

Research Article

Quantized State Based Simulation of Time Invariant and Time Varying Continuous Systems

Xiao Song, Yaofei Ma, Wei Zhang, and Jianguyun Wang

Science and Technology on Aircraft Control Laboratory, School of Automation Science, Beihang University, Beijing 100191, China

Correspondence should be addressed to Yaofei Ma; mayaofeibuaa@163.com

Received 6 June 2014; Revised 13 October 2014; Accepted 13 October 2014

Academic Editor: Kang Li

Copyright © 2015 Xiao Song et al. 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.

Continuous system can be discretized for computer simulation. Quantized state systems (QSS) method has been used to discretize time invariant systems based on the discretization of the state space. A HLA based QSS method is proposed in this paper to address issues of real-time advancements in simulation and an aircraft control example was introduced to illustrate our method. Moreover, to simulate time varying systems, a novel approach is also proposed and exemplified with a practical case.

1. Introduction

Continuous system simulation is a topic which has advanced significantly with the appearance of modern computers. Based on classic methods for numerical resolution of ordinary differential equation (ODE) such as Euler and Runge-Kutta, several variable steps and implicit ODE solver methods were developed. These modern methods which usually make use of iteration rules and symbolic manipulation allow the efficient simulation of complex systems, including differential algebraic equation (DAE) and variable structure systems.

Although there are several differences between the mentioned ODE solver algorithms, all of them share a property: they are based on time discretization. That is, they give a solution obtained from a difference equation system (i.e., a discrete-time model) which is only defined in some discrete instants.

A completely different approach for ODE numerical simulation has been being developed since the end of the 90s. In this new approach, the time discretization is replaced by the state variables quantization and a discrete event simulation model (within the DEVS formalism framework) is obtained instead of a discrete-time one.

Discrete event system specification (DEVS) [1–3] is a formalism which allows representing and simulating any system having a finite number of changes in a finite interval of time. In that way, systems modelled by petri nets, state charts,

event graphs, and even difference equations can be seen as particular cases of DEVS models.

The origin of the quantization based integration methods can be found in the definition of quantized systems and their representation in terms of DEVS models [4]. This idea was reformulated with the addition of hysteresis and it was formalized as a simulation method for ODEs in [5, 6] where the quantized state systems (QSS) were defined. Moreover, there exist some other approaches such as Backward QSS and Centered QSS [7–9]. But all the previous works using QSS to simulate continuous systems did not address the issue how to implement QSS in real-time distributed simulations such as high level architecture (HLA) and distributed interactive simulation (DIS). They did not mention the approach to extend QSS to time varying simulations either. As such, this paper proposes QSS based researches on both real-time and time varying simulation methods, where QSS must be extended for simulation. To accomplish this, we develop a way to keep the real-time synchronization in high level architecture (HLA) distributed simulation environment and propose a method to simulate time varying system.

The paper is structured as follows. In Section 2, key technologies of QSS are introduced and analyzed. Section 3 introduces the proposed distributed QSS framework to address real-time systems and a validation example is designed and results are shown and analyzed. In Section 4, a new method of time varying case is proposed and illustrated with a practical

example. Finally, conclusions are drawn and future works are discussed.

2. Quantized State Systems (QSS)

Consider a continuous system model:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \\ y(t) &= g(x(t), u(t)). \end{aligned} \quad (1)$$

$x(t)$ represents a set of state variables, $u(t)$ represents a set of input variables, $\dot{x}(t)$ represents a set of state derivative variables, $y(t)$ represents a set of output variables, $f(\cdot)$ represents the state transform function, and $g(\cdot)$ represents the output function.

To deduce quantization function, let $D = \{d_0, d_1, \dots, d_r\}$ be a set of real numbers, where $d_{i-1} < d_i$, $1 \leq i \leq r$. Let Ω be the set of piecewise continuous real valued trajectories and $x \in \Omega$ is a continuous trajectory. Let $b : \Omega \rightarrow \Omega$ be a mapping and let $q = b(x)$, where the trajectory q satisfies

$$q(t) = \begin{cases} d_{i+1}, & \text{if } x(t) = d_{i+1} \wedge q(t^-) = d_i \wedge i < r, \\ d_{i-1}, & \text{if } x(t) = d_i - \varepsilon \wedge q(t^-) = d_i \wedge i > 0, \\ d_i, & \text{otherwise.} \end{cases} \quad (2)$$

Then the map b is hysteretic quantization function [2]. $q(t)$ is a set of quantized state variables used to simulate $x(t)$ of (1), where the discrete values d_i are called quantization levels and the distance $\Delta d \triangleq d_{i+1} - d_i$ is defined as quantum, which is usually constant. The width of the hysteresis window is ε . The values d_0 and d_r are the lower and upper saturation values.

