SIMULATION MODELS AND ZENO PATH AVOIDANCE IN A CLASS OF PIECEWISE LINEAR BIOCHEMICAL PROCESSES

Radu Dobrescu, Virginia Ecaterina Oltean, Matei Dobrescu

Faculty of Control and Computers, Politehnica University of Bucharest, Romania ,
radud@isis.pub.ro, ecaterina_oltean@yahoo.com and mateid@isis.pub.ro

Abstract: *Hybrid automata represent a natural modelling abstraction for biochemical interactions. Within discrete states, the local dynamics is usually described by linear differential equations, and generically, the continuous variables are protein concentrations. Transitions between discrete states occur, with possible reset variables, when specific threshold values are reached, thus producing abrupt changes in the right hand side of the differential system. This paper firstly analyses a simple hybrid automaton modelling a repressor mechanism in molecular biology, which presents a hysterezis behaviour. Alternate modelling frameworks, limitations of the model and simulation challenges are discussed. Secondly, a second-order piecewise-linear model of a generic biochemical interaction process is abstracted to a Zeno hybrid automaton and a Zeno path avoidance scheme, based on fixed-step time discretization is proposed.*

Keywords: *biological systems, hybrid automata, hysterezis, piecewise linear systems, Zeno execution*.

1. INTRODUCTION AND MOTIVATION

Biological processes like protein regulatory networks present a hybrid nature, in which the continuous evolution of protein concentrations is activated or deactivated when the protein concentrations reach given threshold values.

Hybrid automata, combining event-driven with time-driven dynamics, have recently emerged as an ideal framework for modelling, analysis and simulation of biochemical interactions [6], [9]. Within locations the system dynamics is modelled by differential equations, as long as invariants, usually represented by algebraic inequality-like restrictions, are satisfied and transitions between locations occur when conditions for flow equations change are satisfied [2].

Despite recent progress, hybrid systems theory still presents open issues [4] and its application in theoretical systems biology is currently subject of intense research [1].

A description of a biochemical process in terms of state space, its dynamics (i.e. rules for ODEs and discrete transitions) and a region in its state

space corresponding to a desired property can be converted to a hybrid automaton and vice versa.

Given such a two-facet model, a *first problem* is the behaviour prediction from a biologically reasonable initial state [Piazza et al., 2005]. An adequate numerical simulation tool may be an answer, but special care must be taken when implementing the discontinuities in the right-hand side of the differential system.

A *second problem* concerns those regimes - characterized by locations in the hybrid automaton - from which a desired regime or state space region can be reached. In brief, reachability is the problem of deciding if a location of a hybrid automaton can be attained in a finite number of steps from a given initial location.

Zeno hybrid automata, executing infinite many transitions in a finite (Zeno-) time interval, may appear in biological modelling and they are *not* decidable for the reachability problem [9], [10]. Zeno hybrid automata have earlier been studied within computer science theory [2]. Although real systems are not Zeno, a hybrid model may be Zeno due to modelling abstraction. Also, simulation of Zeno behaviour may become imprecise, time consuming or it may just get stuck [11].

The attempts to solve the Zeno phenomenon and to extend system executions beyond the so-called Zeno-time are based on approximations of hybrid systems dynamics. Given a Zeno hybrid automaton, the approach called *regularization* [8] proposes the construction of a family of non-Zeno parameterised hybrid automata, which tend to the initial automaton, as the parameter tends to zero, although their executions respectively differ. *Time discretization* is a similar way to obtain such hybrid systems approximations and Mysore [9] analyses three classes of theoretical relations between the discrete time step *h* and the number of allowed discrete transitions. Implementing these relations in simulation schemes is an open issue.

Starting from a simple repressor hybrid automaton with hysterezis behaviour, the paper firstly discusses some related modelling and simulation challenges and proposes a conversion to a Simulink scheme, similar to those encountered in control engineering approaches. Secondly, the relation between the two proteins in the first model is slightly changed, driving to a second-order piecewise-linear model of a generic biochemical interaction process with Zeno behaviour. For Zeno path avoidance, the paper proposes a MATLAB simulation solution based on fixed-step time discretization, which implements the theoretical timed-jump transition relation introduced in [9].

