DOI: 10.3901/JME.2020.03.121

基于量化状态系统的柔性关节机器人 动力学求解方法*

李志华 吴晨佳 江 德 樊志华 倪 敬
(杭州电子科技大学机械工程学院 杭州 310018)

摘要: 量化状态系统(Quantized state system, QSS)方法是一种新的数值积分方法, 它与传统基于时间离散的积分方法不同, QSS 方法对状态变量进行离散, 依次计算每次状态变量跃迁所需要的时间, 从而推进积分。针对柔性关节机器人动力学刚性方程的求解问题, 提出一种基于量化状态系统的步进校正优化算法(Step correction optimization algorithm, SCOA)。它结合 QSS 方法及隐式算法中梯形积分法的思想, 以有效提高动力学求解的精度和效率。通过对四自由度柔性关节机器人动力学刚性方程的仿真求解, 证明了算法的可行性。将 SCOA 算法与传统数值积分方法和 QSS 方法从仿真效率与仿真精度两方面进行性能对比, 结果表明, SCOA 算法在保证仿真效率的同时能有效提高仿真精度, 算法性能优于传统方法和 QSS 方法。

关键词: 量化状态系统; 动力学; 机器人; 数值积分; 仿真

中图分类号: TP391

The Dynamics Solving Method of Flexible Joint Robot Based on Quantized State System

LI Zhihua WU Chenjia JIANG De FAN Zhihua NI Jing
(School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018)

Abstract: Quantized state system (QSS) is a new numerical integration method, which is different from the traditional time discretization integration methods. The QSS method discretizes state variables and calculates the time required for each transition of variables. To solve the dynamics stiff equation of flexible joint robot, a step correction optimization algorithm (SCOA) based on QSS was proposed. This algorithm combines the ideas of the QSS method and the implicit trapezoidal integral method to effectively improve the dynamics solving accuracy and efficiency. The feasibility of this algorithm is proved by the simulation of four-DOF flexible joint robot. The performance of this algorithm is compared with that of the traditional integration methods and QSS method in terms of simulation efficiency and simulation accuracy. The results show that SCOA algorithm can effectively improve the simulation accuracy while ensuring the simulation efficiency, and its performance is better than the traditional methods and QSS method.

Key words: quantized state system; dynamics; robot; numerical integration; simulation

0 前言

柔性关节机器人是指其机械传动系统中的谐波减速器存在柔性关节的一种机器人, 它具有性能优越, 易携带, 易操作, 工作空间大等特点, 不但在空间领域有重要的应用价值, 而且在国防军工、医疗健康、家庭服务、柔性制造等领域拥有巨大的应用前景和发展潜力^[1]。对于多自由度柔性关节机器人, 由于其动力学模型含有各关节变量的耦合项,

以及柔性关节存在弹性变形等因素所导致的各变量运动不稳定现象, 使得其动力学模型甚为复杂^[2]。而对柔性关节机器人动力学的求解是对其进行精确控制的基础^[3]。长期以来, 柔性关节机器人动力学分析一直是一个难以很好解决的问题, 主要表现在数学建模复杂、求解方法繁琐、运算量较大等方面^[3-5]。

数值积分求解是柔性关节机器人动力学研究的一个重要内容^[6]。由于柔性关节机器人的高度非线性、强刚性以及强耦合的特征, 它存在着数值求解的困难, 并且在仿真过程中容易积累误差, 使得仿真轨迹不能完全表示柔性关节机器人系统的真实运动^[5]。其数学模型一般为一组庞大复杂的二阶常微分方程, 目前, 对这类动力学方程的求解方法包括直

* 浙江省自然科学基金(LY18E050008, LY19E050013)和国家重点研发计划专项(2017YFB1301300)资助项目。20190328 收到初稿; 20190909 收到修改稿

接数值求解方法和降阶数值求解方法。其中,直接数值求解方法主要有 Wilson-θ 法和 Newmark 法等^[7-9],这两类方法属于基于时间离散的隐式积分方法,求解需要调用迭代算法,过程繁琐,计算成本随着系统规模的增大而显著增加,仿真效率也随之降低,而且此类方法在仿真精度的提高上存在局限性^[9]。降阶数值求解方法即一阶常微分方程的转化求解,通常采用基于时间离散积分的 Euler、Runge-Kutta 求解^[10]。由于柔性关节机器人动力学方程呈现出的严重非线性和刚性,因此尽管可以采用目前已有的许多求解刚性微分方程的方法^[11-14],但因它们都是基于时间离散的积分方法,求解过程需要显著减少积分步长以确保数值稳定性,而无限限制的缩短步长容易导致误差的积累从而降低仿真的精度^[15]。

量化状态系统(Quantized state system, QSS)方法是一种新的数值积分方法,它与传统基于时间离散的积分方法显著不同,QSS 方法对状态变量进行离散,即系统的状态变量以“量子”为单位跃迁,依次计算每次状态变量跃迁所需要的时间,从而推进积分。这一思想最早由 ZEIGLER^[16]提出,并由 KOFMAN^[17]首次实现。在求解一般动力学方程时,QSS 方法不仅具有强稳定性、高精确度等优点,而且计算过程完全不需要进行迭代,因此仿真效率大大提升^[18]。国外学者后来又陆续提出了 QSS2 和 QSS3,以及 LIQSS 系列算法,这些算法被运用于各个工程领域,取得了不错的效果^[19-22]。

目前,国内学者研究较少,只有北京航空航天大学朱雨童等^[23-24]和清华大学电机系的杨祎、檀添、李帛洋、秦建等^[18,25-27]将 QSS 方法应用在带有间断和刚性的电力电子系统的仿真求解中。

本文以四自由度柔性关节机器人为对象,在借鉴 QSS 方法及隐式梯形积分法的基础上,提出一种基于量化状态系统的步进校正优化算法(Step correction optimization algorithm, SCOA based-on QSS),有效扩展 QSS 方法在刚性方程系统中的应用范围。该算法在 Matlab 平台上予以实现,并与传统数值积分方法和 QSS 方法进行了性能对比。

1 量化状态系统(QSS)

QSS 方法是以状态变量的量化取代传统数值积分的时间离散化。

考虑一组由常微分方程(ODEs)表示的状态方程系统(State equation system, SES)

