Analysis of Timed Properties Using the Jump-Diffusion Approximation

Paolo Ballarini[1], Marco Beccuti[2(Γ)], Enrico Bibbona[3],
Andras Horvath[2(Γ)], Roberta Sirovich[4], and Jeremy Sproston[2]

[1] Laboratoire MICS, CentraleSupèlec, Université Paris Saclay, France
[2] Dipartimento di Informatica, Università di Torino, Italy
[3] Dipartimento di Scienze Matematiche "G. L. Lagrange", Politecnico di Torino, Italy
[4] Dipartimento di Matematica, Università di Torino, Italy
{beccuti,horvath}@di.unito.it

Abstract. Density dependent Markov chains (DDMCs) describe the interaction of groups of identical objects. In case of large numbers of objects a DDMC can be approximated efficiently by means of either a set of ordinary differential equations (ODEs) or by a set of stochastic differential equations (SDEs). While with the ODE approximation the chain stochasticity is not maintained, the SDE approximation, also known as the diffusion approximation, can capture specific stochastic phenomena (e.g., bi-modality) and has also better convergence characteristics. In this paper we introduce a method for assessing temporal properties, specified in terms of a timed automaton, of a DDMC through a jump diffusion approximation. The added value is in terms of runtime: the costly simulation of a very large DDMC model can be replaced through much faster simulation of the corresponding jump diffusion model. We show the efficacy of the framework through the analysis of a biological oscillator.

1 Introduction

Context. Advances in modelling lead to increasingly complex models of concurrent systems whose analysis, consequently, has become a critical issue. In particular the analysis of quantitative aspects of these systems by means of stochastic models (e.g., Markov chains) may be impaired by the combinatorial explosion of their state space. To cope with this problem several approaches have been proposed in the literature including, e.g., decomposition and aggregation, bounding techniques, compact representations of the state space. However, when the model accounts for large groups of individuals (e.g., Internet users, molecule populations) these techniques may turn out to be insufficient, meaning that *discrete event simulation* (DES) is the most practical option for analysing the system's performance. Indeed, DES based approaches do not require the explicit

storage of the state space, but instead exploit a set of sample executions (i.e., traces) in order to devise arbitrarily accurate (statistical) *estimates* of relevant indicators of a model's behaviour.

Fluid approximations. In the case of large interacting populations, an alternative to simulation is to use a *deterministic approximation* in which the behavior is represented by a set of ordinary differential equations (ODE) [17]. However, this approach is not suitable for the study of models where stochasticity (bimodality, high variance) plays an important role even for large population counts. To analyze a model's stochastic nature, in [18] a *diffusion approximation* was proposed, based on a set of stochastic differential equations (SDE), that can be applied up to the first visit of the boundary of the state space. Both the deterministic and the diffusion approximation are such that every state variable is approximated by a continuous variable, i.e., the state variables are made "fluid". Since in real systems the boundaries of the state space often can be visited many times, in [7] we proposed an extension, namely, a *jump diffusion approximation*, to properly approximate the original model at the boundaries as well. A further extension was made in [3] that uses partial fluidification of the state space, which results in a *switching jump diffusion approximation*, allowing us to mimic better the original process in the case of low population counts.

Contribution. Starting from [7], in this paper, we propose a new statistical model checking method based on jump diffusion approximation. This method takes as input a DDMC and, following [9, 16, 10, 11], a formal description of a property, in this paper described as a (deterministic) timed automaton [1]. The jump diffusion approximation of the DDMC is used to generate trajectories of the system and the deterministic timed automaton is used to accept or reject each trace. Based on the proportion of the accepted traces confidence intervals are derived for the probability that the system exhibits the property in question.

Applicability. The applied theoretical framework requires a sequence of DDMCs indexed by a parameter N [17]. The sequence is such that the state space, the transition intensities and also the vector describing the initial state increase as N increases. Four possibly overlapping ranges of values can be identified for N. N can be so small that the corresponding DDMC can be analyzed by analytical approaches. As N grows, analytical analysis becomes unfeasible, but the DDMC can still be evaluated efficiently by simulation. By further increasing N, even simulation of the DDMC becomes impractical but the model can still exhibit important stochastic behavior. This is the range in which the approach we propose is convenient to use: in this range the diffusion approximation provides results with reasonable precision in much shorter time than dealing with the original DDMC. For even larger values of N, the stochastic behavior disappears and the model can be analyzed with a deterministic approximation.