The paper is structured as follows. An overview of hybrid automata basic definitions and related concepts is presented in Section 2. Simulation problems and alternate models of the simple repressor hybrid automaton are discussed in Section 3. Zeno path avoidance for the hybrid automaton of a generic two-state biochemical interaction process is analysed in Section 4, followed by concluding remarks.

2. HYBRID AUTOMATA - AN OVERVIEW

A hybrid automaton H is a collection $H=(Q,X,Init,\boldsymbol{f},Inv,E,G,R)$, where Q is a finite set of discrete variables, X is a finite set of continuous states, with $X\subset\mathbf{R}^n$, $Init\subseteq Q\times X$ is the set of initial states, $\boldsymbol{f}:Q\times X\to TX$ is a vector field, $Inv:Q\to 2^X$ is an invariant set for each $q_i\in Q$, $E\subset Q\times Q$ is a set of edges, $G:E\to 2^X$ is a guard of each edge and $R:E\times X\to 2^X$ is a reset map for each edge. TX denotes the tangent space of X and a state of H is $(q_j,\boldsymbol{x}_j)\in Q\times X$.

A hybrid time trajectory is a finite or infinite sequence of intervals $\tau=\{I_i\}_{i=0}^{N}$, such that: $I_i=[\tau_i,\tau_i^{'}]$ for all $i<N$, if $N<\infty$ then either $I_N=[\tau_N,\tau_N^{'}]$ or $I_N=[\tau_N,\tau_N^{'})$ and $\tau_i\le\tau_i^{'}=\tau_{i+1}$, for all *i*. Denote by $\mathcal{T}$ the set of all hybrid time-trajectories and, for $\tau\in\mathcal{T}$, $<\tau>=\{0,1,\ldots,N\}$ if $N<\infty$ and $<\tau>=\{0,1,\ldots\}$ if $N=\infty$.

An *execution* $\mathcal{X}$ of a hybrid automaton is a collection $\mathcal{X}=(\tau,q,\boldsymbol{x})$, where $\tau\in\mathcal{T}$, $q:<\tau>\to Q$, $\boldsymbol{x}=\{\boldsymbol{x}_i(\cdot):i\in<\tau>\}$, with $\boldsymbol{x}_i:I_i\to X$ satisfies the following conditions:

(i) $(q(0),\boldsymbol{x}_0(\tau_0))\in Init$;

(*ii*) $\boldsymbol{x}_i(\cdot)$ is the solution to $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(q(i), \boldsymbol{x}(t))$ over I_i, starting at $\boldsymbol{x}_i(\tau_i)$ and for all $t \in [\tau_i, \tau_i')$, $\boldsymbol{x}_i(t) \in Inv(q(i))$;
(*iii*) for all $i \in <\tau>$, $i < N$, $e = (q(i), q(i+1)) \in E$, $\boldsymbol{x}_i(\tau_i') \in G(e)$ and $\boldsymbol{x}_{i+1}(\tau_{i+1}) \in R(e_i, \boldsymbol{x}_i(\tau_i'))$.

An execution is *finite* if τ is a finite sequence ending with a closed interval, it is *infinite* if τ is either an infinite sequence or if $\sum_i(\tau_i' - \tau_i) = \infty$.

Given a hybrid automaton *H*, an *infinite execution* started in $(q_0, \boldsymbol{x}_0) \in Init$ is called *Zeno* if $\sum_{i=0}^{\infty}(\tau_i' - \tau_i)$ is bounded and $\tau_\infty = \sum_{i=0}^{\infty}(\tau_i' - \tau_i)$ is the *Zeno time*. A h*ybrid automaton* is *Zeno* if all its infinite executions started in some $(q_0, \boldsymbol{x}_0) \in Init$ are Zeno.

