf{w}(t) + \mathbf{f}(\mathbf{x}, t)) \right] \quad (8)$$

Note that only if $\mathbf{J}_s \mathbf{g}(\mathbf{x})$ is invertible can $\mathbf{u}$ be obtained.

4. The Numerical Algorithm

In this section, a numerical algorithm is proposed to solve fractional differential equation Equation (4). Using discrete time domain and the Grünwald–Letnikov definition Equation (2), an approximation differential equation is obtained as a series, given by

$$\dot{\mathbf{x}}(t) = \bar{\mathbf{A}}\mathbf{x}(t) + \mathbf{B}_1 \mathbf{C}(\mathbf{x}, t) + \mathbf{B}_2 \mathbf{w}(t) + \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad t \in [t_k, t_{k+1}] \quad (9)$$

where $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{B}_1/h^\alpha$ and $t_k = kh$.

By solving the above equation, the approximation solution of Equation (9) is obtained,

$$\mathbf{x}(t_{k+1}) = \mathbf{e}^{\bar{\mathbf{A}}h} \mathbf{x}(t_k) + \int_0^h \mathbf{e}^{\bar{\mathbf{A}}\tau} [\mathbf{B}_1 \mathbf{C}(\mathbf{x}, t) + \mathbf{B}_2 \mathbf{w}(t) + \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})\mathbf{u}] d\tau \quad (10)$$

In order to directly integrate the second term on the right side of the above equation, $\mathbf{C}$, $\mathbf{w}$ and $\mathbf{f}$ need to be transformed appropriately. For $t \in [t_k, t_{k+1}]$, $\mathbf{C}$ and $\mathbf{w}$ can be expressed as first-order Newton interpolation polynomials, given by

$$\mathbf{C}(\mathbf{x}, t) = \mathbf{C}(\mathbf{x}, t_k) + \frac{\mathbf{C}(\mathbf{x}, t_{k+1}) - \mathbf{C}(\mathbf{x}, t_k)}{h} t \quad (11)$$

$$\mathbf{w}(\mathbf{x}, t) = \mathbf{w}(\mathbf{x}, t_k) + \frac{\mathbf{w}(\mathbf{x}, t_{k+1}) - \mathbf{w}(\mathbf{x}, t_k)}{h} t \quad (12)$$

The nonlinear part $\mathbf{f}(\mathbf{x}, t)$ is expanded to Taylor series at $t = kh$ and $\mathbf{x} = \mathbf{x}(t_k)$, given by

$$\mathbf{f}(\mathbf{x}, t) = \sum_{j=0}^{\infty} \frac{(t-t_k)^j}{j!} \mathbf{f}^{(j)}(\mathbf{x}, t_k), \quad t \in [t_k, t_{k+1}] \text{ and } j = 0, 1, 2, \dots \quad (13)$$

where

$$\mathbf{f}(\mathbf{x}, t_k) = \mathbf{f}(\mathbf{x}(t_k), t_k) \quad (14)$$

$$\mathbf{f}^{(j)}(\mathbf{x}, t_k) = \left(\frac{\partial \mathbf{f}^{(j-1)}}{\partial t} \right)_{t=t_k} + \sum_{i=1}^n \left(\frac{\partial \mathbf{f}^{(j-1)}}{\partial x_i} \dot{x}_i \right)_{t=t_k} \quad \text{when } j > 0 \quad (15)$$

Finally, using Equations (11)–(13), Equation (10) can be expressed as

$$\begin{aligned} \mathbf{x}(t_{k+1}) &= \mathbf{e}^{\bar{\mathbf{A}}h} \mathbf{x}(t_k) + \left(\mathbf{H}_0 - \frac{1}{h} \mathbf{H}_1 \right) [\mathbf{B}_1 \mathbf{C}(\mathbf{x}, t_k) + \mathbf{B}_2 \mathbf{w}(t_k)] \\ &\quad + \frac{1}{h} \mathbf{H}_1 \mathbf{B}_1 \mathbf{C}(\mathbf{x}, t_{k+1}) + \frac{1}{h} \mathbf{H}_1 \mathbf{B}_2 \mathbf{w}(t_{k+1}) + \sum_{j=0}^{\infty} \frac{1}{j!} \mathbf{T}_j \mathbf{f}^{(j)}(t_k) + \mathbf{H}_0 \mathbf{g}(\mathbf{x}) \mathbf{u} \end{aligned} \quad (16)$$

