Research Article

A New Numerical Technique for Index-3 DAEs Arising from Constrained Multibody Mechanical Systems

Brahim Benhammouda

Higher Colleges of Technology, Abu Dhabi Men's College, P.O. Box 25035, Abu Dhabi, UAE

Correspondence should be addressed to Brahim Benhammouda; bbenhammouda@hct.ac.ae

Received 27 October 2021; Accepted 23 January 2022; Published 27 February 2022

Academic Editor: Charalampos Skokos

Copyright © 2022 Brahim Benhammouda. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Index-3 differential-algebraic equations (DAEs) are mathematical models for the dynamics of constrained multibody mechanical systems that arise in many applications. These DAEs are known to pose a challenge to numerical methods. The purpose of this paper is to propose a new approach to efficiently solve these equations. This approach relies on an effective combination of the power series method (PSM) with the Adomian polynomials. Here, the PSM is directly applied to these DAEs without using the usual index reduction techniques, which are costly and often lead to nonphysical solutions. We expand the nonlinear terms in a series form using the Adomian polynomials to overcome the limitation of the PSM in collecting the coefficients of the power series solution. This technique has led to a simple and efficient algorithm. The domain of convergence of the power series solution is expanded by developing a multistage PSM (MSPSM). To demonstrate the efficiency of the MSPSM and show its applicability, an index-3 nonlinear DAE problem describing a two-link planar robot arm is solved. The numerical results show that the MSPSM is a powerful tool for solving the index-3 DAEs arising from constrained multibody mechanical systems.

1. Introduction

Constrained multibody systems [1] have applications in many important applications like rocket-towed systems [2], aerospace engineering [3], vehicle industry [4, 5], and robotics [6]. Constrained mechanical multibody systems are collections of rigid and/or flexible bodies that are interconnected by kinematic joints. The role of these kinematic joints is to impose some restrictions on the relative motion of the bodies of the system. As examples of constrained mechanical multibody systems, one can mention vehicles, cranes, slider cranks, and constrained robot arms. To express restrictions on the motion of the bodies, algebraic equations are used. Due to these equations, some constraint forces are present in the system. The forces that act on the multibody system components can come from springs, dampers, and actuator or external forces. The dynamics of these complex mechanical systems is described by nonlinear index-3 differential-algebraic equations (DAEs) systems of the form

$$\begin{cases} u' = v, \\ M(u)v' = f(u, v, t) - G^T(u, t)\lambda, \\ 0 = g(u, t), \end{cases} \quad (1)$$

where $t \in [0, T]$ is time, the dash ($'$) represents the time derivative, and $u(t) \in \mathbb{R}^{n_u}$ and $v(t) \in \mathbb{R}^{n_v}$ specify the positions and the orientations of all bodies and their velocities, respectively. The vector $\lambda \in \mathbb{R}^{n_\lambda}$ is the vector of Lagrange multipliers. The matrix $M(u) \in \mathbb{R}^{n_u \times n_u}$ is a mass matrix. The mapping $f: \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \times [0, T] \rightarrow \mathbb{R}^{n_u}$ defines the applied and internal forces (other than the constraint forces), whereas $g: \mathbb{R}^{n_u} \times [0, T] \rightarrow \mathbb{R}^{n_\lambda}$, ($n_\lambda \leq n_u$) defines the constraints. The term $G(u, t)^T \lambda$ represents the constraint forces, where $G(u, t) = \partial g / \partial u \in \mathbb{R}^{n_\lambda \times n_u}$ denotes the Jacobian of $g(u, t)$. To solve DAE (1) uniquely, some initial conditions

$$\begin{cases} u(0) = \alpha, \\ v(0) = \beta, \end{cases} \quad (2)$$

are required. The given vectors α and β , which specify the initial configuration and initial velocity, must be chosen consistent, that is, to satisfy the algebraic system

$$\begin{aligned} g(\alpha, 0) &= 0, \\ G(\alpha, 0)\beta &= 0. \end{aligned} \quad (3)$$

Note here that no initial condition is required for the unknown $\lambda(t)$ as $\lambda(0)$ is already predetermined by DAE (1) and the initial conditions (2).

Throughout this paper, we assume that $M(u)$, $f(u, v, t)$, and $g(u, t)$ are analytical. A standard assumption on the Jacobian $G(u, t)$ is the full row rank condition

$$\text{rank } G(u, t) = n_\lambda, \quad (4)$$

which means that the constraint equations defined by $g(u, t)$ are linearly independent.

In addition, the matrix $M(u)$ is assumed to be symmetric and positive definite on $\text{Ker } G(u, t)$; that is,

$$\xi^T M(u) \xi > 0, \quad \text{for all } \xi \in \text{Ker } G(u, t). \quad (5)$$

If assumptions (4)-(5) hold, then the matrix

$$\begin{pmatrix} M(u) & G^T(u, t) \\ G(u, t) & 0 \end{pmatrix}, \quad (6)$$

is nonsingular and therefore the DAE (1) is an index-3 DAE. If initial conditions (2) are consistent, then the DAEs system (1) has a unique solution [7]. The index used in this paper is the differential index. It is defined as the minimum number of time differentiation needed to transform the DAE into an ordinary differential equation [8]. Those DAEs with an index greater than one, called also higher index DAEs, are widely known to be difficult to solve. The reason is that these DAEs are in fact ill-posed problems in a certain sense and direct discretizations do not lead to satisfactory results in general [8–11]. There are several methods in the literature for solving DAEs, which include Runge–Kutta methods [8, 12], projected explicit and implicit Taylor series methods [13], hybrid block methods [14], variational methods [15], reduction methods [16, 17], transformation methods [18], Adomian decomposition method [19], and the differential transform method [20]. One can find several techniques for solving DAEs of the form (1) [21–28]. The common way to treat these DAEs is to transform them into lower-index DAEs using successive time differentiation before using a numerical method. However, one major drawback of this technique is that the numerical solution of the transformed DAE may not satisfy the algebraic constraints of original DAE (1). This violation of constraints leads to nonphysical solutions. To control the error in the constraints, other techniques like stabilization or augmented Lagrangian formulation have been constructed [22–24, 29]. The most popular one is the Baumgarte stabilization method [22]. However, the major drawback of this method is the way of selecting the feedback parameters. The great challenge is then to construct efficient approximation methods for the DAE (1), which satisfy all the constraints of the DAE.