3. A REPRESSOR HYBRID AUTOMATON WITH HYSTERESIS BEHAVIOUR – SIMULATION MODELS

Consider the repressor hybrid automaton in Fig.1, with notations adapted from [9]. A protein (2) with concentration x_2 is produced at a rate k_{21} and consumed at a rate k_{22}. Its repressor protein (1) with concentration x_1 is always produced at a rate k_{11}. When the repressor concentration x_1 exceeds a certain threshold level θ_1^2, it begins to repress the production of protein (1). Consequently, the repressor protein is consumed, the repressor concentration x_1 eventually drops below a different threshold level $\theta_1^1 < \theta_1^2$ and the system resumes production of protein (2).

Clearly, from the threshold limit inequality and from the fact that the locations are switched depending only on the values of x_1, it results hysterezis behaviour. Main simulation challenges are hysterezis implementation and an adequate handling of discontinuities in the right hand side of differential equations.

The simple repressor hybrid automaton can be viewed as a two-tank accumulation process in control engineering (Fig.2), with a corresponding state space representation and Simulink scheme in Fig.3a.

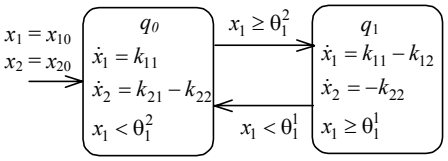


Fig. 1. Hybrid automaton of a repressor mechanism: x_1 repressor and x_2 repressed protein concentrations.

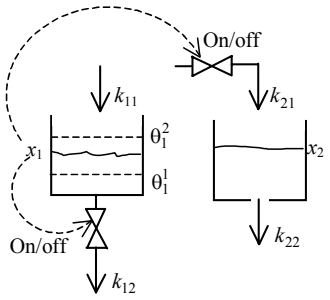
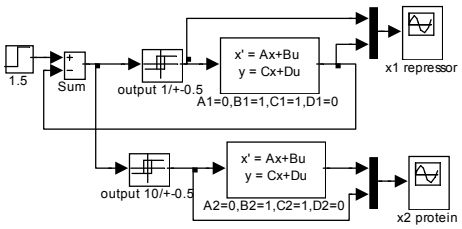
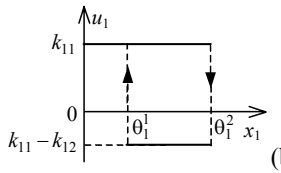


Fig. 2. The repressor hybrid automaton in Fig.1 viewed as an accumulation process in control engineering.



(a)



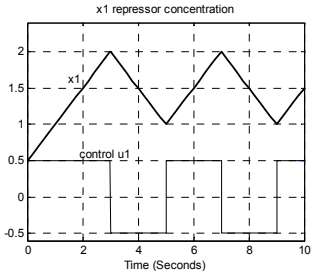
(b)

Fig. 3. Simulink model of the hybrid automaton in Fig.1 (a) and the control law represented as a relay with hysteresis (b).

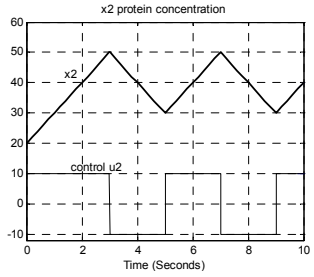
There is no feedback from the controlled variable x_2 and the state equations of the hybrid system can be written as

$$\dot{x}_1 = u_1(x_1, \theta_1^1, \theta_1^2), \ \dot{x}_2 = u_2(x_1, \theta_1^1, \theta_1^2), \tag{1}$$

where $0 < \theta_1^1 < \theta_1^2$. The diagram of the switching control law u_1 in Fig.3b is implemented using the classic relay bloc in Simulink and simulation results are shown in Fig.4.



(a)



(b)

Fig. 4. Simulated evolutions of repressor (a) and protein (b) concentrations, for : $\theta_1^1 = 1$, $\theta_1^2 = 2$, $k_{11} = 0.5$, $k_{12} = 1$, $k_{21} = 20$, $k_{22} = 10$, with $k_{22} = |k_{21} - k_{22}|$.