Organization. The paper is organized as follows. Section 2 introduces DDMCs and Section 3 discusses their approximations. In Section 4 we provide the definition of the applied timed automata. We discuss the issues related to assessing properties through a diffusion approximation in Section 5. In Section 6 a case study is presented. Conclusions are drawn in Section 7.

2 Nearly density dependent Markov chains

Continuous time Markov chains (CTMC) are often used to describe the interaction of groups of identical objects. Informally, such CTMCs are called *density dependent* if the intensities of the interactions can be expressed as a function of the *density of the objects present in the area* (or volume) described by the model (as opposed to being expressed as a function of the number of objects itself).

Definition 1. *Consider a sequence of CTMCs, denoted by $X^{[N]}(t)$, indexed by $N \in \mathbb{N} \setminus \{0\}$ and with state space $\mathcal{S}^{[N]} \subseteq \mathbb{Z}^k$ (i.e., every state is identified by a vector of k integers), that describe the interaction of k groups of identical objects. The sequence $X^{[N]}(t)$ is called* density dependent *if the associated transition intensities, given any two states $r \in \mathcal{S}^{[N]}$ and $r + m \in \mathcal{S}^{[N]}$ that are connected by a transition, can be written in the form*

$$q_{r,r+m}^{[N]} = Nf\left(\frac{r}{N}, m\right) \tag{1}$$

where $f : \mathbb{R}^k \times \mathbb{Z}^k \rightarrow \mathbb{R}_{\geq 0}$ is a bivariate function whose first argument is a vector that provides the density for each group of objects in state r and its second argument is the change in the state due to the transition from state r to state $r + m$ ($\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers).

The indexing parameter N can represent the size of the considered area or volume, or the total number of objects in the model (in this case the vector r/N is a vector of proportions). Note that in Definition 1 a single function, namely f, provides the intensity of every transition of every CTMC of the sequence of CTMCs. This implies that in every CTMC the transitions have the same effect on the state.

The above definition can be relaxed by substituting (1) with

$$q_{r,r+m}^{[N]} = Nf\left(\frac{r}{N}, m\right) + Ng\left(\frac{r}{N}, m, N\right) \tag{2}$$

where $g : \mathbb{R}^k \times \mathbb{Z}^k \times \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is a trivariate function and $g(r/N, m, N) \in O\left(1/N\right)$. Sequences of CTMCs in which the transition intensities are in the form given in (2) are referred to as *nearly density dependent*. The rationale behind the definition is the following. As N grows, thanks to $g(r/N, m, N) \in O\left(1/N\right)$, the term $Ng(r/N, m, N)$, which is *not density dependent*, remains in the order of a constant. The other term instead grows proportionally to N. Accordingly, as N grows the density dependent nature of the process prevails. Indeed, *density dependent* and *nearly density dependent* processes can be studied with the same approximations.

As for notation, the set of possible changes in the state due to a transition will be denoted by C. Formally, a vector $m \in \mathbb{Z}^k$ is in C if and only if there exist two states $r \in \mathcal{S}^{[N]}$ and $r + m \in \mathcal{S}^{[N]}$ such that there is a transition from r to $r + m$. Note that, like the function f, also the set C is shared by every member of a given sequence of DDMC.

Example 1. As an example we consider a simple epidemic model in which two groups are involved, namely, susceptible and infected individuals. Accordingly, each state is described by a pair (i, j) providing the number of susceptible and infected people, respectively. We assume that the modelled individuals are uniformly distributed over an area split into N equally sized cells and that three kinds of events are possible. The number of susceptible individuals grows with an intensity proportional to the number of cells: $q_{(i,j),(i+1,j)}^{[N]} = N\lambda_1$. Due to the contact of two infected and one susceptible person in one of the cells, one susceptible individual becomes infected; this happens with intensity $q_{(i,j),(i-1,j+1)}^{[N]} = \frac{ij(j-1)}{2} \frac{1}{N^3} N\lambda_2$, where the first term is the number of ways the three individuals can be selected, the second term is the probability that the three selected individuals are together in a given cell, and the multiplication by N is due to the fact that the contact can occur in any cell. Infected individuals can become immune independently of each other and independently of the number of cells; the associated intensity is $q_{(i,j),(i,j-1)}^{[N]} = j\lambda_3$. The intensity of the first type of event is independent of the actual state and proportional to N and thus it is a special form of (1). The intensity of the other two kinds of events can be rewritten as

