

2-20-2021

A Quantized State System Method With Adaptive Quantum

Zhihua Li

School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Dongjin Fu

School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Li Guang

School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Zhihua Fan

School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

A Quantized State System Method With Adaptive Quantum

Abstract

Abstract: Quantized state system (QSS) is a new numerical integration method. It has advantages over the traditional time-discrete integration methods in solving general ordinary differential equation (ODE) systems. But it is hard to choose an appropriate quantum for the QSS method. *In order to improve the accuracy and efficiency of QSS, a quantized state system method with adaptive quantum (VQSS) is proposed. Combined with the idea of step control in the variable step Runge-Kutta method, it can adaptively change the quantum in the process of computation.* The feasibility of this algorithm is verified by the simulation of two examples and a vehicle motion case. The results show that VQSS has higher computational accuracy and efficiency than that of traditional QSS, ODE45 and ODE23.

Keywords

quantized state system, adaptive quantum, ordinary differential equation, numerical integration, simulation

Recommended Citation

Li Zhihua, Fu Dongjin, Li Guang, Fan Zhihua. A Quantized State System Method With Adaptive Quantum[J]. Journal of System Simulation, 2021, 33(2): 445-452.

一种带自适应量子的量化状态系统方法

李志华, 傅东金, 李广, 樊志华

(杭州电子科技大学 机械工程学院, 浙江 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, 在求解一般常微分方程(Ordinary Differential Equation, ODE)系统时, 比传统基于时间离散的积分方法更具优势, 但 QSS 方法存在量子选择困难的问题, 为此提出一种带自适应量子的量化状态系统方法(VQSS), 它结合变步长龙格库塔方法中步长控制的思想, 能够在计算过程中对量子进行自适应变化, 以提高 QSS 算法的精度和效率。通过两个算例和一个汽车运动实例的仿真求解, 验证了算法的可行性。将其与传统 QSS, ODE45 和 ODE23 等算法进行比较, 结果表明 VQSS 具有更高的计算精度和计算效率。

关键词: 量化状态系统; 自适应量子; 常微分方程; 数值积分; 仿真

中图分类号: TP391

文献标志码: A

文章编号: 1004-731X (2021) 02-0445-08

DOI: 10.16182/j.issn1004731x.joss.19-0517

A Quantized State System Method With Adaptive Quantum

Li Zhihua, Fu Dongjin, Li Guang, Fan Zhihua

(School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Quantized state system (QSS) is a new numerical integration method. It has advantages over the traditional time-discrete integration methods in solving general ordinary differential equation (ODE) systems. But it is hard to choose an appropriate quantum for the QSS method. *In order to improve the accuracy and efficiency of QSS, a quantized state system method with adaptive quantum (VQSS) is proposed. Combined with the idea of step control in the variable step Runge-Kutta method, it can adaptively change the quantum in the process of computation.* The feasibility of this algorithm is verified by the simulation of two examples and a vehicle motion case. The results show that VQSS has higher computational accuracy and efficiency than that of traditional QSS, ODE45 and ODE23.

Keywords: quantized state system; adaptive quantum; ordinary differential equation; numerical integration; simulation

引言

计算机仿真技术已经成为人类认识和改造客观世界的重要方法。常微分方程(Ordinary Differential Equation, ODE)系统模型主要来自现代机械、电力电子、神经网络、多粒子碰撞动力学等科学与工程领域^[1]。ODE 系统仿真求解采用的积分方法基本都是基于时间离散的数值积分方法, 如

Euler, Runge-Kutta 等, 它们在时间轴上离散模型, 逐(时间)点求得状态变量的值, 从而推进积分计算。

量化状态系统(Quantized State System, QSS)方法是一种新的数值积分方法, 它与传统时间离散的积分方法显著不同, QSS 是对状态变量进行离散, 即状态变量以“量子”为单位进行跃迁, 依次计算每次状态变量跃迁所需的时间, 从而推进积分的计算。这一思想由 Zeigler^[2]于 1998 年提出, 并由

收稿日期: 2019-09-05 修回日期: 2020-01-14

基金项目: 国家重点研发计划(2017YFB1301300), 浙江省自然科学基金(LY18E050008, LY19E050013)

第一作者: 李志华(1966-), 男, 博士, 教授, 研究方向为多领域建模与仿真优化、CAD/CAE。E-mail: D98LZH@263.net

Kofman^[3]于2001年实现。在求解一般ODE系统时，QSS算法不仅具有强稳定性、高精度等优点，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^[1]。