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

式中, $\mathbf{x}(t) \in \mathbf{R}^n$ 为状态向量; $\mathbf{u}(t)$ 为输入向量, QSS

方法将式(1)近似为

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) \quad (2)$$

式中, $\mathbf{q}(t)$ 为状态变量的量化向量,每一个量化变量 $q_i(t)$ 通过式(3)的迟滞量化函数得到

$$q_i(t) = \begin{cases} x_i(t) & |x_i(t) - q_i(t^-)| \geq \Delta Q_i \\ q_i(t^-) & \text{其他} \end{cases} \quad (3)$$

式中, ΔQ_i 为量子,量子是事先给定的,与传统数值积分算法中的时间步长作用类似,量子直接决定了算法误差的大小。从式(3)中可以很容易看出 $q_i(t)$ 遵循分段常数的轨迹,只有当 $x_i(t)$ 的变化超过某个值(即量子 ΔQ_i)才会引起 $q_i(t)$ 的变化,称为发生了一次跃迁,且 $q_i(t)$ 与 $x_i(t)$ 的差值不会超过一个量子,这也保证了算法的稳定性与全局误差界限^[28]。

由于 $q_i(t)$ 轨迹的特殊性,式(2)的数值解很明确,因此 QSS 算法可以很容易得到仿真。

设 $i=1, \dots, n$, $j=1, \dots, n$, T 为设置的仿真时间, Δt_{x_i} 为单个变量每一次跃迁所需的时间, Δt 为系统每推进一次仿真所需的时间;此外,变量只有在发生一次跃迁时,即 $|q_i - x_i| = \Delta Q_i$ 时,才会更新其量化变量 q_i ,此时 $q_i = x_i$; q_i 是否更新决定了变量导数 $\dot{x}_i$ 是否更新;未发生跃迁的状态变量的变化值记为 Δx_i ;系统仿真推进的时间点记为 t ,状态变量在该时间点的值记为 x_i 。

QSS 算法的流程图如图 1 所示。

图 1 QSS 算法流程图

QSS 算法具有很好的全局误差控制性能,这与

量子 ΔQ_j 的取值密不可分, 而且 QSS 算法是实时仿真的优异选择, 因为算法的异步性, 所以算法可以很容易地在并行计算架构上实现^[28]。

2 基于量化状态系统的步进校正优化算法(SCOA based-on QSS)

2.1 SCAO 算法

本文提出的基于量化状态系统的步进校正优化算法(SCOA based-on QSS), 其基本思路为: QSS 方法作为显式算法, 不需要进行迭代求解, 能保证仿真的高效性; 而梯形法作为隐式算法, 能较好地保证仿真的稳定性; 若将两者结合, 则不仅可以较好地保证仿真的稳定性, 而且还保证了算法的高效性。此外, 本算法若不再将量化变量值作为系统状态变量的近似值, 而是作为预测值, 则可以有效地提升仿真精度。故 SCAO 算法如下:

假设系统的量化状态方程如式(2)。

令系统状态变量为 x_j , 则其对应的量化状态变量为 q_j , $j=1, \dots, n$; 当在时间点 t 时, 量化变量 q_j 的预测值有两个, 分别为: q_j^+ (上限值)、 q_j^- (下限值); 它们的计算公式分别为: $q_j^+ = q_j + \Delta Q_j$ 、 $q_j^- = q_j - \Delta Q_j$, ΔQ_j 为量子; 那么系统量化状态变量预测值的导数可分别记为: $\dot{x}_j^+$ 、 $\dot{x}_j^-$; 仿真推进过程中 x_j 的取值一直趋向 q_j 取值的方向, 而 q_j 的取值为

$$q_j = \begin{cases} q_j^+ & \dot{x}_j^+ > 0 \text{ 且 } \dot{x}_j^- > 0 \\ q_j^- & \dot{x}_j^+ < 0 \text{ 且 } \dot{x}_j^- < 0 \\ \hat{q}_j & \dot{x}_j^+ \cdot \dot{x}_j^- < 0 \end{cases} \quad (4)$$

(1) 仿真推进过程中, 如若系统量化状态变量预测值的导数符号一致时, 即 $\dot{x}_j^+ \cdot \dot{x}_j^- > 0$, 则可以确定系统变量的运行轨迹, 此时根据公式(4)可以求得 q_j , 然后根据公式(2)求得 $\dot{x}_j$ 的值。

这样, 状态变量每一次跃迁的时间为

$$\Delta t_j = \frac{\Delta Q_j}{|\dot{x}_j|} \quad (5)$$

此时, 状态变量的仿真推进时间及变量值计算公式为

$$t_j^{(k)} = t_j^{(k-1)} + \Delta t_j \quad (6)$$

$$x_j^{(k)} = x_j^{(k-1)} + \frac{\Delta t_j}{2} \cdot (\dot{x}_j^{(k-1)} + \dot{x}_j^{(k)}) \quad (7)$$

式中 k ——系统仿真进程中执行的步数。

(2) 当系统量化状态变量预测值的导数符号不同时, 即 $\dot{x}_j^+ \cdot \dot{x}_j^- < 0$, 则说明仿真进程中变量的轨迹发生了变化, 那么在这段变量轨迹中必然存在系统

量化状态变量预测值的导数为零, 即 $\dot{x}_j = 0$; 此时

$$\hat{q}_j = -\frac{u_{jj}}{A_{jj}} \quad (8)$$

这里要说明的是 A_{jj} 与 u_{jj} 的取值: A_{jj} 为系统雅可比矩阵的对角线元素, $A_{jj} = \frac{\partial f}{\partial x}$; $u_{jj} = \dot{x}_j - A_{jj} \cdot q_j$;

根据公式(4)计算系统量化状态变量 q_j 的值, 即 $q_j = \hat{q}_j$; 然后根据公式(2)重置系统量化状态变量的导数为 $\hat{x}_j$, 此时, 变量发生跃迁的时间为

$$\Delta t_j = \frac{\Delta Q_j}{|\hat{x}_j|} \quad (9)$$

而变量的值为

$$x_j^{(k)} = \frac{1}{2} \cdot (x_j^{(k-1)} + \hat{q}_j) \quad (10)$$

无论系统量化状态变量预测值的导数符号正负, 整个系统每推进一次仿真的时间为

$$\Delta t = \min(\Delta t_j) \quad (11)$$

2.2 收敛性与稳定性分析

2.2.1 收敛性分析

假定 $\Delta x(t) = q(t) - x(t)$, $\Delta x(t)$ 满足 $\|\Delta x(t)\| \leq \Delta_x$, Δ_x 为预定义的常数, 则公式(2)可以写成

$$\dot{x}(t) = f(x(t) + \Delta x(t), u(t)) \quad (12)$$

令 $\phi(t)$, $\phi_1(t)$ 分别为公式(1)与(12)在初始条件 $x(0) = x_0$ 的解, 且有 $\phi(0) = \phi_1(0)$, 则有

$$\phi_1(t) - \phi(t) = \int_0^t [f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))] d\tau \quad (13)$$

