Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

A METHOD FOR BOUNDING ERROR IN MULTI-RATE AND FEDERATED SIMULATIONS

James Nutaro
Computational Science and Engineering Division
Oak Ridge National Laboratory
One Bethel Valley Road
Oak Ridge, TN 37831, USA

ABSTRACT

This article presents a method for bounding errors that arise from interactions between components in a variety of simulation contexts. The proposed method combines key elements of the quantized state technique for numerical integration and the generalized discrete event system specification. Specifically, this method quantizes a model's output variables while allowing its internal variables to evolve by any suitable technique. This approach bounds the global error in proportion to the quantization threshold for simulations of networks of stable, linear systems. The proposed technique is particularly suitable for combining existing simulation models into federated, multi-rate simulations.

1 INTRODUCTION

Here we combine key elements of the quantized state system technique for numerical integration (Kofman and Junco 2001, Kofman 2002, Kofman 2006, Zeigler et al. 2000) and the generalized discrete event system specification (Giambiasi, Escude, and Ghosh 2000) to produce a robust, general approach for reusing continuous system simulation software in a variety of contexts. The proposed method satisfies the original goals of the quantized state system approach and generalized discrete event system simulation method to bring continuous system models into a discrete event simulation. It goes beyond this to support the construction of federated simulations, multi-rate simulations, and other circumstances where separately developed models need to be combined in a way that bounds errors while permitting loose coupling.

The central idea of the new method is that global error can be bounded if continuous components exchange data when their output variables meet specific threshold levels. This idea itself is not new, and the case where quantization is applied to state variables is examined by Kofman and Junco (2001) and Zeigler, Sarjoughian, and Praehofer (2000) in the context of numerical integration. We extend this idea along the line proposed by Giambiasi, Escude, and Ghosh (2000) in their generalized discrete event system specification to allow any method for evolving the model's internal state variables and placing the threshold requirement only on its output variables.

The approach is illustrated in Fig. 1 for two components connected in a feedback loop. Component A has a single output variable on which B depends and vice versa. Thresholds for interaction are placed in the range spaces of the output variables. The component A sends new data to B when its output variable takes on a value nq, where n is an integer. In the other direction, B sends new data to A under the same condition. Between these events, the components' internal states evolve separately.

In many cases, this technique can be applied to already existing simulation software with only small modifications. In particular, discontinuity locking (see, e.g., Cellier and Kofman (2006)), which is a general solution to the numerical problem of detecting threshold crossings and localizing them in time, is a method conveniently present in almost all continuous system simulation packages. In cases where this is not practical, we show how the sampling rate for a simulation model can be selected such that it satisfies the output threshold requirement.

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

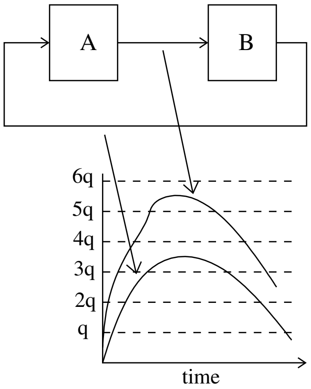

Figure 1: Two simulation components interacting via quantized output variables. New data for the continuous output variable is sent when it changes by a quantity q.

2 QUANTIZED INTERACTIONS

Consider a set of linear systems $1,...,m$. Each system has state vector $\mathbf{x}_i$; output vector $\mathbf{y}_i$; state transition matrix $\mathbf{A}_i$; output matrix $\mathbf{C}_i$; and input matrices $\mathbf{B}_{i,1},...,\mathbf{B}_{i,m}$ where $\mathbf{B}_{i,j}$ multiplies $\mathbf{y}_j$ produced by system j. We will also be considering a vector $\mathbf{f}_i(\mathbf{y}_i)$ of quantization functions for each system i and corresponding quantization error vector ε_i. An elements within a vector is indexed by its system followed by its position within the vector. For example, $y_{i,j}$ is the jth element in the output vector $\mathbf{y}_i$ of the ith system; $f_{i,j}$ is the jth quantization function in the quantization vector $\mathbf{f}_i$ of the ith system.