Kofman等^[4-12]后续又提出了QSS2, QSS3, BQSS, LIQSS等算法及相关应用，它们在一定程度上提高了系统的仿真精度，并解决了ODE系统部分刚性问题。但是这些算法和QSS一样，都存在计算精度与计算效率冲突的问题，即量子越小，计算精度越高，但计算效率越低；量子越大，计算精度越低，但计算效率越高。在不知道仿真轨迹的前提下，几乎不可能选取到一个合适的量子大小来计算^[13]，因此寻找一种能够在精度和效率之间取得平衡的量子方法则显得尤为重要。

国内学者对QSS的研究较少，只有秦建、檀添等将QSS方法应用于电力电子系统的仿真求解中^[14-17]。

针对QSS方法中量子选择困难的问题，本文提出一种带自适应量子的QSS算法(VQSS)，该算法在Matlab平台上予以了实现，对2个典型算例和一个汽车运动实例进行了仿真求解，并与传统QSS, ODE45和ODE23等算法进行了性能对比。

1 量化状态系统

QSS是对状态变量进行量化，而非时间的离散化。

考虑一组由常微分方程表示的状态方程系统(State Equation System, SES):

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

式中： $\mathbf{x}(t) \in R^n$ 为状态向量。QSS方法将式(1)近似为：

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

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

$$q_j(t) = \begin{cases} x_j(t) & |q_j(t) - x_j(t)| \geq \Delta Q_j \\ q_j(t^-) & \text{其他} \end{cases} \quad (3)$$

式中： ΔQ_j 为量子，量子是事先给定的，与传统时间离散积分方法中的时间步长 h 作用类似，量子的大小直接决定了算法误差的大小。从式(3)可以看出， $q_j(t)$ 遵循分段常数的轨迹，只有当 $x_j(t)$ 的变化超过了某个阈值(ΔQ_j)时，才会引起 $q_j(t)$ 的变化，这个行为称之为跃迁，且 $q_j(t)$ 与 $x_j(t)$ 的差值不会超过一个量子，这也保证了算法的稳定性和全局误差界限^[18]。QSS的状态变量 x 和量化变量 q 的轨迹如图1所示，由于 $q_j(t)$ 轨迹的特殊性，因此式(2)具有很明确的数值解，很容易实现QSS算法仿真。

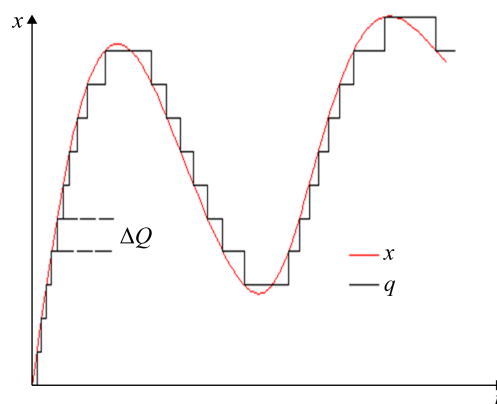


图1 状态变量 x 和量化变量 q 的轨迹图

Fig. 1 Trajectory chart of diagram of state variable x and quantitative variable q

QSS具体实现如下，设 $i=1,2,\dots,n$, $j=1,2,\dots,m$, T 为设置的仿真时间，由式(2)求得 x_j 的导数，再得出 x_j 跃迁需要变化的大小：

$$\Delta x_j = \begin{cases} q_j - x_j + \Delta Q_j & \dot{x}_j > 0 \\ x_j - q_j + \Delta Q_j & \dot{x}_j < 0 \end{cases} \quad (4)$$

根据 \dot{x}_j 和 Δx 求得各变量跃迁的时间 Δt_j ：

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

从而求出系统每推进一次仿真所需的时间 Δt ：

$$\Delta t = \min\{\Delta t_j\} \quad (6)$$

由于变量只有发生跃迁时，即 $|q_j(t) - x_j(t)| = \Delta Q_j$ ，才会更新其量化变量 q_j ，此时 $q_j = x_j$ 。因此仿真时间的推进和变量的计算由式(7)和(8)确定：

$$t^{i+1} = t^i + \Delta t \quad (7)$$

$$x_j^{i+1} = x_j^i + \dot{x}_j \cdot \Delta t \quad (8)$$

式中: q_j 是否更新决定了变量导数 \dot{x}_j 是否更新, 发生跃迁的状态变量的变化值记为 Δx_j , 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_j , 则 QSS 算法的流程图如图 2 所示。