where the parameters $\mathbf{H}_0$, $\mathbf{H}_1$ and $\mathbf{T}_j$ are

$$\begin{aligned} \mathbf{H}_0 &= \bar{\mathbf{A}}^{-1} (\mathbf{e}^{\bar{\mathbf{A}}h} - \mathbf{I}) \\ \mathbf{H}_1 &= -\bar{\mathbf{A}}^{-1} (\mathbf{H}_0 - h \mathbf{e}^{\bar{\mathbf{A}}h}) \\ \mathbf{T}_0 &= \mathbf{H}_0 \\ \mathbf{T}_j &= \bar{\mathbf{A}}^{-1} (j \mathbf{T}_{j-1} - h^j \mathbf{I}) \quad \text{when } j > 0 \end{aligned} \quad (17)$$

5. Violation Corrections

The existence of numerical integration error results in violation, and hence, the algebraic equation will do not hold. In order to ensure the accuracy of calculation, it is necessary to modify the results of numerical calculation.

Assuming that the value of integration in step k is $\mathbf{x}_k$, the constraint violation will occur due to the integration error, that is $\mathbf{s}(\mathbf{x}_k) \neq 0$. In order to improve the calculation accuracy, the value of $\mathbf{x}_k$ needs to be corrected. Let the corrected term be $\Delta \mathbf{x}_k$, and then the corrected value and the algebraic equation are

$$\hat{\mathbf{x}}_k = \mathbf{x}_k + \Delta \mathbf{x}_k \quad (18)$$

$$\mathbf{s}(\hat{\mathbf{x}}_k) = 0 \quad (19)$$

Expanding the above equation in Taylor series at $\mathbf{x}_k$ up to first order, yields

$$\mathbf{s}(\hat{\mathbf{x}}_k) = \mathbf{s}(\mathbf{x}_k) + \mathbf{J}_s(\mathbf{x}_k) \Delta \mathbf{x}_k = 0 \quad (20)$$

Because the Jacobi matrix $\mathbf{J}_s$ is usually not a square matrix, the solution of the above equation is not unique. Since the constraint equations are independent of each other, the generalized inverse of matrix $\mathbf{J}_s$ is

$$\mathbf{J}_s^+ = \mathbf{J}_s^T (\mathbf{J}_s \mathbf{J}_s^T)^{-1} \quad (21)$$

The minimum norm solution is

$$\Delta \mathbf{x}_k = -\mathbf{J}_s^+(\mathbf{x}_k) \mathbf{s}(\mathbf{x}_k) \quad (22)$$

Therefore, the corrected value can be written as

$$\hat{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{J}_s^+(\mathbf{x}_k) \mathbf{s}(\mathbf{x}_k) \quad (23)$$

6. Results

In this section, we first use the present method to calculate an example with an exact solution to verify its validity, and then examine the accuracy and efficiency of the present method by comparing it with other methods. The effect of violation corrections is discussed at last.

6.1. An Example with Exact Solutions

Consider a nonlinear fractional order differential-algebraic equation

$$\begin{cases} \dot{x}_1 = -x_1 + \frac{3}{2}x_2 - x_3 + \frac{1}{\Gamma(2.5)}x_1^{(0.5)} + x_3^{1.5} \\ \dot{x}_2 = x_2 + \frac{1}{2}\frac{1}{x_2} - x_3^{0.5} \\ \dot{x}_3 = \frac{1}{2}x_1 + 1 - \frac{1}{2}x_3^{1.5} \\ x_1 - x_2x_3 = 0 \end{cases} \quad (24)$$

Its exact solutions are

$$\begin{cases} x_1 = t^{1.5} \\ x_2 = t^{0.5} \\ x_3 = t \end{cases} \quad (25)$$

According to Equations (1) and (24), this can be expressed in the matrix form, then we get

$$\mathbf{A} = \begin{bmatrix} -1 & \frac{3}{2} & -1 \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} \frac{1}{\Gamma(2.5)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{f}(\mathbf{x}, t) = \begin{bmatrix} x_3^{1.5} \\ \frac{1}{2}\frac{1}{x_2} - x_3^{0.5} \\ 1 - \frac{1}{2}x_3^{1.5} \end{bmatrix} \quad (26)$$