To simplify the initial analysis, assume that the simulation models for these systems evolve the state vectors without error. Furthermore, assume the subsystems are Moore machines so that the components can be coupled without requiring the solution of a fixed point problem. In the ideal case, the output vectors $\mathbf{y}_i$ are communicated continuously and each component model takes the form

$$\dot{\mathbf{x}}_i = \mathbf{A}_i\mathbf{x}_i + \begin{bmatrix}\mathbf{B}_{i,1} & \dots & \mathbf{B}_{i,m}\end{bmatrix}\begin{bmatrix}\mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m\end{bmatrix} \tag{1}$$

$$\mathbf{y}_i = \mathbf{C}_i\mathbf{x}_i \ . \tag{2}$$

In the simulation, each output variable $y_{i,j}$ in $\mathbf{y}_i$ is quantized by a function $f_{i,j}$. For each $f_{i,j}$ there is a constant $q_{i,j}$ such that

$$|f_{i,j}(y_{i,j}) - y_{i,j}| \le q_{i,j} \ . \tag{3}$$

An example of such a function is

$$f_{i,j}(y_{i,j}) = q_{i,j}\left\lfloor \frac{y_{i,j}}{q_{i,j}} \right\rfloor \ . \tag{4}$$

All simulations report a new value of $y_{i,j}(t)$ at each instant of time that $f_{i,j}(y_{i,j}(t))$ changes its value. To be compact we write the reported output vector as $\mathbf{f}_i(\mathbf{y}_i)$.

With this change, the simulation will not follow the exact solutions $\mathbf{x}_i$ and $\mathbf{y}_i$. Instead, the simulation will produce solutions $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ that evolve according to

$$\dot{\tilde{\mathbf{x}}}_i = \mathbf{A}_i\tilde{\mathbf{x}}_i + \begin{bmatrix}\mathbf{B}_{i,1} & \dots & \mathbf{B}_{i,m}\end{bmatrix}\begin{bmatrix}\mathbf{f}_i(\tilde{\mathbf{y}}_1) \\ \vdots \\ \mathbf{f}_m(\tilde{\mathbf{y}}_m)\end{bmatrix} \tag{5}$$

$$\tilde{\mathbf{y}}_i = \mathbf{C}\tilde{\mathbf{x}}_i \ . \tag{6}$$

We initialize the simulation so that $\tilde{\mathbf{x}}_i(0) = \mathbf{x}_i(0)$, but otherwise the true and computed solutions will differ. This difference is in the form of a state error $\mathbf{x}_i - \tilde{\mathbf{x}}_i$ and an output error $\mathbf{y}_i - \tilde{\mathbf{y}}_i$, and these errors evolve as

$$\dot{\mathbf{x}}_i - \dot{\tilde{\mathbf{x}}}_i = \mathbf{A}_i(\mathbf{x}_i - \tilde{\mathbf{x}}_i) + \begin{bmatrix}\mathbf{B}_{i,1} & \dots & \mathbf{B}_{i,m}\end{bmatrix}\left(\begin{bmatrix}\mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m\end{bmatrix} - \begin{bmatrix}\mathbf{f}_1(\tilde{\mathbf{y}}_1) \\ \vdots \\ \mathbf{f}_m(\tilde{\mathbf{y}}_m)\end{bmatrix}\right) \tag{7}$$

$$\mathbf{y}_i - \tilde{\mathbf{y}}_i = \mathbf{C}_i(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\ . \tag{8}$$

Equation 3 ensures there is always an $\varepsilon_{i,j} \in [-q_{i,j}, q_{i,j}]$ such that

$$f_{i,j}(\tilde{y}_{i,j}) = \tilde{y}_{i,j} + \varepsilon_{i,j} \tag{9}$$

and the $\varepsilon_{i,j}$ for system i can be arranged into the vector ε_i. Hence, we can rewrite the error equations as

$$\dot{\mathbf{x}}_i - \dot{\tilde{\mathbf{x}}}_i = \mathbf{A}_i(\mathbf{x}_i - \tilde{\mathbf{x}}_i) + \begin{bmatrix}\mathbf{B}_{i,1} & \dots & \mathbf{B}_{i,m}\end{bmatrix}\left(\begin{bmatrix}\mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m\end{bmatrix} - \begin{bmatrix}\tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_m\end{bmatrix} - \begin{bmatrix}\varepsilon_1 \\ \vdots \\ \varepsilon_m\end{bmatrix}\right) \tag{10}$$

$$\mathbf{y}_i - \tilde{\mathbf{y}}_i = \mathbf{C}_i(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\ . \tag{11}$$