The power series method (PSM) [30, 31] is a well-known method for solving ordinary differential equations. Although this technique is effective for solving many problems like mechanical systems [27, 32], nonlinear Schrödinger equations [33], partial differential equations [34, 35], nonlinear systems of algebraic equations [36], and epidemiology [37], it was not frequently used for nonlinear problems. The reason is that the method may converge only over smaller intervals and the collection of the coefficients of the power series solution can be a difficult task. We can overcome these drawbacks using the Adomian polynomials and the multistage technique.

In this paper, we present a new technique to solve general strongly nonlinear index-3 DAEs (1) arising from constrained mechanical systems efficiently. This technique is based on an effective combination of the power series method (PSM) with the Adomian polynomials [38], where there are several algorithms for computing these polynomials [39–43]. The novelty in our approach is in the direct application of the approximation method to these DAEs without the use of the usual index reduction techniques which are complex and costly and often lead to nonphysical solutions. The proposed technique has led to a general solution procedure with a simple and efficient algorithm that has three main features. The first one is that the algorithm includes all the DAE algebraic constraints. This ensures that the numerical solution remains on the DAE constraint manifolds. The second feature is that our algorithm requires only the solution of small algebraic systems for the algebraic variables with positive definite coefficient matrices. The third feature is that our technique can cope with any type of nonlinearity in the DAE using the Adomian polynomials. Finally, to enlarge the domain of convergence of the numerical solution, we developed a multistage PSM (MSPSM). This allows us to simulate constrained mechanical multibody systems over large time intervals. The proposed technique has led to a simple and efficient algorithm that can be easily implemented to solve general nonlinear index-3 DAEs. Our technique can be used in designing software for industrial applications and provides a powerful tool for mechanical engineers in particular. To demonstrate the effectiveness and accuracy of the multistage MSPSM, an index-3 DAE problem that models a two-link planar robot arm is solved by implementing the proposed technique in Maple. Numerical simulations show that the multistage MSPSM is a powerful tool to solve the DAE initial-value problem (1)-(2).

This paper is organized as follows: in Section 2, we review the Adomian polynomials. Next, in Section 3, we derive the proposed method for solving the DAE initial-value problem (1)-(2). Then, in Section 4, we apply the developed technique to solve a nonlinear index-3 DAE that models a two-link planar robot arm and give a discussion of numerical results. Finally, a conclusion is given in Section 5.

2. Adomian Polynomials

In this section, we briefly review the Adomian polynomials [38]. These polynomials will be used to expand the nonlinear

terms in the differential equation. A nonlinear term $A(u)$ in a differential equation is often decomposed as

$$A(u) = \sum_{k=0}^{\infty} A_k(u_0, u_1, \dots, u_k), \quad (7)$$

where the Adomian polynomials A_k are generated, for all types of nonlinearities, from

$$A_k := A_k(u_0, u_1, \dots, u_k) = \frac{1}{k!} \frac{d^k}{d\xi^k} \left(A \left(\sum_{i=0}^{\infty} \xi^i u_i \right) \right)_{\xi=0}, \quad k \geq 0. \quad (8)$$

Here u_k , $k = 0, 1, 2, \dots$, denote the components used in the expansion

$$u = \sum_{k=0}^{\infty} u_k. \quad (9)$$

Using (8), one can easily compute the first few Adomian polynomials

$$\begin{aligned} A_0 &= A(u_0), \\ A_1 &= u_1 A^{(1)}, \\ A_2 &= u_2 A^{(1)} + \frac{u_1^2}{2!} A^{(2)}, \\ A_3 &= u_3 A^{(1)} + u_1 u_2 A^{(2)} + \frac{u_1^3}{3!} A^{(3)}, \\ A_4 &= u_4 A^{(1)} + \left(\frac{u_2^2}{2!} + u_1 u_3 \right) A^{(2)} + \frac{u_1^2 u_2}{2!} A^{(3)} + \frac{u_1^4}{4!} A^{(4)}, \end{aligned} \quad (10)$$

where $A^{(k)} := d^k A(u_0)/du_0^k$.

In a similar way, one can generate any desired Adomian polynomial from (8). For a two-variable function $A(u, v)$, the first few Adomian polynomials are

$$\begin{aligned} A_1 &= u_1 A^{(1,0)} + v_1 A^{(0,1)}, \\ A_2 &= u_2 A^{(1,0)} + v_2 A^{(0,1)} + \frac{u_1^2}{2} A^{(2,0)} \\ &\quad + \frac{v_1^2}{2} A^{(0,2)} + u_1 v_1 A^{(1,1)}, \\ A_3 &= u_3 A^{(1,0)} + v_3 A^{(0,1)} + u_1 u_2 A^{(2,0)} + v_1 v_2 A^{(0,2)} \\ &\quad + (u_1 v_2 + u_2 v_1) A^{(1,1)} \\ &\quad + \frac{u_1^3}{6} A^{(3,0)} + \frac{v_1^3}{6} A^{(0,3)} + \frac{u_1^2 v_1}{2} A^{(2,1)} + \frac{u_1 v_1^2}{2} A^{(1,2)}, \end{aligned} \quad (11)$$

where $A^{(k,l)} := \partial^{k+l} A(u_0, v_0)/\partial u_0^k \partial v_0^l$.

