Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 797–819
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

On the stability and performance of discrete event methods for simulating continuous systems ☆

James Nutaro [a,*], Bernard Zeigler [b]

[a] Oak Ridge National Laboratory, Oak Ridge, TN, United States
[b] Arizona Center for Integrative Modeling and Simulation, University of Arizona, Tucson, AZ, United States

Received 29 August 2006; received in revised form 19 July 2007; accepted 17 August 2007
Available online 31 August 2007

Abstract

This paper establishes a link between the stability of a first order, explicit discrete event integration scheme and the stability criteria for the explicit Euler method. The paper begins by constructing a time-varying linear system with bounded inputs that is equivalent to the first order discrete event integration scheme. The stability of the discrete event system is shown to result from the fact that it automatically adjusts its time advance to lie below the limit set by the explicit Euler stability criteria. Moreover, because it is not necessary to update all integrators at this rate, a significant performance advantage is possible. Our results confirm and explain previously reported studies where it is demonstrated that a reduced number of updates can provide a significant performance advantage compared to fixed step methods. These results also throw some light on stability requirements for discrete event simulation of spatially extended systems.

Keywords: Discrete event simulation; DEVS; Stability; Differential automata

1. Introduction

There is a growing interest in continuous system simulation methods that use forward looking predictions of threshold crossings in continuous variables as the primary (in many instances, only) basis for advancing time (see, e.g. [1–7]). For many applications, this kind of purely asynchronous approach results in a relatively easy to program simulation that is stable, computationally efficient, and can be parallelized using

☆ Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the US Department of Energy under Contract No. DE-AC05-00OR22725. The submitted manuscript has been authored by a contractor of the US Government under Contract DE-AC05-00OR22725. Accordingly, the US Government retains a nonexclusive, royalty- free license to publish or reproduce the published form of this contribution, or allow others to do so, for US Government purposes.
* Corresponding author.
E-mail addresses: nutarojj@ornl.gov (J. Nutaro), zeigler@ece.arizona.edu (B. Zeigler).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.08.015

highly effective parallel discrete event simulation algorithms (see, e.g. [8,9,7]). These gains seem to be offset only by the relatively low ($\leqslant 3$) order of accuracy that has been obtained to date with this approach (see, e.g. [2,10–13]).

Theoretical studies of these types of algorithms have received less attention than the development of the algorithms themselves. This is due in part to the apparent lack of well established methods for analysis. The simplicity of the updating conditions (i.e., finding the next anticipated threshold crossing time) is what makes these methods attractive, but this simplicity belies complex dynamics that are exhibited by the algorithms themselves.

Mathematical incarnations of these simulation algorithms closely resemble the hybrid automata considered in control theory (see, e.g. [14,15]). Consequently, analysis of these types of discrete event methods tends to draw more from hybrid systems concepts and less from established numerical theory for adaptive methods (see, e.g. [16–19]). The advantage of purely asynchronous approaches is mainly in its support for large scale parallel computing. By eliminating the need for any kind of global synchronization, it is possible to use highly effective and very scalable parallel discrete event simulation algorithms to simulate many kinds of continuous processes (see [20] for an introduction to parallel discrete event simulation).

This paper makes a contribution to the stability theory for a first order discrete event method. Our theory links known stability criteria for the explicit Euler integration scheme and the first order discrete event method. The major result of the theory is that the resultant of the discrete event system satisfies explicit Euler stability criteria when simulating transients. The explicit Euler stability criteria can be violated only when the simulated system is close to equilibrium. Violations can persist indefinitely only when the system is actually at equilibrium.

One consequence of the theory is that discrete event simulations of linear, stable, time invariant systems produce trajectories that are ultimately bounded (see, e.g. [21]). An accurate description of the bounding region in terms of the system derivatives is derived. However, this bound does not give an estimate of the simulation error (see, for comparison, [22]).

The stability proof first formulates a Discrete Event System Specification (DEVS, see [23]) representation of the first order integration scheme. This representation allows it to be treated as a time-varying linear system with bounded inputs. The stability of this time-varying linear system depends on the resultant of the DEVS satisfying explicit Euler stability criteria. The main result is that the DEVS resultant satisfies this stability criteria everywhere except in a narrow region near equilibrium. Consequently, the system has ultimately bounded trajectories.

The plan for the remaining part of this paper is as follows. Section 2 gives an overview of two previous stability studies that are germane to our topic. Section 3 introduces an appropriate DEVS representation of an integrator network. Section 4 shows that this representation accounts for an ''aggressive'' re-formulation of the standard DEVS interpretation. In Section 5, this re-formulation of the DEVS integrator network is transformed into a time-varying linear system with bounded inputs. In Section 6, it is shown that this system has ultimately bounded trajectories. Moreover, the region that ultimately bounds a trajectory is centered on the system equilibrium, and this bounding region has dimensions that are proportional to the integration quantum. In Section 7, the implications of a time advance limit for error bounds and performance are considered.

Examples of the time advance limit on the DEVS resultant are provided in Section 8. These examples demonstrate the behavior of the time advance during transients, and provide concrete demonstrations of the time advance bound. The consequences of this bound on simulation errors are also demonstrated. Finally, conclusions are offered concerning the significance of the results, their implications for discrete event simulation of spatially extended systems, and the need for further development of a theory for discrete event numerical integration.

2. Related stability studies

There are two previous stability studies that are particularly relevant to the development in this paper. These are a study of quantized state systems, undertaken in [1,22,24], and a study of differential automata presented in [15]. Quantized state systems are a particular type of differential automata, and so these two studies

each provide new information about that class of systems. The theory that will be developed in this paper gives a third vantage point, and with this new view another insight is obtained.

Linear quantized state systems,[1] as presented in [1] and elsewhere (also see Section 3 for an overview), are defined by a linear system equipped with a hysteretic quantization function:

$$\begin{aligned}
&\dot{x}(t) = Az(t),\\
&z_i(t) = b(x_i(t)), \quad \text{and} \quad b(x_i(t)) \text{ quantizes } x_i(t) \text{ by}\\
&b(x_i(t)) = \begin{cases} x_i(0) & \text{if } t = 0,\\ b(x_i(t^-)) + \Delta q & \text{if } x_i(t) = b(x_i(t^-)) + \Delta q,\\ b(x_i(t^-)) - \Delta q & \text{if } x_i(t) = b(x_i(t^-)) - \Delta q,\\ b(x_i(t^-)) & \text{otherwise.}\end{cases}
\end{aligned} \tag{1}$$

Here we assume a uniform quantization. That is, the same Δq is applied to each component of x. While this is not necessary in theory or practice, it does simplify the presentation considerably.

When the *continuous* system $\dot{x} = Ax$ is stable, the quantized state system produces bounded trajectories that can be made arbitrarily close to those of the continuous system. Theorem 1, due to Kofman and Junco, makes this concrete.

Theorem 1 (Kofman and Junco). *Let $\tilde{x}(t)$ be a trajectory of the linear system $\dot{\tilde{x}}(t) = A\tilde{x}(t)$, $x(t)$ a trajectory of its associated quantized state system with a uniform quantization Δq, and $x(0) = \tilde{x}(0)$. If A is a Hurwitz and diagonalizable matrix, then*

$$|\tilde{x}(t) - x(t)| \leqslant |V||\Re(\Lambda)^{-1}\Lambda||V^{-1}|\Delta q,$$

where $A = V\Lambda V^{-1}$ and $|\cdot|$ is the component-wise magnitude.

Theorem 1 states that the error in a simulation of a linear system by a quantized state system is proportional to Δq, with the constant of proportionality being determined by A itself.

The linear quantized state system described above is an instance of a differential automaton [15]. Differential automata are defined by state space equations in the form

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

$$q(t + 0) = \delta(x(t), q(t)), \tag{3}$$

where $x(t) \in \mathbb{R}^n$ and $q(t)$ is a discrete state in some finite set Q. This system follows a trajectory defined by Eq. (2) so long as q remains constant. At switching times t_k, the value of q changes instantaneously from $q(t_k)$ to $q(t_k + 0)$ through Eq. (3) (see [15] for a rigorous treatment of this subject).

We can define the vector form of b by a function $\bar{b}$ whose range is countable. This function applies b to each element of its argument, and its range is a countable (but not finite) subset of $\mathbb{R}^n$. The quantized state system can now be written as the pseudo-differential automaton

$$\dot{x}(t) = Aq(t), \tag{4}$$

$$q(t + 0) = \bar{b}(x(t)). \tag{5}$$

The term 'pseudo-differential automaton' is used to indicate that the range of q is not finite. We can now correct this fact.

The requirement that q have a finite range can be satisfied if, for any $x(0)$, it is possible to find a bounded subset of $\mathbb{R}^n$ such that for all $t \geqslant 0$, $x(t)$ remains in that subset. This subset is called an *invariant set*, and it is denoted by K. When considering any particular initial condition, we can always restrict the range of q to the finite image of K under $\bar{b}$. Therefore, for any initial condition we can construct a differential automaton that has a finite set of discrete states. The next lemma establishes the existence of K when A has eigenvalues with negative real parts.

[1] The definition of quantized state systems can be readily expanded to include non-linear systems, and similar properties hold in the more general case. See [24].

Lemma 2. *Suppose that A is a Hurwitz matrix. Then for every initial condition* $x(0)$, *there exists a bounded set* $K \subset \mathbb{R}^n$ *such that, for all* $t \geqslant 0$, *the resulting trajectory* $x(t)$ *stays inside the set K.*

Proof. If $x(t)$ is bounded, then this is clearly true. Suppose, contrarily, that $x(t)$ is not bounded. Then as $t \to \infty$, the interval between switching times goes to zero. However, it is also true (see Section 6) that $x(t)$ at switching times $t_1, t_2, \ldots$ can be written

$$x(t_{n+1}) = (I + (t_{n+1} - t_n)A)x_n + (t_{n+1} - t_n)Ak_n,$$

where k_n is a vector with component-wise bounded elements. The matrix

$$I + (t_{n+1} - t_n)A$$

has eigenvalues inside of the unit circle when $t_{n+1} - t_n$ is positive but suitably small. But this contradicts our assumption that $x(t)$ grows without bound because, upon reaching some finite limit, $x(t)$ must begin to contract towards $(t_{n+1} - t_n)Ak_n$. □

The following facts are also true for quantized state systems[2]:

(1) The vector Aq is constant between switching times.
(2) It is deterministic; every initial state generates a single trajectory.
(3) The switching surfaces are hypercubes.
(4) The system is legitimate (non-Zeno) (see, e.g. [23,15,25]).