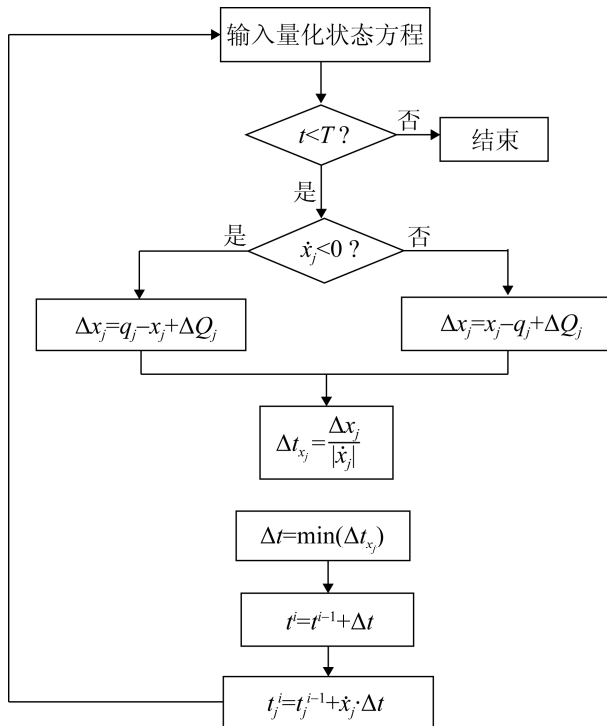


图 2 QSS 算法流程图
Fig. 2 Flow diagram of QSS Algorithm

2 带自适应量子的量化状态系统方法

2.1 VQSS 算法

在 QSS 中, 量子是事先给定的, 量子的大小直接决定了算法误差的大小。在不知道仿真轨迹的前提下, 几乎不可能选取到一个合适的量子大小来计算。因此, 本文提出了带自适应量子的量化状态系统方法, 它结合了变步长龙格库塔方法中步长控制的思想。

在龙格库塔积分方法中, 单从每一步看, 步长越小, 截断误差就越小, 但是随着步长的缩小, 在一定的求解范围内所要完成的步数就增加了, 步数的增加不但引起计算量的增大, 而且可能导致较大

的舍入误差积累, 因此存在一个合适步长的选择问题, 故后来又提出了变步长龙格库塔法。变步长龙格库塔法先以步长 h 求一个近似值记为 $y_{n+1}^{(h)}$, 然后将步长折半, 即取 $h/2$ 为步长, 从 x_n 跨两步到 x_{n+1} , 再求得一个近似值 $y_{n+1}^{(h/2)}$ 。通过检查步长折半前后两次计算结果的偏差 Δ 来判定所选的步长是否合适, 给定精度 ε , 如果 $\Delta > \varepsilon$, 则将步长反复折半进行计算, 直到 $\Delta < \varepsilon$ 为止, 这时取得的结果为最终结果 $y_{n+1}^{(h/2)}$; 如果 $\Delta < \varepsilon$, 则将步长反复加倍, 直到 $\Delta > \varepsilon$ 为止, 这时再将步长折半一次得到所要的结果。

本文提出的 VQSS, 其思路如下:

- (1) 状态变量先以量子 ΔQ 求得一个近似值记为 $y_{n+1}^{(\Delta Q)}$;
- (2) 将量子 ΔQ 折半, 即取 $\Delta Q/2$ 为量子, 从 y_n 跨两步达到 y_{n+1} ;
- (3) 比较求解结果的时间点并加以校正, 求得一个近似值, 记为 $y_{n+1}^{(\Delta Q/2)}$ 。

最后通过检查量子折半前后 2 次计算结果的偏差 er 和误差限 e_{max}, e_{min} 的关系来判定所选的量子是否合适。如果 $er > e_{max}$, 则下一次计算的量子比本次的量子减小一倍; 如果 $er < e_{min}$, 则下一次计算的量子比本次的量子增加一倍; 如果 $e_{min} < er < e_{max}$, 则下一次计算的量子与本次的量子保持相同。

本算法结合了传统 QSS 算法和变步长龙格库塔方法的思想。首先, 传统 QSS 方法作为显式算法, 不需要迭代计算, 减少了仿真时间; 其次变步长龙格库塔方法的思想平衡了计算精度与计算效率的矛盾, 提高了仿真的整体性能。算法实现具体如下:

