

High Performance Modeling with Quantized System

Jong Sik Lee

School of Computer Science and Engineering
Inha University
Incheon 402-751, South Korea
jslee@inha.ac.kr

Abstract. As analyses of system behavior and complexity through computer modeling and simulation have been growing, high performance modeling is noticed to handle behavior and complexity of modern large-scale systems. The high performance modeling focuses on high resolution representation of system and performance improvement. This paper presents a quantized system modeling as an effective high performance modeling. The quantized system modeling represents state resolution using state quantization and improves system performance through message filtering among system components. This paper realizes a practical quantization component which is based on both of Discrete Time System (DTS) and Discrete Event System (DES) and shows usefulness of quantized system in a variety of industrial applications. This paper models a real-world application, e.g. a space traveling system with the quantized system and evaluates system performance, accuracy, and scalability of the quantized system on DES-based modeling and simulation.

1 Introduction

As system behavior and complexity analyzed by computer has been increasing, high performance modeling [1, 2, 3, 4, 5] is demanded to deal with behavior and complexity of large-scale systems. For high performance modeling, high-resolution and large-scale representations of system is needed to handle behavior of large-scale modern system. This paper presents a quantization-based system modeling of a complex and large-scale system to support high-resolution and large-scale representations. The quantized system modeling [6, 7, 8, 9, 10] is based on a quantization of state of system and provides a high performance simulation.

This paper reviews a discrete event-based system modeling specification, which is called DEVS (Discrete Event System Specification) [11, 12] and specifies an existing DTS in sense of strong representation of DEVS. For a realization of the quantized system modeling and DEVS, we present a quantized DEVS integrator [11] which provides behaviors and characteristics of discrete event-based system. To validate a fidelity of the quantized system modeling and DEVS, a kinetics of a spaceship is taken an application. The kinetics maintains an accounting of where ships are and predicting their future destinations. We describe a workable representation of the DTSS (Discrete Time System Specification) [11] formalism in DEVS. For performance evaluation, we model the kinetics on both of DTSS and DEVS formalisms and environments and compare system performance each other. Section 2

describes quantization approach and show how the quantization approach can be applied to improve system performance. Section 3 presents a quantized system and realizes a quantized DEVS integrator. Section 4 presents a modeling of a kinetics of spaceship. Section 5 discusses experiment and performance evaluation. Section 6 is conclusion.

2 Quantization and Performance

Quantization is based on the quantization theory [6, 7] with modeling formalism and system homomorphisms. A continuous trajectory with a finite number of values in a finite time interval is approximated. In order to obtain a discrete time system approximation, discretization of the time base is needed with a finite time interval. The finite number of values is then calculated from the partition of the trajectory into a finite number of segments (each of which has a finite computation). The partition of the trajectory with the finite number of values provides a way to quantize the value space, which is partitioned in every D interval (quantum), and the time space is partitioned in every T interval (time interval).

In discrete event systems, we sample the time values at every quantum interval (D), use discrete values with continuous time, and send the quantum levels out after the sampled time interval. This is called the quantization based on D . In a real application, the state trajectory is represented by the crossings of an equally spaced set of boundaries separated by D . Using quantization, we checks a threshold crossing of output value of a sender whenever an output event occurs and sends the output value to a receiver only when the threshold crossing occurs. The effect of quantization is to reduce the number of messages exchanged between sender and receiver. We can expect to save the communication data and the computation of the receiver from the reduced messages through the message reduction. Considered with scalability of a system, the quantization increases system performance in various ways such as decreasing overall execution time or allowing a larger number of entities to be performed.