These items follow almost directly from the definition of the quantized state system. However, a complete proof requires a lengthy recapitulation of basic definitions, and so we only sketch the arguments here. Item 1 follows directly from Eqs. (5) and (4). Item 2 is also an immediate consequence of the definition of the quantized state system (see [23,1], or [15] for a review of the dynamics associated with this type of discrete event system). Item 3 follows immediately from Eqs. (5) and (1). Finally, to see that the system is legitimate, it is sufficient to note that $\Delta q > 0$ and $\|\dot{x}\|$ is bounded. Therefore, the time separating switching events is strictly positive.

Given facts 1–4 and Lemma 2, the following theorem, due to Matveev and Savkin [15], holds so long as, for all t, $x(t) \neq 0$. Note that if $x(t) = 0$ at some point, then $\dot{x}(t) = 0$, and the system has reached equilibrium.

Theorem 3 (Matveev and Savkin). *The following statements hold*:

(i) *There exists a limit cycle lying in K.*
(ii) *The number of such cycles is finite.*
(iii) *Any limit cycle lying in K is regularly locally asymptotically stable in K.*
(iv) *Any trajectory lying in K regularly converges to one of the above limit cycles.*

To summarize, this theorem states that, as $t \to \infty$, every trajectory becomes periodic. That is to say, *both* the continuous *and* discrete variables are periodic! If we wait long enough, the system will settle into a trajectory with a period T such that

$$x(t) = x(t + T) \quad \text{and} \quad q(t) = q(t + T).$$

In practice, this is often seen to occur in finite time. Moreover, there are only finitely many such periodic trajectories, and so they act as distinct 'equilibrium trajectories' for the system. See [15] for a rigorous definition of the terms used in Theorem 3.

[2] These items address the following pre-conditions for Theorem 3: Item 1 satisfies Assumption 5.2.1 in [15]. Assumption 5.2.2 holds because of item 2. Assumption 5.2.3 is satisfied by items 2 and 3. Assumption 5.2.4 follows from item 3 and the fact that the system trajectories are bounded. Assumption 5.2.5 states that the system must be legitimate. Assumptions 5.2.6–5.2.8 are satisfied by items 2 and 4.

Theorems 1 and 3 (with Lemma 2), and the new development that will be presented in this paper, give three distinct proofs that linear quantized state systems produce bounded trajectories (under, of course, the necessary assumptions concerning A). In the course of constructing these arguments, however, three different facts are uncovered – each being apparent only from the vantage point of one of the three particular theories. These are

(1) [Theorem 1]: The trajectories of a linear quantized state system can be made arbitrarily close to those of a continuous linear system $\dot{x} = Ax$. This makes quantized state systems suitable for use as a numerical integration scheme. Moreover, linear quantized state systems with a stable A matrix produce bounded trajectories irregardless of the quantization parameter Δq.
(2) [Theorem 7; see Section 6]: Limiting changes in $x(t)$ between events forces the time advance of the resultant to satisfy known stability constraints almost everywhere (note this is only proved for the case where A is Hurwitz with real eigenvalues; see Section 6). That is, the quantized state system acts as a kind of self-stabilizing numerical method. This supports the observation, made by Karimabadi et al. in [7], that their discrete event method for simulating plasma flow is self-stabilizing.
(3) [Theorem 3, Lemma 2]: A linear quantized state system either reaches equilibrium at $x(t) = 0$, or it is ultimate attracted to a limit cycle that is periodic in both its continuous state x and discrete state q. This is clearly evident in simulations of linear quantized state systems, and a part of the machinery that produces the limit cycles is the instability of the system near equilibrium (here the matrix $I + (t_{n+1} - t_n)A$ is no longer Hurwitz, and so $x(t)$ can grow – see Section 6). It further suggests that quantized state systems which are used as numerical integration schemes should be equipped with an upper limit on the time advance in order to converge on equilibrium points.

3. DEVS simulation of linear time invariant systems

A first order DEVS integrator with optimal hysteresis is described by (see [1,3])

$$
\begin{aligned}
&S = \{(q, \dot{q}, q_l) | (q, \dot{q}, q_l) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}\},\\
&X = Y = \mathbb{R},\\
&ta((q, \dot{q}, q_l)) = \begin{cases} (q_l + \Delta q - q)/\dot{q}, & \text{if } \dot{q} > 0,\\ (q_l - \Delta q - q)/\dot{q}, & \text{if } \dot{q} < 0,\\ \infty, & \text{otherwise,} \end{cases}\\
&\delta_{\text{int}}((q, \dot{q}, q_l)) = (q_l + \Delta q \cdot \text{sgn}(\dot{q}), \dot{q}, q_l + \Delta q \cdot \text{sgn}(\dot{q})),\\
&\delta_{\text{ext}}((q, \dot{q}, q_l), e, x) = (q + e \cdot \dot{q}, x, q_l),\\
&\lambda((q, \dot{q}, q_l)) = q_l + \Delta q \cdot \text{sgn}(\dot{q}),
\end{aligned}
\tag{6}
$$

where $\text{sgn}(\cdot)$ is the sign function

$$
\text{sgn}(x) = \begin{cases} 1, & \text{if } x > 0,\\ -1, & \text{if } x < 0,\\ 0, & \text{if } x = 0 \end{cases}
$$

and Δq is the state space discretization parameter. For the purposes of this discussion, the initial state always has $q_l = q$. Note that the $\dot{q} < 0$ case of the time advance function could be rewritten more conventionally as

$$
ta((q, \dot{q}, q_l)) = \frac{(q - (q_l - \Delta q))}{|\dot{q}|} \quad \text{if } \dot{q} < 0.
$$

These integrators are coupled through a set of memory-less functions to describe a system of equations (see, e.g. [1,23]). The resultant of this coupled system (see [23]) is an input free DEVS, and this input free DEVS has an alternative representation that is isomorphic to a piecewise continuous system.

This equivalent piecewise continuous system is described by Eqs. (7)–(9) (see [1]). The matrix A and vectors x and z are constrained to be real, and A is further assumed to be diagonalizable. The vector x, with elements $x_1, x_2, \ldots, x_n$, follows a straight line whose slope is described by Az, where z has the elements $z_1, z_2, \ldots, z_n$. The vector z is a discrete approximation of x that is constructed with a quantization function b and quantizer resolution Δq:

$$\dot{x}(t) = Az(t), \tag{7}$$

$$z_i(t) = b(x_i(t)), \quad \text{and} \quad b(x_i(t)) \text{ quantizes } x_i(t) \text{ by} \tag{8}$$

$$b(x_i(t)) = \begin{cases} x_i(0) & \text{if } t = 0, \\ b(x_i(t^-)) + \Delta q & \text{if } x_i(t) = b(x_i(t^-)) + \Delta q, \\ b(x_i(t^-)) - \Delta q & \text{if } x_i(t) = b(x_i(t^-)) - \Delta q, \\ b(x_i(t^-)) & \text{otherwise.} \end{cases} \tag{9}$$

The trajectory of x is a series of connected lines. The slope of a line, given by Az, changes when any component of x changes in magnitude by Δq. One possible phase plot of the x and z variables is shown in Fig. 1. This plot shows the evolution of a particular system with

$$\Delta q = 0.1, \quad A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}, \quad \text{and} \quad z(0) = x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{10}$$

In this instance, the system moves to its equilibrium state and then stops. The trajectory $x(t)$ is shown by the solid line. The points indicate states at which Eq. (9) is satisfied.

The more general case is illustrated in Fig. 2. This phase plot was produced by the same system with initial conditions $z(0) = x(0) = [\,1.33 \quad 1.33\,]^{\mathrm{T}}$. In this case, the trajectory moves steadily towards equilibrium, and it ultimately becomes trapped in a region near equilibrium.

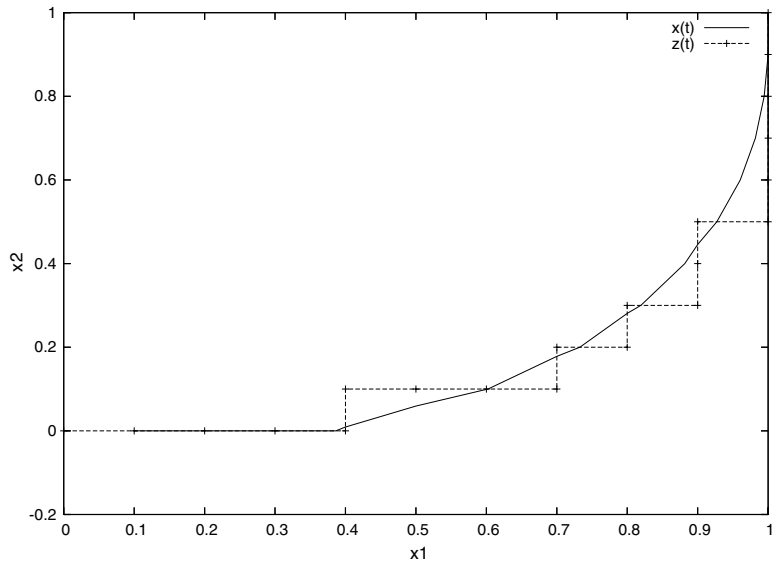

Fig. 1. A phase plot of system (10) with $z(0) = x(0) = [\,1 \quad 1\,]^{\mathrm{T}}$.

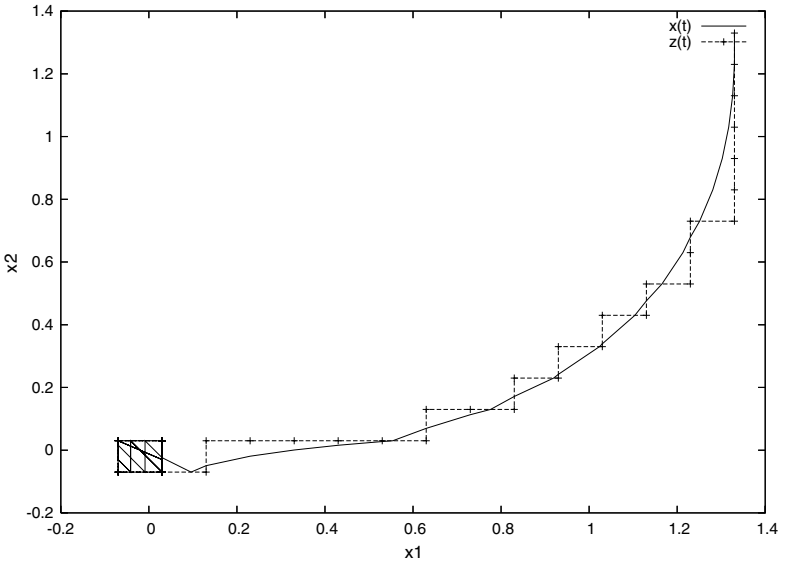

Fig. 2. A phase plot (10) with $z(0) = x(0) = [1.33 \quad 1.33]^{\mathrm{T}}$.