根据欧几里得范数, 公式(13)有

$$\begin{aligned} \|\phi_1(t) - \phi(t)\| &= \left\| \int_0^t [f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))] d\tau \right\| \\ &\Rightarrow \|\phi_1(t) - \phi(t)\| \leq \int_0^t \|f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))\| d\tau \end{aligned} \quad (14)$$

假定 M 为函数 f 的 Lipschitz 常数, 则上式有

$$\begin{aligned} \|\phi_1(t) - \phi(t)\| &\leq \int_0^t \|f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))\| d\tau \\ &\Rightarrow \|\phi_1(t) - \phi(t)\| \leq \int_0^t M \|\phi_1(\tau) + \Delta x(\tau) - \phi(\tau)\| d\tau \\ &\Rightarrow \|\phi_1(t) - \phi(t)\| \leq \int_0^t M \|\phi_1(\tau) - \phi(\tau)\| d\tau + Mt\Delta_x \end{aligned} \quad (15)$$

因为 $\phi(t)$, $\phi_1(t)$ 都是连续的, 且 M 为正数, 则根据 Gronwall-Bellman 不等式可以将上式改为

$$\begin{aligned} \|\phi(t) - \phi(t)\| &\leq M t \Delta_x + \int_0^t M^2 s \Delta_x e^{\int_s^{Mt} ds} ds \\ &\Rightarrow \|\phi(t) - \phi(t)\| \leq (e^{Mt} - 1) \Delta_x \end{aligned} \quad (16)$$

因为 M 与 t 不依赖于 Δ_x , 则有

$$\lim_{\Delta_x \rightarrow 0} \|\phi(t) - \phi(t)\| = 0 \quad (17)$$

因此, 从式(17)可以看出, SCOA 算法具有收敛性。

2.2.2 稳定性分析

假定方程(1)是线性时不变系统, 则(1)可以改写成

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (18)$$

则运用 SCOA 算法可以写成

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t) \quad (19)$$

式中, $\mathbf{A}$ 为可对角化的 Hurwitz 矩阵, 则 SCOA 的误差为

$$|\mathbf{e}(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}(\mathbf{A})^{-1} \cdot \mathbf{A}| \cdot |\mathbf{V}^{-1}| \cdot \Delta Q \quad (20)$$

式中 $\mathbf{A}$ —— 矩阵 $\mathbf{A}$ 的特征值的对角矩阵;

$\mathbf{V}$ —— 矩阵 $\mathbf{A}$ 的特征向量矩阵;

Re —— 取复数的实数部分。

从式(20)可以看出, 全局误差总是有界的, 因此 SCOA 算法具有稳定性。

2.3 SCOA 算法的实现

算法实现(以伪代码的形式)如下:

while($t < T$)//设置仿真时间 T

for 每一个 $x_j \in [1, n]$

until ($|x_j - q_j| = \Delta Q_j$)

if ($\dot{x}_j^+ \cdot \dot{x}_j^- > 0$) then

$$q_j = x_j \pm \Delta Q_j$$

$$\Delta t_j = \frac{\Delta Q_j}{|\dot{x}_j|}$$

$$x_j^{(k)} = x_j^{(k-1)} + \frac{\Delta t_j}{2} \cdot (\dot{x}_j^{(k-1)} + \dot{x}_j^{(k)})$$

//(这步借鉴梯形积分法思想)

$$\text{else } \hat{q}_j = -\frac{u_{jj}}{A_{jj}}$$

$$\Delta t_j = \frac{\Delta Q_j}{|\hat{x}_j|}$$

$$x_j^{(k)} = \frac{1}{2} \cdot (x_j^{(k-1)} + \hat{q}_j)$$

//(对变量的取值进行校正优化)

end if

$$A_{jj} = \frac{(\dot{x}_j^+ - \dot{x}_j^-)}{(q_j - q_j^-)}$$

end for

$\Delta t = \min(\Delta t_j)$ //(取状态变量最小跃迁时间)

$t_j^{(k)} = t_j^{(k-1)} + \Delta t$ //(仿真推进时间)

end while

3 四自由度柔性关节机器人动力学模型

本文以四自由度柔性关节机器人为研究对象, 它主要由腰部回转关节、肩部关节、肘部关节、腕关节和三节刚性连杆构成, 其中, 腰部回转关节记为关节 1, 肩部关节记为关节 2, 肘部关节记为关节 3, 腕关节记为关节 4。其结构如图 2 所示。

图2 四自由度柔性关节机器人结构简图

相关符号含义如下:

N 为关节减速比, 这里, 四个关节具有相同的减速比; J_{mi} 为第 i 个电机转子的转动惯量, 单位为 kgm^2 ; m_{ri} 为第 i 个电机转子的质量, 单位为 kg ; θ_i, θ_{mi} 为第 i 个电机减速前后的转角, 单位为 rad , $\theta_{mi} = N\theta_i$; I_{li} 为第 i 个刚性连杆的转动惯量, 单位: kgm^2 ; m_{li} 为第 i 个刚性连杆的质量, 单位为 kg ; q_i 为第 i 个刚性连杆的转动角度, 即关节转角, 单位为 rad , 以上 $i=1,2,3,4$; d_i 为第 i 个刚性连杆质心到关节质心间的距离, 单位为 m , $i=2,3,4$; l_i 为第 i 个刚性连杆的长度, 单位为 m , $i=2,3,4$ 。

这里要进行假设说明的是, 该机器人模型的柔性关节不考虑电机动力学的影响, 只将电机看作理想的力矩源。

定义两组广义坐标: 关节转角 $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$, 电机经减速器后的输出转角 $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ 。

采用拉格朗日法对四自由度柔性关节机器人进行动力学建模, 得到其动力学方程如下(由于篇幅所限, 具体推导过程和内容可参考文献[29])

$$\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) \\ \mathbf{J}\ddot{\boldsymbol{\theta}} + \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) = \boldsymbol{\tau} \end{cases} \quad (21)$$