Replace $x(t)$ on the right side of (1) with $q(t)$. The approximated model is as the following equation shows:

$$\begin{aligned} \dot{x}(t) &= f(q(t), u(t)) \\ y(t) &= g(q(t), u(t)). \end{aligned} \quad (3)$$

Kofman and Junco [1] proved that if $u(t)$ has piecewise constant and bounded trajectories and $f(\cdot)$ is continuous and bounded, then $q(t)$, $x(t)$, $\dot{x}(t)$, and $y(t)$ have piecewise constant trajectories.

3. Real-Time Simulation of Time Invariant System in HLA

Although previous works [7–9] introduced hysteretic quantization function above, they did not address the issue how to fulfill time management in real-time distributed simulations such as DIS and HLA systems. To tackle this problem, we propose using RTFederate in distributed simulations such as HLA and DIS systems. This method is about real-time management and is generic to all distributed simulations. Therefore we will use HLA as our example system in this section.

3.1. RTFederate: A Time Control Member. As shown in Figure 1, RTFederate mainly has three components: Time

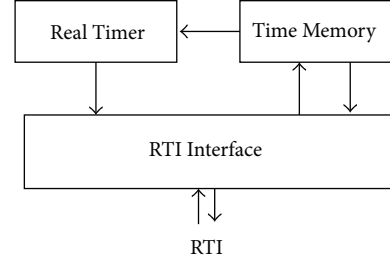


FIGURE 1: The structure of RTFederate module.

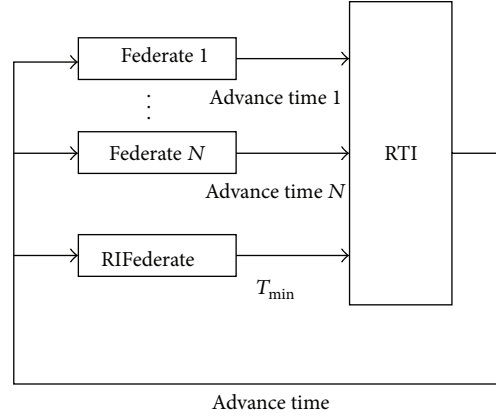


FIGURE 2: The procedure of time advance.

Memory, Real Timer and RTI Interface. They are designed to ensure that the time scale in simulation is identical with that of real world time. RTFederate would be an independent federate in HLA, and its time management strategy is set as both Constrained and Regulation.

For the component of Time Memory, it needs to set a minimum time step denoted as T_{\min} ; it would be a positive integer.

RTFederate module would get the allowed advance time from RTI through RTI Interface and then add it to the Time Memory. If the value T in Time Memory is smaller than T_{\min} , Time Memory would request time advance with T_{\min} through RTI Interface. Otherwise, Time Memory would transfer the value T to Real Timer and then reset the value T to zero. Real Timer would count $\lfloor T \rfloor$ of unit time. $\lfloor T \rfloor$ means the largest integer which is smaller than T . At the end of the count, Real Timer would request time advance with T_{\min} through RTI Interface.

3.2. Real-Time Simulation. Each of the subsystems would be discretized with QSS method and be treated as a federate in HLA. The time management mode of all the federates should be set as “both Constrained and Regulating”. Also, RTFederate module needs to be set with the same mode. In HLA, Runtime Infrastructure (RTI) software would allow time advance only if all of the federates request to advance and the least requested time step would be allowed. Figure 2 shows the procedure of time advance.

Only if RTFederate especially requests to advance, RTI would allow its time advancement. So the simulation speed would be controlled by RTFederate module and then it would support real-time simulation.

3.3. *Example of Time Invariant System.* Consider a vertical control system of a plane as the following equation shows:

$$\begin{bmatrix} \frac{d\Delta\bar{v}}{dt} \\ \frac{d\Delta\alpha}{dt} \\ \frac{d\Delta\theta}{dt} \\ \frac{d\Delta q}{dt} \end{bmatrix} = \begin{bmatrix} -X_v & -X_\alpha & -X_\theta & 0 \\ -Z_v & -Z_\alpha & 0 & 1 \\ 0 & 0 & 0 & 1 \\ -M_v + M_{\dot{\alpha}}Z_v & -M_\alpha + M_{\dot{\alpha}}Z_\alpha & 0 & -M_q - M_{\dot{\alpha}} \end{bmatrix} \cdot \begin{bmatrix} \Delta\bar{v} \\ \Delta\alpha \\ \Delta\theta \\ \Delta q \end{bmatrix} + \begin{bmatrix} -X_{\delta_T} & 0 \\ 0 & -Z_{\delta_e} \\ 0 & 0 \\ -M_{\delta_T} & -M_{\delta_e} + M_{\dot{\alpha}}Z_{\delta_e} \end{bmatrix} \cdot \begin{bmatrix} \Delta\delta_T \\ \Delta\delta_e \end{bmatrix}. \quad (4)$$

The symbols in the equation are listed below.