4. Equivalent representations of the integrator

A stability theory can be constructed for the system described by Eqs. (7)–(9). The relationship between that system and the discrete event system described by (6) can be summarized in the following way. The state trajectories of the components (i.e., x_i, $\dot{x}_i$, and z_i and q_i, $\dot{q}_i$, and $q_{l,i}$) are equal at the event times for those components. Event times are defined in terms of the time advance function $ta(\cdot)$ for the discrete event model, and in terms of the quantization function $b(\cdot)$ for the piecewise continuous model.

The mapping between the discrete event system and the piecewise continuous system is developed in three parts. In the first part, sufficient conditions are developed for an *aggressive* DEVS model to be equivalent to a classic, here called a *lazy* DEVS model. The aggressive DEVS model differs from the lazy DEVS model by defining a state updating function that can be applied in the absence of internal or external events. In the second part, an aggressive form of the DEVS integrator is developed. In the third and final part, the aggressive DEVS integrator is shown to be equivalent to the piecewise continuous system described by Eqs. (7)–(9).

An input and output free coupled model N, with components that are DEVS atomic models, is defined by a structure (see [23])

$N = \langle D, \{z_{i,j}\}, Select\rangle$, where

D is the set of component models,

$\{z_{ij}\}$ is the set of coupling functions $z_{ij} : Y_i \to X_j$,

$Select : 2^D \to D$ is the tie resolution function.

For the purposes of this discussion, the component set D is assumed to contain only atomic DEVS models. The coupling function z_{ij} maps the output of component i to the input of component j, where $i,j \in D$. The *Select* function is used to determine which component's output and internal state transition functions should be evaluated when multiple components are eligible to undergo an internal event.

The coupled model N is associated with an input and output free atomic model called its *resultant* (see [23] for a detailed treatment of this subject). This atomic model, which will here be called a *lazy* DEVS, is defined by

$$\mathrm{DEVS}_{\mathrm{lazy}} = \langle S, \delta_{\mathrm{int}}, ta\rangle.$$

The state set S is

$$S = \prod_{d \in D} Q_d,$$

where Q_d is the set of *total states* (s_d, e_d) of the atomic model d. The time advance function of the resultant is

$$ta(s) = \min\{\sigma_d | d \in D\}, \tag{11}$$

where for each component d, $\sigma_d = ta_d(s_d) - e_d$. That is, σ_d is the time remaining until the next internal event of component d.

The imminent set of the coupled model N that corresponds to the state s of the resultant is given by $IMM(s) = \{d | d \in D \ \& \ \sigma_d = ta(s)\}$. This describes the set of components that have minimum remaining time σ_d (i.e., they are candidates to undergo an internal event).

The internal transition function of the resultant is defined in the following way. Let $s = (\ldots,(s_d, e_d),\ldots)$ and $d^* = Select(IMM(s))$. Then

$$\delta_{\text{int}}(s) = s' = (\ldots, (s'_d, e'_d), \ldots), \quad \text{where}$$

$$(s'_d, e'_d) = \begin{cases} (\delta_{\text{int},d}(s_d), 0), & \text{if } d = d^*, \\ (\delta_{\text{ext},d}(s_d, e_d + ta(s), x_d), 0), & \text{if } x_d \neq \emptyset, \\ (s_d, e_d + ta(s)), & \text{if } x_d = \emptyset, \end{cases}$$

where $x_d = z_{d^*,d}(\lambda_{d^*}(s_{d^*}))$.

When the function $\delta_{\text{int},d}$ is evaluated, it is said that an internal event has occurred at component d. Evaluation of $\delta_{\text{ext},d}$ is called an external event, and the $x_d = \emptyset$ case is called a non-event. The derivation of the resultant can be found in [23].

An aggressive DEVS can be specified over the same set of components as a lazy DEVS. The aggressive DEVS time advance function and state set are identical to those of the lazy DEVS. The internal transition function is defined in terms of a set of functions $\delta_{\emptyset,d} : Q_d \times \mathbb{R} \to Q_d$ and $\delta_{x,d} : Q_d \times X_d \to Q_d$. There is one $\delta_{\emptyset,d}$ function and one $\delta_{x,d}$ function for each component. The aggressive DEVS internal transition function is given by

$$(\ldots, (s'_d, e'_d), \ldots) = \delta_{\text{int}}((\ldots, (s_d, e_d), \ldots)), \quad \text{where}$$

$$(s'_d, e'_d) = \begin{cases} (\delta_{\text{int},d}(s_d), 0), & \text{if } d = d^*, \\ \delta_{x,d}(\delta_{\emptyset,d}((s_d, e_d), ta(s)), x_d), & \text{if } x_d \neq \emptyset, \\ \delta_{\emptyset,d}((s_d, e_d), ta(s)), & \text{if } x_d = \emptyset, \end{cases}$$

where $x_d = z_{d^*,d}(\lambda_{d^*}(s_{d^*}))$.

The initial state of the aggressive and lazy DEVS are constrained to be identical with all $e_d = 0$. The $\delta_{\emptyset,d}$ and $\delta_{x,d}$ functions are constrained such that the following properties hold:

(1) Autonomous behavior is preserved, i.e.,

$$(s'_d, e'_d) = \delta_{\emptyset,d}((s_d, 0), e_d) \ \Rightarrow \ \delta_{\text{int},d}(s'_d) = \delta_{\text{int},d}(s_d) \ \& \ \lambda_d(s'_d) = \lambda_d(s_d).$$

(2) The external transition function is preserved, i.e.,

$$\delta_{x,d}(\delta_{\emptyset,d}((s_d, e_d), t), x) = (\delta_{\text{ext},d}(s_d, e_d + t, x), 0).$$

(3) The time advance function is preserved, i.e.,

$$(s'_d, e'_d) = \delta_{\emptyset,d}((s_d, e_d), t) \ \Rightarrow \ ta_d(s'_d) - e'_d = ta_d(s_d) - (e_d + t).$$

(4) $\delta_{\emptyset,d}$ has the composition property

$$\delta_{\emptyset,d}((s_d, e_d), \alpha + \beta) = \delta_{\emptyset,d}(\delta_{\emptyset,d}((s_d, e_d), \alpha), \beta).$$

If all of these properties are satisfied, then the following is true.

Theorem 4. *Let $s(t)$ denote the state trajectory of the lazy DEVS and $\tilde{s}(t)$ the state trajectory of the aggressive DEVS. Let t^* denote any time at which some component model of the lazy DEVS undergoes an internal or*

external event. Let $s_d(t)$ denote the trajectory of that component as it evolves as part of the lazy DEVS, and $\tilde{s}_d(t)$ the corresponding component trajectory as it appears as part of the aggressive DEVS. Then

$$s_d(0) = \tilde{s}_d(0) \quad \textit{and} \quad s_d(t^*) = \tilde{s}_d(t^*).$$

Proof. The proof is accomplished by induction on the number of internal and external events that occur in the execution of the lazy DEVS.

Base case: The statement $s_d(0) = \tilde{s}_d(0)$ follows directly from the assertion that $s(0) = \tilde{s}(0)$. Let t_1 denote the first event occurring in the lazy DEVS. So $t_1 = ta(s(0)) = ta(\tilde{s}(0))$. Hence, the first event time of the aggressive and lazy DEVS coincide.

Let d be a component that undergoes an internal or external event. If it is an internal event, then $ta(s) = ta_d(s_d)$ and both the aggressive and lazy DEVS update s_d using $\delta_{\text{int},d}$. If it is an external event, then it follows from property 2 that $\delta_{x,d}(\delta_{\emptyset,d}((s_d, 0), ta(s)), x) = (\delta_{\text{ext},d}(s_d, ta(s), x), 0)$. So $s_d(t_1) = \tilde{s}_d(t_1)$.

Inductive step: Assume that the theorem holds for every subsequent lazy DEVS event time $t_2, t_3, \ldots, t_n$ and consider the next event at time t_{n+1}.

Let d be a component undergoing an internal or external event at time t_{n+1} as part of the lazy DEVS evolution. Let the total state of this component at time t_n be (s_d, e_d). From the induction hypothesis, $s_d(t_n - e_d) = \tilde{s}_d(t_n - e_d)$. That is, they agree on the state of component d when it last changed as part of the lazy DEVS evolution. Using this fact and the composition property of $\delta_{\emptyset,d}$, the corresponding total state of the aggressive DEVS component at time t_n can be written as $(s'_d, e'_d) = \delta_{\emptyset,d}((s_d, 0), e_d)$.