Substituting the $\mathbf{y}_j - \tilde{\mathbf{y}}_j$ in Eqn. 10 with the corresponding $\mathbf{C}_j(\mathbf{x}_j - \tilde{\mathbf{x}}_j)$ and rearranging terms we arrive at the expression

$$\dot{\mathbf{x}}_i - \dot{\tilde{\mathbf{x}}}_i = \mathbf{A}_i(\mathbf{x}_i - \tilde{\mathbf{x}}_i) + \begin{bmatrix}\mathbf{B}_{i,1}\mathbf{C}_1 & \dots & \mathbf{B}_{i,m}\mathbf{C}_m\end{bmatrix}\begin{bmatrix}\mathbf{x}_1 - \tilde{\mathbf{x}}_1 \\ \vdots \\ \mathbf{x}_m - \tilde{\mathbf{x}}_m\end{bmatrix} - \begin{bmatrix}\mathbf{B}_{i,1} & \dots & \mathbf{B}_{i,m}\end{bmatrix}\begin{bmatrix}\varepsilon_1 \\ \vdots \\ \varepsilon_m\end{bmatrix}\ . \tag{12}$$

With this expression, it is apparent how a global system of error equations can be composed from the component errors. To do so, define

$$\Delta\mathbf{x} = \begin{bmatrix}\mathbf{x}_1 - \tilde{\mathbf{x}}_1 \\ \vdots \\ \mathbf{x}_m - \tilde{\mathbf{x}}_m\end{bmatrix}, \Delta\mathbf{y} = \begin{bmatrix}\mathbf{y}_1 - \tilde{\mathbf{y}}_1 \\ \vdots \\ \mathbf{y}_m - \tilde{\mathbf{y}}_m\end{bmatrix}, \varepsilon = \begin{bmatrix}\varepsilon_1 \\ \vdots \\ \varepsilon_m\end{bmatrix} \tag{13}$$

and the matrices

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 + \mathbf{B}_{1,1}\mathbf{C}_1 & \mathbf{B}_{1,2}\mathbf{C}_2 & \dots & \mathbf{B}_{1,m}\mathbf{C}_m \\ \mathbf{B}_{2,1}\mathbf{C}_1 & \mathbf{A}_2 + \mathbf{B}_{2,2}\mathbf{C}_2 & \dots & \mathbf{B}_{2,m}\mathbf{C}_m \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{B}_{m,1}\mathbf{C}_1 & \dots & \mathbf{B}_{m,m-1}\mathbf{C}_{m-1} & \mathbf{A}_m + \mathbf{B}_{m,m}\mathbf{C}_m\end{bmatrix}, \tag{14}$$

$$\mathbf{B} = \begin{bmatrix}\mathbf{B}_{1,1} & \dots & \mathbf{B}_{1,m} \\ \vdots & \vdots & \vdots \\ \mathbf{B}_{m,1} & \dots & \mathbf{B}_{m,m}\end{bmatrix}, \text{ and} \tag{15}$$

$$\mathbf{C} = \begin{bmatrix}\mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_m\end{bmatrix}\ . \tag{16}$$

With these matrices and vectors, the global error can be written as

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} - \mathbf{B}\varepsilon \tag{17}$$

$$\Delta\mathbf{y} = \mathbf{C}\Delta\mathbf{x} \ . \tag{18}$$

For any matrix or vector, let $|\cdot|$ indicate the same matrix or vector with its elements replaced by their modulus. If the matrix $\mathbf{A}$ is Hurwitz and it can be decomposed into its eigenvectors $\mathbf{S}$ and eigenvalues Λ such that

$$\mathbf{A} = \mathbf{S}\Lambda\mathbf{S}^{-1} \tag{19}$$

then, recalling that $|\varepsilon| < \max_{i,j} q_{i,j}$ and $\mathbf{x}_i(0) = \tilde{\mathbf{x}}_i$, it follows from Theorem 3 of (Kofman 2005) (pg. 3; note in our case $\Delta x_{max} = 0$ and $\Delta u_{max} = \max_{i,j} q_{i,j}$) that

$$|\Delta\mathbf{x}| \le |\mathbf{S}||\mathbb{R}\mathrm{e}(\Lambda)^{-1}\mathbf{S}^{-1}\mathbf{B}| \max_{i,j} q_{i,j} \ . \tag{20}$$

