Journal of Computational Science &Engineering 40 (2019) 1051-1055

Available at http://www.asocse.org ISSN 1710-4068

Particle Swarm Optimization for Dynamic Differential-Algebraic Equations of Multibody Systems

Xiaoxiao Zhang[1], Jieyu Ding [1,2*]

[1]School of Mathematics and Statistics, Qingdao University, Qingdao 266071, China

[2]Center for Computational Mechanics and Engineering Simulation, Qingdao University, Qingdao 266071, China

Abstract: **Aiming at solving the differential-algebraic equations of multi-body system dynamics, the solution method based on particle swarm optimization is studied. Firstly, Lagrange interpolation is performed on generalized coordinates and generalized velocity, and the Gauss numerical integration method is combined to transform the problem of solving differential algebraic equations into solving optimization problems. Then the particle swarm algorithm is used to solve the problem optimally. Finally, through the multi-body system simulation experiment of the planar two-link manipulator, it is verified that the particle swarm optimization algorithm not only maintains the constraint but also guarantees the energy precision in solving the dynamic equation. The results show that the intelligent optimization algorithm has a good application prospect in solving multi-body dynamics problems.**

Keywords: Multibody system dynamics; Differential algebraic equations; Particle swarm optimization

Received 9 March 2019, Revised 8 April 2019, Accepted 10 April 2019

*Corresponding Author: Jieyu Ding, djy@qdu.edu.cn

1. Introduction

Dynamic simulation of multi-body system has become one of the important contents of Engineering simulation, and has been widely used in mechanical, aerospace, vehicle, robotics and biomechanics fields. The dynamics of multi-body systems is usually described as differential-algebraic equations, and the study of numerical integration method is an important part of the computational dynamics of multi-body systems. The traditional methods for solving differential-algebraic equations in this field are extended from the numerical solution of ordinary differential equations. The focus of the study is to achieve the stability of differential-algebraic equations by constrained stability [1,2]. In recent years, some methods developed gradually, such as generalized-alpha method [3], symplectic algorithm [4], energy method [5,6], symplectic preserved variable fraction integral method [7], Lie group method [8], have achieved good simulation results.

Particle swarm optimization (PSO) is a group-based stochastic optimization technique proposed by Eberhart and Kennedy in 1995 [9]. Subsequently, Kennedy and Eberhart proposed a particle swarm optimization algorithm for discrete space [10]. Clerc proposed a particle swarm algorithm with contraction factor [11,12]. The idea of this method is to simulate the social behavior of flocks and fish schools. A particle swarm is randomly generated, where each particle represents a possible solution to an optimization problem, and the position of each particle is evaluated by the objective function. The particle finds the optimal solution through the process of iterative process. Compared with other optimization algorithms, particle swarm optimization algorithm has the advantages of simple algorithm, fast convergence, high optimization efficiency and strong robustness. At present, it has achieved very good results in various multidimensional continuous space optimization problems. Production scheduling, robotic intelligent control and image processing have been widely used.

In this paper, the Lagrange interpolation method is combined with the Gauss numerical integration method to transform the differential-algebraic equation problem into an optimization problem, and then the particle swarm optimization algorithm is designed for the optimization problem. Finally, the simulation experiment of the planar two-link manipulator shows that the algorithm has high precision while maintaining the constraint.

2. DAEs of Multibody system Dynamics

The DAEs of index-3 in multibody systems dynamics can be described as follows:

$$\begin{cases} M(q,t)\ddot{q} + \Phi_q^T \lambda = F(\dot{q},q,t) \\ \Phi(q,t) = 0 \end{cases} \#(1)$$

where $q \epsilon R^n$ is the generalized coordinate, $\dot{q} \epsilon R^n$ is the generalized velocity, λ is the Lagrange multiplier, Φ is the constraint equation of the generalized coordinate q,Φ_q is the Jacobian matrix of the constraint equation.