Suppose that component d is selected by the lazy DEVS to undergo an internal event at time t_{n+1}. So

$$t_{n+1} = t_n + ta_d(s_d) - e_d.$$

It follows from property 3 that the corresponding next event time of the aggressive DEVS component is

$$t_n + ta_d(s'_d) - e'_d = t_n + ta_d(s_d) - (0 + e_d) = t_n + ta_d(s_d) - e_d = t_{n+1}$$

and so it too will undergo an internal event at time t_{n+1}. Moreover, it follows from property 1 that

$$\delta_{\text{int},d}(s_d) = \delta_{\text{int},d}(s'_d), \quad \text{so } s_d(t_{n+1}) = \tilde{s}_d(t_{n+1}).$$

Notice that t_{n+1} defines the next event time of both the aggressive and lazy DEVS. Also, the induction hypothesis requires that t_n be the previous event time for both DEVS. Consequently, $ta(s(t_n)) = ta(\tilde{s}(t_n))$, or to be brief, $ta(s) = ta(\tilde{s})$ at all event times. If d undergoes an external event, then it follows from this fact, properties 1, 4, and 2 that

$$x_d = z_{d^*,d}(\lambda_{d^*}(s_{d^*})) = z_{d^*,d}(\lambda_{d^*}(s'_{d^*})) \quad \text{and} \quad (\delta_{\text{ext},d}(s_d, e_d + ta(s), x_d), 0) = \delta_{x,d}(\delta_{\emptyset,d}((s_d, 0), e_d + ta(s)), x_d),$$

where d^* refers to the component that was selected for an internal event. It was previously shown that $(s'_{d^*}, e'_{d^*}) = \delta_{\emptyset,d^*}((s_{d^*}, 0), e_{d^*})$. Therefore, $s_d(t_{n+1}) = \tilde{s}_d(t_{n+1})$ in this case as well. □

The lazy DEVS integrator defined by system (6) can be written as an equivalent aggressive DEVS. To define this aggressive DEVS integrator, let

$$\delta_\emptyset(((q, \dot{q}, q_l), e), \tau) = ((q + (e + \tau) \cdot \dot{q}, \dot{q}, q_l), 0) \quad \text{and} \quad \delta_x(((q, \dot{q}, q_l), e), x) = ((q, x, q_l), 0).$$

Properties 1–4 can be verified as follows:

(1) Preservation of autonomous behavior

$$\begin{aligned}\delta_\emptyset(((q, \dot{q}, q_l), 0), e) = ((q + e \cdot \dot{q}, \dot{q}, q_l), 0) \;&\Rightarrow\; \delta_{\text{int}}((q, \dot{q}, q_l)) \\ &= (q_l + \Delta q \cdot \operatorname{sgn}(\dot{q}), \dot{q}, q_l + \Delta q \cdot \operatorname{sgn}(\dot{q})) = \delta_{\text{int}}((q + e \cdot \dot{q}, \dot{q}, q_l)) \\ \&\;\; \lambda((q, \dot{q}, q_l)) &= q_l + \Delta q \cdot \operatorname{sgn}(\dot{q}) = \lambda((q + e \cdot \dot{q}, \dot{q}, q_l)).\end{aligned}$$

(2) Preservation of the external transition function

$$\begin{aligned}\delta_x(\delta_\emptyset(((q, \dot{q}, q_l), e), t), x) &= \delta_x(((q + (e + t) \cdot \dot{q}, \dot{q}, q_l), 0), x) = ((q + (e + t) \cdot \dot{q}, x, q_l), 0) \\ &= (\delta_{\text{ext}}((q, \dot{q}, q_l), e + t, x), 0).\end{aligned}$$

(3) Preservation of the time advance function

$$\begin{aligned}\delta_\emptyset(((q,\dot{q},q_l),e),t) &= ((q+(e+t)\cdot\dot{q},\dot{q},q_l),0) \;\Rightarrow\; ta((q,\dot{q},q_l)) - (e+t) = \frac{q_l \pm \Delta q - q}{\dot{q}} - (e+t)\\ &= \frac{q_l \pm \Delta q - q}{\dot{q}} - \frac{(e+t)\cdot\dot{q}}{\dot{q}} = \frac{q_l \pm \Delta q - q - (e+t)\cdot\dot{q}}{\dot{q}} = \frac{q_l \pm \Delta q - (q+(e+t)\cdot\dot{q})}{\dot{q}}\\ &= ta((q+(e+t)\cdot\dot{q},\dot{q},q_l)).\end{aligned}$$

(4) Composition property for $\delta_\emptyset$

$$\begin{aligned}\delta_\emptyset(\delta_\emptyset(((q,\dot{q},q_l),e),\alpha),\beta) &= \delta_\emptyset(((q+(\alpha+e)\cdot\dot{q},\dot{q},q_l),0),\beta) = ((q+(\alpha+e+\beta)\cdot\dot{q},\dot{q},q_l),0)\\ &= \delta_\emptyset(((q,\dot{q},q_l),e),\alpha+\beta).\end{aligned}$$

The aggressive and lazy DEVS integrator networks have components whose state trajectories agree at their lazy DEVS event times. A similar relationship can be established between the aggressive DEVS integrator network and the system described by Eqs. (7)–(9).

Theorem 5. *Consider the resultant of a network of aggressive DEVS integrators. The integrators are coupled through a set of memory-less functions that correspond to the rows of Az in Eq. (7). Let the function $f_d(\cdot)$ denote the dth row of Az. Let t^* denote an event time for the aggressive DEVS network, i.e., a time at which the aggressive DEVS internal transition function is evaluated. Let the state trajectory of component d be denoted by $q_d(t)$, $\dot{q}_d(t)$, and $q_{l,d}(t)$. Then*

$$q_d(t^*) = x_d(t^*), \quad q_{l,d}(t^*) = z_d(t^*), \quad \text{and} \quad \dot{q}_d(t^*) = f_d(z(t^*)) = \dot{x}_d(t^*).$$

Proof. The theorem can always be made to hold at $t = 0$ by setting $q_d(0) = x_d(0) = z_d(0) = q_{l,d}(0)$. Suppose the theorem holds at some time t^*, and consider the event at the next event time $t^* + ta(s)$.

Let component d undergo an internal event at this time. The definition of the aggressive DEVS requires that $e_d = 0$, and so

$$ta(s) = ta_d(s_d) = \begin{cases}(q_{l,d} + \Delta q - q_d)/\dot{q}, & \text{if } \dot{q} > 0,\\ (q_{l,d} - \Delta q - q_d)/\dot{q}, & \text{if } \dot{q} < 0,\\ \infty, & \text{otherwise.}\end{cases}$$

Without loss of generality, consider the case where $\dot{q} > 0$. In this case

$$q_d(t^* + ta(s)) = q_{l,d}(t^*) + \Delta q = q_d(t^*) + \frac{q_{l,d}(t^*) + \Delta q - q_d(t^*)}{\dot{q}(t^*)}\cdot\dot{q}_d(t^*) = q_d(t^*) + \int_{t^*}^{t^*+ta(s)} \dot{q}_d(t^*)\mathrm{d}\tau. \tag{12}$$

From the induction hypothesis, $\dot{q}_d(t^*) = \dot{x}_d(t^*)$, $q_{l,\,d}(t^*) = z_d(t^*)$, and $q_d(t^*) = x_d(t^*)$. Moreover, $\dot{x}_d(t)$ is constant until the next discrete change in $z(t)$, and so this time can be computed as

$$\frac{\Delta q + z_d(t^*) - x(t^*)}{\dot{x}(t^*)} = ta(s).$$

By making the appropriate substitutions into Eq. (12), it can be verified that $x_d(t^* + ta(s)) = q_d(t^* + ta(s))$. It follows immediately from this fact, Eqs. (8) and (9) that $\dot{q}_d(t^* + ta(s)) = \dot{x}_d(t^* + ta(s))$ and $q_{l,d}(t^* + ta(s)) = z_d(t^* + ta(s))$.

If d undergoes an external event at time $t^* + ta(s)$, then $q_{l,d}$ and z_d remain unchanged, and consequently, $q_{l,d}(t^* + ta(s)) = z_d(t^* + ta(s))$. It is also true that

$$x_d(t^* + ta(s)) = x_d(t^*) + \int_{t^*}^{t^*+ta(s)} \dot{x}_d(t^*)\mathrm{d}\tau = x_d(t^*) + ta(s)\cdot\dot{x}_d(t^*) = q_d(t^*) + ta(s)\cdot\dot{x}_d(t^*)$$

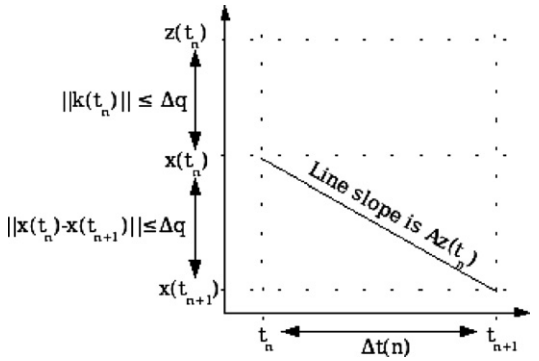

Fig. 3. Parts of Eq. (14).

and so $x_d(t^* + ta(s)) = q_d(t^* + ta(s))$. Moreover, the agreement of the imminent model state with the piecewise continuous system and property 1 ensures that $\dot{q}_d(t^* + ta(s)) = f_d(z(t^* + ta(s))) = \dot{x}_d(t^* + ta(s))$. □

5. Discrete time linear model of the DEVS integrator

The system described by Eqs. (7)–(9) can be rewritten as a time-varying linear system with bounded input. This is intuitively clear upon observing that the continuous formulation describes a sequence of line segments. Moreover, the time interval separating two different segments can be readily determined from Eq. (9) and the component-wise slope $Az(t)$ of the line segment.

These observations can be formalized in the following way. Let $\Delta t(n)$ be the time separating any two sequential discrete values of z (e.g. two adjacent points on the $z(t)$ curves in Figs. 1 and 2). To be precise, let t_n and t_{n+1} denote the times at which these sequential values occur. Then $\Delta t(n) = t_{n+1} - t_n$.

The function $z(t)$ is, by construction, constant in the interval $[t_n, t_{n+1})$. The behavior of $x(t)$ in this interval can be determined directly from Eq. (7) to be

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + \int_{t_n}^{t_{n+1}} Az(t_n)\mathrm{d}t = x(t_n) + Az(t_n)\int_{t_n}^{t_{n+1}} \mathrm{d}t = x(t_n) + (t_{n+1} - t_n)Az(t_n) \\ &= x(t_n) + \Delta t(n)Az(t_n). \end{aligned} \tag{13}$$

Letting $z(t_n) = x(t_n) + k(t_n)$, Eq. (13) can be rewritten as

$$x(t_{n+1}) = x(t_n) + \Delta t(n)Ax(t_n) + \Delta t(n)Ak(t_n) = (I + \Delta t(n)A)x(t_n) + \Delta t(n)Ak(t_n). \tag{14}$$

Eq. (9) requires that $\|k(t_n)\|_\infty \leqslant \Delta q$, and so this is a time-varying linear system with bounded input. This formulation of the system is illustrated in Fig. 3.

6. Stability of the linear system model

In everything that follows, *the matrix A is assumed to have real, negative eigenvalues*. The stability of system (13) results from the step size $\Delta t(n)$ satisfying the stability criteria

$$\Delta t(n) < \frac{2}{|\lambda_{\max}|},$$

when the system derivatives Az_n are suitably large. This forces system (13) to contract when it is away from equilibrium, just as the explicit Euler time step limit forces a simulation of a linear system to contract until it reaches equilibrium. The bound on $\Delta t(n)$ is a direct result of restricting individual components to change by no more than Δq in any step of the simulation algorithm. In this manner, the time advance is automatically adjusted in such a way that the DEVS model satisfies the explicit Euler stability criteria, and it is therefore stable.