The corresponding bound on the output error is

$$|\Delta\mathbf{y}| = |\mathbf{C}\Delta\mathbf{x}| \le |\mathbf{C}||\Delta\mathbf{x}| \ . \tag{21}$$

2.1 Numerical integration errors

In practice, the internal state variables $\mathbf{x}_i$ will evolve via some numerical integration scheme and this will introduce errors in addition to those produced by quantization. We can model these errors with bounded perturbation vectors $\mathbf{k}_i$. This change causes Eqns. 5 and 6 to be replaced with

$$\dot{\tilde{\mathbf{x}}}_i = \mathbf{A}_i(\tilde{\mathbf{x}}_i + \mathbf{k}_i) + \mathbf{B}\begin{bmatrix} \mathbf{f}_i(\tilde{\mathbf{y}}_1) \\ \vdots \\ \mathbf{f}_m(\tilde{\mathbf{y}}_m) \end{bmatrix} \tag{22}$$

$$\tilde{\mathbf{y}}_i = \mathbf{C}(\tilde{\mathbf{x}}_i + \mathbf{k}_i) \ . \tag{23}$$

Deriving the global error term as before, it follows from Theorem 3 of (Kofman 2005) that the corresponding bound on the global error is

$$|\Delta\mathbf{x}| \le |\mathbf{S}|(|\mathbb{R}\mathrm{e}(\Lambda)^{-1}\Lambda||\mathbf{S}^{-1}| \max_{i,j} k_{i,j} + |\mathbb{R}\mathrm{e}(\Lambda)^{-1}\mathbf{S}^{-1}\mathbf{B}| \max_{i,j} q_{i,j}) \ . \tag{24}$$

Unlike the quantization constants $q_{i,j}$, the values of the $k_{i,j}$ will not be known explicitly because they are determined by the error of the numerical methods being used. However, the $k_{i,j}$ can be made arbitrarily small through the choice of numerical method and the overriding error will be due to quantization of the model outputs. Therefore in practice we can ignore the numerical integration error when calculating an error bound for the quantized simulation.

3 EXAMPLE A

A simple example of a multi-rate system is the pair

$$\dot{x}_1 = 17x_1 - 9x_2 \tag{25}$$

$$\dot{x}_2 = -28x_2 + 54x_1 \ . \tag{26}$$

The global $\mathbf{A}$ and $\mathbf{B}$ matrices for this model are

$$\mathbf{A} = \begin{bmatrix} 17 & -9 \\ 54 & -28 \end{bmatrix} , \mathbf{B} = \begin{bmatrix} 0 & -9 \\ 54 & 0 \end{bmatrix} \ . \tag{27}$$

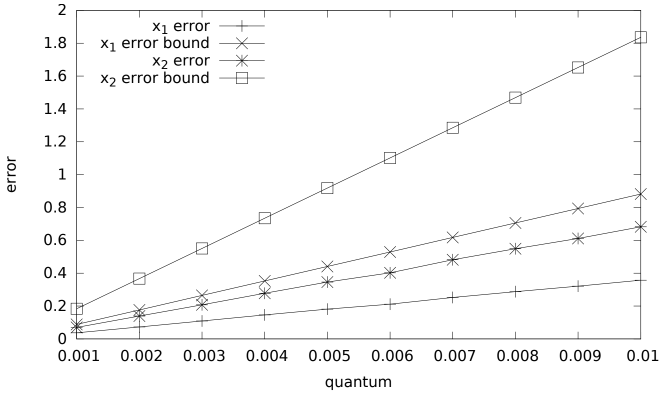

Figure 2: Comparison of actual and anticipated errors for Example A.

and the solution to this set of equations is

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0.31623 & 0.44721 \\ 0.94868 & 0.89443 \end{bmatrix} \begin{bmatrix} e^{-10t} & 0 \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} -6.3246 & 3.1623 \\ 6.7082 & -2.2361 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} . \tag{28}$$

The initial values are $x_1(0) = x_2(0) = 1$.