The constraint equation $\Phi(q,t) = 0$ is used to find the second-order derivative. Equation (1) can be reduced from 3 to index 1, and the equation form is as follows:

$$\begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} F \\ -(\Phi_q \dot{q})_q \dot{q} \end{bmatrix} \#(2)$$

The simulation time $[0, T]$ is equally divided into several cells $[t_i, t_{i+1}]$, $i = 0,1,\dots,N$. In the time interval $[t_i, t_{i+1}]$,use Lagrangian interpolation to approximate $q(t)$:

$$q(t) = \sum_{j=0}^{m} l_j(t) q_j \#(3)$$

$$l_j(t) = \prod_{k=0,k\neq j}^{m} \frac{(t - t_k)}{(t_j - t_k)} \#(4)$$

Then

2019@ The American Computational Science Society. All rights reserved.

$$\dot{q}(t) = \sum_{j=0}^{m} \dot{l}_j(t) q_j \#(5)$$

$$\ddot{q}(t) = \sum_{j=0}^{m} \ddot{l}_j(t) q_j \#(6)$$

Interpolation node q_i satisfies the constraint $\Phi(q,t) = 0$. The interpolation functions $q(t), \dot{q}(t), \ddot{q}(t)$ are brought into the index 1 differential-algebraic equation to obtain the equation (7):

$$f(t) = \begin{bmatrix} \ddot{q}(t) \\ \lambda \end{bmatrix} - \begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix}^{-1} \begin{bmatrix} F(\dot{q}(t), q(t), t) \\ -(\Phi_q \dot{q})_q \dot{q} \end{bmatrix} \#(7)$$
$$\forall t \in [t_i, t_{i+1}]$$

If $f(t) = 0$ for $\forall t \in [t_i, t_{i+1}]$, the interpolation function $q(t)$ is the exact solution of equation (1) in $[t_i, t_{i+1}]$. Obviously, the equation (7) is a nonlinear equation, and the analytical solution cannot be normally found. This means that when equation (7) is close to zero, $q(t)$ is the solution of equation (7) in $[t_i, t_{i+1}]$.Can be converted into the following optimization problems:

$$\min_{q_i} \int_{t_i}^{t_{i+1}} f^T f dx \#(8)$$

The Gauss numerical integration of the above formula can be transformed into the following form:

$$\min_{q_i} \sum_{s=1}^{S} A_s f^T(t_s) f(t_s) \#(9)$$

where A_s is the Gauss integral coefficient, and t_s is the Gauss integral node.

Obviously, equation (9) is a non-convex optimization problem [13]. The existing research shows that the traditional mathematical optimization method is difficult to solve such problems effectively. The intelligent optimization algorithm is an effective algorithm for solving such problems. Particle swarm optimization is an intelligent algorithm that mimics population motion and provides new theories and methods for solving engineering problems.

3. Design of particle swarm optimization algorithm

The traditional particle swarm optimization algorithm consists of a particle swarm system with M particles $X = \{X_1, X_2, X_3, \dots, X_M\}$, $X_i \in R^D$. The position and velocity of the i particle at time k : $X_i(k) = (X_{i1}(k), X_{i2}(k), \dots, X_{iD}(k))$ and $V_i(k) = (V_{i1}(k), V_{i2}(k), \dots, V_{iD}(k))$, $(i = 1,2,3, \dots, M)$. The historical optimal position of a certain particle X_i is $pbest_i$. The overall optimal position of the particle swarm system X is gbest. During the iteration, the particle position and velocity update formula is as follows:

$$V_{ij}^{t+1} = \omega V_{ij}^t + c_1 r_1 (pbest_j^t - X_{ij}^t) + c_2 r_2 (gbest_j^t - X_{ij}^t)$$
$$X_{ij}^{t+1} = X_{ij}^t + V_{ij} \#(10)$$

where ω is the inertia weight; c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are random numbers uniformly distributed between (0, 1).

The particle swarm optimization algorithm is used to solve the interpolation function $q(t)$ of the interval $[t_i, t_{i+1}]$. The initial point $q_0 = q(t_i) \in R^n$ is known, and the interpolation node $q_j, j = 1,2 \dots m$ Is the node to be optimized, and q_j satisfies the constraint $\Phi(q,t) = 0$. Define the objective function as follows:

$$\varphi(q_j) = \sum_{s=1}^{S} A_s f^T(t_s) f(t_s) \#(11)$$

If $\varphi(q_j) = 0$, then $q_j, j = 1,2 \dots m$ is the exact value. Since the objective function φ is a nonlinear function, there is generally no analytical solution, so $\varphi(q_j)$ can only be as close as possible to the exact solution, and the convergence precision φ_{min} and the maximum iteration step g_{max} are set. If the convergence precision is reached or the maximum iteration step is reached, then output the optimization result.

The solution steps of the particle swarm algorithm are as follows:

1) Set the initial parameters of the particle swarm algorithm: population size, particle dimension d, range of particle position, maximum number of iterations, and randomly generate the initial population, defining the objective function φ.

2) Calculate the fitness value of each particle according to the set objective function.

3) Compare the fitness of each particle, determine the historical optimal position of each particle and the historical optimal position of the population.