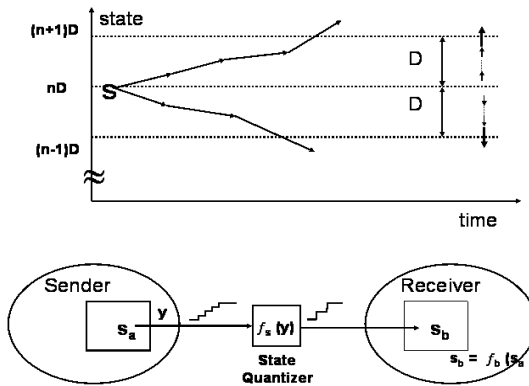


Fig. 1. Quantization and Performance Improvement through Message Traffic Reduction

This paper introduces an actually realized quantization approach to improve system performance through message traffic reduction among system components. As Fig. 1 illustrates, the quantization approach applies when a sender component is updating a receiver component on a numerical, real-valued, state variable, which is a dynamically changing attribute. A quantizer is applied to the sender's output, which checks for threshold (boundary) crossings whenever a change in the variable occurs. Only when such a crossing occurs, a new value of the variable is sent across the network to the receiver. The quantization reduces the number of messages sent and incurs some local computation at the sender.

3 Quantization Component

A quantized integrator [11], which is a quantization component, basically performs a linear extrapolation. The time to the next boundary crossing is the quantum size divided by the input (derivative). The boundary is predicted either to be one up or one down according to the sign of the derivative. When an input event is received, the state is updated using the old input before recalculating the predicted crossing, which provides an important correction for error reduction. A quantized integrator accepts DEVS input segments and produces quantized output.

If we are on a boundary, the time advance computation merely divides quantum interval (D) by the current input x (the derivative or slope). If we reach the upper boundary $(n+1)D$ or lower boundary $(n-1)D$, we output and update the state accordingly. As long as the input remains the same, the time to cross the successive boundaries ($(n+1)D$ or $(n-1)D$) will be the same. When a new input is received, we update the state using the old input and the elapsed time. From this new state (q), the new time to reach either the upper or lower boundary is computed.

Comparison of Time Trajectory of Quantized Integrator : (Discrete Event System vs. Discrete Time System)

The DEVS stores a state of a system and its last input by definition of DEVS, $M = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta)$. A DTSS system can be strongly represented by the DEVS with four functions, $ta(q,x)$, $\lambda(q,x)$, $\delta_{int}(q,x)$, and $\delta_{ext}((q,x), e, x')$.

The *time advance* is the time to the next system output.

$$ta(q,x) = \min(t | (q, x_{t'}) \neq \phi)$$

The output of the DEVS at the next internal event is the corresponding system output

$$\lambda(q,x) = \lambda(q, x_{ta(q,x)})$$

Unless there is an external input, the DEVS will update its state to the state of the system at the next output.

$$\delta_{int}(q,x) = \delta_{int}(q, x_{ta(q,x)})$$

If there is an external input after an elapsed time, e , the DEVS will immediately update its state to the corresponding system state and will also store the new input.

$$\delta_{ext}((q,x), e, x') = \delta_{ext}((q, x_{e'}), e, x')$$

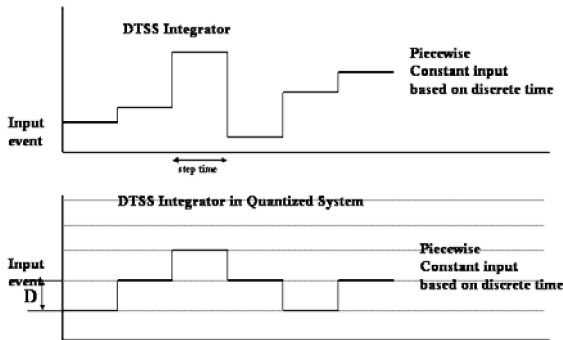


Fig. 2. Input Time Trajectory (DTSS integrator vs. Quantized DTSS Integrator)

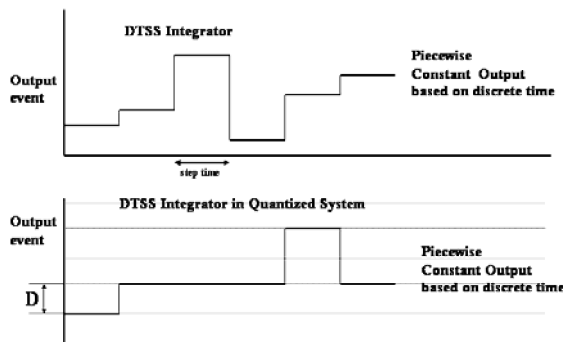


Fig. 3. Output Time Trajectory (DTSS integrator vs. Quantized DTSS Integrator)