When u is expanded in a power series $u = \sum_{k=0}^{\infty} u_k t^k$, the nonlinear term $A(u)$ can be expanded using the Adomian polynomials as follows [44]:

$$A \left(\sum_{k=0}^{\infty} u_k t^k \right) = \sum_{k=0}^{\infty} A_k(u_0, u_1, \dots, u_k) t^k. \quad (12)$$

One silent property of the Adomian polynomials which we will exploit when developing our algorithm is the fact that A_k is affine with respect to u_k for all $k \geq 1$.

3. Derivation of the Proposed Technique

In this section, we construct our method for the solution of DAEs (1). These systems are index-3 DAEs which are known to be difficult to solve even numerically. The technique we propose here is an effective combination of the power series method (PSM) with the Adomian polynomials [38]. To make our algorithm robust and efficient, we apply the PSM directly to the DAE (1) and do not use index reduction techniques which are complex and computationally expensive. Then, we expand the nonlinear terms in power series using the Adomian polynomials [44]. To incorporate the algebraic constraints in the solution procedure, we use the fact that the Adomian polynomials A_k are affine polynomials with respect to u_k for all $k \geq 1$. All this has simplified our algorithm and led to the solution of only small $n_\lambda \times n_\lambda$ nonsingular linear algebraic systems. This has also forced the numerical solution to satisfy all the algebraic constraints of the original DAE. Our algorithm is simple and can be easily implemented in Maple or Mathematica. The following new theorem gives some important results for the Adomian polynomials in the case of vectors and matrices.

Theorem 1. Let $q = (q_1, \dots, q_n)^T \in \mathbb{R}^n$, $h = (h_1, \dots, h_n)^T \in \mathbb{R}^n$, and $u = (u_1, \dots, u_n)^T \in \mathbb{R}^n$ be three vectors and $M := (M(u)_{i,j}) \in \mathbb{R}^n \times \mathbb{R}^n$ be an invertible matrix. Let $q = \sum_{k=0}^{\infty} q_k$ and $h = \sum_{k=0}^{\infty} h_k$ with $q_k = (q_{1,k}, \dots, q_{n,k})^T$ and $h_k = (h_{1,k}, \dots, h_{n,k})^T$, where $q_{i,k}$ and $h_{i,k}$ are the k th Adomian polynomials of the components q_i and h_i , $i = 1, \dots, n$, respectively. Let $M_k = (M_{i,j,k})$, where $M_{i,j,k} = M_{i,j,k}(u_{1,0}, \dots, u_{n,0}), \dots, (u_{1,k}, \dots, u_{n,k})$ denotes the k th Adomian polynomial of the entry $M_{i,j}$. Then

$$\begin{aligned} (a) \quad h &= Mq \Rightarrow h_k = \sum_{l=0}^k M_{k-l} q_l \\ (b) \quad q &= M^{-1}h \Rightarrow M_0 q_k = h_k - \sum_{l=0}^{k-1} M_{k-l} q_l, \quad k = 0, 1, \dots \end{aligned}$$

Proof. (a) Let $h = Mq$; then,

$$\begin{aligned} h_k &= (Mq)_k = \begin{pmatrix} \sum_{j=1}^n (M_{1,j} q_j)_k \\ \vdots \\ \sum_{j=1}^n (M_{n,j} q_j)_k \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^n \sum_{l=0}^k M_{1,j,k-l} q_{j,l} \\ \vdots \\ \sum_{j=1}^n \sum_{l=0}^k M_{n,j,k-l} q_{j,l} \end{pmatrix} = \sum_{l=0}^k M_{k-l} q_l. \end{aligned} \quad (13)$$

(b) Let $q = M^{-1}h$; then $h = Mq$. Using (a), we have

$$\begin{aligned} h_k &= (Mq)_k = \sum_{l=0}^k M_{k-l}q_l \\ &= M_0q_k + \sum_{l=0}^{k-1} M_{k-l}q_l. \end{aligned} \quad (14)$$

Therefore, q_k is given by

$$M_0q_k = h_k - \sum_{l=0}^{k-1} M_{k-l}q_l. \quad (15)$$

This completes the proof of the theorem.

We start our technique by writing first the second equation of DAE (1) as

$$v' = M^{-1}(u)(f(u, v, t) - G^T(u, t)\lambda). \quad (16)$$

Now, let

$$h(u, v, \lambda, t) = f(u, v, t) - G^T(u, t)\lambda \quad (17)$$

and

$$q(u, v, \lambda, t) = M^{-1}(u)h(u, v, \lambda, t). \quad (18)$$

Then, (16) can be written as

$$v' = q(u, v, \lambda, t). \quad (19)$$

Then, we expand the solution components $u(t)$, $v(t)$, and $\lambda(t)$ in power series as

$$\begin{aligned} u(t) &= \sum_{k=0}^{\infty} u_k t^k, \\ v(t) &= \sum_{k=0}^{\infty} v_k t^k, \\ \lambda(t) &= \sum_{k=0}^{\infty} \lambda_k t^k, \end{aligned} \quad (20)$$

where the unknown vectors $u_k \in \mathbb{R}^{n_u}$, $v_k \in \mathbb{R}^{n_v}$, and $\lambda_k \in \mathbb{R}^{n_\lambda}$, $k = 0, 1, 2, \dots$, will be determined later by our method.

The nonlinear term $q(u, v, t)$ is expanded using the Adomian polynomial as

$$q\left(\sum_{k=0}^{\infty} u_k t^k, \sum_{k=0}^{\infty} v_k t^k, t\right) = \sum_{n=0}^{\infty} q_k t^k, \quad (21)$$

where q_k denotes the vector of the Adomian polynomials of the components of q .