The Symbols in (4). The symbols are as follows: $\Delta\bar{v}$: change value of plane velocity, state variable; $\Delta\alpha$: change value of attack angle, state variable; $\Delta\theta$: change value of pitch angle, state variable; Δq : change value of pitch rate, state variable; X_v : derivative coefficient of x -axis force (body axis) caused by velocity; X_α : derivative coefficient of x -axis force (body axis) caused by attack angle; X_θ : derivative coefficient of x -axis force (body axis) caused by pitch angle; X_{δ_T} : derivative coefficient of x -axis force (body axis) caused by throttle control lever; Z_v : derivative coefficient of z -axis force (body axis) caused by velocity; Z_α : derivative coefficient of z -axis force (body axis) caused by attack angle; Z_{δ_e} : derivative coefficient of z -axis force (body axis) caused by the deflection of elevator; M_v : derivative coefficient of rolling moment caused by velocity; M_α : derivative coefficient of rolling moment caused by attack angle; $M_{\dot{\alpha}}$: derivative coefficient of rolling moment caused by changing rate of attack angle; M_q : the derivative coefficient of rolling moment caused by pitch rate; M_{δ_T} : the derivative coefficient of rolling moment caused by throttle control lever; M_{δ_e} : the derivative coefficient of rolling moment caused by the deflection of elevator; $\Delta\delta_e$: change of deflection angle of elevator; and $\Delta\delta_T$: changes of the position of thrust lever.

The system is discretized with QSS method, the quantum unit is $q = 0.001$ and so as the hysteresis, and the initial conditions are $\Delta\bar{v} = \Delta\alpha = \Delta\theta = \Delta q = 0$, $\Delta\delta_e = 2^\circ$, $\Delta\delta_T = 0^\circ$.

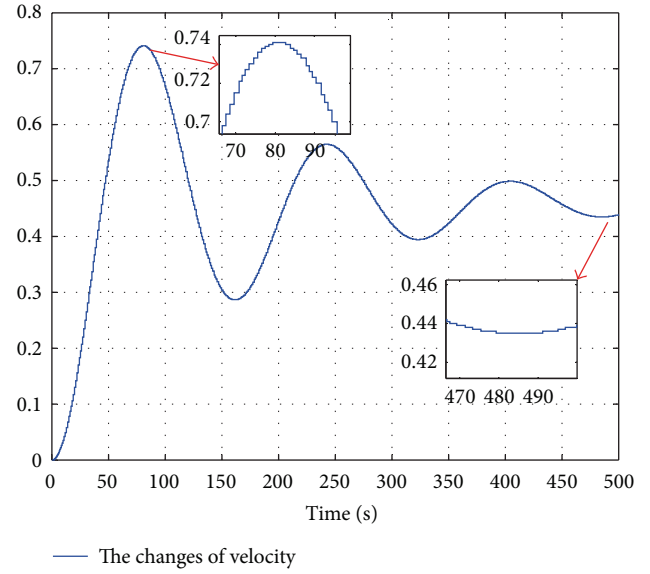


FIGURE 3: Curve of $\Delta\bar{v}$ in the simulation with QSS method.

TABLE 1: The step intervals around the first peak value.

Step width (s)	2.179262	1.010354	1.010355	...	1.09538	1.095208
$\Delta\bar{v}$	0.733	0.736	0.738	...	0.741	0.74

TABLE 2: The step intervals around the relative stable region.

Step width (s)	2.111777	3.908981	11.875489	...	1.871779	2.88458
$\Delta\bar{v}$	0.436	0.435	0.436	...	0.438	0.439

Figure 3 shows the response of $\Delta\bar{v}$, which is a long period transition process.

As seen from Figure 3, the step intervals became smaller when $\Delta\bar{v}$ gets close to peak values, where the values change severely. On the other hand, the step intervals get bigger when $\Delta\bar{v}$ reaches a relatively stable status, where the value does not change much.

Table 1 shows the step interval around the first peak value, and Table 2 shows the step intervals in relative stable status. The results indicate that the QSS method reduces the redundant calculation and improves the simulation efficiency.