4) Update all particles according to formula (10).

5) If φ(gbest) < φ_{min} or reaches the maximum iteration step g_{max}, the optimal solution gbest is output, otherwise steps 2)-step 5) are repeated until the preset termination condition is satisfied.

4. Numerical Example

The slider-double link system is taken as an example to verify and analyze the above method. As shown in Figure 1, the length of the link are L_1 and L_2 and the masses are m_1 and m_2. The corners are denoted by θ_1 and θ_2. Suppose the system is only affected by gravity.

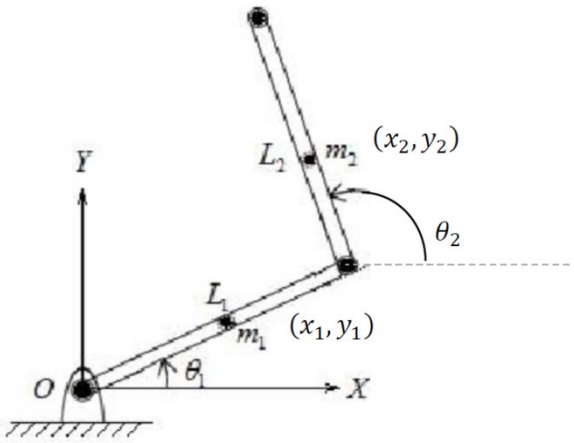

Figure 1.Three-degree-of-freedom robotic arm.

2019@ The American Computational Science Society. All rights reserved.

Taking the state variable $q=(x_1,y_1,\theta_1,x_2,y_2,\theta_2)$, the system dynamics index 3 differential-algebraic equation is:

$$\begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix}\begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix}=\begin{bmatrix} F \\ -\left(\Phi_q \dot{q}\right)_q \dot{q} \end{bmatrix} \#(11)$$

where $M=\mathrm{diag}(m_1,m_1,J_1,m_2,m_2,J_2)$ is a generalized mass matrix, $F=(0,-m_1g,0,0,-m_2g,0)$ is a generalized force vector, $J_1=\frac{1}{12}m_1L_1^2$ and $J_2=\frac{1}{12}m_2L_2^2$ are the moments of inertia of the connecting rod.

$$\Phi_q=\begin{pmatrix} 1 & 0 & \frac{L_1}{2}\sin\theta_1 & 0 & 0 & 0 \\ 0 & 1 & -\frac{L_1}{2}\cos\theta_1 & 0 & 0 & 0 \\ -1 & 0 & \frac{L_1}{2}\sin\theta_1 & 1 & 0 & \frac{L_2}{2}\sin\theta_2 \\ 0 & -1 & -\frac{L_1}{2}\cos\theta_1 & 0 & 1 & -\frac{L_2}{2}\cos\theta_2 \end{pmatrix} \#(12)$$

$$\left(\Phi_q\dot{q}\right)_q=\begin{pmatrix} 0 & 0 & \frac{L_1}{2}\cos\theta_1\,\dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & \frac{L_1}{2}\sin\theta_1\,\dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & \frac{L_1}{2}\cos\theta_1\,\dot{\theta}_1 & 0 & 0 & \frac{L_2}{2}\cos\theta_2\,\dot{\theta}_2 \\ 0 & 0 & \frac{L_1}{2}\sin\theta_1\,\dot{\theta}_1 & 0 & 0 & \frac{L_2}{2}\sin\theta_2\,\dot{\theta}_2 \end{pmatrix} \#(13)$$

Approximate $q(t)$ with three-point Lagrange interpolation:

$$q(t)=\frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)}q_0+\frac{(t-t_0)(t-t_2)}{(t_1-t_0)(t_1-t_2)}q_1+\frac{(t-t_1)(t-t_0)}{(t_2-t_1)(t_2-t_0)}q_2$$

$$\dot{q}(t)=\frac{(t-t_1)+(t-t_2)}{(t_0-t_1)(t_0-t_2)}q_0+\frac{(t-t_0)+(t-t_2)}{(t_1-t_0)(t_1-t_2)}q_1+\frac{(t-t_1)+(t-t_0)}{(t_2-t_1)(t_2-t_0)}q_2$$

$$\ddot{q}(t)=\frac{2}{(t_0-t_1)(t_0-t_2)}q_0+\frac{2}{(t_1-t_0)(t_1-t_2)}q_1+\frac{2}{(t_2-t_1)(t_2-t_0)}q_2$$