We can solve these equations numerically by using corrected Euler to advance the state variables and bisection to locate the quantum thresholds. The truncation error for corrected Euler with step size h is proportional to h^2. When solving for x_1 we use a step size of 0.01 and for x_2 a step size of 0.001. The order of magnitude difference in the step sizes is commensurate with the difference in the eigenvalues.

The output variables are quantized in increments of q. To realize the quantization requires extra variables q_1 and q_2 for each model to record the most recent quantization level; initially $q_1 = x_1$ and $q_2 = x_2$. Four quantization functions, u_1, u_2, d_1, and d_2 are used to trigger output at the appropriate times, with q_1 or q_2 being set appropriately at each event. These functions are

$$u_1(x_1) = x_1 - (q_1 + q) \ , \tag{29}$$

$$d_1(x_1) = x_1 - (q_1 - q) \ , \tag{30}$$

$$u_2(x_2) = x_2 - (q_2 + q) \ , \text{ and} \tag{31}$$

$$d_2(x_2) = x_2 - (q_2 - q) \ . \tag{32}$$

Data is exchanged at the roots of these equations.

The integrator induced part of the global error will be on the order of 10^{-4}. If we choose q to be much larger than this then the global error will be dominated by quantization. In this case, Eqn. 20 predicts that the error will be bounded such that

$$|\Delta x_1| \leq 88.2q \tag{33}$$

$$|\Delta x_2| \leq 183.6q \ . \tag{34}$$

Figure 2 compares the numerical error in simulations of the above model with bounds derived from Eqn. 20. In this case, the bounds are somewhat conservative but the linear dependence of the error on the quantum is immediately apparent.

4 EXAMPLE B

In this example we consider two identical systems that follow one another. Each system is described by two state variables, its position x and velocity v, one output variable y that maps its internal representation of position to an outward representation, and an input z that is its view of the other system's position. The equations governing each system are

$$\begin{bmatrix} \dot{v} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} z \tag{35}$$

$$y = 0.5x \; . \tag{36}$$

The initial positions of the systems are $x_1 = 0$ (the first system) and $x_2 = 10$ (the second system) and the initial velocities are zero.

If we fill in the implicit zero matrices required to put this system into the form of Eqns. 1 and 2, then the global $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ matrices are

$$\mathbf{A} = \begin{bmatrix} -1 & -1 & 0 & 0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 0.5 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \; , \tag{37}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \; , \text{and} \tag{38}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \; . \tag{39}$$

The error bound on the state variables when using a quantum q is

$$|\Delta v| \leq 5.0193q \tag{40}$$

$$|\Delta x| \leq 5.7889q \; . \tag{41}$$

and the error in the output variable is

$$|\Delta y| \leq 2.8944q \; . \tag{42}$$

To test these error bounds, the system was simulated as described in Sect. 3 using a step size of $h = 0.001$ for each system. The error in this calculation was compared with an exact solution obtained by diagonalizing the global $\mathbf{A}$ matrix. The results are shown in Fig. 3. As before, the error bounds are conservative and the linear dependence of the error on the quantum size is clearly apparent.

5 POLYNOMIAL QUANTIZATION

A series of quantization functions proposed for the quantized state system integrators (see Kofman (2002), Kofman (2006)) are naturally applicable to the proposed method. The advantage of these functions is a reduction in the number of messages exchanged (see Castro and Kofman (2015)). The general approach is for each model to record the previous n reported values of its input and output variables. These data are fit to polynomials of degree $n-1$. Models receiving data use these polynomials to extrapolate future values. Models sending data track the difference between their output polynomial and actual output value, and the model sends new data when the polynomial and actual value disagree by a quantum.

