Quantized System Modeling and Performance Evaluation

Jong Sik Lee
School of Computer Science and Engineering
Inha University
jslee@inha.ac.kr

Keywords:
Quantization-based message-filtering, DEVS-based integrator, State update transmission, DEVSJAVA simulation

This paper reviews existing message-filtering schemes andpresents a quantization-based message-filtering approach which reduces state update transmission and message traffic requirement. For a realization of the approach, we develop a DEVS-based integrator which provides behavior and characteristic of the quantization-based approach. We take a spaceship and space traveling system as a case study to evaluate performance of the quantization-based approach. The approach is validated by DEVSJAVA simulations of the case study. The comparison of message traffic requirement between DTSS (Discrete Time System Specification)-based and DEVS-based systems apparently shows system performance improvement through the quantization-based approach.

1 Introduction

As system behavior and complexityanalyzed by computer has been increasing, high performance computing is demanded to deal with behavior and complexity of large-scale systems. For high performance computing, 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 [1, 2] is based on a quantization of state of system and provides a high performance simulation. To evaluate the quantized system, a kinetics of a spaceship is taken an application for modeling and simulation. The kinetics maintains an accounting of where ships are and predicting their future destinations. In addition, we describe a workable representation of the DTSS (Discrete Time System Specification) formalism in DEVS (Discrete Event System Specification) [3, 4, 5]. 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 DEVS Interfaces. 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. Background: DEVS and DEVS Interfaces

Strong Representation

The DEVS formalism can represent a formalism at the coupled system level if for every coupled model specifiable in the formalism, there is a DEVS coupled model, which simulates it with a coupled system morphism. We'll call this *strong, local* or *component-wise* simulation, since we don't have to have solved the model, i.e., reduced its level of specification to get a DEVS simulation. Instead, we convert the components of the model individually to a DEVS representation and then run the resulting DEVS coupled model to generate the original behavior.

Weak Representation

Contrast to strong representation, *weak* or *global*representation in which we only know how to create a morphism at the I/O system level for any system in a formalism but not at the coupled system level. Then, any coupled model has to be simulated or solved first before being translated to a DEVS equivalent. This is rather impractical for complex coupled models. Nevertheless, it is useful to know whether a DEVS can represent a formalism at the lower levels since such information is needed for universality and uniqueness proofs.

DEVS-like system

The system with DEVS Interface is called as DEVS-like system [3]. DEVS-like system is DEVS system in a very strong sense. Every DEVS-like system can homomorphically be simulated by a DEVS I/O System by Universality Theorem for Systems with DEVS Interfaces. The DEVS universality theorem for DEVS-like systems states that all such systems are strongly simulated by DEVS.

When $S = (T, X, , Y, Q, ,)$ is supposed to be a DEVS-like system, where is a set of DEVS segments and the output segments are also DEVS segments, we can take $S = Q \, x \, X$, i.e., the DEVS stores the state of the system and its last input by definition of DEVS, $M = (X, Y, S, ext, int,, ta)$. Therefore this system can be strongly simulated by DEVS with four functions, $ta(q,x)$, (q,x),$int(q,x)$, and $ext((q,x), e ,x')$. For example, DTSS (Discrete Time System Specification) and DESS (Differential Equation System Specification) are DEVS-like systems.

3. Quantized System

To simulate systems with a digital computer requires that we approximate a continuous trajectory with a finite number of values in a finite time interval. One way, is to discretize the time base to obtain a DTSS approximation. Alternatively, rather than discretization ofthe time base we may partition the trajectory into a finite number of segments each of which has a finite computation associated with it. One way to do this is to quantize the value space. Quantization is a very general concept applicable to any value set, X. we can assign a partition x on X with a finite number of partition blocks (equivalence classes).

This paper suggests a Quantized DEVS integrator component which realizes a quantization concept. We represent four functions of DEVS formalism to construct an integrator in Quantized DEVS.If there is an external event after an elapsed time, e, the external transition function will immediately update its state using the current input, and store the new input from the input quantizer.

$$\delta ext((q,x,n),e,x') = (q+x*e,x',n)$$

Unless there is an external input, the internal transition function will update its state after the next output.

$$\delta int(q,x,n) = (n + D*sign(x), x, n + sign(x))$$

This function occurs when input and output come to the system at same time. In first, the external transition function operates and the internal transition function is second.

$$\delta con((q,x,n), x') = \delta int((q,x,n), ta(q,x,n), x')$$

The output is the DEVS at the next internal event is the quantized of output of state at that time.

$$\lambda(q,x) = n + D*sign(x)$$

Time advance function vary whether the state may be inside a block (hence not a multiple of D) or not. The external transition handles in case the state would be inside a block (hence not a multiple of D).

