Analysis of unilateral event-continuous systems

Yu V Shornikov\textsuperscript{1} and E A Popov\textsuperscript{2}

Automated Control Systems Department, Novosibirsk State Technical University, 20 Prospekt K. Marksa, Novosibirsk, 630073, Russia

E-mail: shornikov@inbox.ru\textsuperscript{1}, e.popov@corp.nstu.ru\textsuperscript{2}

Abstract. Numerical analysis of unilateral event-continuous processes in comparison with traditional simulation algorithms for discrete-continuous systems is addressed. An original integration step size control algorithm guaranteeing correct detection of unilateral events is given. The algorithm is tested on a typical problem. Simulations are carried out in modern computer environments for modeling and simulation of complex dynamic systems of the mentioned class. A comparative analysis of the application of the software environments to event-continuous processes with unilateral events is presented.

1. Introduction

Complex systems with dynamic processes involving both discrete and continuous components are faced in many research and engineering areas. Analyzing such a system, one has to take into account the features of the system behavior on the discontinuity surface. In the modern terminology, systems with these properties are referred to as event-continuous systems (ECSs) \cite{1–3}. Contrary to traditional discrete-continuous systems, ECSs are studied using modern numerical analysis methods, which are emphasized in this paper. Unique characteristics of ECSs limit the application of traditional symbolic analysis methods. Therefore, systems of this kind have to be studied with original numerical algorithms implemented in simulation environments.

ECSs operate in different modes depending on certain conditions. Each local mode is described by a mathematical model of a given class. For the sake of simplicity, without loss of generality, let an ECS mode behavior belong to the class of explicit systems of ordinary differential equations

\begin{equation}
y' = f\left(t, y\right), \quad y(t_0) = y_0, \quad t \in [t_0, t^*],
\end{equation}

where $y \in \mathbb{R}^N$ is the state vector, $f: \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a nonlinear vector function satisfying the Lipschitz conditions over $t \in [t_0, t^*]$, $y_0 \in \mathbb{R}^N$ is the vector of initial conditions, $t^*$ is the mode's end time or the switching time.

The conditions of a local mode's existence are defined by a so-called event function $g\left(t, y\right): \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}^s$. Hence, the global behavior of an ECS with (1) is

\begin{equation}
y' = f\left(t, y\right), \quad y(t_0) = y_0, \\
pr : g\left(t, y\right) < 0, \quad t \in [t_0, t_e],
\end{equation}

where $pr$ is the condition for the event occurrence.
where \( f(t, y): \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N \) is a nonlinear vector function satisfying the Caratheodory conditions over the simulation interval \([t_0, t_k]\), \( g(t, y): \mathbb{R}^N \rightarrow \mathbb{R}^s \) is a differentiable nonlinear vector function, \( pr: \mathbb{R}^N \rightarrow \{false, true\} \), \( t_k > t^* \) is the simulation's end time.

Note that the right-hand sides of local modes have discontinuities of the first kind at the transition times \( t^* \). That is why the Lipschitz conditions do not hold, and the application of traditional analysis methods to systems (2) is limited. Correct computation of the event times \( t^* \), which determine the solution of system of equations (2), is associated with some difficulties. This complicated problem is solved differently in different simulation environments [3–6]. More challenging is simulation of a problem with unilateral events, when

\[
\lim_{n \to \infty} g(t_n, y_n) = 0, \quad n = 1, 2, \ldots \quad (3)
\]

Condition of unilateral events (3) along with the mode predicate in (2) are the joint conditions of a mode change.

The ISMA environment [7] designed for modeling and simulation of hybrid dynamical systems from different applications ensures satisfaction of (3) by controlling the integration step size according to the formula

\[
h_{n+1} = (\gamma - 1) g_n \left[ \frac{\partial g_n}{\partial y_n} \cdot f_n + \frac{\partial g_n}{\partial t} \right]^{-1}, \quad \gamma \in [0, 1),
\]

where \( g_n = g(t_n, y_n) \), \( f_n = f(t_n, y_n) \).

Let us consider a typical ECS of class (2), (3).

2. Test problem

There is a ball perfectly elastic bouncing off a perfectly rigid surface. Let the initial position of the ball above the surface be denoted by \( H \). The system of equations involving the vertical velocity projection \( v_y \) and the distance to the surface \( y \) is

\[
y' = v_y, \quad y(t_0) = H, \quad v'_y = -q, \quad v_y(t_0) = 0,
\]

\[
\text{if} \ (y \leq 0 \ \text{and} \ v_y < 0) \ \text{then} \ v_y = -v_y,
\]

where \( q \) is the gravitational acceleration.