Substituting (20) into the first equation of the DAE (1) and (19), we obtain

$$\sum_{k=1}^{\infty} k u_k t^{k-1} = \sum_{k=0}^{\infty} v_k t^k \quad (22)$$

and

$$\sum_{k=1}^{\infty} k v_k t^{k-1} = \sum_{k=0}^{\infty} q_k t^k. \quad (23)$$

Equations (22) and (23) lead to the recursions

$$v_{k-1} = k u_k, \quad k \geq 1, \quad (24)$$

and

$$k(k-1)u_k = q_{k-2}, \quad k \geq 2. \quad (25)$$

Further, the nonlinear terms $f(u, v, t)$, $g(u, t)$, $M(u)$, and $G(u, t)$ are also expanded in power series using the Adomian polynomials as

$$\begin{aligned} f\left(\sum_{k=0}^{\infty} u_k t^k, \sum_{k=0}^{\infty} v_k t^k, t\right) &= \sum_{k=0}^{\infty} f_k t^k, \\ g\left(\sum_{k=0}^{\infty} u_k t^k, t\right) &= \sum_{k=0}^{\infty} g_k t^k, \end{aligned} \quad (26)$$

and

$$\begin{aligned} M\left(\sum_{k=0}^{\infty} u_k t^k\right) &= \sum_{k=0}^{\infty} M_k t^k, \\ G\left(\sum_{k=0}^{\infty} u_k t^k, t\right) &= \sum_{k=0}^{\infty} G_k t^k. \end{aligned} \quad (27)$$

where $f_k := f_k(u_0, v_0, \dots, u_k, v_k)$ and $g_k := g_k(u_0, \dots, u_k)$ denote the vectors of the Adomian polynomials of the components of the vectors f and g , respectively. The matrices $M_k(u_0, \dots, u_k)$ and $G_k(u_0, \dots, u_k)$ are the matrices of the Adomian polynomials of the entries of the matrices M and G .

Then, from (18) and the result (b) of the previous theorem, we have

$$q_k = M_0^{-1} \left(h_k - \sum_{l=0}^{k-1} M_{k-l} q_l \right). \quad (28)$$

Now, using (17) and the result (a) of the previous theorem, we obtain

$$h_k = f_k - \sum_{l=0}^k G_{k-l}^T \lambda_l. \quad (29)$$

Substituting equation (29) into equation (28), we deduce that

$$q_k = M_0^{-1} \left(f_k - \sum_{l=0}^k G_{k-l}^T \lambda_l - \sum_{l=0}^{k-1} M_{k-l} q_l \right), \quad (30)$$

which can be written as

$$q_k = -M_0^{-1} G_0^T \lambda_k + M_0^{-1} \left(f_k - \sum_{l=0}^{k-1} (G_{k-l}^T \lambda_l + M_{k-l} q_l) \right). \quad (31)$$

Equation (31) can be written as

$$q_{k-2} = -M_0^{-1} G_0^T \lambda_{k-2} + r_{k-2}, \quad (32)$$

where

$$r_{k-2} = M_0^{-1} \left(f_{k-2} - \sum_{l=0}^{k-3} (G_{k-2-l}^T \lambda_l + M_{k-2-l} q_l) \right). \quad (33)$$

Here, it is important to note that r_{k-2} depends only on $u_0, \dots, u_{k-1}$, $v_0, \dots, v_{k-1}$, and $\lambda_0, \dots, \lambda_{k-3}$.

Equations (25) and (32) lead to the algebraic linear system for u_k and λ_{k-2} .

$$k(k-1)u_k + M_0^{-1} G_0^T \lambda_{k-2} = r_{k-2}, \quad k \geq 2. \quad (34)$$

Using the expansion of g from (26) in the last equation of DAE (1), we have

$$\begin{aligned} g_k &= 0, \\ k &= 0, 1, \dots \end{aligned} \quad (35)$$

We split g_k as

$$g_k = -s_k + G_0 u_k, \quad k \geq 2. \quad (36)$$

From the definition of the Adomian polynomial g_k of g , the term s_k depends only on $u_0, \dots, u_{k-1}$.

Using now equation (35), we obtain the algebraic linear system for u_k .

$$G_0 u_k = s_k, \quad k \geq 2. \quad (37)$$

Finally, the unknown vectors u_k and λ_k are determined by solving the system linear algebraic system formed by (34) and (37). Then, the unknown vector v_k is obtained from (24). Since the Jacobian $G(u, t)$ has full row rank and the matrix $M(u)$ is positive definite, then the linear algebraic system formed by (34) and (37) has a unique solution. To solve this system, we multiply system (34) from the left by G_0 then use (37) to obtain the following linear algebraic system for λ_{k-2} .

$$G_0 M_0^{-1} G_0^T \lambda_{k-2} = G_0 r_{k-2} - k(k-1)s_k, \quad k \geq 2. \quad (38)$$

Since the rank condition (4) holds and $M(u)$ is positive definite, then the matrix $G_0 M_0^{-1} G_0^T$ is nonsingular and system (38) determines λ_{k-2} uniquely for $k \geq 2$. Substituting the expression of λ_{k-2} into system (34), we can compute the unknown vector u_k , for $k \geq 2$. Using now (24), we find the unknown vector v_k , for $k \geq 1$. The initial terms for the recursion (38), (34), and (24) are $u_0 = u(0)$, $v_0 = v(0)$.

Finally, we obtain an approximate solution for the DAE initial-value problem (1)-(2) as

$$\begin{aligned} u(t) &= \sum_{k=0}^K u_k t^k, \\ v(t) &= \sum_{k=0}^{K-1} v_k t^k, \\ \lambda(t) &= \sum_{k=0}^{K-2} \lambda_k t^k, \end{aligned} \quad (39)$$

where K is the order of approximation.

The process of the PSM for solving the DAE initial-value problem (1)-(2) is given by (Algorithm 1).

The PSM may lead to solutions of the DAE initial-value problem (1)-(2) that converge only for small intervals. Therefore, we extend the interval of convergence of the PSM by developing a multistage PSM (MSPSM). The process of the MSPSM is described as follows (Algorithm 2). $\square$

4. Numerical Simulation and Discussion

In this section, an example of DAEs (1)-(2) taken from [25] is solved to illustrate and demonstrate the effectiveness of our technique. To this end, a maple code is developed and we obtained the numerical results discussed in this section. The example used here describes the dynamics of a two-link planar robot arm subject to a prescribed trajectory (see Figure 1).