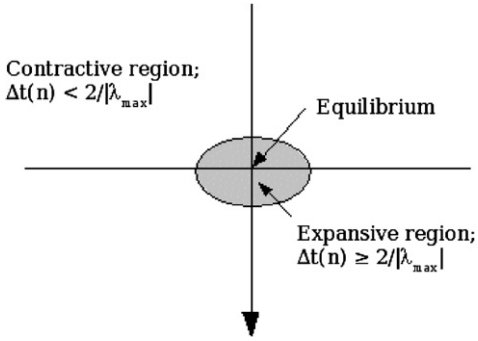

Fig. 4. Contractive and expansive regions of a two-dimensional state space.

In effect, the state space can be separated into two distinct regions. There is a contractive region in which the stability criteria is satisfied. This contractive region occupies all of the state space except for a small region near the equilibrium. Near equilibrium, there is an expansive region in which the stability criteria can be violated. The expansive region is inside of the contractive region, and this containment assures that the trajectories of (13) are bounded. Fig. 4 illustrates this partitioning of the state space.

The step size $\Delta t(n)$ for system (13) is equivalent to the time advance of the corresponding aggressive DEVS resultant. The time advance is given by Eq. (11), where the σ_d are bounded by the time advances of the individual components. The time advance of the resultant is, consequently, bounded by the time advance of its fastest component. An upper bound on the time advance of the resultant can be stated explicitly for any particular state of the resultant.

Theorem 6. *$\Delta t(n)$ is bounded by the inequality*

$$\Delta t(n) \leqslant \frac{2\Delta q}{\|Az_n\|_\infty}. \tag{15}$$

Proof. The component σ_ds are bounded by Theorem 5 and Eq. (6) as

$$\sigma_d \leqslant \frac{2\Delta q}{|(Az_n)_d|} = \frac{2\Delta q}{|\dot{q}_d|},$$

where $(Az_n)_d$ is the dth component of the vector Az_n. The time advance of the resultant is the minimum of the σ_d. It is therefore bounded by the least upper bound of the set of $\{\sigma_d | d \in D\}$. The least upper bound of this set is given by

$$\mathrm{lub}\{\sigma_d | d \in D\} \leqslant \frac{2\Delta q}{\|Az_n\|_\infty}$$

and so

$$\Delta t(n) \leqslant \mathrm{lub}\{\sigma_d | d \in D\} \leqslant \frac{2\Delta q}{\|Az_n\|_\infty},$$

which completes the proof. □

This bound on the time advance of the resultant forces the eigenvalues of $I + \Delta t(n)A$ to have magnitudes less than 1 when the system is sufficiently far from its equilibrium. More specifically, if the system is a suitable distance from equilibrium, then $\Delta t(n)$ satisfies

$$\Delta t(n) < \frac{2}{|\lambda_{\max}|},$$

where $|\lambda_{\max}|$ is the largest magnitude of the eigenvalues belonging to A.

Theorem 7. *If*

$$\|Az(t_n)\|_\infty > \Delta q|\lambda_{\max}|,$$

then

$$\Delta t(n) < \frac{2}{|\lambda_{\max}|}.$$

Proof. Let

$$\|Az(t_n)\|_\infty > \Delta q|\lambda_{\max}|.$$

Inequality (15) can be rearranged to show that

$$\frac{2\Delta q}{\Delta t(n)} \geqslant \|Az_n\|_\infty.$$

Therefore,

$$\frac{2\Delta q}{\Delta t(n)} > \Delta q|\lambda_{\max}|$$

and it follows that

$$\Delta t(n) < \frac{2}{|\lambda_{\max}|}. \qquad \square$$

The link between explicit Euler stability (see, e.g. [26], or any introductory numerical analysis textbook) and the stability of the DEVS integrator network can be stated explicitly as a corollary to Theorem 7.

Corollary 8. *Let h_{limit} be the maximum time step in a stable, explicit Euler simulation of a stable linear system. If $\|Az_n\|_\infty > \Delta q|\lambda_{\max}|$, then $\Delta t(n) < h_{\text{limit}}$.*

It follows Theorem 7 that Eq. (14) has ultimately bounded trajectories.

Theorem 9. *Consider the system described by Eq. (14). It for all n*

$$\Delta t(n) < \frac{2}{|\lambda_{\max}|},$$

then

$$\|x(t_n)\|_\infty < \frac{2\Delta q\beta}{|\lambda_{\max}|}\|A\|_\infty$$

as $n \to \infty$. The constant $0 < \beta < \infty$ is specific to the system under consideration.

Proof. Let $\phi(t_n, t_m) = (I + \Delta t(n-1)A)(I + \Delta t(n-2)A)\cdots(I + \Delta t(m)A)$, or, more concisely

$$\phi(t_n, t_m) = \prod_{p=1}^{n-m}(I + \Delta t(n-p)A).$$

It is well known from the theory of linear systems (see, e.g. [21]) that

$$x(t_n) = \phi(t_n, 0)x(0) + \sum_{q=0}^{n-1}(\phi(t_n, t_{q+1})\Delta t(q)Ak(t_q)).$$

It follows from the hypothesis and well known properties of the infinity-norm that

$$\begin{aligned}\|x(t_n)\|_\infty &\leqslant \|\phi(t_n,0)x(0)\|_\infty + \sum_{q=0}^{n-1}(\|\phi(t_n,t_{q+1})\Delta t(q)Ak(t_q)\|_\infty)\\ &\leqslant \|\phi(t_n,0)x(0)\|_\infty + \sum_{q=0}^{n-1}(\|\phi(t_n,t_{q+1})\|_\infty \cdot |\Delta t(q)| \cdot \|A\|_\infty \cdot \|k(t_q)\|_\infty)\\ &< \|\phi(t_n,0)x(0)\|_\infty + \frac{2\Delta q}{|\lambda_{\max}|}\|A\|_\infty \sum_{q=0}^{n-1}\|\phi(t_n,t_{q+1})\|_\infty.\end{aligned}$$

The upper bound on $\Delta t(n)$ constrains the eigenvalues of $I + \Delta t(n)A$ to have magnitudes less than 1. Consequently, as $n \to \infty$, $\|\phi(t_n,0)x(0)\|_\infty$ vanishes and

$$\sum_{q=0}^{n-1}\|\phi(t_n,t_{q+1})\|_\infty < \infty.$$

Moreover,

$$\frac{\|\phi(t_n,t_k)\|_\infty}{\|\phi(t_n,t_{k-1})\|_\infty} < 1$$

and so, by the ratio test, the sum converges as $n \to \infty$. Let β be this limit to complete the proof. □

Theorem 9 shows that $x(t)$ is ultimately contained in a bounded region, provided that $\|Az(t_n)\|_\infty$ is sufficiently large, or, more specifically, if it satisfies Theorem 7. It remains to show that violations of Theorem 7 cannot persist. That is, if a system enters the expansive region, then it must remain in that region (e.g. if $z(t) = 0$) or once again enter the contractive region.

Suppose that, for some t_m, the hypothesis of Theorem 9, and consequently, Theorem 7, fails to hold. That is, for some time t_m

$$\|Az(t_m)\|_\infty \leqslant \Delta q|\lambda_{\max}|. \tag{16}$$

If inequality (16) is satisfied at all times subsequent to t_m (e.g. if $z(t_m) = 0$) then (16) describes a bound on $x(t)$ that is proportional to Δq. More specifically, $\forall n > m$, $x(t_n)$ must satisfy

$$\|A(x(t_n) + k(t_n))\|_\infty \leqslant \Delta q|\lambda_{\max}|,$$

where Eq. (9) constrains $k(t_n)$ such that $\|k(t_n)\|_\infty \leqslant \Delta q$.

Otherwise, for some $n > m$, the hypothesis of Theorem 9 holds. In this case, $x(t_n)$ is bounded by the region described in Theorem 9 or the smallest value of $\|x(t)\|_\infty$ for which Theorem 7 holds. More specifically, there is a constant $K > 1$ such that

$$\|A(x(t_n) + k(t_n))\|_\infty \leqslant K\Delta q|\lambda_{\max}|.$$

In either case, $\|x(t_n)\|_\infty$ is ultimately contained in a region whose dimensions are proportional to Δq. The conclusion of this argument is restated as Theorem 10.

Theorem 10. *Consider the system described by Eq. (14). As $t_n \to \infty$, $x(t_n)$ must satisfy*

$$\|A(x(t_n) + k(t_n))\|_\infty \leqslant K\Delta q|\lambda_{\max}|,$$

where $K > 1$, or

$$\|x(t_n)\|_\infty < \frac{2\Delta q\beta}{|\lambda_{\max}|}\|A\|_\infty,$$

where $0 < \beta < \infty$ is a constant specific to the system under consideration.

Theorem 7 defines the outer edge of the expansive region, and (9) gives a conservative estimate of the inner edge of the contractive region. Theorem 10 states that the system must ultimately be trapped within one of these bounds. In particular, the system becomes trapped within the expansive region if it happens to reach equilibrium. Otherwise, the system will wander in an area contained within the conservative bound on the inner edge of the contractive region.

7. Comparing QSS and forward Euler

A synchronous form of the first order discrete event method can be constructed by updating every state variable simultaneously. This derived method uses an adaptive time step that is chosen so that the fastest component changes its magnitude by Δq.[3]

This method simulates a set of ordinary differential equations in the form

$$\dot{x} = f(x)$$

with a recursive function

$$x_{n+1} = x_n + h(n) f(x_n), \quad \text{where } h(n) = \frac{\Delta q}{\|f(x_n)\|_\infty}. \tag{17}$$

The method stops if $\|f(x_n)\|_\infty = 0$ (i.e., if an equilibrium is reached).

When simulating a linear system $\dot{x} = Ax$, Eq. (17) becomes

$$x_{n+1} = x_n + \frac{\Delta q}{\|Ax_n\|_\infty} x_n = \left(I + \frac{\Delta q}{\|Ax_n\|_\infty}\right) x_n.$$

This system is the same as system (13) with $k_n = 0$. The step size at the nth step is

$$h(n) = \frac{\Delta q}{\|Ax_n\|_\infty},$$

where (13) has a step size

$$\Delta t(n) < \frac{2\Delta q}{\|Az_n\|_\infty}.$$

Within trajectory segments that are monotonically increasing or decreasing, $\Delta t(n)$ will, in fact, be bounded by $\Delta q/\|Az_n\|_\infty$. The $2\Delta q$ term applies only when hysteresis can cause the system to move a distance further than Δq between events (see [1]).