假设系统的量化状态方程如式(2)所示, 令系统状态变量为 x_j , 则对应的量化状态变量为 $q_j, j=1, 2, \dots, m$ 。

首先使用传统的 QSS 方法计算(后文称为量子改变前的第 1 次计算), 按照式(2)~(8)得到第 1 次的结果, x_j^*, q_j^*, t^* 。

然后将跃迁变量的量子减半,再由本次初值开始重新按照式(2)~(8)再计算 2 次,得到量子改变后第 1 次计算结果,量子改变后 $x_j^{(1)}, q_j^{(1)}, t^{(1)}$; 第 2 次计算结果 $x_j^{(2)}, q_j^{(2)}, t^{(2)}$;

根据 3 次求解结果进行时间校正:

$$x_j = \begin{cases} x_j^{(1)} + \dot{x}_j^{(2)} \cdot (t^* - t^{(1)}) & t^* < t^{(2)} \\ x_j^{(2)} + \dot{x}_j^{(3)} \cdot (t^* - t^{(2)}) & t^* > t^{(2)} \end{cases} \quad (9)$$

式中: x_j 为校正后的状态变量值; $x_j^{(1)}$ 为量子改变后第 1 次计算结果的状态变量值; $x_j^{(2)}$ 为量子改变后第 2 次计算结果的状态变量值; $t^{(1)}$ 为量子改变后第 1 次计算结果的时间值; $t^{(2)}$ 为量子改变后第 2 次计算结果的时间值; $\dot{x}_j^{(2)}$ 为量子改变后第 2 次计算结果的状态变量值; $\dot{x}_j^{(3)}$ 为量子改变后第 3 次计算结果的状态变量值;

计算校正值和第一次实际计算值的误差 er :

$$er = |x_j - x_j^*| \quad (10)$$

设 e_{\max} 为最大误差限, e_{\min} 为最小误差限,比较误差 er 和误差限的关系得出量子是否变化:

$$\Delta Q_j = \begin{cases} 2\Delta Q_j^* & er < e_{\min} \text{ 且 } er \neq 0 \\ \Delta Q_j^* & \text{其他} \\ \frac{1}{2}\Delta Q_j^* & er > e_{\max} \end{cases} \quad (11)$$

式中: ΔQ_j 为本次计算起始量子大小; ΔQ_j^* 为下次计算起始量子大小;

这种自适应调整量子大小的方法虽然增加了局部的计算量,但从总体上考虑,计算成本往往是合算的,它较好地解决了计算精度与计算效率之间的矛盾。

计算的最终结果由式(12)给定:

$$x_{jac} = \begin{cases} x_j & er > e_{\max} \\ x_j^* & \text{其他} \end{cases} \quad (12)$$

该方法依旧使用传统 QSS 方法中的时间推进方法来进行仿真时钟的推进,每次得到 x_{jac} 后,对跃迁变量的 q_j 更新, $q_j = x_j$ 。之后再重复这一过程,实现仿真求解。

2.2 算法收敛性及稳定性分析

2.2.1 收敛性分析

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

$$\dot{x}(t) = f(x(t) + \Delta x(t), u(t)) \quad (13)$$

令 $\phi(t)$, $\phi_1(t)$ 分别为公式(1)与(13)在初始条件 $x(0) = x_0$ 的解,且有 $\phi(0) = \phi_1(0)$,则有

$$\phi_1(t) - \phi(t) = \int_0^t [f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))] d\tau \quad (14)$$

根据欧几里得范数,公式(14)有

$$\begin{aligned} \|\phi_1(t) - \phi(t)\| &= \left\| \int_0^t [f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))] d\tau \right\| \Rightarrow \\ \|\phi_1(t) - \phi(t)\| &\leq \int_0^t \|f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))\| d\tau \quad (15) \end{aligned}$$

假定 M 为函数 f 的 Lipschitz 常数,则式(15)有

$$\begin{aligned} \|\phi_1(t) - \phi(t)\| &\leq \int_0^t \|f(\phi_1(\tau) + \Delta x(\tau), u(\tau)) - f(\phi(\tau), u(\tau))\| d\tau \Rightarrow \\ \|\phi_1(t) - \phi(t)\| &\leq \int_0^t M \|\phi_1(\tau) + \Delta x(\tau) - \phi(\tau)\| d\tau \Rightarrow \\ \|\phi_1(t) - \phi(t)\| &\leq \int_0^t M \|\phi_1(\tau) - \phi(\tau)\| d\tau + Mt\Delta_x \quad (16) \end{aligned}$$