The robot arm is composed by two rigid links. The left end of the first link (link on the left) is fixed at the origin of the x y -plane but allowed to rotate. The angle of rotation of this end is measured from the x -axis and is denoted by θ_1 . The right end of this first link is connected to the left end of the second link (link on the right) at (x_1, y_1) . This second link is allowed to rotate in the x , y -plane and the angle of rotation is denoted by θ_2 as shown in Figure 1. The masses and lengths of the links are denoted by m_i and l_i , $i = 1, 2$, respectively. The coordinates (x_1, y_1) of the middle joint of the robot arm are given by

$$\begin{aligned} x_1 &= l_1 \cos \theta_1, \\ y_1 &= l_1 \sin \theta_1, \end{aligned} \quad (40)$$

and the coordinates (x_2, y_2) of the end-effector of the robot arm are

$$\begin{aligned} x_2 &= x_1 + l_2 \cos(\theta_1 + \theta_2), \\ y_2 &= y_1 + l_2 \sin(\theta_1 + \theta_2). \end{aligned} \quad (41)$$

If the end-effector of the robot arm is constrained to move along the x -axis, then $y_2 = 0$ and we obtain the following constraint on the angles θ_1 and θ_2 .

$$l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) = 0. \quad (42)$$

For this robot arm example, the mass matrix is

```

input
  read  $K$  (order of approximation);
  read  $n_u$  and  $n_\lambda < n_u$ ,  $M \in \mathbb{R}^{n_u \times n_u}$ ,  $f \in \mathbb{R}^{n_u}$  and  $g \in \mathbb{R}^{n_\lambda}$ ;
  read  $u_0 \in \mathbb{R}^{n_u}$  and  $v_0 \in \mathbb{R}^{n_\lambda}$  (consistent initial conditions);
output
  the approximate solution  $u(t), v(t), \lambda(t)$ ;
initialization
   $u_0 := u(0), v_0 := v(0), u_1 := v_0$ ;
while  $0 \leq k \leq K$  do
  compute  $M_k, G_k, f_k, g_k$ .
end
   $H := G_0 M_0^{-1} G_0^T$ ;
while  $2 \leq k \leq K$  do
   $s_k := -g_k + G_0 u_k$ ;
   $r_{k-2} := M_0^{-1} (f_{k-2} - \sum_{l=0}^{k-3} (G_{k-2-l}^T \lambda_l + M_{k-2-l} q_l))$ ;
  Solve  $H \lambda_{k-2} = G_0 r_{k-2} - k(k-1)s_k$  for  $\lambda_{k-2}$ ;
   $q_{k-2} := r_{k-2} - M_0^{-1} G_0^T \lambda_{k-2}$ ;
   $u_k := q_{k-2} / (k(k-1))$ ;
   $v_{k-1} := k u_k$ 
end

```

ALGORITHM 1: PSM Algorithm.

- (1) Choose the number N of subdivisions and the order of approximation K .
- (2) Subdivide the interval $[0, T]$ into N equal subintervals $[t_{i-1}, t_i]$, $i = 1, \dots, N$ of size $h = T/N$, where $t_i = ih$.
- (3) Let $i = 1$, apply the PSM to the DAE (1) on $[0, t_1]$ using $u_0^1 = u(0)$, $v_0^1 = v(0)$ to obtain the approximate solution $u^1(t) = \sum_{k=0}^K u_k^1 t^k$, and $v^1(t) = \sum_{k=0}^{K-1} v_k^1 t^k$, $\lambda^1(t) = \sum_{k=0}^{K-2} \lambda_k^1 t^k$, $0 \leq t \leq t_1$.
- (4) From $i = 2$ until N , apply the PSM to DAE (1) on $[t_{i-1}, t_i]$ using $u_0^i = u^{i-1}(t_{i-1})$, $v_0^i = v^{i-1}(t_{i-1})$ to obtain the approximate solution $u^i(t) = \sum_{k=0}^K u_k^i (t - t_{i-1})^k$, $v^i(t) = \sum_{k=0}^{K-1} v_k^i (t - t_{i-1})^k$, and $\lambda^i(t) = \sum_{k=0}^{K-2} \lambda_k^i (t - t_{i-1})^k$, $t_{i-1} \leq t \leq t_i$.
- (5) Repeat step 4.
- (6) The approximate solution of DAE initial-value problem (1)-(2) on $[0, T]$ is $u(t) = \begin{cases} u^1(t), & 0 \leq t \leq t_1 \\ u^2(t), & t_1 \leq t \leq t_2 \\ \vdots \\ u^N(t), & t_{N-1} \leq t \leq T \end{cases}$,
 $v(t) = \begin{cases} v^1(t), & 0 \leq t \leq t_1 \\ v^2(t), & t_1 \leq t \leq t_2 \\ \vdots \\ v^N(t), & t_{N-1} \leq t \leq T \end{cases}$ and $\lambda(t) = \begin{cases} \lambda^1(t), & 0 \leq t \leq t_1 \\ \lambda^2(t), & t_1 \leq t \leq t_2 \\ \vdots \\ \lambda^N(t), & t_{N-1} \leq t \leq T \end{cases}$.

ALGORITHM 2: MSPSM algorithm.