Using the sliding mode control theory, we convert the fractional order differential-algebraic equation into an equivalent fractional differential equation, and obtain the sliding mode surface, given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1\mathbf{x}^{(0.5)}(t) + \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (27)$$

$$\mathbf{s}(\mathbf{x}) = x_1 - x_2x_3 = 0 \quad (28)$$

Considering the controllability condition, the control matrix is chosen as $\mathbf{g}(\mathbf{x}) = (0, 1, 0)^T$. The parameters of sliding mode are set to be $\varepsilon = 5$ and $\lambda = 5$. The control vector $\mathbf{u}$ is consistent with the control law, Equation (8).

We start with the initial condition $\mathbf{x}(\tau) = (0.001, 0.1, 0.01)^T$, ($\tau \leq 0$). Equation (27) is solved numerically by using the present method, and the numerical solutions are studied by comparing with the exact solutions in the following.

First, the numerical solutions are obtained by the present method with time step $h = 0.001$. Figure 1 shows the numerical solution calculated by the present method and the exact solution obtained by Equation (25). As shown in the figure, the numerical results have a good agreement with the exact results. Figure 2 shows the percentage difference between the numerical solutions and exact solutions. As shown in Figure 2, the difference is very small and decreases monotonically with the increase in t . Note that the maximum occurs at $t = 1$ when $1 \leq t \leq 20$, and its magnitude is less than 7%. The range $0 \leq t < 1$ is not taken into consideration, because the exact solution tends towards 0 when $t \rightarrow 0$, which results in very large percentage difference, even for a small error.

Figure 1. Comparison between the exact solutions and the numerical solutions obtained by the present method.

Figure 2. Percentage differences between the exact solutions and the present method.

Next, the effect of time step on the accuracy of the present method is studied. Figures 3–5 show the percentage difference between the numerical solutions and the exact solutions for different time steps, $h = 0.001, 0.005, 0.01$ and 0.02 . As shown in the figures, for all cases of time steps, the percentage difference increases when $t \rightarrow 1$, but within 5% when $5 < t < 35$. For the case of $t > 25$, the percentage difference increases with the increase in t . Note that for all solutions, the larger the step is, the larger the error is. For the case of $t > 35$, the accuracy of small step is still good (when $h = 0.001$, the maximum error at $t = 50$ is about 1%), while the accuracy of large step drops sharply.

Figure 3. Percentage difference of x_1 between the exact solutions and the present method for different time steps h .

Figure 4. Percentage difference of x_2 between the exact solutions and the present method for different time steps h .

Figure 5. Percentage difference of x_3 between the exact solutions and the present method for different time steps h .

6.2. Comparisons with Other Methods

Chen et al. [26] proposed a numerical method for solving fractional differential equations by using the Grünwald–Letnikov definition. Wang et al. [20] introduced the sliding mode control method to the FDAEs, and obtained the numerical solution by using the predictor-corrector method [23]. In this section, we aim to study the characteristics of the present method by comparing with the two numerical methods.

Consider a numerical example, given by

$$\begin{cases} \dot{x}_1 = -x_1 + x_2 - x_1^{(\alpha)} + x_2x_3 \\ \dot{x}_2 = 0.5x_1 - x_2 \\ \dot{x}_3 = x_1 - x_3 \\ x_2 - x_1x_3 = 0 \end{cases} \quad (29)$$

Assume that the initial condition is $x(\tau) = (1, 1, 1)^T$, ($\tau \leq 0$) and the order of fractional calculus is $\alpha = 0.25$. According to Equation (1), the above equation can be expressed in matrix form, and hence, the coefficient matrices and the nonlinear part are

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 0 \\ 0.5 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{f}(x, t) = \begin{bmatrix} x_2x_3 \\ 0 \\ 0 \end{bmatrix}. \quad (30)$$

Note that Equation (29) has no external input, hence $\mathbf{w}(t) = 0$ and $\mathbf{B}_2 = 0$. According to the sliding mode theory, we convert Equation (29) into fractional differential equation as Equation (4). The control matrix $\mathbf{g}(\mathbf{x})$ should satisfy two conditions as mentioned above, and then we let $\mathbf{g}(\mathbf{x}) = (0, 1, 0)^T$. The sliding mode surface is $\mathbf{s} = x_2 - x_1x_3 = 0$, and we set the parameters of sliding mode $\varepsilon = \lambda = 2$.