We will demonstrate this approach using Example B and a linear extrapolation function that employs the two most recently reported values of y. These are indicated by $q(t_1)$ and $q(t_0)$, with $t_1 - t_0 = \tau$ being the interval between the most recent reports. The extrapolation function is

$$p(t) = \frac{q(t_1) - q(t_0)}{\tau}(t - t_1) + q(t_1) \tag{43}$$

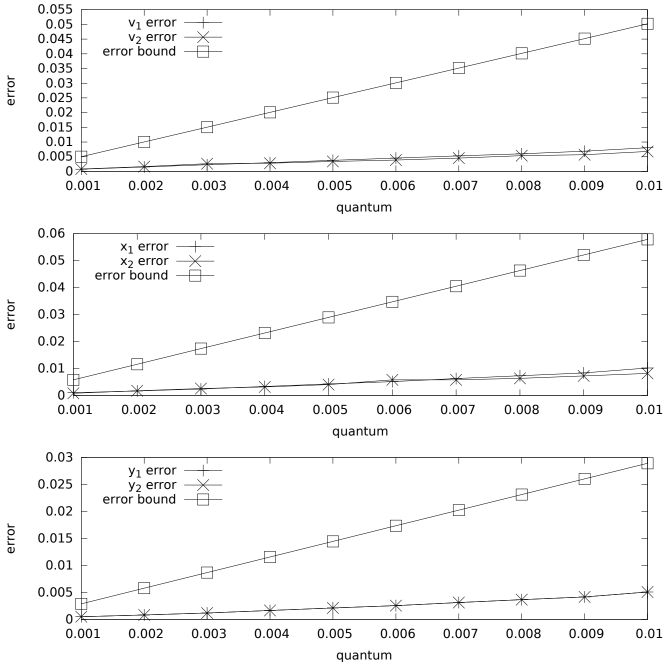

Figure 3: Comparison of actual and anticipated errors for Example B.

Table 1: Total number of messages exchanged in simulations of Example B using the quantization functions described in Sect. 4 (y), Sect. 5 ($p(t)$), and Sect. 6 (small h).

q	y	$p(t)$	small h
0.001	9114	370	59998
0.002	4553	260	29998
0.003	3032	214	19999
0.004	2276	182	14999
0.005	1820	160	11998
0.006	1513	150	9998
0.007	1296	137	8570
0.008	1135	129	7499
0.009	1007	123	6666
0.01	904	111	5999

and the quantization functions are

$$u(y,t) = y - p(t) - q \text{ and} \tag{44}$$

$$d(y,t) = y - p(t) + q \ . \tag{45}$$

Figure 4 shows the errors produced by this scheme. As expected, the errors are very similar to those in Fig. 3 and satisfy the expected error bounds. However, the number of communicated points, which is tallied in Table 1, is much smaller because linear extrapolation tends to keep the actual and reported values close over relatively long periods of time.

6 SMALL COMMUNICATION INTERVALS

In some instances, it is impractical to retrofit a model for precisely handling update events. In many cases, these models are designed to use a step size h that is simply subject to a maximum value, though it might grow or shrink for the purpose of controlling internal errors or stability. Generally, output from such a model will be generated at each time step. If we have knowledge of the largest values taken by the output

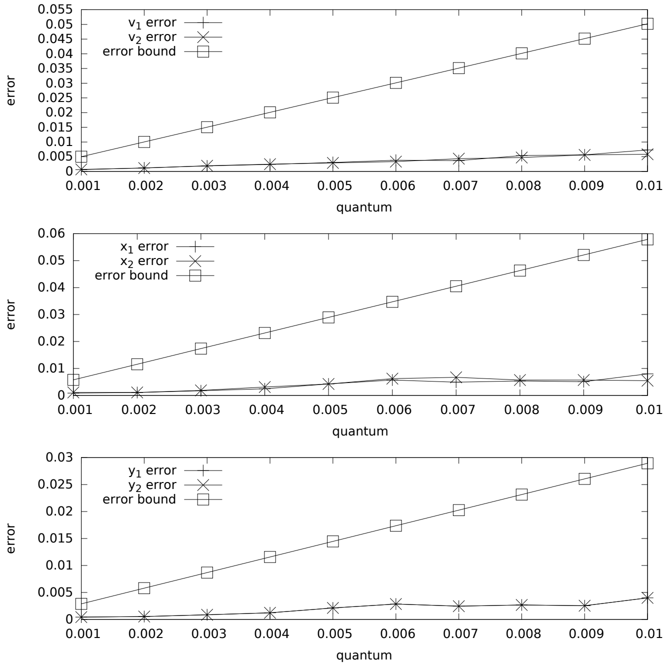

Figure 4: Comparison of actual and anticipated errors for Example B when using linear extrapolation for the quantization function.