The numerical properties of the synchronous DEVS integrator are similar to those of the asynchronous form. The synchronous integrator will, in general, require fewer simulation cycles (i.e., fewer state changes of the resultant) to cover an interval of time. However, it will update every component at each simulation cycle. Consequently, the total number of component updates could be (very) large when compared with the asynchronous scheme, even if the asynchronous scheme requires more simulation cycles. The synchronous integrator will also create smaller errors than the asynchronous form because it forces the slower components to update at state space intervals smaller than Δq.

The asynchronous form trades error for, potentially, reduced execution time. If individual components are loosely coupled[4] and their rates vary widely, then the asynchronous form could require significantly fewer component updates than the synchronous form. The additional errors in the asynchronous method result from evaluating slower components less often than would be done using the synchronous method.

[3] The synchronous DEVS model ensures that every component sees the most recently computed state values for its neighbors. This is unlike the aggressive DEVS model in which components only see new neighbor values when the neighbors q variable reaches a threshold crossing (i.e., $|q - q_l| = \Delta q$).

[4] By *loosely coupled* we mean that the number of integrator inputs influenced by an integrator output is small relative to the dimension of the network. In linear systems, this is caused by a sparse transition matrix. An example will be given in Section 8.

Table 1
Relative errors of synchronous and asynchronous DEVS methods with respect to explicit Euler

Euler time step	Relation to maximum DEVS error
$\min\{h(n)\}$	Maximum Euler error < maximum DEVS error
$\max\{h(n) \mid h(n) < 2/\lvert\lambda_{\max}\rvert\}$	Maximum Euler error > maximum DEVS error

The synchronous integrator is useful for comparing the relative properties of discrete event and other fixed and adaptive time stepping methods. It closely mimics the error and stability properties of the asynchronous model. If the synchronous model has error, stability, and/or performance properties that look attractive relative to other methods, then the asynchronous model could provide an additional performance boost via its locally adaptive time advance. Similarly, if the synchronous model looks unattractive for some particular problem, then the extra performance obtained with the asynchronous method will need to be very significant for it to be considered a viable alternative.

Eq. (17) is an adaptive form of the explicit Euler method

$$x_{n+1} = x_n + hf(x_n). \tag{18}$$

If h is fixed to be the smallest $h(n)$ observed over a simulation interval (e.g. as is done in [5]), then Eq. (18) will require a larger number of steps than both the synchronous and asynchronous DEVS methods. However, Eq. (18) will use smaller time steps to evaluate slower dynamics, thereby giving it better accuracy than either adaptive method. If h is fixed to be the largest $h(n) < 2/|\lambda_{\max}|$, then Eq. (18) will need fewer steps than the adaptive methods but will, in general, exhibit larger errors.

The errors exhibited by the asynchronous method will, in general, be larger than those of the synchronous method. However, for systems with loosely coupled components and a large range of rates, the asynchronous method may perform better by reducing the total number of component updates.

These observations, applied to linear systems, are summarized in Table 1. Where the time advance of the DEVS schemes is limited by Theorem 7, the explicit Euler method can be used to bound the errors that will be observed in a first order discrete event simulation of a linear system. The largest explicit Euler error that occurs when using a time step $\min\{h(n)\}$ gives a lower bound on the maximum error that will be observed in the discrete event simulation. The largest explicit Euler error that occurs when using a time step $\max\{h(n) \mid h(n) < 2/|\lambda_{\max}|\}$ gives an upper bound on the maximum error that will be observed in the discrete event simulation.

8. Examples

The examples in this section demonstrate the error and stability properties developed in the previous sections. It also highlights the computational advantage of the first order QSS scheme, relative to the other first order methods described in Section 7, for simulating large, loosely coupled systems. Specifically, the first order QSS method generates errors comparable to other first order accurate methods while realizing a substantial reduction in computational costs for large, loosely coupled systems. We anticipate that this advantage will carry over to higher order QSS schemes (see, e.g. [2,10,11,13]).

Consider the system $\dot{x} = -\alpha x$, where $\alpha > 0$. This system has a single state variable. The synchronous and asynchronous DEVS methods are identical for a system with a single variable. A (synchronous or asynchronous) discrete event simulation of this system is described exactly by

$$x_{n+1} = x_n - h(n)\alpha x_n, \quad \text{where } h(n) = \frac{\Delta q}{|\alpha x_n|}.$$

Whenever $|x_n| \geqslant \Delta q/2$, then $h(n) \leqslant 2/\alpha$. Noting that α is the eigenvalue of this system, we see that a discrete event simulation of this system is stable regardless of the choice of Δq. The first order quantization method automatically adjusts its time step to maintain stability.

The relative errors in a discrete event and fixed step explicit Euler simulation of this system are shown in Fig. 5. This example was computed using $x_0 = 1$, $\alpha = 1$, and $\Delta q = 0.1$. The simulation is stopped at the first step after the discrete event scheme terminates at time $t \approx 3$.

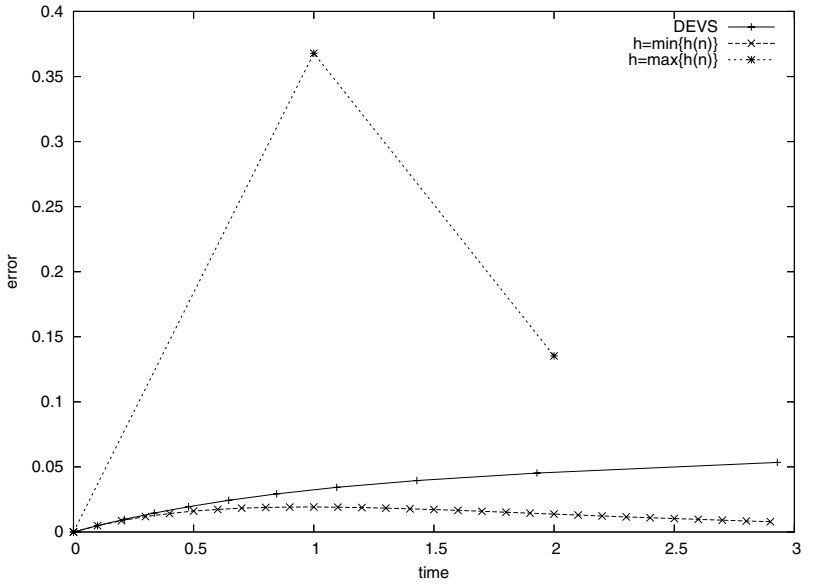

Fig. 5. Comparison of errors produced by DEVS and explicit Euler when simulating $\dot{x} = -x$.

Fig. 5 plots the errors in the discrete event scheme, explicit Euler with $h = \min\{h(n)\}$ and $h = \max\{h(n)\}$. The discrete event scheme requires 10 steps to complete the simulation; with $h = \min\{h(n)\} = 0.1$, explicit Euler requires 30 steps; with $h = \max\{h(n)\} = 1$, explicit Euler requires three steps.

Consider the system

$$A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}. \tag{19}$$

The phase plot of this system, starting from initial state $x = [\,1 \quad 1\,]^{\mathrm{T}}$ and simulated with $\Delta q = 0.1$, is shown in Fig. 1. Fig. 6 shows the contractive and expansive regions for this system with $\Delta q = 0.1$. This bound describes the z vectors that satisfy the inequality $\|Az\|_\infty < \Delta q|\lambda_{\max}| = 0.2$. This inequality is satisfied when

$$|-z_1 + z_2| < 0.2,$$
$$|-2z_2| < 0.2.$$

Fig. 7 shows the time advance of the resultant as a function of time during a simulation of this system. Fig. 8 shows the output trajectories z_n for the same simulation using the asynchronous DEVS integrator. The explicit Euler stability limit for this system is 1. The time advance of the resultant satisfies this whenever the system state is outside of the expansive region.

Figs. 9 and 10 show a comparison of the errors in the simulated trajectories of this two state variable system for four different integration methods; the synchronous DEVS, asynchronous DEVS, explicit Euler with h equal to the smallest time advance of the synchronous DEVS, and with h equal to the largest stable time advance of the synchronous DEVS. In this example, the smallest time advance of the synchronous DEVS is 0.05 and the largest is 0.721397. The expected ranking with respect to observed errors appears can be clearly observed; i.e., explicit Euler with small time step, synchronous DEVS, asynchronous DEVS, and then explicit Euler with the largest time step. The largest observed errors with each method are shown in Table 2.

The total number of steps (i.e., simulation cycles) and total number of component updates for each method are summarized in Table 3. The component update count for the Euler simulations and synchronous DEVS simulation can be computed by multiplying the step count by the number of components. The component update count for the asynchronous method is the sum of the number of internal and external transition function evaluations. In this case, there are only two state variables, and therefore, the asynchronous method has a very difficult time exploiting the different component rates.

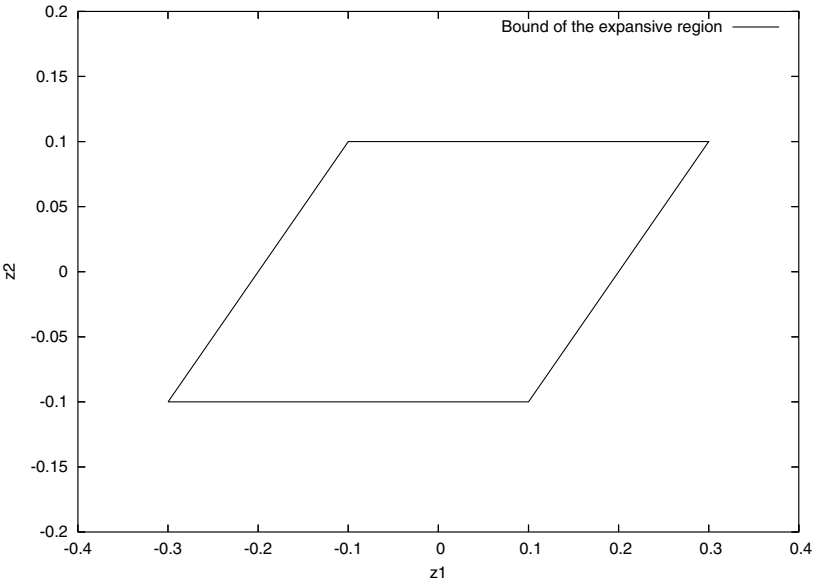

Fig. 6. Expansive and contractive regions for (19) with $\Delta q = 0.1$.