To be simulated in distributed environment each of the state variables would be treated as a federate and in addition to RTFederate module it has five federates in total. Considering the procedure of time advance shown in Figure 4, we have the following time advance equations for these federates:

$$\text{Federate1: } d\Delta\bar{v}/dt = -X_v \cdot \Delta\bar{v} - X_\alpha \cdot \Delta\alpha - X_\theta \cdot \Delta\theta - X_{\delta_T} \cdot \Delta\delta_T;$$

$$\text{Federate2: } d\Delta\alpha/dt = -Z_v \cdot \Delta\bar{v} - Z_\alpha \cdot \Delta\alpha + \Delta q - Z_{\delta_e} \cdot \Delta\delta_e;$$

$$\text{Federate3: } d\Delta\theta/dt = \Delta q;$$

$$\text{Federate4: } d\Delta q/dt = (-M_v + M_{\dot{\alpha}}Z_v) \cdot \Delta \bar{v} + (-M_{\alpha} + M_{\dot{\alpha}}Z_{\alpha}) \cdot \Delta \alpha - (M_q + M_{\dot{\alpha}}) \cdot \Delta q - M_{\delta_T} \cdot \Delta \delta_T + (M_{\dot{\alpha}}Z_{\delta_e} - M_{\delta_e}) \cdot \Delta \delta_e.$$

4. QSS for Time Varying System

4.1. QSS Method for Time Varying System. In the previous section, a QSS method of time invariant system is proposed. However, in practical systems, there exist time varying state variables. And in this section, we proceed to propose a QSS approach of time varying system (TVS).

Consider the TVS as follows:

$$\dot{x} = A(t)x. \quad (5)$$

When $t = t_i$ ($i \geq 0$), (5) follows to be

$$\dot{x} = A(t_i)x. \quad (6)$$

At this point, state derivative is \dot{x}_i and simulation advances to t'_{i+1} . When $t = t'_{i+1}$, (5) can be rewritten as

$$\dot{x} = A(t'_{i+1})x \quad (7)$$

and the state derivative is \dot{x}_{i+1} . If \dot{x}_i and \dot{x}_{i+1} have the same signs, the state derivative of $[t_i, t'_{i+1}]$ is regarded as $(1/2)(\dot{x}_i + \dot{x}_{i+1})$ and simulation is advanced to t_{i+1} based on selected quantum of x . If \dot{x}_i and \dot{x}_{i+1} have different signs, it means the sign changes during $[t_i, t'_{i+1}]$; then advance time should be shortened half as $t''_{i+1} = 0.5 * (t + t'_{i+1})$:

$$\dot{x} = A(t''_{i+1})x. \quad (8)$$

Reconsider \dot{x}_{i+1} ; if \dot{x}_i and \dot{x}_{i+1} have the same signs, then state derivative of $[t_i, t''_{i+1}]$ is $(1/2)(\dot{x}_i + \dot{x}_{i+1})$ and simulation advances to t_{i+1} . Otherwise the advance time is shortened half again until the two state variables have the same signs.

4.2. Example of Time Varying System. Consider

$$\dot{x} = -2t(t+1)^{-2}x. \quad (9)$$

Equation (9) is an instance of TVS, with the initial conditions: $x_0 = 1$, $t_0 = 1$. At initial time t_0 , $\dot{x}_0 = -2 \times 1 \times (1+1)^{-2} \times 1 = -0.5$. The quantum of x is set to 0.001 and the advancing step is $0.001/(0.5) = 0.002$. In the first step, the simulation time advances to $t'_{i+1} = 1 + 0.002 = 1.002$ and $\dot{x}_1 = -2 \times 1 \times (1.002+1)^{-2} \times 1 = -0.499$. As \dot{x}_0 and \dot{x}_1 have the same signs, state derivative of $t \in [1, 1.002]$ is $\dot{x} = (1/2)(-0.5 - 0.499) = -0.4995$.

The analytical solution of (9) is

$$x = e^{-2(\ln(t+1) - \ln 2 + 1/(t+1)) - 0.5} \quad (10)$$

and Figure 5 shows the trajectories of (9) produced by QSS method and the analytical solution, respectively. As we can see, they are almost identical with each other, except the difference in step intervals. The QSS method is more computationally efficient.

The trajectory values at some time points are listed in Table 3. We can also get the conclusion that the result of our method corresponds to analytic curve with minor errors.