derivatives $y^{(1)}$, ..., $y^{(n)}$, then we can estimate the maximum step size to ensure new values are reported at intervals smaller than q. This is easily done by solving for h in the truncated Taylor series

$$q = \sum_{i=1}^{n} \frac{h^n}{n!} |y^{(n)}| \; . \tag{46}$$

This technique is particularly useful when a bound on speed, acceleration, and possibly higher order derivatives can be gleaned from physical considerations. In these cases, an appropriately small step size can be used to control the global error in a simulation. We will demonstrate this approach using Example B with the simulator modified to exchange data at fixed intervals in time. The maximum velocity obtained by either system is approximately 3 and so $y^{(1)} = 0.5\dot{x}_{\max} = 0.5 v_{\max} = 1.5$. Therefore we fix the communication interval to be $q/1.5$.

Figure 5 shows the errors produced by this scheme. The errors are similar to those in Fig. 3 and they satisfy the expected bound. However, the number of communicated points, shown in Table 1, is large because the communication interval is a conservative estimate based on the maximum velocity over the course of the entire simulation. In practice, this estimate would be more conservative still because estimates of maximum speed would be based on the limits of the system's performance rather than what actually occurs in a particular simulation experiment.

For this particular example, the execution time of the model is reduced along with the message count when using the quantized output relative to using a small step size. This implies that the computational effort imposed by the zero crossing algorithm is compensated for by the reduction in the number of simulation steps that quantization enables. The simulations use the corrected Euler method for integrating the equations through time with an adaptive step size that keeps the truncation error near 0.01; root finding is done using the secant method but with a fixed lefthand side (Nutaro 2014) and tolerance of 10^{-6}. The mean and standard deviation of the execution times using ten replications with $q = 0.001$, which is the most computationally intensive of the cases considered, are as follows: for small step size, a mean of 0.0492 s and standard deviation 0.00222 s; for the Sect. 4 quantization function, a mean of 0.0337 s and standard

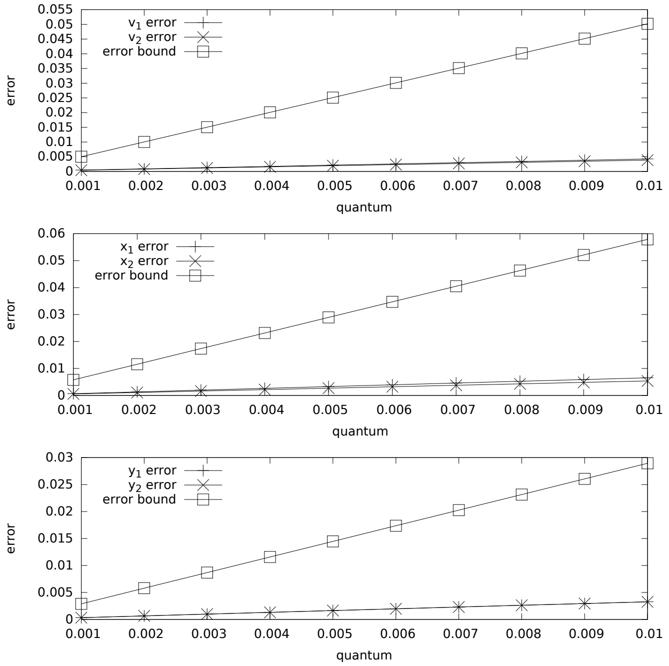

Figure 5: Comparison of actual and anticipated errors for Example B when using a small reporting interval.

deviation 0.00193; and for the Sect. 5 quantization function, a mean of 0.0198 s and standard deviation 0.00206.