因为 $\phi(t)$, $\phi_1(t)$ 都是连续的,且 M 为正数,则根据 Gronwall-Bellman 不等式可以将式(16)改为

$$\begin{aligned} \|\phi_1(t) - \phi(t)\| &\leq Mt\Delta_x + \int_0^t M^2 s \Delta_x e^{\int_s^t M d\tau} ds \Rightarrow \\ \|\phi_1(t) - \phi(t)\| &\leq (e^{Mt} - 1)\Delta_x \quad (17) \end{aligned}$$

因为 M 与 t 不依赖于 Δx ,则有

$$\lim_{\Delta_x \rightarrow 0} \|\phi_1(t) - \phi(t)\| = 0 \quad (18)$$

因此,从式(18)可以看出, VQSS 算法具有收敛性。

2.2.2 稳定性分析

假定式(1)是线性时不变系统,则式(1)可改写为:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (19)$$

则运用量化后可写为:

$$\dot{x}(t) = Aq(t) + Bu(t) \quad (20)$$

式中: A 为可对角化的 Hurwitz 矩阵,则误差为:

$$|e(t)| \leq |V| \cdot |\operatorname{Re}(A)^{-1} \cdot A| \cdot |V^{-1}| \cdot \Delta Q \quad (21)$$

式中: A 为矩阵 A 的特征值的对角矩阵; V 为矩阵 A 的特征向量矩阵; Re 为取复数的实数部分。

可以看到, 误差限与量子成正比, 即对于任何量子误差总是有界的, 所以 VQSS 算法是稳定的。

VQSS 伪代码如下:

```

while( $t < t_f$ ) // 仿真终止时间  $t_f$ 
 $\Delta t = \min\left(\frac{\Delta x_j}{|\dot{x}_j|}\right)$  // 求最小跃迁时间和确定跃迁
变量
 $x_j^* = x_{jac} + \dot{x}_{jac} \cdot \Delta t$  // 用传统 QSS 确定比较量
 $t^* = t + \Delta t$  // 用传统 QSS 确定比较时间节点
 $\Delta Q_j = \frac{1}{2} \Delta Q_j$  // 跃迁变量的量子减半
While( $i < 2$ )
 $\Delta t^{(i)} = \min\left(\frac{\Delta x_j^{(i)}}{|\dot{x}_j^{(i)}|}\right)$  // 变量子后的跃迁时间
 $x_j^{(i)} = x_j^{(i-1)} + \dot{x}_j^{(i-1)} \cdot \Delta t^{(i)}$  // 计算变量子后的 2 次
结果
end while
If ( $t^* < t^{(2)}$ ) // 对跃迁变量的时间节点进行校正
 $x_j = x_j^{(1)} + \dot{x}_j^{(2)} \cdot (t^* - t^{(1)})$ 
else
 $x_j = x_j^{(2)} + \dot{x}_j^{(3)} \cdot (t^* - t^{(2)})$ 
end if
If ( $er < e_{\min}$  &  $er \neq 0$ ) // 根据误差值范围改变量
子大小
 $\Delta Q_i = 2 \Delta Q_i^*$  // 误差足够小, 量子翻倍
 $x_{jac} = x_j^*$  // 第 1 次计算结果可以选用
else if ( $er > e_{\max}$ )
 $\Delta Q_i = \frac{1}{2} \Delta Q_i^*$  // 误差过大, 量子减半
 $x_{jac} = x_j$  // 取校正后的结果为本次计算结果
else
 $x_{jac} = x_j^*$  // 量子不变, 第 1 次计算结果可以选
用
end if

```

$t = t + \Delta t$ // 仿真时间推进

end while

3 算例

3.1 仿真背景

本节引入了 2 个典型的 ODE 算例和一个工程仿真实例, 以其解析解(或 DASSL 解)为基准值, 将 VQSS 与传统时间离散方法(包括 ODE45, ODE23)和传统 QSS 方法从仿真精度和仿真效率两方面进行性能对比。

(1) VQSS 算法及 ODE45, ODE23, QSS 算法全部在 Matlab 平台上实现;

(2) 计算机硬件配置为 Intel i5-4260 处理器, CPU 运行速度为 1.4 GHz, 操作系统为 Windows7;

(3) 对所有的算例设置相同的误差容限, 即 $\operatorname{Rel.Tol} = \operatorname{Abs.Tol} = 10^{-3}$;