From the initial value $q(t_0)=q_0$, $\dot{q}(t_0)=\dot{q}_0$:

$$q_1=\left(\frac{h\dot{q}_0+3q_0+q_2}{4}\right),\ h=t_{i+1}-t_i \#(14)$$

The interpolation node q_i, $i=0,1,2$ satisfies the constraint $\Phi(q,t)=0$:

$$q_i=\begin{pmatrix} \frac{L_1}{2}\cos\theta_1 \\ \frac{L_1}{2}\sin\theta_1 \\ \theta_1 \\ L_1\cos\theta_1+\frac{L_2}{2}\cos\theta_2 \\ L_1\sin\theta_1+\frac{L_2}{2}\sin\theta_2 \\ \theta_2 \end{pmatrix} \#(15)$$

Two-point Gauss numerical integration method is used to solve the objective function.

$$\min_{q_2}\left(\sum_{s=1}^{2}A_s f^T(t_s)f(t_s)\right) \#(16)$$

$$\begin{cases} A_1=A_2=\dfrac{\mathrm{h}}{2} \\ t_1=\dfrac{-\sqrt{3}}{3}\cdot\dfrac{h}{2}+\dfrac{t_{i+1}+t_i}{2} \\ t_2=\dfrac{\sqrt{3}}{3}\cdot\dfrac{h}{2}+\dfrac{t_{i+1}+t_i}{2} \end{cases}$$

Take the mass of the connecting rod $m_1=1kg, m_2=2kg$, the length of the rod $L_1=1m, L_2=\sqrt{3}m$, the acceleration of gravity $g=9.81$, the step length $\mathrm{h}=0.002s$. Initial state of the connecting rod: $\theta_1=\frac{\pi}{3},\theta_2=-\frac{\pi}{6}$, generalized speed $\dot{q}=0$. Particle swarm algorithm parameter setting [14]: population size $M=120$, $c_1=c_2=2$, $\omega=0.1$, convergence precision $\varphi_{min}=10^{-15}$, maximum iteration step g_{max} =500. The simulation results are shown in the figure:

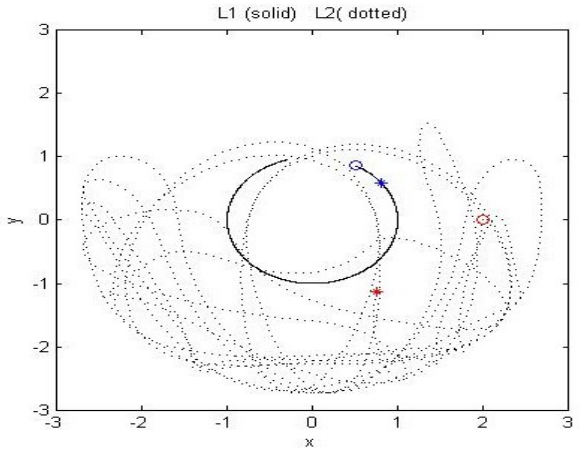

Figure 2.The trajectory of the end of connecting rod.

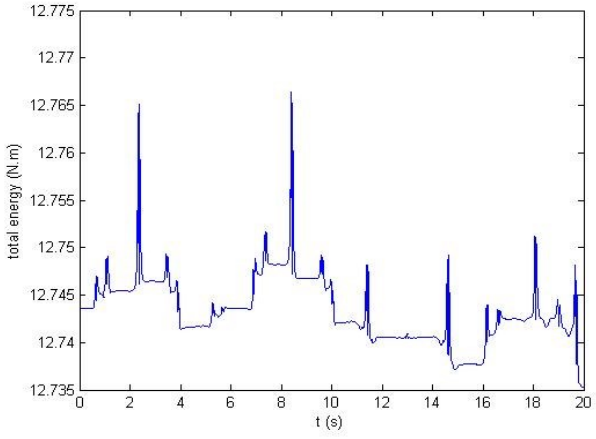

Figure 3.Total System Energy.

2019@ The American Computational Science Society. All rights reserved.

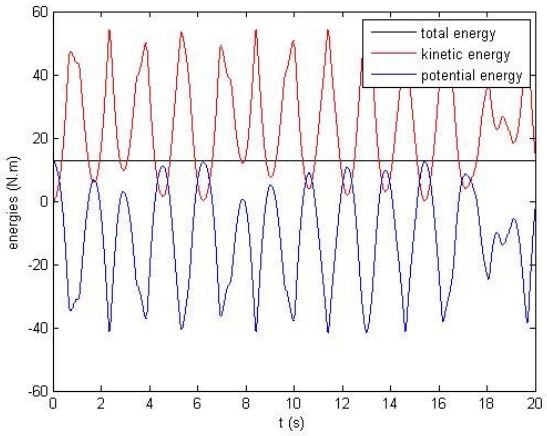

Figure 4.System Kinetic Energy and Potential Energy.