Fig. 2 and Fig.3 compare input and output trajectories between of DTSS integrator and Quantized DTSS integrator and show the difference between two integrators that are mentioned with representation of each formalism previously. The integrator designed to fit at DTSS Simulation Environment, after every step time, puts output event and gets input event basically. While the quantized integrator can put output event when output value crosses the boundary of quantum based partition block. And this integrator can get input event when input event occurs for this quantized integrator. Input value is quantized by an input quantizer.

Fig. 4 and Fig. 5 compare input and output trajectories between of DTSS integrator and Quantized DEVS integrator and show the difference between two integrators that are mentioned with representation of each formalism previously. The integrator of DTSS Simulation Environment, after every step time, puts output event and gets input event basically. While the quantized DEVS integrator can put output event at the time from time advance function, $ta()$. Time from time advance function is when state of system crosses the boundary of quantum based partition block of state. That means that the crossing of the partition block boundaries are implemented as state events. So, time from time advance function depends on a quantum, current input, and current state. And this quantized DEVS integrator can get input event when input event occurs for this integrator.

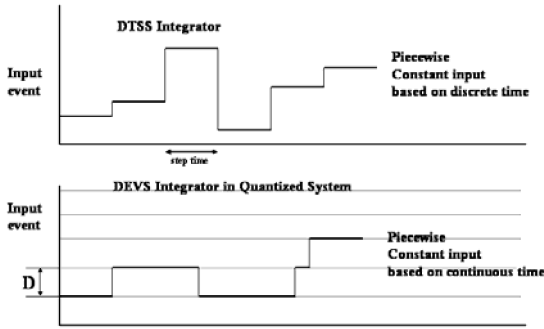


Fig. 4. Input Time Trajectory (DTSS integrator vs. Quantized DEVS Integrator)

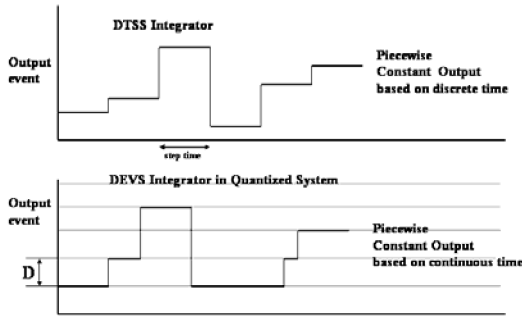


Fig. 5. Output Time Trajectory (DTSS integrator vs. Quantized DEVS Integrator)

4 Kinetics of Spaceship

This section presents a kinetics part of spaceship as an application to evaluate performance of quantized system. We develop a spaceship model as a quantized system. From this model, we construct an abstraction that is for maintaining an accounting of where ships are and predicting their future destinations. Thus our overall modeling objectives are to construct a space travel scheduling and test it. The modeling is based on the differential equations which are based on Newtonian mechanics [13, 14].

4.1 Circulation on an Ideal Circle

Circulation of spaceship on an ideal circle is a part of the kinetics. In order to maintain an ideal circular orbit with radius D and speed v around a massive body, a spaceship is required that a centripetal force, mv^2/d , which equals to the force of gravity. The force of gravity pulls along the line joining the two centers and has magnitude $F = GMm/d^2$, where G is the gravitational constant, M and m are the masses. The distance of a ship with center at (x,y) to the center of gravity of a massive body (x_0,y_0) is $d = ((x - x_0)^2 + (y - y_0)^2)^{1/2}$. The force is projected in the x and y

directions in proportions, $p_x = x/d$ and $p_y = y/d$, respectively. In a ideal orbit with $d = D$ (constant), the coordinate dynamics separate into two independent 2nd order linear oscillators.