$$q_{(i,j),(i-1,j+1)}^{[N]} = N \left(\frac{\lambda_2}{2} \frac{i}{N} \left(\frac{j}{N} \right)^2 \right) - N \left(\frac{1}{N} \frac{\lambda_2}{2} \frac{i}{N} \frac{j}{N} \right), \quad q_{(i,j),(i,j-1)}^{[N]} = N\lambda_3 \frac{j}{N}$$

where the first intensity is nearly density dependent while the second is density dependent. The set of possible state changes is $C = \{(1,0), (-1,1), (0,-1)\}$.

3 Approximations of nearly density dependent CTMCs

All approximations we describe in the following use a process with a continuous state space and thus are considered "fluid" approximations. In order to proceed we need to introduce the sequence of normalized CTMCs given by $Z^{[N]}(t) = X^{[N]}(t)/N$, called also the density process. The reason to use $Z^{[N]}$ instead of the original process is that normalization brings all CTMCs of a given density dependent sequence to the same scale, making them comparable.

The first approximation we consider uses a set of ODEs in which there is one equation per group. Accordingly, the original stochastic behavior is approximated by a deterministic process. The set of ODEs used in the approximation is provided by the following result of Kurtz [17]. Given a nearly density dependent sequence of CTMCs $X^{[N]}(t)$ with initial state that tends to z_0 as N tends to infinity, i.e., $\lim_{N \to \infty} Z^{[N]}(0) = \lim_{N \to \infty} X^{[N]}(0)/N = z_0$, if the function $\sum_{l \in C} lf(y, l)$ satisfies some relatively mild conditions, then the density process $Z^{[N]}(t)$ converges to a deterministic function $z(t)$. The function $z(t)$ is the solution of the following set of ODEs

$$dz(t) = \sum_{l \in C} lf(z(t), l) \, dt, \quad z(0) = z_0. \tag{3}$$

We note that (3) is equivalent to the more familiar form $\frac{dz(t)}{dt} = \sum_{l \in C} l f(z(t), l)$; however we prefer the form in (3) because it has more in common with the other approximations that we introduce later.

The approximation given by $z(t)$ has the following property:

$$\lim_{N \to \infty} \mathbb{P} \left\{ \sup_{t \leq T} \left| Z^{[N]}(t) - z(t) \right| > \delta \right\} = 0, \tag{4}$$

for every $\delta > 0$ and where T is the upper limit of the considered finite time horizon. Moreover, it was shown in [17] that the difference between the deterministic approximation and the original stochastic behavior is characterized by

$$\sup_{t \leq T} \left| Z^{[N]}(t) - z(t) \right| = O\left(1/\sqrt{N} \right) \tag{5}$$

The practical meaning of (5) is that the error of the deterministic approximation decreases as $1/\sqrt{N}$.

Another approximation of a density dependent sequence $X^{[N]}$, which is based on stochastic differential equations and thus it preserves the stochastic nature of the original process, was proposed in [18, 19]. This approximation, denoted by $Y^{[N]}(t)$, is obtained by the following set of SDEs:

$$dY^{[N]}(t) = \sum_{l \in C} l f\left(Y^{[N]}(t), l \right) dt + \sum_{l \in C} \frac{l}{\sqrt{N}} \sqrt{f\left(Y^{[N]}(t), l \right)} \, dW_l(t) \tag{6}$$

where the $W_l(t)$ with $l \in C$ are independent standard one-dimensional Brownian motions. The approximation holds up to the first time $Y^{[N]}(t)$ reaches a boundary of the state space. In (6) the first term is the same used by the deterministic approximation in (3), while the second term is a noise that mimics the stochasticity of the original CTMCs.

For what concerns the relation of the diffusion approximation and the original density process, in [18] it has been proven that, for any finite N, we have

$$\sup_{t \leq T} \left| Z^{[N]}(t) - Y^{[N]}(t) \right| = O\left(\log N/N \right) \tag{7}$$

In practice, one uses $N \cdot z(t)$ or $N \cdot Y^{[N]}(t)$ to approximate the original CTMC $X^{[N]}(t)$. The difference between $N \cdot z(t)$ and $X^{[N]}(t)$ according to (5) is in the order of $N(1/\sqrt{N}) = \sqrt{N}$. Between $N \cdot Y^{[N]}(t)$ and $X^{[N]}(t)$ according to (7) it is instead $N(\log N/N) = \log N$ which is much lower than $\sqrt{N}$.