This model is a typical example of an ECS with unilateral events and instantaneous changes of state variables every time the ball hits the bounce surface, when \( y \leq 0 \) and \( v_y < 0 \). An ECS mode can be constrained by one or several conditions. In the problem at hand, the mode Flying corresponds to two simultaneously true conditions: 1) the vertical velocity projection is not negative, that is \( g_1(t, v_y) = -v_y < 0 \); 2) the distance to the surface is positive, that is \( g_2(t, v_y) = -y < 0 \). Hence, the two components of the event function satisfy condition of unilateral events (3), which must not be violated according to definition (4).

Let us model and simulate system (4) in different dynamic process modeling and simulation environments using 4th order integration algorithms and conduct a comparative analysis of the obtained simulation results. In order to simplify the comparison of numerical errors, we can assign the initial conditions such values that the time of falling to the surface would be one unit of time. This goal is achieved by setting \( H = q/2 \).
Indeed, the local behavior of (4) is of the form

\[ y' = v_y, \quad y(t_0) = q / 2, \]
\[ v' = -q, \quad v_y(t_0) = 0. \]

Hence, \[ v_y(t) = -qt + c = \text{const}. \] For \( t_0 = 0 \) the solution to (5) is

\[ v_y(t) = -qt, \]
\[ y(t) = \frac{q}{2} (1 - t^2), \quad t \in [0, t^*_1]. \]

At the event time, \( y(t^*_1) = 0 \). From the equation \( \frac{q}{2} (1 - (t^*_1)^2) = 0 \), we obtain \( t^*_1 = 1 \).

When the bounce occurs, the velocity's direction instantly changes to the opposite one, and the next local behavior is therefore

\[ y' = v_y, \quad y(t^*_1) = 0, \]
\[ v' = -q, \quad v_y(t^*_1) = q. \]

The solution to new local mode (6) is

\[ v_y(t) = q(2 - t), \]
\[ y(t) = -\frac{q}{2} t^2 + 2qt - \frac{3q}{2}, \quad t \in [t^*_1, t^*_2]. \]

where \( t^*_2 \) is the time of the next bounce, when \( y(t^*_2) = 0 \). Then, from the equation

\[ -\frac{q}{2} (t^*_2)^2 + 2qt^*_2 - \frac{3q}{2} = 0, \]

we have \( t^*_2 = 3 \) and so on.

3. Simulation results

A textual program model of (4) in the basic input language of ISMA (LISMA) is given in figure 1.

```plaintext
const q = 0.81;
y' = vy;
y(t0) = q / 2.0;
v' = -q;
v(t0) = 0.0;
state flying(y <= 0 AND vy < 0)
{
    set vy = -vy;
}
from init, flying;
```

*Figure 1.* Program model in LISMA.
Figure 2 depicts the simulation results obtained over the whole simulation interval $[0; 20]$ using a modified 4th order Runge-Kutta-Merson algorithm with the minimum step size $h_{\text{min}} = 10^{-12}$, maximum step size $h_{\text{max}} = 10^{-2}$, accuracy $\varepsilon_{\text{tol}} = 10^{-4}$.

Figure 3 demonstrates the solution in the neighborhood of the first bounce time. Note that the bounce occurs above the surface. This is due to the event detection algorithm decreasing the step size according to the event function behavior as the ball approaches the ground, which ensures asymptotic approaching to the mode surface without crossing it.

SimInTech is a software suite intended for designing control systems and developed by the Russian company "3V Servis" [8].

A model of (4) in the Programming Language of SimInTech is shown in figure 4.
The model was simulated using a modified Runge-Kutta-Merson algorithm with $h_{\text{min}} = 10^{-12}$, $h_{\text{max}} = 10^{-2}$, the relative error tolerance $\varepsilon_{\text{reltol}} = 10^{-4}$, absolute error tolerance $\varepsilon_{\text{abstol}} = 10^{-6}$ (figure 5).

The simulation results over the whole simulation interval are quite identical to those shown in figure 2. Although, the magnified graph shows that the bounce actually occurs below the surface. OpenModelica is an open modeling and simulation environment employing the object-oriented language Modelica [3]. A textual model of the system in Modelica is presented in figure 6.