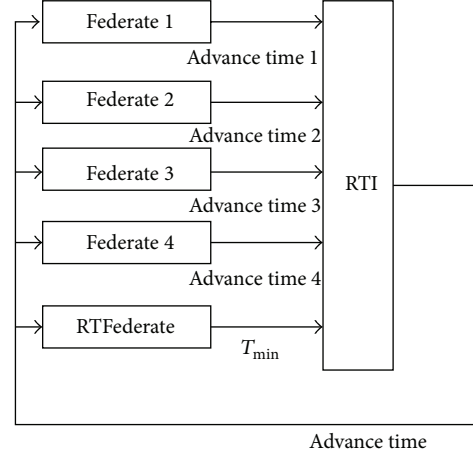


FIGURE 4: Procedure of time advance.

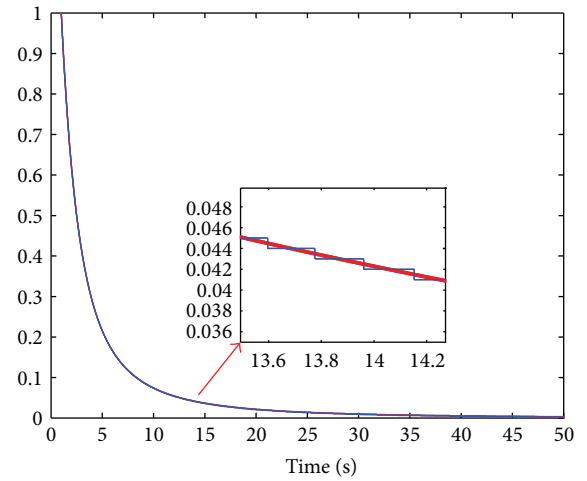


FIGURE 5: The solution trajectories of QSS method (the stair line) and the analytical solution (the line).

TABLE 3: Comparing QSS with analytic result.

Time (s)	1.001	1.003	1.005	...	1.017	1.019
QSS solution	1	0.999	0.998	...	0.992	0.991
Analytical solution	0.999500	0.998501	0.997503		0.991536	0.990545

5. Conclusions and Future Work

The paper discretizes the continuous system using QSS method which is based on the quantization of the state space to improve the simulation efficiency. QSS method transforms the continuous system to the discrete event model. The paper introduces both time invariant and time varying methods to support different system simulations. QSS method guarantees the stability and convergence under some conditions and it is not adapted to all the continuous system simulations yet. We need to extend the application range in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research in this paper was supported by Grants 61473013 and 61104057 from the Natural Science Foundation of China and funding of the Science and Technology on Complex Land Systems Simulation Laboratory (63963). The authors thank the reviewers for their comments. This work is based on one of our conference papers of AsiaSim 2012 [10]. The simulation section is extended to cover time varying systems and the results of time invariant system are further explained and corrected.

References

- [1] E. Kofman and S. Junco, "Quantized state systems. A DEVS approach for continuous system simulation," Tech. Rep. LSD0004, LSD, FCEIA, UNR, Transactions of SCS, 2000.
- [2] E. Kofman, "A third order discrete event method for continuous system simulation," *Latin American Applied Research*, vol. 36, no. 2, pp. 101–108, 2006.
- [3] B. P. Zeigler, T. G. Kim, and H. Praehofer, *Theory of Modeling and Simulation*, Academic Press, New York, NY, USA, 2nd edition, 2000.
- [4] B. P. Zeigler and J. S. Lee, "Theory of quantized systems: formal basis for DEVS/HLA distributed simulation environment," in *Enabling Technology for Simulation Science II*, vol. 3369 of *Proceedings of SPIE*, pp. 49–58, Orlando, Fla, USA, April 1998.
- [5] E. Kofman, "Discrete event simulation of hybrid systems," Tech. Rep. LSD0205, LSD, UNR, Transactions of SCS, 2002.
- [6] E. Kofman, "Discrete event simulation of hybrid systems," *SIAM Journal on Scientific Computing*, vol. 25, no. 5, pp. 1771–1797, 2004.
- [7] F. Bergero, E. Kofman, and F. Cellier, "A novel parallelization technique for DEVS simulation of continuous and hybrid systems," *Simulation*, vol. 89, no. 6, pp. 663–683, 2013.
- [8] E. Kofman, "Quantization-based simulation of differential algebraic equation systems," *Simulation*, vol. 79, no. 7, pp. 363–376, 2003.
- [9] E. Kofman, J. S. Lee, and B. P. Zeigler, "DEVS representation of differential equation systems: review of recent advances," in *Proceedings of the 13th European Simulation Symposium*, pp. 591–595, 2001.
- [10] W. Zhang and J. Wang, "Quantization based real-time simulation of continuous system in distributed environment," in *Proceedings of the 2012 Asia Simulation Conference*, pp. 363–369, Shanghai, China, October 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