FIGURE 1: A constrained two-link planar robot arm.

$$M(\theta_1, \theta_2) = \begin{pmatrix} m_1 \frac{l_1^2}{3} + m_2 \left(l_1^2 + \frac{l_2^2}{3} + l_1 l_2 \cos \theta_2 \right) & m_2 \left(\frac{l_2^2}{3} + 0.5 l_1 l_2 \cos \theta_2 \right) \\ m_2 \left(\frac{l_2^2}{3} + 0.5 l_1 l_2 \cos \theta_2 \right) & m_2 \frac{l_2^2}{3} \end{pmatrix}. \quad (43)$$

The force term is

$$f\left(\theta_1, \theta_2, \frac{d\theta_1}{dt}, \frac{d\theta_2}{dt}, t\right) = \begin{pmatrix} (l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2))(d\theta_1/dt) - 3\theta_1 \\ l_2 \cos(\theta_1 + \theta_2)(d\theta_1/dt) + (1 - 1.5 \cos \theta_2)\theta_1 \end{pmatrix}. \quad (44)$$

And the constraint function is

$$g(\theta_1, \theta_2, t) = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2). \quad (45)$$

In this numerical simulation, we take $l_1 = l_2 = 1$ and $m_1 = m_2 = 3$. Using the notation of the DAE system (1), with $n_u = 2$, $n_\lambda = 1$, $u = (u_1, u_2)^T = (\theta_1, \theta_2)^T$, and $v = du/dt$, we have

$$M(u) = \begin{pmatrix} 5 + 3 \cos u_2 & 1 + 1.5 \cos u_2 \\ 1 + 1.5 \cos u_2 & 1 \end{pmatrix} \quad (46)$$

and

$$f(u, v, t) = \begin{pmatrix} (\cos u_1 + \cos(u_1 + u_2))v_1 - 3u_1 \\ \cos(u_1 + u_2)v_1 + (1 - 1.5 \cos u_2)u_1 \end{pmatrix}. \quad (47)$$

The constraint function becomes

$$g(u, t) = \sin u_1 + \sin(u_1 + u_2). \quad (48)$$

And its Jacobian $G(u, t) = g'_u(u, t) = (\cos u_1 + \cos(u_1 + u_2), \cos(u_1 + u_2))$ is full row rank $n_\lambda = 1$. It is easy to check that the exact solution of the DAE corresponding to this robot arm problem is $u(t) = (\sin t, -2 \sin t)^T$, $v(t) = (\cos t, -2 \cos t)^T$, and $\lambda(t) = \cos t$. This solution corresponds to the initial conditions

$$\begin{aligned} u(0) &= (0, 0)^T, \\ v(0) &= (1, -2)^T. \end{aligned} \quad (49)$$

These initial conditions are consistent, since they satisfy $g(u(0), 0) = 0$ and $g'_u(u(0), 0)v(0) = 0$. They correspond to the position where the robot arm initially lies along the x -axis and starts moving with the initial velocity $v(0) = (1, -2)^T$. The DAE corresponding to this robot arm example is an index-3 DAE, therefore being difficult to solve even numerically. To solve it using our technique, we have implemented Algorithm 1 presented in Section 3 in Maple 15. Taking the number of approximation terms $K = 10$, our technique was able to compute the first terms of the Maclaurin power series of the exact solution as

$$\begin{aligned} u(t) &= t \begin{pmatrix} 1 \\ -2 \end{pmatrix} - \frac{t^3}{3!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \frac{t^5}{5!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} - \frac{t^7}{7!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \frac{t^9}{9!} \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \\ v(t) &= \begin{pmatrix} 1 \\ -2 \end{pmatrix} - \frac{t^2}{2!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \frac{t^4}{4!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} - \frac{t^6}{6!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \frac{t^8}{8!} \begin{pmatrix} 1 \\ -2 \end{pmatrix} \end{aligned} \quad (50)$$

and

$$\lambda(t) = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \frac{t^8}{8!}. \quad (51)$$

This truncated power series may give accurate solutions only on small intervals.

To extend the domain of convergence of the numerical solution, we implemented Algorithm 2 of the multistage PSM (MSPSM) given in Section 3. For the numerical simulations, we applied the MSPSM over the time interval $[0, T] = [0, 12]$, with an order of approximation $K = 5$, and a number of interval subdivisions $N = 240$. Starting the recursion with $u_0 = u(0) = (0, 0)^T$, $v_0 = v(0) = (1, -2)^T$ and using the MSPSM, we obtain the numerical results shown in Figures 1 to 9. These figures show the solution components together with their corresponding absolute errors. In the figures on the left, the solid line represents the numerical solution and the dashed line represents the exact solution. Figure 2 on the left shows the numerical solution for the angle $\theta_1 = u_1$ together with the exact angle θ_1 whereas the figure on the

right shows the absolute error for the angle θ_1 . From Figure 2, we can see that there is a good agreement between the computed and exact values for the angle θ_1 and the absolute error is 2×10^{-7} . In Figure 3, the left figure shows the numerical solution for the angle $\theta_2 = u_2$ together with the exact angle θ_2 whereas the figure on the right shows the absolute error for the angle θ_2 . From Figure 3, we can see that there is a good agreement between the computed and exact values for the angle θ_2 and the absolute error is 5×10^{-7} . In Figure 4, the left figure shows the numerical solution for the velocity of the angle θ_1 together with the exact velocity of this angle whereas the figure on the right shows the absolute error for the velocity of this angle. From Figure 4, we can see that there is a good agreement between the computed and exact velocities for this angle θ_1 . The absolute error is 2×10^{-7} . Similarly, in Figure 5, the left figure shows the numerical solution for the velocity of the angle θ_2 together with the exact velocity of this angle whereas the figure on the right shows the absolute error for the velocity of this angle. From Figure 5, we can see that there is

FIGURE 2: Approximate solution component u_1 (a) and absolute error e_{u_1} (b). FIGURE 3: Approximate solution component u_2 (a) and absolute error e_{u_2} (b). FIGURE 4: Approximate solution component v_1 (a) and absolute error e_{v_1} (b). FIGURE 5: Approximate solution component v_2 (a) and absolute error e_{v_2} (b).