A limitation of the previous approach based on SDEs is that it can be applied only to models where the probability of reaching a boundary of the area of the process is negligible. In order to overcome this limitation, in [7] we introduced a jump diffusion process in which the jumps are used to capture the behavior of the process at the boundaries. We provide here a brief description of the jump diffusion process, denoted by $J^{[N]}(t)$; for a detailed treatment, see [7, 3].

The main idea is to split the transitions of the model into two sets depending on the current state. In particular, we denote by $C^\circ(y)$ the set of transitions

that change one or more components of the state which are at the boundary in state y. The jump diffusion process is defined then by

$$dJ^{[N]}(t) = \sum_{l \in C - C^\circ(J^{[N]}(t))} l f\left(J^{[N]}(t), l\right) dt + \qquad (8)$$

$$\sum_{l \in C - C^\circ(J^{[N]}(t))} \frac{l}{\sqrt{N}} \sqrt{f\left(J^{[N]}(t), l\right)} dW_l(t) + \sum_{l \in C^\circ(J^{[N]}(t))} \frac{l}{N} dM_l^{[N]}(t)$$

where the first two terms are analogous to those in (6) but are restricted to those transitions that change components away from the boundaries. If none of the components are at the boundary of the state space then $J^{[N]}(t)$ behaves exactly as $Y^{[N]}(t)$. The term $M_l^{[N]}(t)$ corresponds to Poisson counting processes that gives rise to jumps that mimic the behavior of the original CTMC at the boundaries. In other words, when the process reaches a boundary then discrete jumps regulated by a Poisson process make it jump back eventually to the inner part of the state space. The intensity associated with $dM_l^{[N]}(t)$ is $\mu_l(t) = N f\left(J^{[N]}(t), j\right)$, i.e., it is taken directly from the original CTMC (note that $J^{[N]}(t)$ provides directly a vector of densities as required by f). Then $dM_l^{[N]}(t)$ is multiplied by l/N because that is the effect of the transition in the normalized state space.

Recent studies [8] have shown that the jump diffusion approximation has similar characteristics to those of the pure diffusion approximation and, in particular, that the approximation it introduces is as good as that of the "pure" diffusion process, that is:

$$\sup_{t \leq T} \left| Z^{[N]}(t) - J^{[N]}(t) \right| = O\left(\log N / N\right) \qquad (9)$$

Numerical evaluation of the goodness of the jump diffusion approximation has been illustrated instead in [7, 3].

4   Timed Automata

In this section, we introduce a timed automata-based formalism for the specification of timed properties of CTMCs. As is standard when using timed automata for the specification of properties of stochastic systems (e.g., [13, 12, 20, 10, 11]), we use *deterministic* timed automata (DTA): that is, each input sequence of the timed automaton (which in our context is a trajectory, i.e., a function from time to the state space of the CTMC, representing a particular behavior of the CTMC or of its diffusion approximation) corresponds to a single run of the timed automaton. In order to provide a uniform framework for timed properties interpreted on CTMC and on jump diffusion approximations, our DTA are labeled with constraints both on clocks and on variables characterizing the state space, but are not labeled with actions corresponding to individual CTMC transitions (which have no meaning in the jump-diffusion diffusion approximation setting).

Edges of our variant of DTA are *urgent*: they are taken as soon as they are enabled. Urgency of edges allows for a natural interpretation of our DTA not only on behaviors of CTMC, but also on trajectories of their diffusion approximations.

We denote by $\mathcal{S} \subseteq \mathbb{Z}^k$ the state space and by $\mathcal{V} = \{\vartheta_1, ..., \vartheta_k\}$ a set of k variables, where we interpret ϑ_i as a variable corresponding to the i-th element of the vector representing a state. Let $\mathcal{C}$ be a finite set of variables called *clocks*.