Note that the system is not asymptotically stable and the lack of feedback from x_2 acts like model incompleteness. A modification of the production rate of protein (2), such that $k_{22} \neq |k_{21} - k_{22}|$, driving to an unsymmetrical relay for u_2, generates unbounded evolution of protein concentration (Fig.5).

Remark 1. Using the notations in the *piecewise-linear model of genetic regulatory networks dynamics* in [5], firstly introduced by [7], equations (1) can be rewritten as

$$\dot{x}_1 = k_{11} - k_{12} s^+(x_1, \theta_1^1), \dot{x}_2 = u_2(x_1, \theta_1^1, \theta_1^2), \quad (2)$$

where $k_{ij} > 0$, $i, j = 1:2$, $0 < \theta_1^1 < \theta_1^2$ and $s^{\pm}(x_1, \theta_1^j)$ are step functions, with $s^+(x_1, \theta_1^j) = 0$, for $x_1 < \theta_1^j$, $s^+(x_1, \theta_1^j) = 1$ for $x_1 > \theta_1^j$ and $s^-(x_1, \theta_1^j) = 1 - s^+(x_1, \theta_1^j)$, $j = 1:2$. In the original piecewise-linear model of genetic regulatory networks dynamics in [5], there are *exponential protein degradations* rates in the right hand side of the ODEs, $-\gamma_i x_i$ with $\gamma_i > 0$, so the process is *asymptotically stable*. In (1), the exponential protein degradations are neglected and thus the system is a pure integrator (Fig.5).

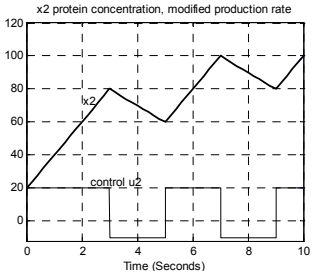


Fig. 5. Unrealistic evolution of protein concentration with same repressor evolution and parameters as in Fig.4, except $k_{21} = 30$, thus $k_{22} \neq |k_{21} - k_{22}|$.

The decays arise from the sign of the difference between the coefficients k_{i2} and k_{i2}, $i = 1:2$. ◊

4. TIME DISCRETIZATION FOR A ZENO HYBRID MODEL OF A GENERIC BIO-CHEMICAL INTERACTION PROCESS

Consider the biochemical interaction process modelled by the hybrid automaton in Fig.6, obtained by modifying the model in Fig.1: the variables x_i, $i = 1:2$, are protein concentrations and this time each one is a repressor for the other. This toy-model generalizes the Zeno hybrid automaton proposed in [4], with following numerical values: $x_1(0) = 0$, $x_2(0) = 4$, $k_{11} = k_{22} = 1$, $k_{21} = k_{12} = 2$, $\theta_2^1 = \theta_1^1 = 0$. The system switches from one location to another, when one of the two concentrations falls to the lowest level zero.

4.1 Behaviour analysis of the Zeno hybrid automaton.

At $t_0 = 0$, the hybrid automaton starts from location q_0 with the initial continuous state and it evolves generating the state trajectories $x_1(t) = t$, $x_2(t) = 4 - 2t$, until, at $t_1 = 4/2 = 2\,\text{sec}$, the enabling condition $x_2(t_1) = 0$ for transition $q_0 \to q_1$ is satisfied and the system instantly switches to q_1. The time spent in q_0 is $\Delta_0 = 2\,\text{sec}$.

Similarly, from t_1 the system evolves in the state q_1, with the continuous behaviour $x_1(t) = 2 - 2t$, $x_2(t) = t$, until, at $t_2 = 2/2 = 1\,\text{sec}$, the enabling condition

$x_1(t_2)=0$ for transition $q_1 \to q_0$ is satisfied, and the system instantly switches to q_0. The time spent in q_1 is $\Delta_1 = 1\,\text{sec}$. The process continues and the time elapsed after the first k switching is given by

$$T_k = \sum_{i=0}^{k} \Delta_i = 2\left(1 + 2^{-1} + \cdots + 2^{-k}\right) = 4\left(1 - 2^{-k}\right). \quad (3)$$

Zeno behaviour results from the fact that time does not "progress" with k, and the Zeno-time is

$$\tau_\infty = \lim_{k \to \infty} T_k = 4\,\text{sec}. \quad (4)$$

4.2 Time discretization as hybrid systems approximation technique - background

Time discretization is an approach for hybrid systems approximation, in view of model checking. Consider a hybrid automaton H with continuous dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(q_i, \boldsymbol{x})$, $\boldsymbol{x} \in \mathbf{R}^n$, $q_i \in Q$ and a time step $h > 0$.