a good agreement between the computed and exact velocities for this angle θ_2 . The absolute error is 5×10^{-7} . In Figure 6, the left figure shows the numerical solution for the Lagrange multiplier λ together with the exact value of λ whereas the figure on the right shows the absolute error for this algebraic variable λ . From Figure 6, we can see that there is a good agreement between the computed and exact values for λ . The absolute error for this algebraic variable is 3×10^{-7} . In Figure 7, the left figure shows the x component x_1 of the computed position (x_1, y_1) of the middle joint of the robot arm together with the exact values of x_1 . The figure on the right shows the absolute error in this variable. From Figure 7, we can see that there is a good agreement between the computed and exact values of x_1 . The absolute error for this variable is 10^{-7} . Similarly, in Figure 8, the left figure shows the

x -component x_2 of the computed position (x_2, y_2) of the end-effector of the robot arm together with the exact values of x_2 . The figure on the right shows the absolute error in this variable. From Figure 8, we can see that there is a good agreement between the computed and exact values of x_2 . The absolute error for this variable is 2×10^{-7} . The numerical values of the y -coordinate of the end-effector of the robot arm are found to be $y_2 = 0$ with an error of 5×10^{-18} . Therefore, the robot arm end-effector follows the prescribed trajectory (the x -axis) with an accuracy of 7×10^{-18} . Finally, in Figure 9, the left figure shows the y -component y_1 of the computed position (x_1, y_1) of the middle joint of the robot arm together with the exact values of y_1 . The figure on the right shows the absolute error in this variable. From Figure 9, we can see that there is a good agreement between the

FIGURE 6: Approximate solution component λ (a) and absolute error e_λ (b). FIGURE 7: Approximate solution component x_1 (a) and absolute error e_{x_1} (b). FIGURE 8: Approximate solution component x_2 (a) and absolute error e_{x_2} (b). FIGURE 9: Approximate solution component y_1 (a) and absolute error e_{y_1} (b).

computed and exact values of y_1 . The absolute error for this variable is 2×10^{-7} . We have also computed the drift (error) in the position and the velocity constraints $g(u, t) = 0$ and $g'_u(u, t)v = 0$ to test the accuracy of our new algorithm. These errors are found to be 5×10^{-18} and 4×10^{-18} , respectively. These results show that DAE constraints are satisfied with a high accuracy and that the end-effector of the robot arm is following the prescribed path with high accuracy. These numerical results also show that the MSPSM has computed the solution accurately for large intervals without violating the constraints. Finally, to increase accuracy in the approximate solution, one can increase the order of approximation K or increase the number of interval subdivisions N .

5. Conclusion

The differential and algebraic equations (DAEs) with index-3 are known to be difficult to solve numerically. In this paper, we propose a novel technique to solve the index-3 DAEs (1). Our technique avoids transformation techniques like index reductions which are known to be costly and complex. Our algorithm instead applies the approximation method directly to the original DAE (1) using the power series method and the Adomian polynomials. This has led to a simple and efficient algorithm that can be easily implemented in Maple or Mathematica to solve the index-3 DAEs (1). It should be emphasized here that only small linear algebraic systems are solved in this technique. The PSM has computed the exact

solution in a power series form. In order to extend the domain of convergence of the solution, we have constructed a multistage PSM (MSPSM). To illustrate the effectiveness of our technique, a nonlinear DAE problem describing a two-link planar robot arm is solved. The numerical simulations show that the MSPSM has successfully handled this nonlinear index-3 DAE over large time intervals without violating the DAE constraints. The numerical simulations also show that the robot arm end-effector follows the prescribed trajectory with an accuracy of 5×10^{-18} . The method can provide a much greater accuracy by adding few terms into the power series or more time interval subdivisions. Future research will be to generalize this technique to more implicit higher-index DAE systems.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

- [1] A. A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, New York, NY, USA, 2013.
- [2] F. Han, Z. Qiao, and F. Chen, "Dynamic modeling, simulation, and flight test of a rocket-towed net system," *Mathematical Problems in Engineering*, vol. 2019, Article ID 1523828, 21 pages, 2019.
- [3] R. Bordephong, I. G. Derek, S. Michiel et al., "Multibody system modelling of unmanned aircraft system collisions with the human head," *International Journal of Crashworthiness*, vol. 25, no. 6, pp. 689–707, 2020.
- [4] B. Matteo, M. Giacomo, and R. Giulio, "Dynamic handling characterization and set-up optimization for a formula SAE race car via multi-body simulation," *J. Machines*, vol. 9, no. 6, 2021.
- [5] C. A. Manrique-Escobar, C. M. Pappalardo, and D. A. Guida, "Multibody system Approach for the systematic development of a closed-chain kinematic model for two-wheeled vehicles," *Machines*, vol. 9, 2021.
- [6] G. Moritz, H. David, Z. Yonas, B. Moritz, T. Benrhnard, and C. Stelian, "Analytically differentiable dynamics for multi-body systems with friction contact," *ACM Transactions on Graphics*, vol. 39, no. 6, 2020.
- [7] E. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods*, Allyn & Bacon, Massachusetts, MA, USA, 1989.
- [8] K. E. Brennan, S. L. Campbell, and L. R. Petzold, *The Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, New York, NY, USA, 1989.
- [9] J. T. Betts, S. L. Campbell, and K. C. Thompson, "Direct transcription solution of optimal control problems with differential algebraic equations with delays," in *Proceedings of the 14 th. IASTED, International Symposium on Intelligent Systems and Control (ISC 2013)*, pp. 166–173, Marina del Rey, USA, 2013.
- [10] S. Campbell and P. Kunkel, "Solving higher index DAE optimal control problems," *Numerical Algebra, Control and Optimization*, vol. 6, no. 4, pp. 447–472, 2016.
- [11] S. L. Campbell and J. T. Betts, "Comments on direct transcription solution of DAE constrained optimal control problems with two discretization approaches," *Numerical Algorithms*, vol. 73, no. 3, pp. 807–838, 2016.
- [12] V. H. Linh and N. D. Truong, "Runge-Kutta methods revisited for a class of structured strangeness-free differential-algebraic equations," *Electronic transactions on numerical analysis ETNA*, vol. 48, pp. 131–155, 2018.
- [13] D. Estévez Schwarz and R. Lamour, "Projected explicit and implicit Taylor series methods for DAEs," *Numerical Algorithms*, vol. 88, pp. 615–646, 2021.
- [14] B. I. Akinukaw, O. A. Akinfenwa, and S. A. Okunuga, "Hybrid block algorithm for solving differential-algebraic equation with Hessenberg index 3," *FUTA Journal of Research in Sciences*, vol. 15, no. 1, pp. 32–39, 2019.
- [15] F. A. Ghazwa and R. A. Zaboon, "Approximate solution of a reduced-type index-hessenberg differential-algebraic control system," *Journal of Applied Mathematics*, vol. 2021, Article ID 9706255, 13 pages, 2021.
- [16] S. Thota, "On solving system of linear differential-algebraic equations using reduction algorithm," *International Journal of Mathematics and Mathematical Sciences*, vol. 2020, Article ID 6671926, 10 pages, 2020.
- [17] S. Thota, "On Maple implementation of a reduction algorithm for differential-algebraic systems," in *Proceedings of the International Conference on Mathematical Advances and Applications 2020*, Istanbul, Turkey, November 2020.
- [18] M. Golchian, M. Gachpazan, and S. H. Tabasi, "A new approach for computing the exact solutions of DAEs in generalized Hessenberg forms," *International Journal of Nonlinear Analysis and Applications*, vol. 11, no. 1, pp. 199–206, 2020.
- [19] B. Benhammouda, "A novel technique to solve nonlinear higher-index Hessenberg differential-algebraic equations by Adomian decomposition method," *Springer Plus*, vol. 5, no. 1, pp. 1–14, 2016.
- [20] L. Yang and S. Kai, "Solving power system differential algebraic equations using the differential transformation," *IEEE transaction on power*, vol. 35, no. 3, 2020.
- [21] M. P. Carmine, L. Antonio, and G. Domenico, "Stability analysis of rigid multibody mechanical systems with holonomic and nonholonomic constraints," *Archive of Applied Mechanics*, vol. 90, pp. 1961–2005, 2020.
- [22] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computational Methods in Applied Mechanics and Engineering*, vol. 1, pp. 1–16, 1972.
- [23] M. Neto and J. Ambrósio, "Stabilization methods for the integration of DAE in the presence of redundant constraints," *Multibody System Dynamics*, vol. 10, pp. 81–105, 2003.
- [24] U. Ascher and L. Petzold, "Stability of computational methods for constrained dynamics systems," *SIAM Journal on Scientific Computing*, vol. 14, pp. 95–120, 1993.
- [25] U. Ascher and P. Lin, "Sequential regularization methods for nonlinear higher-index DAEs," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 160–181, 1997.
- [26] B. Benhammouda and H. Vazquez-Leal, "Analytical solution of a nonlinear index-three DAEs system modelling a slider-crank mechanism," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 206473, 14 pages, 2015.