式中, $\mathbf{M}(\mathbf{q})$ 表示广义质量矩阵; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ 包含哥氏

力和向心力的矩阵; $\mathbf{g}(\mathbf{q})$ 表示重力项矩阵; $\mathbf{K}$ 表示刚度矩阵; $\mathbf{J}$ 表示转子经减速器后的转动惯量矩阵; $\boldsymbol{\tau}$ 表示电机力矩矩阵。

为了方便 SCOA 算法求解, 将式(21)化为式(1)所示的常微分方程形式, 并整理

$$\begin{aligned}
 \dot{x}_1 &= (k(x_{13} - x_9) - C_{11}x_1 - C_{12}x_2 - C_{13}x_3 - C_{14}x_4) / M_{11} \\
 \dot{x}_2 &= ((kM_{33}(x_{14} - x_{10}) - kM_{23}(x_{15} - x_{11})) \times \\
 &\quad (M_{43}^2 - M_{44}M_{33}) - (kM_{43}(x_{15} - x_{11}) - \\
 &\quad kM_{44}(x_{16} - x_{12}))(M_{24}M_{33} - M_{34}M_{23}) - \\
 &\quad ((C_{21}M_{33} - C_{31}M_{23})x_1 + (C_{22}M_{33} - C_{32}M_{23})x_2 + \\
 &\quad (C_{23}M_{33} - C_{33}M_{23})x_3 + (C_{24}M_{33} - C_{34}M_{23})x_4 + \\
 &\quad g_2M_{33} - g_3M_{23})(M_{43}^2 - M_{44}M_{33}) + \\
 &\quad (M_{24}M_{33} - M_{34}M_{23})((C_{31}M_{43} - C_{41}M_{44})x_1 + \\
 &\quad (C_{32}M_{43} - C_{42}M_{44})x_2 + (C_{33} - M_{43} - C_{43}M_{44})x_3 + \\
 &\quad (C_{34}M_{43} - C_{44}M_{44})x_4 + g_3M_{43} - g_4M_{44})) \div \\
 &\quad ((M_{43}^2 - M_{44}M_{33})(M_{22}M_{33} - M_{23}^2) - \\
 &\quad (M_{23}M_{43} - M_{42}M_{33})(M_{24}M_{33} - M_{34}M_{23})) \\
 \dot{x}_3 &= ((kM_{32}(x_{14} - x_{10}) - kM_{22}(x_{15} - x_{11})) \times \\
 &\quad (M_{34}M_{42} - M_{44}M_{32}) - (kM_{42}(x_{15} - x_{11}) - \\
 &\quad kM_{32}(x_{16} - x_{12}))(M_{24}M_{32} - M_{34}M_{22}) - \\
 &\quad (M_{32}(C_{21}x_1 + C_{22}x_2 + C_{23}x_3 + C_{24}x_4 + g_2) - \\
 &\quad M_{22}(C_{31}x_1 + C_{32}x_2 + C_{33}x_3 + C_{34}x_4 + g_3)) \times \\
 &\quad (M_{34}M_{42} - M_{44}M_{32}) + (M_{42}(C_{31}x_1 + C_{32}x_2 + C_{33}x_3 + \\
 &\quad C_{34}x_4 + g_3) - M_{32}(C_{41}x_1 + C_{42}x_2 + C_{43}x_3 + \\
 &\quad C_{44}x_4 + g_4)) \times (M_{24}M_{32} - M_{34}M_{22})) \div \\
 &\quad ((M_{34}M_{42} - M_{44}M_{32})) \times (M_{23}^2 - M_{33}M_{22}) - \\
 &\quad (M_{33}M_{42} - M_{43}M_{32}) \times (M_{24}M_{32} - M_{34}M_{22})) \\
 \dot{x}_4 &= ((kM_{33}(x_{14} - x_{10}) - kM_{23}(x_{15} - x_{11})) \times \\
 &\quad (M_{23}M_{43} - M_{42}M_{33}) - (M_{22}M_{33} - M_{23}^2) \times \\
 &\quad (kM_{43}(x_{15} - x_{11}) - kM_{44}(x_{16} - x_{12})) - \\
 &\quad ((C_{21}M_{33} - C_{31}M_{23})x_1 + (C_{22}M_{33} - C_{32}M_{23})x_2 + \\
 &\quad (C_{23}M_{33} - C_{33}M_{23})x_3 + (C_{24}M_{33} - C_{34}M_{23})x_4 + \\
 &\quad g_2M_{33} - g_3M_{23})(M_{23}M_{43} - M_{42}M_{33}) + \\
 &\quad (M_{22}M_{33} - M_{23}^2)((C_{31}M_{43} - C_{41}M_{44})x_1 + \\
 &\quad (C_{32}M_{43} - C_{42}M_{44})x_2 + (C_{33}M_{43} - C_{43}M_{44})x_3 + \\
 &\quad (C_{34}M_{43} - C_{44}M_{44})x_4 + g_3M_{43} - g_4M_{44})) \div \\
 &\quad ((M_{24}M_{33} - M_{34}M_{23})(M_{23}M_{43} - M_{42}M_{33}) - \\
 &\quad (M_{22}M_{33} - M_{23}^2)(M_{43}^2 - M_{44}M_{33})) \\
 \dot{x}_5 &= (\tau_1 - k(x_{13} - x_9)) / (N^2 \cdot J_{m1}) \\
 \dot{x}_6 &= (\tau_2 - k(x_{14} - x_{10})) / (N^2 \cdot J_{m2}) \\
 \dot{x}_7 &= (\tau_3 - k(x_{15} - x_{11})) / (N^2 \cdot J_{m3}) \\
 \dot{x}_8 &= (\tau_4 - k(x_{16} - x_{12})) / (N^2 \cdot J_{m4}) \\
 \dot{x}_9 &= x_1 \\
 \dot{x}_{10} &= x_2 \\
 \dot{x}_{11} &= x_3 \\
 \dot{x}_{12} &= x_4 \\
 \dot{x}_{13} &= x_5 \\
 \dot{x}_{14} &= x_6 \\
 \dot{x}_{15} &= x_7 \\
 \dot{x}_{16} &= x_8
 \end{aligned} \tag{22}$$