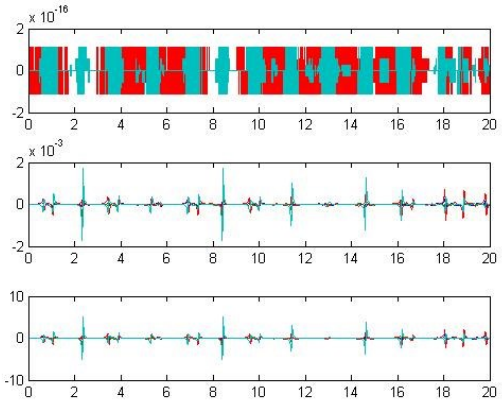

Figure 5.Planar manipulator restraint.

As shown in Figure 3 and Figure 4, the maximum error of the total energy of the system is $\max_{0<i<n}|H(t_i) - H(t_0)| = 2.2838354400690 \times 10^{-2}$. The average error of the total energy of the system is $2.727479433485 \times 10^{-3}$, Kinetic energy and potential energy change stable. It can be seen from Figure 5 that the system guarantees displacement constraints, and the speed and acceleration constraints are large.

5. Conclusion

In this paper, the particle swarm optimization algorithm is used to solve the multi-body system dynamics differential-algebraic equations, and the planar two-link manipulator system is simulated. The experimental results show that the energy error of the system is small when the constraint of index 1 is satisfied. The particle swarm algorithm has few parameters and is simple and does not need the derivative information of the objective function. It has achieved good results for solving the differential-algebraic equations of multi-body systems, and illustrates the feasibility of applying the intelligent optimization algorithm to multi-body dynamics simulation. However, this method has the disadvantages of large computational complexity and long running time of the algorithm. In the future, it is a major research direction in effectively reducing the running time and improving the accuracy of the algorithm.

Acknowledgments

The financial support was provided by the National Natural Science Foundation of China(NNSFC) through grant Nos 11772166，11472143.

References

[1] Simenon B. Computational flexible multibody dynamics-A differential algebraic approach. Berlin: Springer，2013

[2] Fon-Llagunes J M. Multibody dynamics computational methods and applications. Switzerland: Springer，2016.

[3] Bruls O, Cardona A, Arnold M. Lie group generalized-alpha time integration of constrained flexible multibody systems. Mechanism & Machine Theory, 2012, 48(1):121-137.

[4] Jay L. Symplectic Partitioned Runge-Kutta Methods for Constrained Hamiltonian Systems. SIAM Journal on Numerical Analysis, 1996, 33(1):368-387.

[5] Lens E, Alberto Cardona. An energy preserving/decaying scheme for nonlinearly constrained multibody systems. Multibody System Dynamics, 2007, 18(3):435-470.

[6] Betsch P, Uhlar S. Energy-momentum conserving integration of multibody dynamics. Multibody System Dynamics, 2007, 17(4):243-289.

[7] West J E M M. Discrete mechanics and variational integrators. Acta Numerica, 2001, 10.

[8] Iserles A, Munthekaas H Z, Nørsett S P, et al. Lie-group methods. Acta Numerica, 2000, 9(2):215--365.

[9] J.Kennedy, R.C.Eberhart. A discrete binary version of the particle swarm algorithm. IEEE International Conference on Systems.1997,5(5):4104-4108.

[10] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm. IEEE International Conference on Systems,Man,and Cybernetics. Computational Cybernetics and Simulation. IEEE,1997.

[11] M.Clerc. The swarm and the queen:towards a deterministic and adaptive particle swarm optimisation. Congress on Evolutionary Computation. 2002,3(3):1957.

[12] R C Eberhart, Y Shi. Comparing inertia weights and constriction factors in particle swarm optimization. Congress on Evolutionary Computation. 2002,1(5):84-88.

[13] Vasant P, Ganesan T, Elamvazuthi I. An

2019@ The American Computational Science Society. All rights reserved.

improved PSO approach for solving non-convex optimization problems. Ninth International Conference on Ict & Knowledge Engineering. IEEE, 2012.

[14] Xu Z, Du L, Wang H, et al. Particle swarm optimization-based algorithm of a symplectic method for robotic dynamics and control. Applied Mathematics and Mechanics(English Edition), 2019, 40(01):113-128.

2019@ The American Computational Science Society. All rights reserved.