With the notations in [9], H can execute:

- a *discrete transition* $(q_0, x_0) \xrightarrow[D]{0} (q', x')$, in $h = 0$ time (i.e. instantly), denoted shortly $q_0 \xrightarrow[D]{0} q'$ or
- a *continuous transition* $(q_0, \boldsymbol{x}_0) \xrightarrow[C]{h} (q'(=q_0), \boldsymbol{x}')$, with $\boldsymbol{x}' = \boldsymbol{x}_0 + \int_0^h \boldsymbol{f}(q_0, \boldsymbol{x}(t))dt$.

Numerical simulation techniques based on *variable time-step discretization* implement discrete-time transition relations allowing the system to take steps less than h:

$$q_0 \xrightarrow{h} q' = \{\exists q'', h_1 : 0 < h_1 < h \wedge q'' \xrightarrow[C]{h_1} q' \wedge q'' \xrightarrow[D]{0} q''\} \vee \{q \xrightarrow[C]{h} q'\}. \quad (5)$$

Thus, the system is allowed (a) to evolve continuously for the full-time h in the current state *or* (b) to evolve to an intermediate point $\le h$ and then take a transition. In this case, the number of iterations does *not* correlate with the elapsed simulation time: some of the iterations correspond to discrete transitions, while some continuous steps are of length less than h. Also, *Zeno-path* are *not excluded* [9].

Since Zeno-behaviour arises from zero time switching, a natural refinement of the transition relation (5) is, according to Mysore [9], imposing a *minimum* on the time spent in each step. In this regard, *the timed-jump model* allows only one state transition at the beginning of the time-step:

$$q_0 \xrightarrow{h} q' = \{\exists q'' : q \xrightarrow[D]{0} q'' \wedge q'' \xrightarrow[C]{h} q'\} \vee \{q \xrightarrow[C]{h} q'\}. \quad (6)$$

There are several limitations of this approach. Firstly, the *equality* transition conditions may never be satisfied, as they are checked at the start/end of the time-step h [9], so they have to be replaced with *inequalities*, which capture the same dynamics. Secondly, the system is not allowed to take multiple jumps at a step h, so h has to be "small" enough, but without increasing too much the system complexity.

4.3 MATLAB implementation of the timed-jump model: fixed-step time discretization and Zeno-path avoidance

In order to avoid effects of Zeno behaviour in the vicinity of Zeno time, a *discrete time approach* is proposed, based on the *linearity of the local continuous dynamics* in the hybrid automaton in Fig.6. Recall that given a system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \quad (7)$$

and the sampling step $h > 0$, the corresponding discrete-time linear system is

$$\boldsymbol{x}_d(k+1) = \boldsymbol{A}_d \boldsymbol{x}_d(k) + \boldsymbol{B}_d \boldsymbol{u}_d(k), \quad (8)$$

with k the integer time variable (iteration), $\boldsymbol{x}_d(k) = \boldsymbol{x}(kh)$, $\boldsymbol{u}_d(k) = \boldsymbol{u}(kh)$ and

$$\boldsymbol{A}_d = \exp(h\boldsymbol{A}), \ \boldsymbol{B}_d = \int_0^h \exp(\theta \boldsymbol{A}) d\theta \cdot \boldsymbol{B}. \quad (9)$$

The two continuous integrators, in variables x_1 and x_2, are discretized (using the routine `c2d`, for example) and the dynamical equations are:

$$x_{1d}(+1) = x_{1d} + h \cdot u_1, x_{2d}(+1) = x_{2d} + h \cdot u_2. \quad (10)$$