Basically, frequency $\omega = (GM/d^3)^{1/2}$ would be to maintain to circulate, however we use a gain value instead of frequency. The gain controls a degree of movement of spaceship. As the gain changes, system performance will be measured and compared since the gain decides stability, accuracy, and execution time. For spaceship traveling, one of the strong influences is gravity. Gravity is a force exerted on a body by all other bodies in relation to their distance away. The center of gravity allows us to aggregate particles in a rigid body into a single point that represents their gravitational interaction with any other body to consider the forces acting at the centers of gravity of interacting bodies.

5 Experiment and Performance Evaluation

We develop a kinetics model of spaceship. The kinetics model has a total of four integrators and is developed on DEVSJAVA modeling and simulation environment [6, 7]. We develop two different systems: Quantized DTSS and Quantized DEVS. Quantized DTSS system includes quantized DTSS integrators and Quantized DEVS system includes quantized DEVS integrators. Fig. 6, Fig. 7, and Fig. 8 compare system accuracy and performance between quantized DTSS and quantized DEVS systems with three performance measures: average of error, number of message passing, and system execution time. As shown in Fig. 6, average of error is increasing apparently as quantum size, D , is increasing. The quantized DEVS system shows the more accuracy than that of the quantized DTSS since DEVS is basically based on a continuous time modeling, thus the error from DEVS is not accumulated. Meanwhile, DEVS is basically based on a discrete time modeling.

The number of message passing and system execution time measure system performance improvement through execution cost reduction. In both of quantized DTSS and quantized DEVS, as quantum size, D , is increasing, number of message passing and system execution time are decreasing apparently in Fig. 7 and Fig. 8. However, the average of error is increasing in Fig. 6. There exists a tradeoff between execution cost reduction and error increment. We should control the quantum size, D , and reduce execution cost within a tolerable error. In comparison between quantized DTSS and quantized DEVS, the quantized DEVS apparently reduces number of message passing and system execution time with the smaller error.

6 Conclusion

This paper presents a quantized system with DTSS and DEVS representations. The quantized system reduces an amount of computation in a complex and large-scale systems, thus reduces a total execution time. Especially, the quantized system apparently reduces the number of message passing among components. That means that the quantized system reduces data transmission requirement, naturally. The quantized system is able to provide high performance modeling in distributed system

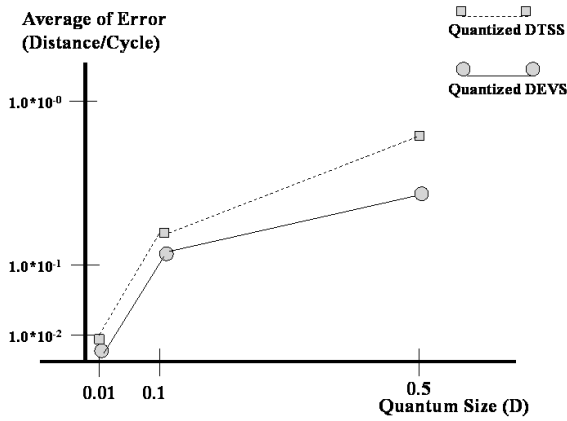


Fig. 6. Average of Error (Quantized DTSS System vs. Quantized DEVS system)

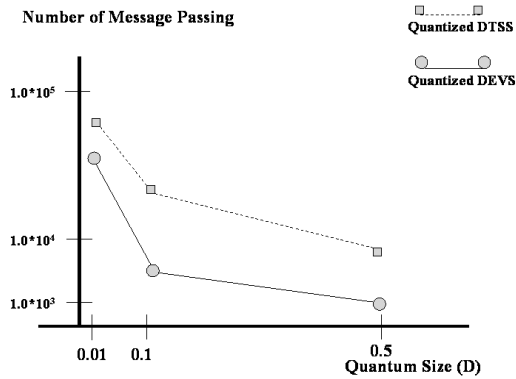


Fig. 7. Number of Message Passing (Quantized DTSS System vs. Quantized DEVS system)