式中, $[x_9, x_{10}, x_{11}, x_{12}]^T$ 为广义坐标 $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$; $[x_{13}, x_{14}, x_{15}, x_{16}]^T$ 为广义坐标 $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T$; $[x_1, x_2, x_3, x_4]^T$ 为 $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4]^T$; $[x_5, x_6, x_7, x_8]^T$ 为 $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$ 。其余为

$$\begin{aligned}
 M_{11}(\mathbf{q}) &= I_{l1} + (m_{l2}d_2^2 + (m_{r3} + m_{l3} + m_{r4} + m_{l4})l_2^2) \times \\
 &\quad (\sin(x_{10}))^2 + (m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2) \times \\
 &\quad (\sin(x_{10} + x_{11}))^2 + m_{l4}d_4^2(\sin(x_{10} + x_{11} + x_{12}))^2 + \\
 &\quad 2(m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3) \times \\
 &\quad (\sin(x_{10}))(\sin(x_{10} + x_{11})) + \\
 &\quad 2m_{l4}l_2d_4(\sin(x_{10}))(\sin(x_{10} + x_{11} + x_{12})) + \\
 &\quad 2(m_{l4}l_3d_4)(\sin(x_{10} + x_{11}))(\sin(x_{10} + x_{11} + x_{12})) \\
 M_{12}(\mathbf{q}) &= M_{13}(\mathbf{q}) = M_{14}(\mathbf{q}) = \\
 M_{21}(\mathbf{q}) &= M_{31}(\mathbf{q}) = M_{41}(\mathbf{q}) = 0 \\
 M_{22}(\mathbf{q}) &= (m_{l2}d_2^2 + (m_{r3} + m_{l3} + m_{r4} + m_{l4})l_2^2) + \\
 &\quad (m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2) + m_{l4}d_4^2 + (I_{l2} + J_{m3}) + \\
 &\quad (I_{l3} + J_{m4}) + I_{l4} + 2(m_{l3}l_2d_3 + \\
 &\quad (m_{r4} + m_{l4})l_2l_3)(\cos(x_{10} + x_{11})) + \\
 &\quad 2(m_{l4}l_2d_4)(\cos(x_{11} + x_{12})) + \\
 &\quad 2(m_{l4}l_3d_4)(\cos(x_{12})) \\
 M_{23}(\mathbf{q}) &= (I_{l3} + J_{m4}) + I_{l4} + (m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2) + \\
 &\quad m_{l4}d_4^2 + (m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\cos(x_{11})) + \\
 &\quad (m_{l4}l_2d_4)(\cos(x_{11} + x_{12})) + 2(m_{l4}l_3d_4)(\cos(x_{12})) \\
 M_{24}(\mathbf{q}) &= I_{l4} + m_{l4}d_4^2 + \\
 &\quad m_{l4}l_2d_4(\cos(x_{11} + x_{12})) + m_{l4}l_3d_4(\cos(x_{12})) \\
 M_{32}(\mathbf{q}) &= M_{23}(\mathbf{q}) \\
 M_{33}(\mathbf{q}) &= I_{l3} + J_{m4} + I_{l4} + (m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2) + \\
 &\quad m_{l4}d_4^2 + 2m_{l4}l_3d_4(\cos(x_{12})) \\
 M_{34}(\mathbf{q}) &= I_{l4} + m_{l4}d_4^2 + m_{l4}l_3d_4(\cos(x_{12})) \\
 M_{42}(\mathbf{q}) &= M_{24}(\mathbf{q}) \\
 M_{43}(\mathbf{q}) &= I_{l4} + m_{l4}d_4^2 + m_{l4}l_3d_4(\cos(x_{12})) \\
 M_{44}(\mathbf{q}) &= I_{l4} + m_{l4}d_4^2 \\
 C_{11} &= ((m_{l2}d_2^2 + (m_{r3} + m_{l3} + m_{r4} + m_{l4})l_2^2) \times \\
 &\quad (\sin(x_{10}))(\cos(x_{10})) + (m_{l3}d_3^2 + \\
 &\quad (m_{r4} + m_{l4})l_3^2)(\sin(x_{10} + x_{11})) \times \\
 &\quad (\cos(x_{10} + x_{11})) + (m_{l4}d_4^2) \times \\
 &\quad (\sin(x_{10} + x_{11} + x_{12}))(\cos(x_{10} + x_{11} + x_{12})) + \\
 &\quad (m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\sin(x_{10} + x_{11} + x_{12})) + \\
 &\quad (m_{l4}l_2d_4)(\sin(x_{10} + x_{11} + x_{12})) + \\
 &\quad (m_{l4}l_3d_4)(\sin(2x_{10} + 2x_{11} + x_{12})))x_2 + \\
 &\quad ((m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2)(\sin(x_{10} + x_{11})) \times \\
 &\quad (\cos(x_{10} + x_{11})) + (m_{l4}d_4^2)(\sin(x_{10} + x_{11} + x_{12})) \times \\
 &\quad (\cos(x_{10} + x_{11} + x_{12}))) + (m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3) \times
 \end{aligned}$$

$$\begin{aligned}
& (\sin(x_{10}))(\cos(x_{10} + x_{11})) + (m_{l4}l_2d_4)(\sin(x_{10})) \times \\
& (\cos(x_{10} + x_{11} + x_{12})) + (m_{l4}l_3d_4) \times \\
& (\sin(2x_{10} + 2x_{11} + x_{12})))x_3 + \\
& ((m_{l4}d_4^2)(\sin(x_{10} + x_{11} + x_{12}))(\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l4}l_2d_4)(\sin(x_{10}))(\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l4}l_3d_4)(\sin(x_{10} + x_{11}))(\cos(x_{10} + x_{11} + x_{12})))x_4
\end{aligned}$$

$$\begin{aligned}
C_{12} = & ((m_{l2}d_2^2 + (m_{r3} + m_{l3} + m_{r4} + m_{l4})l_2^2) \times \\
& (\sin(x_{10}))(\cos(x_{10})) + (m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2) \times \\
& (\sin(x_{10} + x_{11}))(\cos(x_{10} + x_{11})) + \\
& (m_{l4}d_4^2)(\sin(x_{10} + x_{11} + x_{12}))(\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\sin(2x_{10} + x_{11})) + \\
& (m_{l4}l_2d_4)(\sin(2x_{10} + x_{11} + x_{12})) + \\
& (m_{l4}l_3d_4)(\sin(2x_{10} + 2x_{11} + x_{12})))x_1
\end{aligned}$$