$$\begin{aligned} ta(q,x,n) &= |((n+1)D - q)/x| \text{ if } x > 0 \text{ and } (n+1)D - q > 0 \\ &= |(q - nD)/x| \text{ if } x < 0 \text{ and if } |q-nD| > 0 \\ &= \text{otherwise (i.e., } x = 0) \end{aligned}$$

While the internal transition handles in case the state would be on a boundary.

$$ta(q,x,n) = |D/x| \text{ if } x \,!\!= 0$$

When a new input is received, we update the state using the old input and the elapsed time in external transition function. From this new state, q, the new time to reach either the upper or lower boundary is computed. Even if we start on a boundary, the state may eventually be inside a block (hence not a multiple of quantum size D) as a result of an external transition. If a state ison a boundary, the time advance computation merely divides D by the current input x which is the derivative in internal transition function. If we reach the upper boundary (n+1) or lower boundary (nl) we output and update the state accordingly. While the input remains the same, the time to cross successive boundaries will be the same.

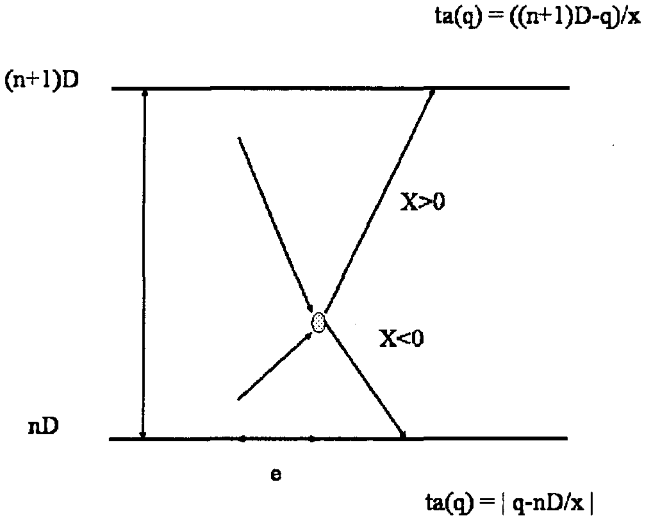

Figure 1 Time advance function, when state is inside a block, in external transition function

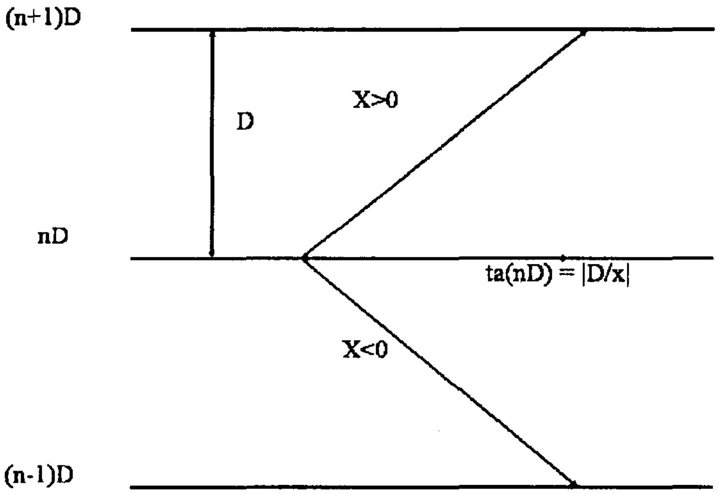

Figure 2 Time advance function, when state is on boundary, in internal transition function

Figure 3 compares a time trajectoryof 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, 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.

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.

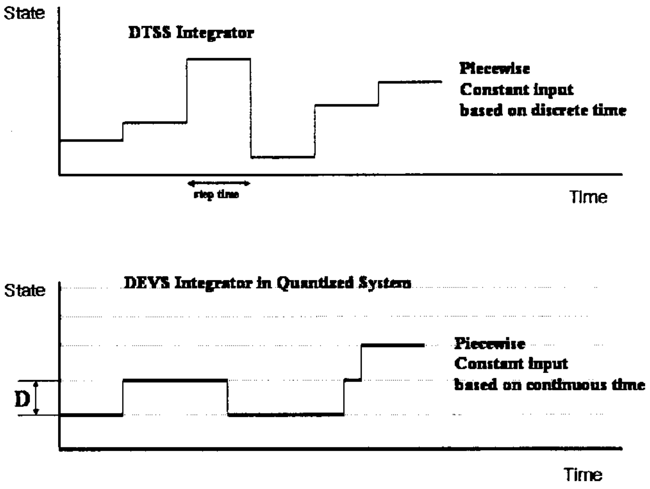

Figure 3 Time trajectory between of DTSS integrator and Quantized DEVS integrator