7 CONCLUSIONS

Linear dependence of the error bound on the quantum size justifies the use of quantum thresholds for controlling global error. However, in many circumstances it will be infeasible to compute an actual bound on the error. This may be because the parameters of a simulation model are not known (e.g., when integrating complex components), the model is nonlinear (and, hence, we rely on it being approximately linear between output events), or both. Given this difficulty, it becomes important to create new, practical methods for choosing a quantum size that balances simulation errors and communication overheads. One possible approach to this problem is online monitoring of the error caused by quantization. First steps in this direction have been taken by Bolduc and Vangheluwe (2003), but it remains an important topic for future research.

A key simplifying assumption in our analysis concerned the form of the components' output functions, which do not depend on the present input. This avoids the issue of handling algebraic loops and is appropriate in many circumstances. Nonetheless, this assumption restricts the scope of applicability for the proposed method. The problem of algebraic loops in component based models has been considered by Kübler and Schielen (2000) for a simulation procedure that uses periodic synchronization between components, and it may be possible to modify their method to accommodate quantized exchanges of data. The related problem of solving algebraic loops in conjunction with quantized state integration techniques is examined by Kofman (2003).

It is intuitively appealing to think that errors proportional to $q_{i,j}^n$, $n > 1$ are possible with some suitable choice of quantization function. Unfortunately, the theory underlying our approach appears to prohibit such an improvement; this specific problem is explored in relation to numerical integration by Kofman (2006) and (2002), Kofman and Junco (2001), and Castro and Kofman (2015). Some reduction of the error can be had by placing an upper limit on the time between communicated points (Nutaro 2007), but this has the effect of increasing the number of messages that must be sent. The creation of efficient, high order accuracy methods is a topic for future work.

ACKNOWLEDGMENTS

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

REFERENCES

Bolduc, J. S., and H. Vangheluwe. 2003, July. "Mapping ODEs to DEVS: Adaptive quantization". In *Proceedings of the 2003 Summer Simulation MultiConference (SCSC'03)*, 401–407. Montreál, Canada: The Society for Modeling and Simulation International.

Castro, R., and E. Kofman. 2015. "Activity of order n in continuous systems". *SIMULATION* 91 (4): 337–348.

Cellier, F. E., and E. Kofman. 2006. *Continuous System Simulation*. Springer.

Giambiasi, N., B. Escude, and S. Ghosh. 2000. "GDEVS: A generalized discrete event specification for accurate modeling of dynamic systems". *SIMULATION* 17 (3): 120–134.

Kofman, E. 2002. "A Second-Order Approximation for DEVS Simulation of Continuous Systems". *SIMULATION* 78 (2): 76–89.

Kofman, E. 2003. "Quantization-Based Simulation of Differential Algebraic Equation Systems". *SIMULATION* 79 (7): 363–376.

Kofman, E. 2005. "Non-conservative ultimate bound estimation in LTI perturbed systems". *Automatica* 41 (10): 1835 – 1838.

Kofman, E. 2006. "A third order discrete event method for continuous system simulation". *Latin American applied research* 36 (2): 101–108.

Kofman, E., and S. Junco. 2001, September. "Quantized-state Systems: A DEVS Approach for Continuous System Simulation". *Transactions of the Society for Computer Simulation International* 18 (3): 123–132.

Kübler, R., and W. Schielen. 2000. "Two methods of simulator coupling". *Mathematical and computer modelling of dynamical systems* 6 (2): 93–113.

Nutaro, J. 2007. "A Second Order Accurate Adams-Bashforth Type Discrete Event Integration Scheme". In *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation*, PADS '07, 25–31. Washington, DC, USA: IEEE Computer Society.

Nutaro, J. 2014. "An extension of the OpenModelica compiler for using Modelica models in a discrete event simulation". *SIMULATION* 90 (12): 1328–1345.

Zeigler, B. P., H. Sarjoughian, and H. Praehofer. 2000. "Theory of quantized systems: DEVS simulation of perceiving agents". *Cybernetics and Systems* 31 (6): 611–647.

AUTHOR BIOGRAPHIES

JAMES NUTARO is a Distinguished Research and Development Scientist in the Computational Sciences and Engineering Division of Oak Ridge National Laboratories. He holds a PhD in Computer Engineering from the University of Arizona. His email address is nutarojj@ornl.gov.