First, for comparison, we calculate Equation (29) using the numerical method [26], the predictor-corrector method [20] and the present method, respectively. However, the numerical method [26] cannot solve Equation (29) directly, since there is an algebraic equation. Therefore, we convert Equation (29) by substituting the algebraic equation $x_2 - x_1x_3 = 0$ into the fractional differential equations, and obtain

$$\begin{cases} \dot{x}_1 = -x_1 - x_1^{(\alpha)} + x_1x_3 + x_1x_3^2 \\ \dot{x}_3 = x_1 - x_3 \end{cases} \quad (31)$$

The above equations are equivalent to Equation (29) and suitable for the numerical method [26].

Next, we compare the solutions calculated by the numerical method [26], the predictor-corrector method [20] and the present method. Figure 6 shows the numerical results of three methods with the time step $h = 0.01$. As shown in this figure, the curves of results are not in good agreement when $t = 1$ to 10. The results of the present method are basically smaller than those of other method. Figure 7 shows the numerical results of three methods with time step $h = 0.001$. From the figure, the results of three methods are in good agreement for the case of $h = 0.001$. Therefore, the validity of the present method can be proved. Note that the results of the two methods [20,26] are very close, even for different h . However, Figures 6 and 7 cannot examine the accuracy of the present method since there is no exact solutions for comparison.

Figure 6. Numerical solutions of x_1 , x_2 and x_3 for the case of $h = 0.01$.

Figure 7. Numerical solutions of x_1 , x_2 and x_3 for the case of $h = 0.001$.

Next, in order to illustrate the accuracy of the present method, we compare its results with those obtained by extremely small step h . Figure 8 shows the relative error of the present method and the numerical method [26] comparing to the solution of the numerical method [26] with $h = 0.0001$. Because there is no exact solutions for Equation (29), the results calculated by the numerical method [26] with a sufficiently small step ($h = 0.0001$) are regarded as high-precision solutions. In this section, we only consider the value of x_1 and examine its accuracy. Note that for the both cases of $h = 0.01$ and 0.001 , the present method has higher precision than the numerical method [26].

Figure 8. Percentage difference of x_1 obtained by the present method and the numerical method [26] comparing to the numerical method [26] with $h = 0.0001$.

Table 1 lists the maximum percentage differences of three methods for different steps h . It clearly indicates that the accuracy of the numerical method [26] and the predictor-corrector method [20] is equivalent. However, the present method is much more accurate than the other two methods. For the case of $h = 0.001$, the maximum percentage difference of the present method is less than 1%, while those of the other methods are about 4.5%. For the case of $h = 0.0005$, the accuracy of the present method is sufficiently high, and the error is less than 0.07%.

Table 1. The maximum percentage differences for three methods by comparing with the results of the numerical method [26] of $h = 0.0001$.

No.	Method	Maximum Difference (%)			
		$h = 0.01$	$h = 0.001$	$h = 0.0005$	$h = 0.0001$
1	The numerical method [26]	36.47	4.56	2.31	0
2	The predictor-corrector method [20]	36.94	4.57	2.31	0
3	The present method	11.93	0.85	0.066	−0.70

Table 2 lists the time consumptions of the three methods for different time steps. Note that the numerical method [26] takes the longest time for the same case of h , which is about twice as long as the predictor-corrector method [20]. This illustrates that the computing efficiency is low when solving FDAEs using the numerical method of fractional differential equations. For all cases of h , the predictor-corrector method is the most efficient algorithm, and the present method is very close to it. In fact, the time differences between the two methods can be negligible in most cases. However, according to Table 1, the present method with $h = 0.001$ is much more accurate than the predictor-corrector method with $h = 0.0005$. It only takes 1.60 s for the present method to yield solutions with maximum error about 0.85%. From this point of view, the present method is the most efficient one, because it takes the shortest CPU time but obtains the highest precision.

Table 2. Time consumptions of numerical computing for different methods.