(4) 系统各个状态变量的相对误差如式(22):

$$e_{rr} = \sqrt{\frac{\sum (u_m[k] - u[k])^2}{\sum u[k]^2}} \quad (22)$$

式中: $u_m[k]$ 为各个算法求得的解; $u[k]$ 为解析解或者是 DASSL 求解器在很小误差设定(10^{-9})下求得的解, 在此作为基准值。

3.2 算例验证

算例 1: 线性二阶系统^[19]

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -3x_1(t) - 4x_2(t) + 1 \end{cases} \quad (23)$$

式中: $t \in [0, 20]$, $x_1(0) = 0$, $x_2(0) = 0$;

算例 2: 蹦极跳动力学模型(非线性二阶系统)^[20]

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{580 - 20x_1 - x_2 - |x_2| x_2^3}{58} \end{cases} \quad (24)$$

式中: $t \in [0, 10]$, $x_1(0) = -30$, $x_2(0) = 0$ 。

算例 3: 汽车转弯模型^[21]

如图 3 所示, δ 表示前轮与汽车中轴线的夹角, θ 表示汽车中轴线与水平轴线的夹角, v 表示前桥

中心的运动速度, (x, y) 表示后桥中心的坐标。

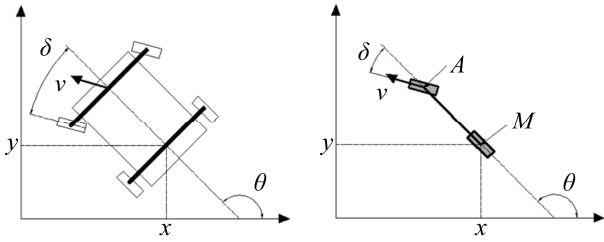


图3 汽车在平地上运动的俯视图

Fig. 3 Top view of a car moving on level ground

假设汽车前后轴距 $L=3$ m, 则有:

$$\begin{cases} v \cos \delta = v_M \\ v \sin \delta = L\dot{\theta} \end{cases} \quad (25)$$

于是可得:

$$\dot{\theta} = v \sin \delta / L \quad (26)$$

并且有:

$$\begin{cases} \dot{x} = v_M \cos \theta = v \cos \delta \cos \theta \\ \dot{y} = v_M \sin \theta = v \cos \delta \sin \theta \end{cases} \quad (27)$$

所以汽车的状态方程可以写成:

$$\begin{cases} \dot{x}_1 = x_4 \cos x_5 \cos x_3 \\ \dot{x}_2 = x_4 \cos x_5 \sin x_3 \\ \dot{x}_3 = \frac{x_4 \sin x_5}{L} \\ \dot{x}_4 = u_1 \\ \dot{x}_5 = u_2 \end{cases} \quad (28)$$

其中: $[x_1, x_2, x_3, x_4, x_5] = [x, y, \theta, v, \delta]$, u_1, u_2 为输入常量。

初始值设为: $[x_1, x_2, x_3, x_4, x_5] = [0, 0, 0, 7 \text{ m}, 0.2 \text{ rad}]$, 该系统仿真时长设为 20 s。

3.3 仿真结果

用 VQSS 算法对算例 1 进行仿真求解, 得到如图 4 所示轨迹。

各算法对算例 1 的求解结果如表 1 所示, 从表 1 可以看出, 对比 ODE45, VQSS 在精度上提高了 1 倍, 效率上提高了 147 倍; 对比 ODE23, VQSS 在精度上提高了 1.6 倍, 效率上提高了 64 倍; 对比传统 QSS, VQSS 在精度上提高了 3.3 倍, 效率上提高了 20 倍。

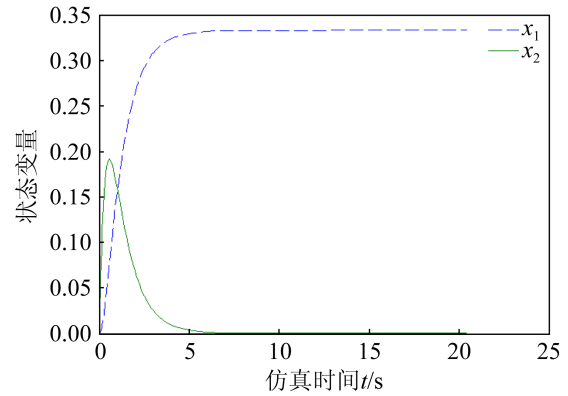


图4 VQSS 对算例 1 的仿真轨迹

Fig. 4 Simulation results of example 1 by VQSS

表1 算例 1 的仿真结果

Tab. 1 Simulation results of example 1

仿真算法	误差容限	相对误差	CPU 仿真 时间/s	仿真步数
ODE45	Tol=10 ⁻³	4.081×10 ⁻³	0.277 874	121
ODE23	Tol=10 ⁻³	5.107×10 ⁻³	0.121 716	48
QSS	Tol=10 ⁻³ , ΔQ=10 ⁻³	8.225×10 ⁻³	0.039 68	330
VQSS	Tol=10 ⁻³ , ΔQ=10 ⁻³	1.934×10 ⁻³	0.001 868	69