$$\begin{aligned}
C_{13} = & ((m_{l3}d_3^2 + (m_{r4} + m_{l4})l_3^2)(\sin(x_{10} + x_{11})) \times \\
& (\cos(x_{10} + x_{11})) + (m_{l4}d_4^2)(\sin(x_{10} + x_{11} + x_{12})) \times \\
& (\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\sin(x_{10})) \times \\
& (\cos(x_{10} + x_{11})) + \\
& (m_{l4}l_2d_4)(\sin(x_{10}))(\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l4}l_3d_4)(\sin(x_{10} + x_{11} + x_{12})))x_1
\end{aligned}$$

$$\begin{aligned}
C_{14} = & ((m_{l4}d_4^2)(\sin(x_{10} + x_{11} + x_{12}))(\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l4}l_2d_4)(\sin(x_{10}))(\cos(x_{10} + x_{11} + x_{12})) + \\
& (m_{l4}l_3d_4)(\sin(x_{10} + x_{11}))(\cos(x_{10} + x_{11} + x_{12})))x_1
\end{aligned}$$

$$C_{21} = -C_{12}$$

$$\begin{aligned}
C_{22} = & -((m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\sin(x_{11}))x_3 + \\
& (m_{l4}l_2d_4)(\sin(x_{11} + x_{12}))(x_3 + x_4) + \\
& (m_{l4}l_3d_4)(\sin(x_{12})))x_4
\end{aligned}$$

$$\begin{aligned}
C_{23} = & -((m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\sin(x_{11}))(x_2 + x_3) + \\
& (m_{l4}l_2d_4)(\sin(x_{11} + x_{12}))(x_2 + x_3 + x_4) + \\
& 2(m_{l4}l_3d_4)(\sin(x_{12})))x_4
\end{aligned}$$

$$\begin{aligned}
C_{24} = & -(m_{l4}l_2d_4(\sin(x_{11} + x_{12}))(x_2 + x_3 + x_4) + \\
& m_{l4}l_3d_4(\sin(x_{12}))(x_2 + x_4))
\end{aligned}$$

$$C_{31} = -C_{13}$$

$$\begin{aligned}
C_{32} = & ((m_{l3}l_2d_3 + (m_{r4} + m_{l4})l_2l_3)(\sin(x_{11})) + \\
& (m_{l4}l_2d_4)(\sin(x_{11} + x_{12})))x_2
\end{aligned}$$

$$C_{33} = -m_{l4}l_3d_4(\sin(x_{12}))x_4$$

$$C_{34} = -m_{l4}l_3d_4(\sin(x_{12}))(2x_2 + x_3 + x_4)$$

$$C_{41} = -C_{14}$$

$$\begin{aligned}
C_{42} = & (m_{l4}l_2d_4(\sin(x_{10} + x_{11} + x_{12})) + \\
& m_{l4}l_3d_4(\sin(x_{12})))x_2
\end{aligned}$$

$$C_{43} = m_{l4}l_3d_4(\sin(x_{12}))(2x_2 + x_3)$$

$$C_{44} = 0$$

$$g_1 = 0$$

$$\begin{aligned}
g_2 = & -g((m_{l2}d_2 + (m_{r3} + m_{l3} + m_{r4} + m_{l4})l_2)(\sin(x_{10})) + \\
& (m_{l3}d_3 + (m_{r4} + m_{l4})l_3)(\sin(x_{10} + x_{11})) + \\
& (m_{l4}d_4)(\sin(x_{10} + x_{11} + x_{12})))
\end{aligned}$$

$$\begin{aligned}
g_3 = & -g((m_{l3}d_3 + (m_{r4} + m_{l4})l_3)(\sin(x_{10} + x_{11})) + \\
& (m_{l4}d_4)(\sin(x_{10} + x_{11} + x_{12})))
\end{aligned}$$

$$g_4 = -g((m_{l4}d_4)(\sin(x_{10} + x_{11} + x_{12})))$$

4 仿真结果

4.1 仿真条件说明

(1) SCOA 算法、传统数值积分方法(包括 Euler 法、Runge-Kutta 系列法 ODE45、ODE23s、ODE15s) 和 QSS 算法全部在 Matlab 平台上实现;

(2) 计算机硬件配置为 Intel i5-5257 处理器, CPU 运行速度为 2.70 GHz, 操作系统为 Windows8.1;

(3) 对所有的算法设置相同的误差容限, 即 Rel.Tol=Abs.Tol=10⁻⁵;

(4) 状态变量的相对误差计算公式为

$$e_{rr} = \sqrt{\frac{\sum (u_m[k] - u_{dassl}[k])^2}{\sum u_{dassl}[k]^2}} \quad (23)$$

式中, $u_m[k]$ 为各个算法求的状态变量值; $u_{dassl}[k]$ 为 DASSL 求解器在很小误差设定 10⁻⁹ 下求的状态变量值, 在此作为基准值;

(5) 四自由度柔性关节机器人动力学方程中各参量值如表 1 所示, 状态变量初值设为 $[x_1, x_2, \dots, x_{16}]^T = [0, 0, 0, 0, 0, 0, 0, 0, -\frac{\pi}{3}, \frac{\pi}{9}, \frac{\pi}{2}, 0, -\frac{\pi}{3}, \frac{\pi}{9}, \frac{\pi}{2}]^T$

4.2 仿真结果对比分析

采用 SCOA 算法对四自由度柔性关节机器人动力学模型进行仿真求解, 得到图 3 所示的关节转角轨迹图和图 4 所示的关节角速度轨迹图。从图 3 中可以直观看到柔性关节机器人四个关节转角的运动轨迹, 它们从初始角变化到终止角, 然后稳定在终止角, 运行轨迹都很平稳, 并未因各个变量之间存在的强耦合而出现振荡。从图 4 中可以看到四个关节角速度的变化过程, 加速—匀速—减速; 从局部放大图可以看出, 在达到某一仿真时间点后, 柔性关节机器人的关节角速度只在一定范围内小幅波动, 并最终趋于稳定。由此可见, SCOA 算法具有较好的稳定性和收敛性, 能够胜任复杂机器人动力学刚性方程的实际求解。