- [27] B. Benhammouda, "Approximate analytical solution of index-2 DAEs arising from constrained multibody systems," *Asian Research Journal of Mathematics*, vol. 8, no. 3, pp. 1–15, 2018.
- [28] B. Benhammouda, "Solution of constrained mechanical multibody systems using Adomian decomposition method," 2006, <https://arxiv.org/abs/1605.09041>.
- [29] E. Bayo and A. Avello, "Singularity-free augmented Lagrangian algorithms for constrained multibody dynamics," *Nonlinear Dynamics*, vol. 5, pp. 209–231, 1994.
- [30] E. A. Coddington, *An Introduction to Ordinary Differential Equations*, Dover Publications, New York, NY, USA, 1989.
- [31] E. Kreyszig, *Advanced Engineering Mathematics*, Wiley & Sons, New York, NY, USA, 1999.
- [32] C. P. Filipich, M. B. Rosales, and F. Buezas, "Some nonlinear mechanical problems solved with analytical solutions," *J. Latin American Appl. Research*, vol. 34, pp. 101–109, 2004.
- [33] L. Al Sakka and U. Al Khawaja, "High accuracy power series method for solving scalar, vector, and inhomogeneous nonlinear Schrödinger equations," 2021, <https://arxiv.org/abs/2108.13174>.
- [34] M. Kurulay and M. Bayram, "A novel power series method for solving second order partial differential equations," *European Journal Of Pure And Applied Mathematics*, vol. 2, no. 2, pp. 268–277, 2009.
- [35] A. S. Nuseir and A. Al-Hasson, "Power series solution for nonlinear system of partial differential equations," *Applied Mathematical Sciences*, vol. 6, no. 104, pp. 5147–5159, 2012.
- [36] M. M. Khader and M. Adel, "Approximate solutions for the nonlinear systems of algebraic equations using the power series method," *applications and applied mathematics*, *International Journal*, vol. 15, no. 2, 2020.
- [37] H. M. Srivastava, I. Area, and J. J. Nieto, "Power-series solution of compartmental epidemiological models," *MBE*, vol. 18, no. 4, pp. 3274–3290, 2021.
- [38] R. Rach, "A new definition of the adomian polynomials," *Kybernetes*, vol. 37, pp. 910–955, 2008.
- [39] R. Rach, "A convenient computational form for the adomian polynomials," *Journal of Mathematical Analysis and Applications*, vol. 102, pp. 415–419, 1984.
- [40] A. M. Wazwaz, "A new algorithm for calculating adomian polynomials for nonlinear operators," *Applied Mathematics and Computation*, vol. 111, pp. 53–69, 2000.
- [41] J. S. Duan, "Recurrence triangle for adomian polynomials," *Applied Mathematics and Computation*, vol. 216, pp. 1235–1241, 2010.
- [42] J. S. Duan, "An efficient algorithm for the multivariable adomian polynomials," *Applied Mathematics and Computation*, vol. 217, pp. 2456–2467, 2010.
- [43] J. S. Duan, "Convenient analytic recurrence algorithms for the adomian polynomials," *Applied Mathematics and Computation*, vol. 217, pp. 6337–6348, 2011.
- [44] R. Rach and G. Adomian, "Transformation of series," *Applied Mathematics Letters*, vol. 4, no. 4, pp. 69–71, 1991.