No.	Method	CPU Time (s)			
		$h = 0.01$	$h = 0.001$	$h = 0.0005$	$h = 0.0001$
1	The numerical method [26]	0.10	2.23	7.16	190.95
2	The predictor-corrector method [20]	0.05	1.10	3.66	96.16
3	The present method	0.12	1.60	4.36	102.25

6.3. The Effect of Violation Corrections

The violation corrections can extremely improve the accuracy of the present method. In this section, we study the effect of violation corrections on the accuracy of the present method by using the example Equation (29). Additionally, we consider only the value of x_1 for comparison in this section. Figure 9 shows the comparison between the present method with violation correction and that without violation correction for different time steps. As shown in the figure, the percentage difference increases with increasing h . It illustrates that the violation corrections are more effective for large step h . Figure 10 shows the percentage difference of four methods with $h = 0.01$ comparing to the numerical method [26] with $h = 0.0001$. As shown in the figure, the present method has the highest precision under such conditions. Moreover, it can be seen that even without violation correction, the accuracy of the present method is higher than that of the other two methods.

Figure 9. Percentage difference between the present method with violation correction and that without violation correction for different time steps h .

Figure 10. Percentage difference of four methods with $h = 0.01$ comparing to the numerical method [26] with $h = 0.0001$.

7. Conclusions

This paper presents a numerical algorithm of FDAEs based on the theory of sliding mode control. The direct violation correction method is used to improve the accuracy of the present method. An example with exact solutions is given to illustrate the validity and precision of the present method. It proves that the solution technology of this paper can be applied in a wide range and possesses a very high accuracy. Additionally, the present method is compared to other two numerical methods. It is found that for the same time step, the present method is more accurate, and its time consumption of computing is close to the fastest method. The effect of violation corrections is investigated for the

representative example of FDAEs and is found to be very effective for improving the accuracy of the present method.

Author Contributions: Software, L.W.; validation, Z.F. and J.X.; writing—review and editing, Y.T.; supervision, N.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Jiangsu Province (China), grant number BK20160933 and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China, grant number 17KJB120008.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Podlubny, I.; Chechkin, A.; Skovranek, T.; Chen, Y.Q.; Jara, B.M.V. Matrix approach to discrete fractional calculus II: Partial fractional differential equations. *J. Comput. Phys.* **2009**, *228*, 3137–3153. [\[CrossRef\]](#)
- Moshrefi-Torbati, M.; Hammond, J.K. Physical and geometrical interpretation of fractional operators. *J. Frankl. Inst.* **1998**, *335*, 1077–1086. [\[CrossRef\]](#)
- Xu, J.; Chen, Y.; Tai, Y.; Xu, X.; Shi, G.; Chen, N. Vibration analysis of complex fractional viscoelastic beam structures by the wave method. *Int. J. Mech. Sci.* **2020**, *167*, 105204. [\[CrossRef\]](#)
- Giona, M.; Cerbelli, S.; Roman, H.E. Fractional diffusion equation and relaxation in complex viscoelastic materials. *Phys. A Stat. Mech. Appl.* **1992**, *191*, 449–453. [\[CrossRef\]](#)
- Ingman, D.; Suzdalnitsky, J. Application of differential operator with servo-order function in model of viscoelastic deformation process. *J. Eng. Mech.* **2005**, *131*, 763–767. [\[CrossRef\]](#)
- Povstenko, Y. Fractional heat conduction in a space with a source varying harmonically in time and associated thermal stresses. *J. Therm. Stresses* **2016**, *39*, 1442–1450. [\[CrossRef\]](#)
- Sun, H.G.; Zhang, Y.; Baleanu, D.; Chen, W.; Chen, Y.Q. A new collection of real world applications of fractional calculus in science and engineering. *Commun. Nonlinear Sci. Numer. Simul.* **2018**, *64*, 213–231. [\[CrossRef\]](#)
- Podlubny, I. Fractional-order systems and PI/sup /spl lambda//D/sup /spl mu//-controllers. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [\[CrossRef\]](#)
- Vinagre, B.; Petráš, I.; Podlubny, I.; Chen, Y.Q. Using fractional order adjustment rules and fractional order reference models in model-reference adaptive control. *Nonlinear Dyn.* **2002**, *29*, 269–279. [\[CrossRef\]](#)
- Chen, N.; Tai, Y.; Chen, N. Nonlinear suspension of fractional order calculus in the application of the variable structure sliding mode control. *Dyn. Control* **2009**, *7*, 258–263.
- Gad, S.; Metered, H.; Bassuiny, A.; Ghany, A.M.A. Multi-objective genetic algorithm fractional-order PID controller for semi-active magnetorheologically damped seat suspension. *J. Vib. Control* **2017**, *23*, 1248–1266. [\[CrossRef\]](#)
- Mehrmann, V.; Wunderlich, L. Hybrid systems of differential-algebraic equations—Analysis and numerical solution. *J. Process. Control* **2009**, *19*, 1218–1228. [\[CrossRef\]](#)
- Louca, L.S. Modal analysis reduction of multi-body systems with generic damping. *J. Comput. Sci.* **2014**, *5*, 415–426. [\[CrossRef\]](#)
- Carpinelli, M.; Gubitosa, M.; Mundo, D.; Desmet, W. Automated independent coordinates' switching for the solution of stiff DAEs with the linearly implicit Euler method. *Multibody Syst. Dyn.* **2016**, *36*, 67–85. [\[CrossRef\]](#)
- Ghanbari, F.; Ghanbari, K.; Mokhtary, P. Generalized Jacobi-Galerkin method for nonlinear fractional differential algebraic equations. *Comput. Appl. Math.* **2018**, *37*, 5456–5475. [\[CrossRef\]](#)
- Ding, X.L.; Jiang, Y.L. Nonnegativity of Solutions of Nonlinear Fractional Differential-Algebraic Equations. *Acta Math. Sci.* **2018**, *38*, 756–768. [\[CrossRef\]](#)
- Liu, H.; Fu, Y.; Li, B. Discrete Waveform Relaxation Method for Linear Fractional Delay Differential-Algebraic Equations. *Discret. Dyn. Nat. Soc.* **2017**, *2017*, 6306570. [\[CrossRef\]](#)
- Ghanbari, F.; Mokhtary, P.; Ghanbari, K. Numerical solution of a class of fractional order integro-differential algebraic equations using Muntz-Jacobi Tau method. *J. Comput. Appl. Math.* **2019**, *362*, 172–184. [\[CrossRef\]](#)