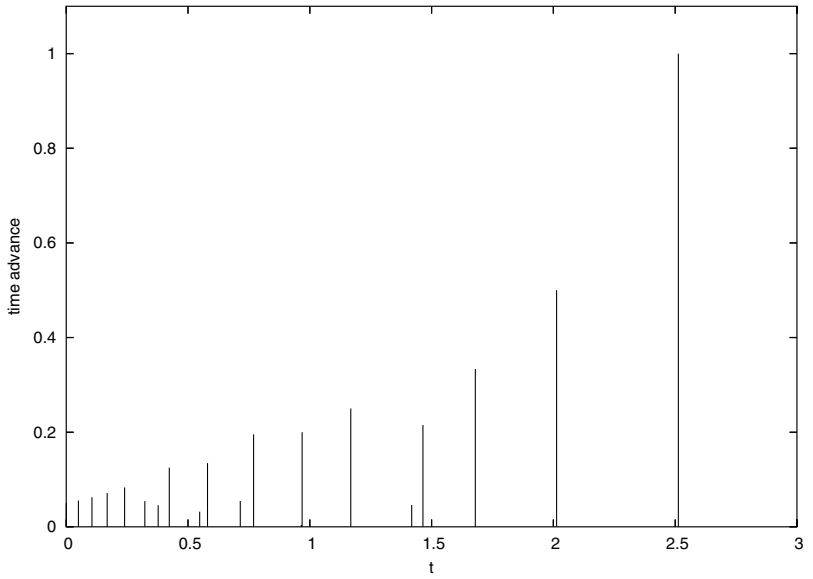

Fig. 7. Time advance of the resultant for (19) using $\Delta q = 0.1$.

Lastly, consider a large, loosely coupled system[5] with N components whose derivatives are

$$\dot{x}_i = \begin{cases} x_{i-1} - 2x_i + x_{i+1}, & \text{if } 1 < i < N, \\ x_{N-1} - 2x_N + x_1, & \text{if } i = N, \\ x_N - 2x_1 + x_2, & \text{if } i = 1. \end{cases} \tag{20}$$

[5] The system is loosely coupled because each integrator updates itself and its two nearest neighbors when an output is generated.

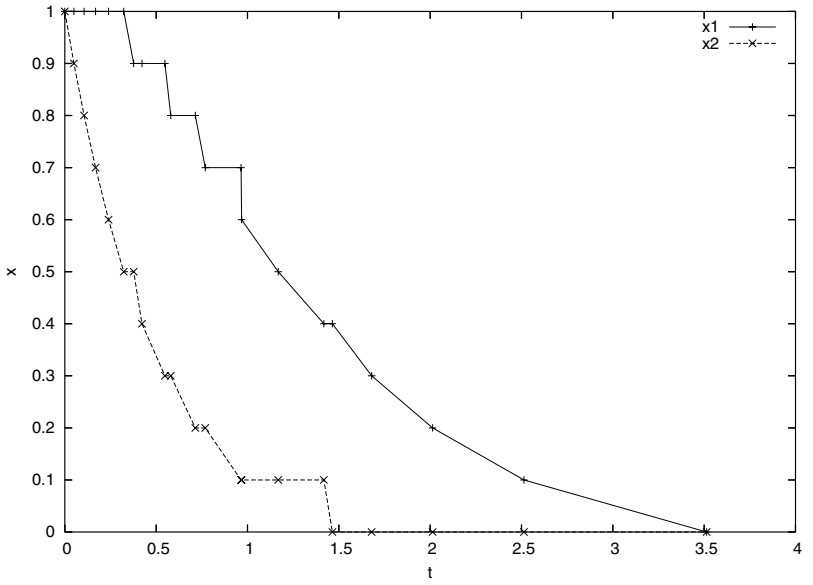

Fig. 8. Simulated trajectories for (19) using $\Delta q = 0.1$.

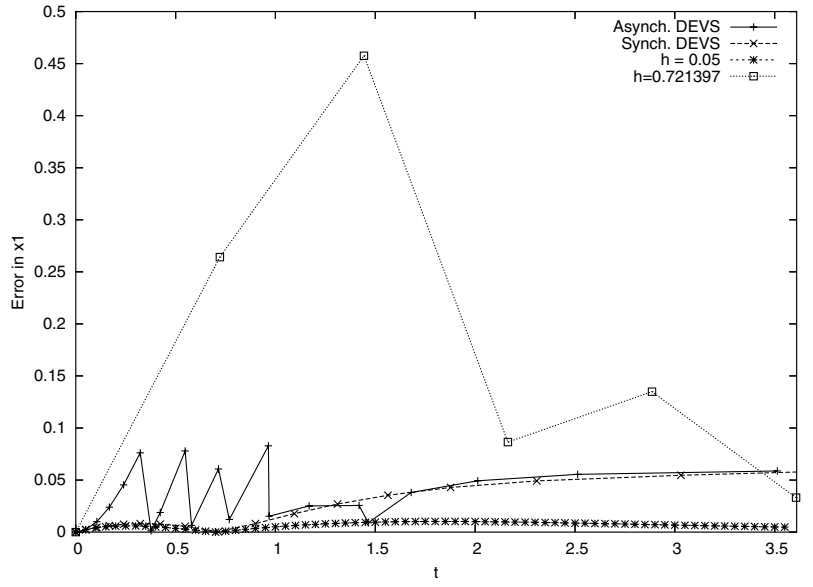

Fig. 9. Error in variable x_1 of system (19).

This system is similar to the heat equation with periodic boundary conditions if the spatial derivatives are approximated by finite differences. An explicit Euler simulation of this system is stable if $h < 0.5$. For the sake of illustration, the initial conditions are taken to be

$$x_i(0) = \begin{cases} 1, & \text{if } i = N/2, \\ 0, & \text{otherwise.} \end{cases}$$

This choice ensures that the system has relatively wide a range of local rates.

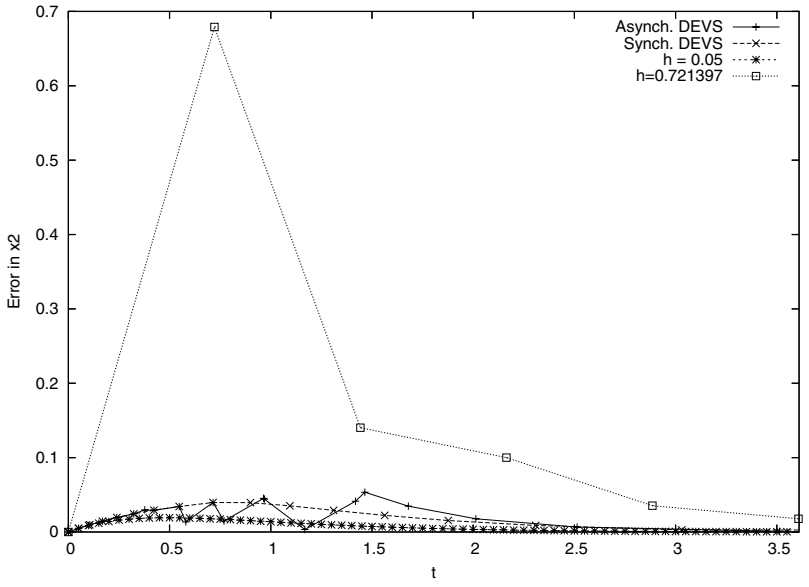

Fig. 10. Error in variable x_2 of system (19).

Table 2
Maximum observed errors

Method	Maximum x_1 error	Maximum x_2 error
Euler with $h_{\min}$	0.0102459	0.019201
Synch. DEVS (contractive region)	0.0491482	0.039556
Asynch. DEVS (contractive region)	0.0829385	0.0587184
Euler with $h_{\max}$	0.457539	0.679061

Table 3
Simulation steps and component update counts for system (19)

Method	Simulation steps	Component updates
Euler with $h = \min\{h(n)\}$	71	142
Euler with $h = \max\{h(n)\}$	5	10
Synch. DEVS	16	32
Asynch. DEVS	20	32

In this context, the potential for the asynchronous DEVS to reduce the number of actual component evaluations is demonstrated by simulating Eq. (20) with $\Delta q = 0.1$ and several choices for N. When N is suitably large, in this case $N \geqslant 50$, the asynchronous DEVS model has a time advance of ∞ at time 26.2023. Moreover, every method uses a fixed number of steps in that time interval. Table 4 shows the number of steps and component updates for each simulation method as N varies. For this model, $h_{\min} = 0.05$ and $h_{\max} = 0.498787$.

Fig. 11 shows $h(n)$, $\|\dot{x}\|_\infty$, the maximum stable time step, and outer edge of the contractive region for the 50 state variable version of Eq. (20). Note that $h_{\max}$ is *not* the largest time advance of the synchronous model; only the largest stable time advance. The largest time advance is 2.92591, with other time advances greater than 0.5 occurring throughout the simulation. However, these time advances always occur when $\|\dot{x}\|_\infty \leqslant 0.199156$. The largest eigenvalue magnitude in system (20) is 4. The expansive region begins when the largest derivative is smaller than $4\Delta q = 0.4$. Notice that $\|\dot{x}\|_\infty \leqslant 0.199156 < \Delta q|\lambda_{\max}| = 0.4$.