用 VQSS 算法对算例 2 进行仿真求解, 得到如图 5 所示轨迹。

各算法对算例 2 的求解结果见表 2。从表 2 可以看出, 对比 ODE45, VQSS 在精度上提高了 35 倍, 效率上提高了 4.5 倍; 对比 ODE23, VQSS 在精度上提高了 42 倍, 效率上提高了 1.6 倍; 对比传统 QSS, VQSS 在精度上提高 5.5 倍, 效率上提高了 10.7 倍。

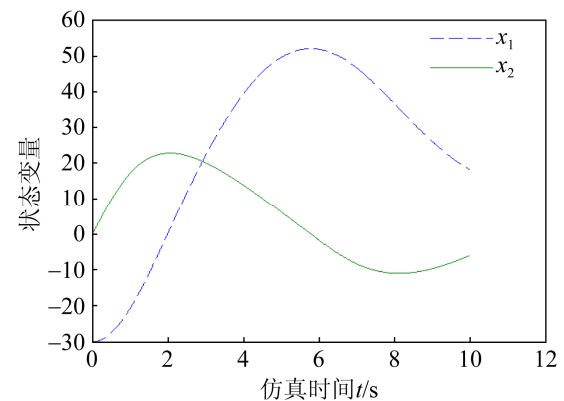


图5 VQSS 对算例 2 的仿真轨迹

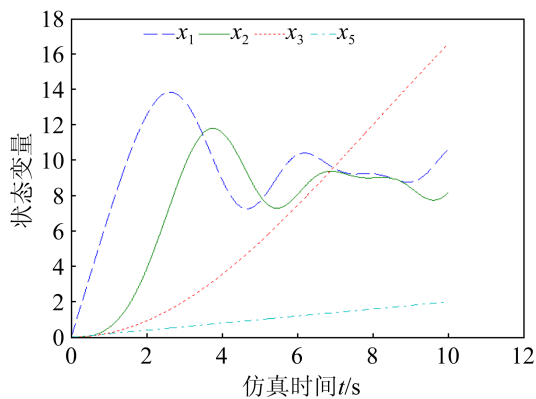
Fig. 5 Simulation results of example 2 by VQSS

表 2 算例 2 的仿真结果
Tab. 2 Simulation results of example 2

仿真算法	误差容限	相对误差	CPU 仿真 时间/s	仿真步数
ODE45	$Tol=10^{-3}$	1.152×10^{-3}	0.404 275	69
ODE23	$Tol=10^{-3}$	1.386×10^{-3}	0.189 827	34
QSS	$Tol=10^{-3}$, $\Delta Q=10^{-2}$	2.049×10^{-4}	0.864 51	17 735
VQSS	$Tol=10^{-3}$, $\Delta Q=10^{-2}$	3.17×10^{-5}	0.073 714	3 705

用 VQSS 算法对算例 3 进行仿真求解, 得到如图 6 所示轨迹。

各算法对算例 3 的求解结果如表 3 所示。从表 3 可以看出, 对比 ODE45, VQSS 在精度上提高了 12 倍, 效率上提高了 5.7 倍; 对比 ODE23, VQSS 在精度上提高了 17 倍, 效率上提高了 1.7 倍; 对比传统 QSS, VQSS 在精度上提高 2.7 倍, 效率上提高了 24 倍。

图 6 VQSS 对算例 3 的仿真轨迹
Fig. 6 Simulation results of example 3 by VQSS表 3 算例 3 的仿真结果
Tab. 3 Simulation results of example 3

仿真算法	误差容限	相对误差	CPU 仿真 时间/s	仿真步数
ODE45	$Tol=10^{-3}$	5.72×10^{-3}	0.450 777	69
ODE23	$Tol=10^{-3}$	7.99×10^{-3}	0.181 184	57
QSS	$Tol=10^{-3}$, $\Delta Q=10^{-3}$	1.65×10^{-3}	1.681 114	18 983
VQSS	$Tol=10^{-3}$, $\Delta Q=10^{-3}$	4.43×10^{-4}	0.067 285	2 607