表 1 四自由度柔性关节机器人参量值

参量	数值
关节减速比 N	160
电机 1、2 转动惯量 $J_{m1}, J_{m2}/(\text{kg}\cdot\text{m}^2)$	9.5×10^{-6}
电机 3、4 转动惯量 $J_{m3}, J_{m4}/(\text{kg}\cdot\text{m}^2)$	3.0×10^{-6}
连杆 1 转动惯量 $I_{11}/(\text{kg}\cdot\text{m}^2)$	0.045
连杆 2 转动惯量 $I_{12}/(\text{kg}\cdot\text{m}^2)$	0.43
连杆 3 转动惯量 $I_{13}/(\text{kg}\cdot\text{m}^2)$	0.30
连杆 4 转动惯量 $I_{14}/(\text{kg}\cdot\text{m}^2)$	0.007
电机 1、2 的转子质量 $m_{r1}, m_{r2}/\text{kg}$	0.146
电机 3、4 的转子质量 $m_{r3}, m_{r4}/\text{kg}$	0.042
连杆 1 的质量 m_{11}/kg	4.30
连杆 2 的质量 m_{12}/kg	7.73
连杆 3 的质量 m_{13}/kg	6.64
连杆 4 的质量 m_{14}/kg	2.01
连杆 2 的长度 l_2/m	0.53
连杆 3 的长度 l_3/m	0.39
连杆 4 的长度 l_4/m	0.10
连杆 2 质心到关节质心距离 d_2/m	0.25
连杆 3 质心到关节质心距离 d_3/m	0.17
连杆 4 质心到关节质心距离 d_4/m	0.03
各关节刚度系数 $k/(\text{N}\cdot\text{m}/\text{rad})$	500
各电机驱动力矩 $\Delta Q/(\text{N}\cdot\text{m})$	2

图 3 关节转角轨迹图

图 4 关节角速度轨迹图

各算法对四自由度柔性关节机器人动力学模型的求解结果如表 2 所示，从表中可以看出，传统数

值求解方法中的 Euler 算法和 ODE45 算法都无解，除此之外，QSS 算法也无解，这是因为这些算法是显式算法，不能应用于刚性方程的求解。对比结果最好的传统算法 ODE15s，SCOA 算法在仿真效率略高的情况下，仿真精度提高了 34 倍。由此可见，SCOA 算法在精度和效率方面比其它算法更具优势。

为了研究量子的大小对 SCOA 算法性能的影响，本文选取了两个不同的量子值来仿真求解。由表 2 可以看出，当减小量子的大小时，SCOA 算法在几乎不增加 CPU 求解时间的同时，仿真精度提高了近 3 倍。

表 2 四自由柔性关节机器人动力学方程仿真结果对比表

仿真算法	误差容限	相对误差	CPU 时间 /s	仿真步数
Euler	$\text{Tol}=10^{-5}$	—	—	—
ODE45	$\text{Tol}=10^{-5}$	—	—	—
ODE15s	$\text{Tol}=10^{-5}$	4.599×10^{-3}	3.913345	468
ODE23s	$\text{Tol}=10^{-5}$	8.372×10^{-3}	4.273836	432
QSS	$\text{Tol}=10^{-5}, \Delta Q=0.1$	—	—	—
SCOA	$\text{Tol}=10^{-5}, \Delta Q=0.1$	1.294×10^{-4}	3.193783	446
	$\text{Tol}=10^{-5}, \Delta Q=0.01$	0.326×10^{-4}	3.875411	571

5 结论

针对柔性关节机器人动力学刚性方程的求解问题，本文尝试运用不同于传统的时间离散数值积分方法，提出了一种基于量化状态系统的步进校正优化算法(SCOA)。通过对机器人动力学的求解，得到如下结论：

(1) Euler 和 ODE45 是显式算法，稳定性较差，很难满足复杂动力学刚性方程在实际求解中的需要。

(2) ode15s 和 ode23s 是传统数值积分方法中求解刚性方程的较好算法，本文提出的 SCOA 算法在仿真精度与仿真效率两方面比 ode15s 和 ode23s 更具优势。

(3) QSS 算法不适合求解动力学刚性方程，SCOA 算法在保证仿真效率的同时能有效提高仿真精度，算法性能优于传统方法和 QSS 方法。

(4) 适当减小量子大小能在几乎不增加 CPU 求解时间的同时，有效提高仿真求解精度。

参考文献

- [1] 王田苗, 陶永. 我国工业机器人技术现状与产业化发展战略[J]. 机械工程学报, 2014, 50(9): 2-13.
WANG Tianmiao, TAO Yong. The present situation and