Table 4
Simulation steps/component update counts for system (20)

N	$h = \min\{h(n)\}$	$h = \max\{h(n)\}$	Synch. DEVS	Asynch. DEVS
50	525/26,250	53/2650	48/2400	67/536
500	525/262,500	53/26,500	48/24,000	67/986
5000	525/2,625,000	53/265,000	48/240,000	67/5486
50000	525/26,250,000	53/2,650,000	48/2,400,000	67/50,486

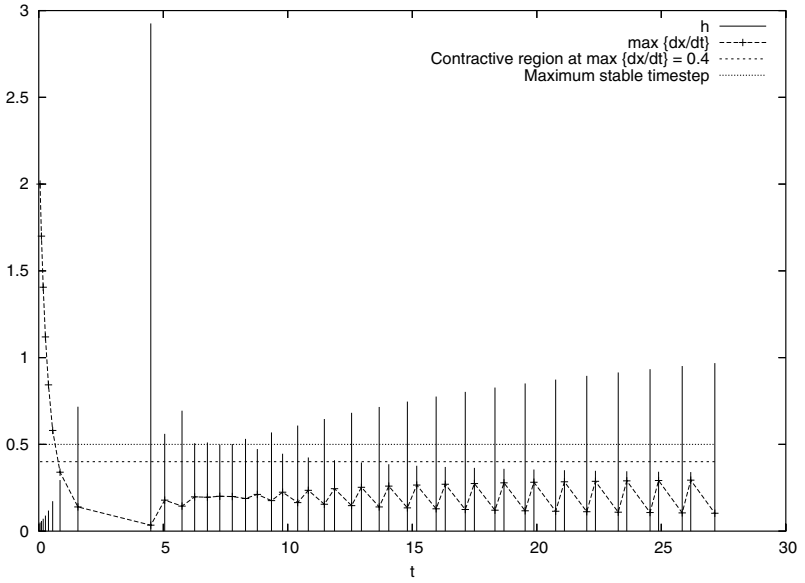

Fig. 11. Maximum derivative and steps size selection for a synchronous DEVS simulation of Eq. (20) with $N = 50$.

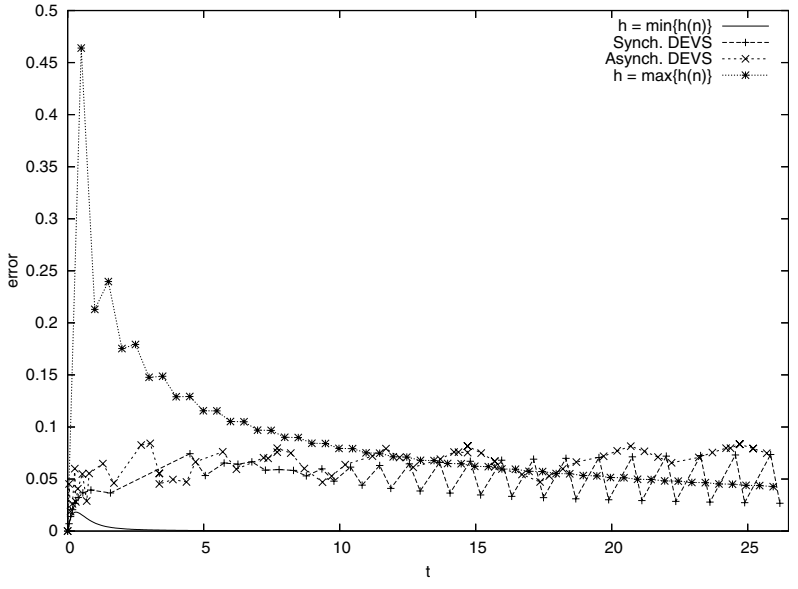

Fig. 12. Maximum error as a function of time for a simulation of (20) with $N = 50$.

Fig. 12 shows the maximum error created by each simulation scheme as a function of time. The expected ranking of maximum errors can be readily observed. The Euler scheme with minimum time step has a maximum error of 0.0182232, the synchronous DEVS scheme 0.0743780, the asynchronous DEVS scheme 0.0842123, and Euler with maximum stable time step has a maximum error of 0.4639601.

For large systems with many loosely coupled components that have varying rates, the QSS method realizes a substantial reduction in the number of calculations required to complete a simulation. The small, two variable system demonstrates that the additional error introduced by updating individual components asynchronously is small. For the two variable system, an (internal) event in one component causes an (external) event in the other component, and so there is no hope of an actual performance improvement. None the less, it is indicative of the modest additional error introduced by asynchrony.

9. Conclusions

The theory developed in this paper complements previous studies of the stability of linear quantized state systems. When the A matrix satisfies typical stability constraints, this class of systems exhibits four interesting properties:

(1) Trajectories are bounded,
(2) they can be made arbitrarily close to those of a continuous system $\dot{x} = Ax$,
(3) all trajectories are ultimately periodic or, after some time, $x(t) = 0$, and
(4) switching times (i.e., the time advance of the resultant) are constrained in such a way that the system is self-stabilizing.

This paper contributes the last item to this litany of facts. By doing so, we show that the first order QSS integration scheme satisfies a well-established stability constraint for its most closely related time stepping method. This constraint is satisfied automatically by the QSS, without any explicit attempt to control error or maintain stable operation. Our result suggests that any explicit, discrete event integration schemes will satisfy classical stability constraints (as described, e.g. in [26,27]).

These two properties – self-stabilization and absence of global synchronization points – make the proposed asynchronous scheme, and its higher order accurate relatives, attractive for simulating many types of spatially distributed systems. There are two main advantages in this context. First, the method provides a simple algorithm that intrinsically tracks the active computational domain (see, e.g. [4,5]). Second, the algorithm can be parallelized using scalable parallel discrete event simulation algorithms (see, e.g. [7–9]).

Of the two advantages, the second is perhaps most significant and, certainly, most distinguishes synchronous and asynchronous methods. By avoiding any kind of global considerations to ensure overall stability, asynchronous schemes offer the performance advantage of locally adaptive time stepping with very little computational overhead. Moreover, this provides an immediate segue into massively parallel computing through the use of well established parallel discrete event simulation algorithms (see, e.g. [20]).

References

[1] E. Kofman, Discrete event simulation of hybrid systems, SIAM Journal on Scientific Computing 25 (5) (2004) 1771–1797.
[2] J. Nutaro, Constructing multi-point discrete event integration schemes, in: Proceedings of the 37th Winter Simulation Conference, Winter Simulation Conference, 2005, pp. 267–273.
[3] B.P. Zeigler, H. Sarjoughian, H. Praehofer, Theory of quantized systems: DEVS simulation of perceiving agents, Cybernetics and Systems 31 (6) (2000) 611–647.
[4] A. Muzy, A. Aiello, P.-A. Santoni, B.P. Zeigler, J.J. Nutaro, R. Jammalamadaka, Discrete event simulation of large-scale spatial continuous systems, Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, IEEE, Hawaii, USA, 2005, pp. 2991–2998.
[5] J.J. Nutaro, B.P. Zeigler, R. Jammalamadaka, S.R. Akerkar, Discrete event solution of gas dynamics within the DEVS framework, in: P.M.A. Sloot, D. Abramson, A.V. Bogdanov, J. Dongarra, A.Y. Zomaya, Y.E. Gorbachev (Eds.), International Conference on Computational Science, Lecture Notes in Computer Science, vol. 2660, Springer, Melbourne, Australia, 2003, pp. 319–328.
[6] G.A. Wainer, N. Giambiasi, Cell-DEVS/GDEVS for complex continuous systems, Simulation 81 (2) (2005) 137–151.

[7] H. Karimabadi, J. Driscoll, Y. Omelchenko, N. Omidi, A new asynchronous methodology for modeling of physical systems: breaking the curse of courant condition, Journal of Computational Physics 205 (2) (2005) 755–775.
[8] J. Nutaro, Parallel discrete event simulation with application to continuous systems, Ph.D. thesis, University of Arizona, Tuscon, Arizona, 2003.
[9] Y. Tang, K. Perumalla, R. Fujimoto, H. Karimabadi, J. Driscoll, Y. Omelchenko, Optimistic parallel discrete event simulations of physical systems using reverse computation, in: Proceedings of the 2005 Workshop on Principles of Advanced and Distributed Simulation (PADS 2005), 2005, pp. 26–35.
[10] E. Kofman, A third order discrete event method for continuous system simulation. Part I: theory, Tech. Rep. LSD0501, School of Electronic Engineering, Universidad Nacional de Rosario, Rosario, Argentina, 2005.
[11] E. Kofman, A third order discrete event method for continuous system simulation. Part II: applications, Tech. Rep. LSD0502, School of Electronic Engineering, Universidad Nacional de Rosario, Rosario, Argentina, 2005.
[12] J. Nutaro, 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), San Diego, CA, 2007, pp. 25–31.
[13] E. Kofman, A second-order approximation for DEVS simulation of continuous systems, Simulation 78 (2) (2002) 76–89.
[14] N. Lynch, R. Segala, F. Vaandrager, Hybrid I/O automata, Information and Computation 185 (1) (2003) 105–157.
[15] A.S. Matveev, A.V. Savkin, Qualitative Theory of Hybrid Dynamical Systems, Birkhauser, Boston, 2000.
[16] A. Logg, Multi-adaptive Galerkin methods for ODEs I, SIAM Journal on Scientific Computing 24 (6) (2002) 1879–1902.
[17] J.M. Esposito, V. Kumar, An asynchronous integration and event detection algorithm for simulating multi-agent hybrid systems, ACM Transactions on Modeling and Computer Simulation 14 (4) (2004) 363–388.
[18] A. Lew, J. Marsden, M. Ortiz, M. West, Asynchronous variational integrators, Archive for Rational Mechanics and Analysis 167 (2003) 85–146.
[19] M. Crouzeix, F. Lisbona, The convergence of variable-stepsize, variable-formula, multistep methods, SIAM Journal on Numerical Analysis 21 (3) (1984) 512–534.
[20] R.M. Fujimoto, Parallel and Distributed Simulation Systems, Wiley-Interscience, 1999.
[21] F. Szidarovszky, A.T. Bahill, Linear Systems Theory, second ed., CRC Press LLC, Boca Raton, Florida, 1998.
[22] E. Kofman, Non-conservative ultimate bound estimation in LTI perturbed systems, Automatica 41 (10) (2005) 1835–1838.
[23] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of Modeling and Simulation, second ed., Academic Press, San Diego, CA, 2000.
[24] E. Kofman, S. Junco, Quantized-state systems: a DEVS Approach for continuous system simulation, Transactions of the Society for Computer Simulation International 18 (3) (2001) 123–132.
[25] J. Zhang, K.H. Johansson, J. Lygeros, S. Sastry, Dynamical systems revisited: hybrid systems with Zeno executions, in: Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control (HSCC 2000), Springer-Verlag, London, UK, 2000, pp. 451–464.
[26] A. Ralston, P. Rabinowitz, A First Course in Numerical Analysis, second ed., Dover Publications, Mineola, New York, 1978.
[27] G. Strang, Introduction to Applied Mathematics, Wellesley-Cambridge Press, Wellesley, MA, 1986.