Then the transition condition tested at step k is:

if $x_{1d}(+1) \le 0$ *then* $u_1 = 1$, $u_2 = -2$
else if $x_{2d}(+1) \le 0$ *then* $u_1 = -2$, $u_2 = 1$
end

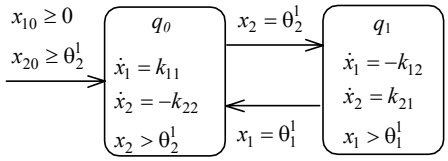


Fig. 6. A Zeno hybrid automaton interpreted as a biochemical interaction process: x_1 and x_2 are protein concentrations, each one repressing the other.

Remark 2. The initial equality transition conditions are replaced by inequalities as follows. In the hybrid automaton in Fig.6, $Inv(q_0) = \{x_2 : x_2 > 0\}$ and

$$x_2 = \theta_2^1 (= 0) \Rightarrow q_0 \overset{0}{\underset{D}{\rightarrow}} q_1,$$

while in the proposed discrete-time model the transition condition is:

$$x_{2d}(+1) \leq \theta_2^1 (= 0) \Rightarrow q_0 \overset{0}{\underset{D}{\rightarrow}} q_1.$$

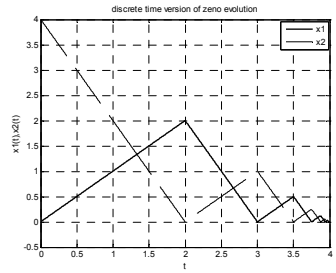
A similar condition is introduced for testing if the transition $q_1 \rightarrow q_0$ occurs. At each step, the new state values are tested. ◊

Simulation results, for $h = 10^{-4}$ and simulation time equal to Zeno-time, are depicted in Fig.7a. Simulation of the discrete time version *over* Zeno time does *not* get stuck, and a continuation of evolution is obtained (Fig.7b). However, after Zeno-time, the values of the simulated discrete variables loose their physical significance. As already emphasized, another problem with this approach is the (rather empirical) choice of sampling step *h*. The plot in Fig.8 compared to the one in Fig.7a illustrates this.

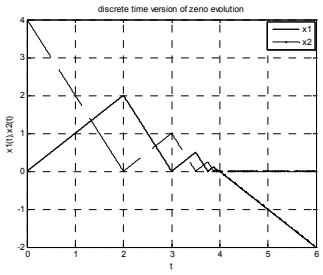
5. CONCLUDING REMARKS

Hybrid automata theory has recently emerged as an important framework for modelling and prediction in systems biology, such that crucial problems like reachability analysis or model checking may get a systematic answer.

However, hybrid systems simulation is difficult because it must properly deal with switching among a collection of continuous evolution laws, with eventual resetting maps over initial conditions of differential equations and with testing transition conditions, such as "zero-crossing", among others.



(a)



(b)

Fig. 7. Evolution of the discrete-time system approximating the Zeno automaton in Fig.6, starting in $(0,4)$, with $h = 0.0001$ and simulation time $t_{sim} = \tau_\infty$ (a) and $t_{sim} = 1.5\tau_\infty$ (b).

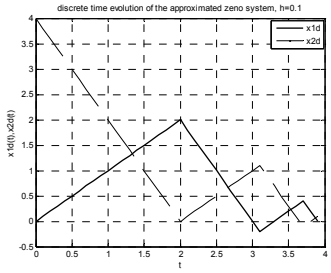


Fig. 8. Evolution of the discrete-time system approximating the Zeno hybrid automaton in Fig.6, with $h = 0.1$ and simulation time $t_{sim} = T$.

The toy example repressor hybrid automaton with hysterezis discussed in Section 3 cannot be simulated using classic variable-step integration methods without special care. If the models are simple enough – as it was the case - they can be converted into Simulink schemes, which successfully capture also special discontinuities, like relays with hystersis. However, if the model complexity increases or if the system parameters are imprecisely known, this is no more a solution and alternate qualitative simulation approaches have to be considered [5].