- industrialization development strategy of industrial robot technology in China[J]. Journal of Mechanical Engineering, 2014, 50(9): 2-13.
- [2] LIU Huashan, HUANG Yong. Bounded adaptive output feedback tracking control for flexible-joint robot manipulators[J]. Journal of Zhejiang University-Science (Applied Physics & Engineering), 2018, 19(7): 557-578.
- [3] 李成刚, 崔文, 尤晶晶. 多连杆柔性关节机器人的神经网络自适应反演控制[J]. 上海交通大学学报, 2016, 50(7): 1096-1101.
- LI Chenggang, CUI Wen, YOU Jingjing. Neural network adaptive inversion control of multi-link flexible joint robot[J]. Journal of Shanghai Jiaotong University, 2016, 50(7): 1096-1101.
- [4] 梁明轩, 李正刚, 唐任仲. 基于柔性多体动力学的机械臂结构优化设计[J]. 中国机械工程, 2017, 28(21): 2562-2566.
- LIANG Mingxuan, LI Zhenggang, TANG Renzhong. Optimal design of manipulator structure based on flexible multibody dynamics[J]. China Mechanical Engineering, 2017, 28(21): 2562-2566.
- [5] 王海, 付邦晨, 薛彬. 六自由度柔性关节机械臂的动力学分析[J]. 中国机械工程, 2016, 27(8): 1096-1101.
- WANG Hai, FU Bangchen, XUE Bin. Dynamic analysis of six-dof flexible joint manipulator[J]. China Mechanical Engineering, 2016, 27(8): 1096-1101.
- [6] 孔向东, 钟万勰. 柔性机械臂动力学方程的精细时程积分法[J]. 机器人, 1998, 20(5): 379-381.
- KONG Xiangdong, ZHONG Wanxie. Precise time-integration method for dynamic equation of flexible manipulator[J]. Robot, 1998, 20(5): 379-381.
- [7] WASFY T, NOOR A. Computational strategies for flexible multibody systems[J]. Advances in Mechanics, 2006, 56(6): 553-613.
- [8] SCHIEHLEN W. Computational dynamics: theory and applications of multibody systems[J]. European Journal of Mechanics, 2006, 25(4): 566-594.
- [9] BROGLIATO B, TEN A, PAOLI L, et al. Numerical simulation of finite dimensional multibody nonsmooth mechanical systems[J]. Applied Mechanics Reviews, 2016, 55(2): 107-149.
- [10] HAIRER E, WANNER G. Solving ordinary differential equations I: Nonstiff Problems [M]. Springer, Berlin, 1991.
- [11] KULIKOV G, WEINER R. Doubly quasi-consistent fixed-step size numerical integration of stiff ordinary differential equations with implicit two-step peer methods[J]. Journal of Computational and Applied Mathematics, 2018, 340(1): 256-275.
- [12] BULUT H, INC M. Two step method for the numerical integration of stiff differential equations[J]. International Journal of Computer Mathematics, 2000, 73(3): 333-340.
- [13] 王钟明, 陈澎我, 林国重. 一种改进型刚性常微分方程数值解法: T-R 预估-校正法[J]. 北京理工大学学报, 1990, 10(1): 18-25.
- WANG Zhongming, CHEN Shuwo, LIN Guozhong. An improved treanor's method for solving stiff ordinary differential equations --T-R predictor-corrector method[J]. Journal of Beijing Institute of Technology, 1990, 10(1): 18-25.
- [14] 廖文远, 李庆扬. 一类求解刚性常微分方程的半隐式多步 RK 方法[J]. 清华大学学报(自然科学版), 1999, 39(6): 1-4.
- LIAO Wenyuan, LI Qingyang. One class of semi-implicit multistep Runge-Kutta method for stiff ODEs[J]. Journal of Tsinghua University (Science and Technology), 1999, 39(6): 1-4.
- [15] 王斌锐, 方水光, 金英连. 综合关节和杆件柔性的机械臂刚柔耦合建模与仿真[J]. 农业机械学报, 2012, 43(2): 212-215.
- WANG Binrui, FANG Shuiguang, JIN Yinglian. Modeling and simulation of rigid-flexible coupling of manipulator with joint and bar flexibility[J]. Transactions of The Chinese Society of Agricultural Machinery, 2012, 43(2): 212-215.
- [16] ZEIGLER B, LEE J. Theory of quantized systems: formal basis for DEVS/HLA distributed simulation environment[J]. In Proceedings of SPIE-The International Society for Optical Engineering, 1998, 3369(1): 49-58.
- [17] KOFMAN E, JUNCO S. Quantized-State Systems: a DEVS approach for continuous system simulation[J]. Transactions of the Society for Modeling and Simulation International, 2001, 18(1): 2-8.
- [18] 杨祎, 赵争鸣, 檀添, 等. 离散状态事件驱动仿真方法及自适应预估校正算法[J]. 电工技术学报, 2017, 32(12): 34-41.
- YANG Yi, ZHAO Zhengming, TAN Tian, et al. Discrete State Event Driven Method and Self-Adapted Predictor-Corrector Algorithm[J]. Transactions of China Electrotechnical Society, 2017, 32(12): 34-41.
- [19] CELLIER E, KOFMAN E. Continuous system simulation[M]. Springer, New York, 2006.
- [20] KOFMAN E. A third order discrete event simulation method for continuous system simulation[J]. Latin American Applied Research, 2006, 36(2): 101-108.
- [21] MIGONI G, BORTOLOTTI M, KOFMAN E, et al. Linearly implicit quantization-based integration methods for stiff ordinary differential equations[J]. Simulation Modelling Practice & Theory, 2013, 35(6): 118-136.
- [22] BERGERO F, CASELLA F, KOFMAN E, et al. On the

- efficiency of quantization-based integration methods for building simulation[J]. Building Simulation, 2017, 11(2): 1-14.
- [23] 朱雨童, 王江云, 韩亮. 复杂航天器虚拟样机模型描述与集成[J]. 计算机集成制造系统, 2010, 16(11): 2363-2368.
ZHU Yutong, WANG Jiangyun, HAN Liang. Description and integration of complex spacecraft virtual prototype model[J]. Computer Integrated Manufacturing Systems, 2010, 16(11): 2363-2368.
- [24] 朱雨童, 王江云, 韩亮. 基于量化状态积分的空间目标温度计算[J]. 红外与激光工程, 2011, 40(12): 2345-2348.
ZHU Yutong, WANG Jiangyun, HAN Liang. Space target temperature calculation based on quantized state integration[J]. Infrared & Laser Engineering, 2011, 40(12): 2345-2348.
- [25] 檀添, 赵争鸣, 李帛洋, 等. 基于离散状态事件驱动的电力电子瞬态过程仿真方法[J]. 电工技术学报, 2017, 32(13): 41-50.
TAN Tian, ZHAO Zhengming, LI Boyang, et al. A transient process simulation method for power electronics based on discrete state event - driven [J]. Transactions of China Electrotechnical Society, 2017, 32(13): 41-50.
- [26] 李帛洋, 赵争鸣, 檀添, 等. 后向离散状态事件驱动电力电子仿真方法[J]. 电工技术学报, 2017, 32(12): 42-49.
LI Boyang, ZHAO Zhengming, TAN Tian, et al. A Backward Discrete State Event Driven Simulation Method for Power Electronics Based on Finite State Machine[J]. Transactions of China Electrotechnical Society, 2017, 32(12): 42-49.
- [27] 秦建, 沈沉, 陈颖, 等. 基于量子化状态驱动的时空自律仿真方法[J]. 中国电机工程学报, 2017, 37(10): 2869-2877.
QIN Jian, SHEN Chen, CHEN Ying, et al. A spatio-temporally self-regulated simulation method based on quantized state system[J]. Proceedings of the Csee, 2017, 37(10): 2869-2877.
- [28] KOFMAN E. Relative error control in quantization based integration[J]. Latin American Applied Research, 2009, 39(3): 231-237.
- [29] 张丹丹. 柔性关节机器人动力学建模与控制[D]. 南京: 南京理工大学, 2017.
ZHANG Dandan. Dynamics modeling and control of flexible joint robot[D]. Nanjin: Nanjing University of Science and Technology, 2017.

作者简介: 李志华(通信作者), 男, 1966 年出生, 博士, 教授。主要研究方向为复杂机电系统多领域建模与仿真优化、CAD/CAE。
E-mail: d98Lzh@263.net