The model was simulated using the classical 4th order Runge-Kutta method with the constant step size $h = 10^{-2}$. Figure 7 demonstrates that the first bounce happens below the surface. The simulation results obtained over the whole simulation interval are almost identical to figure 2.
Simulink is a software tool for complex system modeling and simulation developed by the American company MathWorks [9]. It has an add-on named Stateflow intended for modeling hybrid dynamical systems. The hybrid automaton constructed in Stateflow is given in figure 8.

The model was simulated using the ode45 algorithm based on the 4th order Runge-Kutta-Dormand-Prince method with $h_{\text{min}} = 10^{-12}$, $h_{\text{max}} = 10^{-2}$, $\varepsilon_{\text{reltol}} = 10^{-4}$, $\varepsilon_{\text{abstol}} = 10^{-6}$. According to figure 9, the first bounce occurs below the surface.
Table 1 contains numerical results obtained in the modeling and simulation environments, namely the differences between the exact and simulated times of the first \((t_1^* - \tilde{t}_1)\) and tenth \((t_{10}^* - \tilde{t}_{10})\) bounces and the corresponding simulated ball positions \((y_1^*\) and \(y_{10}^*)\). Since the exact ball position at the time of a bounce \(y^* = 0\), a positive error \(y - y^*\) at such a time means a bounce above the surface, whereas a negative value corresponds to a bounce below the surface violating the condition of unilateral events.

Table 1. Event detection in different simulation environments.

| Environment   | \(t_1^* - \tilde{t}_1\) | \(y_1^*\)     | \(t_{10}^* - \tilde{t}_{10}\) | \(y_{10}^*\)  |
|---------------|-----------------|----------------|-----------------|----------------|
| ISMA          | \(7 \cdot 10^{-14}\) | \(7 \cdot 10^{-13}\) | \(-2 \cdot 10^{-11}\) | \(7 \cdot 10^{-13}\) |
| SimInTech     | \(-10^{-12}\)   | \(-6 \cdot 10^{-12}\) | \(-10^{-6}\) | \(-5 \cdot 10^{-12}\) |
| OpenModelica  | \(-10^{-11}\)   | \(-10^{-10}\) | \(-10^{-10}\) | \(-10^{-10}\) |
| Simulink      | \(-10^{-14}\)   | \(-10^{-13}\) | \(-10^{-12}\) | \(-2 \cdot 10^{-13}\) |

Analyzing the simulation results, we can conclude that condition of unilateral events (3) is strictly satisfied only in ISMA because of the original step size control algorithm enabling correct detection of events in models of the aforementioned class. The other environments detect so-called accuracy critical events in this experiment, which contradicts the physical principles behind the problem. ISMA and Simulink have demonstrated the smallest event detection errors.

References

[1] Goebel R, Sanfelice R G and Teel A R 2012 Hybrid Dynamical Systems: Modeling Stability and Robustness (Princeton, USA: Princeton University Press)
[2] Shornikov Y V and Dostovalov D N 2018 Fundamentals of event-continuous system simulation theory (Novosibirsk, Russia: NSTU Publisher)
[3] Urquía Moreleda A and Martín Villalba C 2018 Modeling and simulation in Engineering using Modelica (Madrid, Spain: UNED Editorial)
[4] Shornikov Y V, Uatay B U and Novikov E A 2010 Methods for solution of stiff problems, hybrid systems and applications (Almaty, Kazakhstan: Kazakh–British Technical University)
[5] Park T and Barton P I 1996 State event location in differential-algebraic models ACM Trans. Model. Comput. Simul. 6(2) 137-65
[6] Esposito J M and Kumar M 2007 A state event detection algorithm for numerically simulating hybrid systems with model singularities ACM Trans. Model. Comput. Simul. 17(1) 1
[7] Shornikov Y V and Dostovalov D N 2017 Features of the ISMA modeling and simulation environment Proc. int. multi-conf. on engineering, comp. and informations sciences (SIBIRCON) (Novosibirsk, Russia) (Novosibirsk: IEEE) 332-7
[8] Baum F I, Kozlov O S, Parshikov I A, Petuhov V N, Timofeev K A and Shchekaturov A M 2013 SimInTech software for programming control-system devices Atomic Energy 113(6) 443-6
[9] Klee H and Allen R 2010 Simulation of dynamic systems with MATLAB and Simulink (USA: CRC Press)