Zeno behaviour, characterized by infinitely many switching executed in a finite time interval, may arise from sometimes-inevitable model simplifications. The direct implementation of Zeno models in simulation

programs drives to improper simulated evolution: the simulation either gets stuck, or it generates incorrect results. The important reachabilty problem, which requires behaviour prediction – is *not decidable* for Zeno systems. A solution is to *approximate* the Zeno system in a convenient way, by a model that is easier to be checked.

Time discretization is, in this respect, an adequate tool for Zeno path avoidance, and several theoretical discretization schemes are proposed in the literature [9]. The implementation approach proposed in Section 4 for approximating a Zeno toy model of a biochemical interaction process is based on converting the continuous linear state equations of the hybrid model to a *fixed-step discrete time model*, using the MATLAB routine `c2d` with *zero-order hold on the inputs*. The switching conditions are tested at each discrete time step and simulation does *not* get stuck when the computed Zeno time is reached. For same initial conditions, the simulated trajectories do *not* coincide with the trajectories of the Zeno hybrid automaton, but if the simulation step is "small enough", the trajectories of the discrete time model reflect adequately the behaviour of the original Zeno hybrid system. The equality-like transition conditions in the original Zeno system are converted, in the program, into inequalities conditions, which capture the original dynamics.

Three basic problems have to be solved when implementing a time discretization scheme: the adequate choice (*i*) of the discretization method, (*ii*) of the discretization step and (*iii*) the proper modification of transition conditions. Also, a prior model analysis and evaluation of Zeno time is usually necessary, for the choice of the simulation time. Adequate interpretation of results is mandatory if simulation time exceeds Zeno time.

Implementation of various time discretization schemes for Zeno path avoidance and integrating them in simulation tools is subject of future research.

REFERENCES

[1] Aderem, A. Systems biology. Its practice and challenges. *Cell*, 121(4), p.511-513, 2005.

[2] Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P-H, Nicollin, X., Olivero, A., Sifakis, J., Yovine, S. The algorithmic analysis of hybrid systems. *Theoretical Comp. Science*, Elsevier, p.138, 3-34, 1995.

[3] Alur, R., Dill, L.D. *Automata for modeling real time systems*. In: *ICALP'90*, LNCS **443**, p.322-335, Springer Verlag, 1990.

[4] Branicky, M.S. *Introduction to Hybrid Systems*. In: Hristu-Varsakelis D., Levine W.S. (Eds.), *Handbook of Networked and Embedded Control Systems*, pp.91-116, Springer Verlag, Birkhäuser, 2005.

[5] de Jong H., Geiselmann, J., Hernandez, C. and Page, M. (2003). Generic Network Analyser: quantitative simulation of genetic regulatory networks. *Bioinformatics*, Oxford University Press, **19**, *3*, p.336-344.

[6] Ghosh, R. and Tomlin, Claire. Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modeling: Delta-notch protein signalling. *IEE Trans. on Systems Biology*, 1(1), p.170-183, 2004.

[7] Glass, L., Kauffman, S. The logical analysis of continuous non-linear biochemical control networks. *J. Theor. Biol.*, **39**, p.103-129, 1973.

[8] Johansson, K. H., Egerstedt, M., Lygeros, J. and Sastry, S. On the regularization of Zeno hybrid automata, *Systems and Control Letters*, **38**, p.141-150, 1999.

[9] Mysore, V.P. *Algorithmic Algebraic Model Checking: Hybrid Automata and Systems Biology*. PhD dissertation, Dept. of Computer Sciences, New York University, 2006.

[10] Piazza, C., Antoniotti, M., Mysore, V., Policriti, A., Winkler, F. and Mishra, B. Algorithmic Algebraic Model Checking I: Challenges from Systems Biology. In: *Proc. of the 17th Int. Conf. on Computer Aided Verification* (The University of Edinburgh, Scotland, UK, July 6-10 2005), CAV 2005, p.5-19, 2005.

[11] Zhang, J., Johansson, K. H., Lygeros J. and Sastry S. Zeno Hybrid Systems, *International Journal of Robust and Nonlinear Control*, **11,** *5*, p.435-451, 2001.