4. Application: Kinetics of spaceship

We take a kinetics part of spaceship as an application to evaluate performance of quantized system. Spaceships are traveling from earth to other planets and moon. We develop a spaceship model as a quantized system. From this model, we construct an abstraction that is for maintaining an accounting of whereships are and predicting their future destinations. Thus our overall modeling objectives are to construct a space travel scheduling and test it. The modelingis based on the differential equations which are based on Newtonian mechanics. Figure 4 gives us information about the movement of spaceship and about how to know where ships are and predict their future destinations with several parameters.

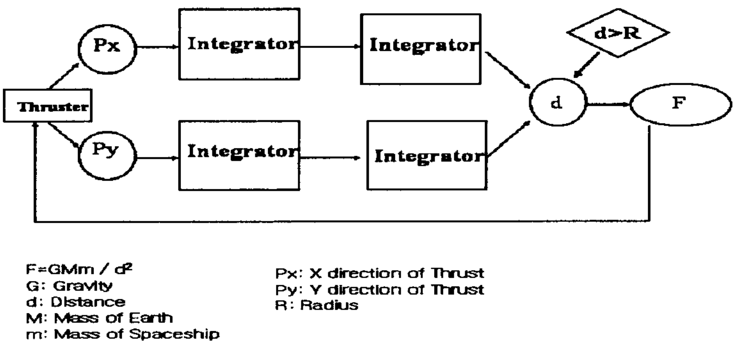

Figure 4 Modeling for kinetics of spaceship

The center of gravity coordinates, mass, M, and escape radius, Rare parameters that characterize earth component. When the distance from the center of gravity exceeds the threshold, R, a state event is generated which sends the ship to another model component.

5. Experiment and Performance Evaluation

We develop a kinetics model of spaceship as shown Figure 4. The kinetics model has a total of four integratorsand is developed on DEVSJAVA modeling and simulation environment. 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. Table 1 compares the performance of quantized DTSS and quantized DEVS systems with three performance measures: number of passed messages, system execution time, and average of error.The number of passed messages and system execution time measures performance improvement through execution cost reduction. The average of error measures system accuracy. In both of quantized DTSS and quantized DEVS, as quantum size,D, is increasing, number of passed messages and system execution time are decreasing apparently. However, the average error is increasing. 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 passed messages and system execution time in the same error of the quantized DTSS.

Model	Average of Error	Number of message passing	System Execution time (second)
Quantized DTSS (D= 0.01)	0.00381394	8769	40916
Quantized DTSS (D= 0.1)	0.0635957	610	29271
Quantized DTSS (D= 0.5)	0.36135	121	23901
Quantized DEVS (D= 0.01)	0.00323472	6489	11237
Quantized DEVS (D= 0.1)	0.0691659	656	10186
Quantized DEVS (D= 0.5)	0.328735	137	9930

Table 1 System performance of Quantized DTSS and Quantized DEVS systems (D: Quantum Size)

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 the total execution time. Especially, the quantized system apparently reduces the number of massage passing among components. That means that the quantized system reduces data transmission requirement, naturally. The quantized system is able to provide high performance computing in distributed system by reducing data transmission requirement among distributed components. The execution of large scale distributed system is achieved with high performance computing in limited communication and computing resources. This paper suggests a quatization-based integrator to realize a quantized system. We develop two types of integrators: quantized DTSS and quantized 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 performance improvement with a tradeoff from system accuracy. With this limitation, the quantized system should be applied within a tolerable error.

Reference

1. Zeigler, B.P. and J.S. Lee. *Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment.* in *Enabling Technology for Simulation Science(II), SPIE AeoroSense 98.* 1998. Orlando, FL

2. Zeigler, B.P., *DEVS Theory of Quantization*, . 1998, DARPA Contract N6133997K-0007: ECE Dept., UA, Tucson, AZ.

3. Zeigler, B.P., T.G. Kim, and H. Praehofer, *Theory of Modeling and Simulation.* 2 ed. 1998, New York, NY: Academic Press

4. Zeigler, B.P., *et al., The DEVS Environment for High-Performance Modeling and Simulation.* IEEE C S & E, 1997. 4(3): p. 61-71.

5. Zeigler, B.P., *et al., DEVS Framework for Modelling, Simulation, Analysis, and Design of Hybrid Systems*, in *Hybrid II, Leture Notes in CS*, P. Antsaklis and A. Nerode, Editors. 1996, Springer-Verlag: Berlin. p. 529-551.