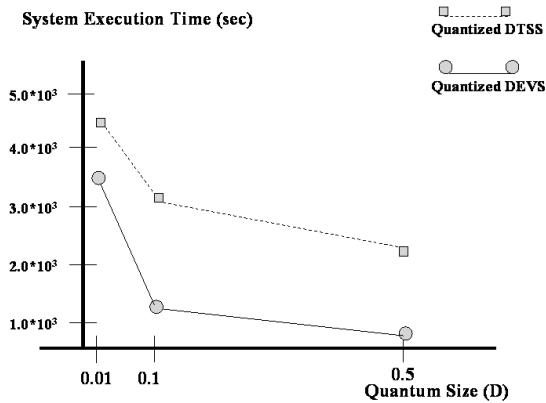


Fig. 8. System Execution Time (Quantized DTSS System vs. Quantized DEVS system)

by reducing data transmission requirement among distributed components. The execution of large-scale distributed system is achieved with high performance modeling in limited communication and computing resources. To realize a quantized system, this paper suggests two types of quantized integrators: quantized DTSS and quantized DEVS. In addition, this paper represents the DTSS formalism in a strong sense of the DEVS, thus this representation shows both continuous and discrete processes can be modeled and executed by the DEVS. Quantized DTSS system is developed with DTSS formalism by using quantized DTSS integrators. Quantized DEVS system is developed with DEVS formalism by using quantized DEVS integrators. The empirical result from the Quantized DTSS and DEVS system shows system performance improvement with a tradeoff from system accuracy. With this limitation, the quantized system should be applied within a tolerable error.

Reference

1. Averill M. Law, and W. David Kelton: *Simulation Modeling and Analysis*. McGraw-Hill. Inc. (1982)
2. Bernard. P. Zeigler and D. Kim: *Design of High Level Modelling / High Performance Simulation Environments*. 10th Workshop on Parallel and Distributed Simulation. Philadelphia (1996)
3. Yoonkeon Moon: *High Performance Simulation Based Optimization Environment: Modeling Spatially Distributed Large Scale Ecosystems*. Ph.D. Dissertation. The University of Arizona (1996)
4. Bernard P. Zeigler, Y. Moon, D. Kim, and J.G. Kim: *C++ DEVS: A High Performance Modeling and Simulation Environment*. 29th Hawaii International Conference on System Sciences (1996)
5. Zeigler, B.P., et al.: *The DEVS Environment for High-Performance Modeling and Simulation*. IEEE C S & E (1997) 61–71
6. Zeigler, B.P. and J.S. Lee: *Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment*. Enabling Technology for Simulation Science(II). SPIE AeoroSense 98. Orlando FL (1998)
7. Zeigler, B.P.: *DEVS Theory of Quantization*. DARPA Contract N6133997K-0007: ECE Dept. UA Tucson AZ (1998)
8. Ernesto Kofman, Sergio Junco: *Quantized-State Systems, a DEVS Approach for Continuous System Simulation*. Transactions of SCS (2001)
9. Bernard. P. Zeigler, H. Sarjoughian, and H. Praehofer: *Theory of Quantized Systems: DEVS Simulation of Perceiving Agents*. J. Sys. & Cyber, Vol. 16, No. 1 (2000)
10. G. Wainer, and B.P. Zeigler: *Experimental Results of Timed Cell-DEVS Quantization, AI and Simulation*. AIS 2000. Tucson AZ
11. Zeigler, B.P., T.G. Kim, and H. Praehofer: *Theory of Modeling and Simulation*. 2ed. New York NY: Academic Press (2000)
12. Zeigler, B.P., et al.: *DEVS Framework for Modeling, Simulation, Analysis, and Design of Hybrid Systems: Hybrid II*, Lecture Notes in CS, P. Antsaklis and A. Nerode, Editors. Springer-Verlag, Berlin (1996) 529–551
13. Roger R. Bate, Donald D. Mueller, Jerry E. White: *Fundamentals of Astrodynamics*. Dover Publications. New York (1971)
14. Erwin Kreyszig: *Advanced Engineering Mathematics: Seventh Edition*. John Wiley & Sons Inc. New York (1993)