4 结论

本文基于量化状态系统方法和变步长龙格库塔中步长控制的思想, 提出一种自适应量子的量化状态系统(VQSS)新方法, 通过对三个算例的仿真求解, 得到如下结论:

(1) ODE45 和 ODE23 是求解一般(即非刚性)ODE 系统的经典算法, 而 VQSS 无论是在计算精度上还是计算效率上都更胜一筹。

(2) 由于 VQSS 中的量子能够在计算过程中自适应变化到合理的大小, 所以相对于传统 QSS, VQSS 具有更高的计算精度和计算效率。

未来将考虑将自适应量子策略应用到 LIQSS 等算法中, 以更好地求解刚性 ODE 系统。

参考文献:

- [1] Cellier E, Kofman E. Continuous System Simulation[M]. New York: Springer, 2006.
- [2] Zeigler B, Lee J S. Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment[J]. In Proceedings of SPIE (S0277-786X), 1998, 3369: 49-58.
- [3] Kofman E, Junco S. Quantized-state Systems: a DEVS Approach for Continuous System Simulation[J]. Transactions of the Society for Modeling and Simulation International (S0740-6797), 2001, 18(1): 2-8.
- [4] Kofman E. Quantization-Based Simulation of Differential Algebraic Equation Systems[J]. Simulation (S0037-5497), 2003, 79(7): 363-376.
- [5] Kofman E. A Second-order Approximation for DEVS Simulation of Continuous Systems[J]. Simulation (S0037-5497), 2002, 78(2): 76-89.
- [6] Kofman E. A Third Order Discrete Event Method for Continuous System Simulation[J]. Latin American applied research (S0327-0793), 2006, 36(2): 101-108.
- [7] Migoni G, Kofman E. Linearly Implicit Discrete Event Methods for Stiff ODE's[J]. Latin American Applied Research (S0327-0793), 2009, 39(3): 245-254.
- [8] Fernandez J, Kofman E, Bergero F. A Parallel Quantized State System Solver for ODEs[J]. Journal of Parallel and Distributed Computing (S0743-7315), 2017, 106: 14-30.
- [9] Bergero F, Fernández J, Kofman E, et al. Time Discretization Versus State Quantization in the Simulation of a One-dimensional Advection-diffusion-

- reaction Equation[J]. Simulation (S0037-5497), 2016, 92(1): 47-61.
- [10] Federico M, Caselia F, Kofman E, et al. On the Efficiency of Quantization-based Integration Methods for Building Simulation[J]. Building Simulation (S1996-3599), 2017, 11(2): 1-14.
- [11] Di P F, Migoni G. Improving Linearly Implicit Quantized State System Methods[J]. Simulation (S0037-5497), 2019, 95(2): 127-144.
- [12] Castro R, Kofman E. Activity of Order n in Continuous Systems[J]. Simulation (S0037-5497), 2015, 91(4): 337-348.
- [13] Kofman E. Discrete Event Simulation of Hybrid Systems[J]. SIAM Journal on Scientific Computing (S1064-8275), 2004, 25(5): 1771-1797.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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): 33-41.
Yang Yi, Zhao Zhengming, Tan Tin, et al. Discrete State Event Driven Method and Self-Adapted Predictor-Corrector Algorithm[J]. Transactions of China Electrotechnical Society, 2017, 32(12): 33-41.
- [18] Kofman E. Relative Error Control in Quantization Based Integration[J]. Latin American applied research (S0327-0793), 2009, 39(3): 231-237.
- [19] 王积伟. 现代控制理论工程[M]. 北京: 高等教育出版社, 2003: 99.
Wang Jiwei. Modern Control Theory Engineering[M]. Beijing: Higher Education Press, 2003: 99.
- [20] 黎明安, 钱利. Matlab/Simulink 动力学系统建模与仿真[M]. 北京: 国防工业出版社, 2015: 7-8.
Li Mingan, Qian Li. Matlab/Simulink Dynamic System Modeling and Simulation[M]. Beijing: National Defense Industry Press, 2015: 7-8.
- [21] 吕克·若兰. 机器人自动化: 建模、仿真与控制[M]. 北京: 机械工业出版社, 2017: 8-9.
Luc Jaulin. Robot Automation: Modeling, Simulation and Control[M]. Beijing: China Machine Press, 2017: 8-9.