19. Ghanbari, F.; Mokhtary, P.; Ghanbari, K. On the numerical solution of a class of linear fractional integro-differential algebraic equations with weakly singular kernels. *Appl. Numer. Math.* **2019**, *144*, 1–20. [[CrossRef](#)]
20. Wang, L.; Chen, N. The predictor-corrector solution for fractional order differential algebraic equation. In Proceedings of the 2014 International Conference on Fractional Differentiation and Its Applications (ICFDA 2014), Catania, Italia, 23–25 June 2014; pp. 1–6.
21. Mortezaee, M.; Ghovatmand, M.; Nazemi, A. Solving variable-order fractional differential algebraic equations via generalized fuzzy hyperbolic model with application in electric circuit modeling. *Soft Comput.* **2020**. [[CrossRef](#)]
22. Ibis, B.; Bayram, M. Numerical comparison of methods for solving fractional differential-algebraic equations (FDAEs). *Comput. Math. Appl.* **2011**, *62*, 3270–3278. [[CrossRef](#)]
23. Diethelm, K.; Ford, N.J.; Freed, A.D. A Predictor-Corrector Approach for the Numerical Solution of Fractional Differential Equations. *Nonlinear Dyn.* **2002**, *29*, 3–22. [[CrossRef](#)]
24. Podlubny, I. *Fractional Differential Equations*; ACADEMIC PRESS: San Diego, CA, USA, 1999; p. 198.
25. Hung, J.Y.; Gao, W.; Hung, J.C. Variable structure control: A survey. *IEEE Trans. Ind. Electron.* **1993**, *40*, 2–22. [[CrossRef](#)]
26. Chen, N.; Chen, N.; Tai, Y. A Numerical Scheme for Nonlinear Dynamic System with Fractional Feedback Control. In Proceedings of the ENOC, Rome, Italy, 24–29 July 2011; Volume 7, pp. 24–29.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).