Definition 2. *A* constraint *is defined by the following grammar:*

$$\Phi ::= \varphi \leq \varphi \mid \mathsf{c} \leq \lambda \mid \mathsf{c} \geq \lambda \mid \Phi \wedge \Phi \,,$$
$$\varphi ::= \varphi + \varphi \mid \varphi - \varphi \mid \varphi * \varphi \mid \varphi / \varphi \mid \vartheta_i \mid \lambda \,,$$

where $\mathsf{c} \in \mathcal{C}$ *is a clock,* $\vartheta_i \in \mathcal{V}$ *and* $\lambda \in \mathbb{Q}$ *is a rational constant. A* guard constraint *is a constraint* Φ *such that, for each* $a \in \mathcal{V} \cup \mathcal{C}$, *there is at most one subformula of* Φ *featuring* a. *An* invariant constraint *is a guard constraint* Φ *in which there is no subformula of the form* $\mathsf{c} \geq \lambda$. *We write* $\mathrm{Guards}(\mathcal{V}, \mathcal{C})$ *and* $\mathrm{Invariants}(\mathcal{V}, \mathcal{C})$ *to denote the set of guard constraints and invariant constraints, respectively, over* $\mathcal{V}$ *and* $\mathcal{C}$.

Examples of invariant constraints include $\vartheta_1 \leq 10 \wedge \vartheta_2 \geq \vartheta_3$ and $\vartheta_1 \geq 3 \wedge \mathsf{c}_1 \leq 15$, whereas $\mathsf{c}_1 \geq 3 \wedge \mathsf{c}_2 \leq 10 \wedge \vartheta_1 \geq 3$ is an example of a guard constraint that is not an invariant constraint (due to the conjunct $\mathsf{c}_1 \geq 3$).

A function $v : \mathcal{C} \to \mathbb{R}_{\geq 0}$ is referred to as a *clock valuation*, and the set of all clock valuations is denoted by $\mathrm{Val}(\mathcal{C})$. For any $v \in \mathrm{Val}(\mathcal{C})$, $\gamma \in \mathbb{R}_{\geq 0}$ and $\mathsf{C} \subseteq \mathcal{C}$, we use $v + \gamma$ to denote the clock valuation that increments all clock values in v by γ (that is, $(v + \gamma)(\mathsf{c}) = v(\mathsf{c}) + \gamma$ for all $\mathsf{c} \in \mathcal{C}$), and $v[\mathsf{C}{:=}0]$ to denote the clock valuation in which clocks in C are reset to 0 (that is, $v[\mathsf{C}{:=}0](\mathsf{c}) = 0$ for $\mathsf{c} \in \mathsf{C}$, and $v[\mathsf{C}{:=}0](\mathsf{c}) = v(\mathsf{c})$ for $\mathsf{c} \in \mathcal{C} \setminus \mathsf{C}$). The clock valuation that assigns 0 to all clocks in $\mathcal{C}$ is denoted by $\mathbf{0}$. Let Φ be a constraint, let $y \in \mathcal{S}$ be a state and let $v \in \mathrm{Val}(\mathcal{C})$ be a clock valuation. Then we write $(y, v) \models \Phi$ if and only if substituting ϑ_i by y_i (where y_i is the i-th element of the vector y) and c by $v(\mathsf{c})$ in Φ results in Φ resolving to true. For example, for y such that $y_1 = 4$ and v such that $v(\mathsf{c}_1) = 12.1$, we write $(y, v) \models \vartheta_1 \geq 3 \wedge \mathsf{c}_1 \leq 15$.

Definition 3. *A* timed automaton *is a tuple* $(\mathcal{L}, \ell_{\mathrm{init}}, \mathcal{F}, \mathcal{C}, \mathrm{Inv}, \mathcal{E})$ *comprising: (1) a finite set* $\mathcal{L}$ *of* locations, *with an* initial location $\ell_{\mathrm{init}} \in \mathcal{L}$ *and a set* $\mathcal{F} \subseteq \mathcal{L}$ *of* final locations; *(2) a finite set* $\mathcal{C}$ *of* clocks; *(3) an* invariant condition $\mathrm{Inv} : \mathcal{L} \to \mathrm{Invariants}(\mathcal{V}, \mathcal{C})$; *(4) a set* $\mathcal{E} \subseteq \mathcal{L} \times \mathrm{Guards}(\mathcal{V}, \mathcal{C}) \times 2^{\mathcal{C}} \times \mathcal{L}$ *of* edges, *where each edge* $(\ell, \Phi, \mathsf{C}, \ell') \in \mathcal{E}$ *comprises a source location* ℓ, *an enabling condition* Φ, *a set* C *of clocks to be reset to 0, and a target location* ℓ'. *A timed automaton is* deterministic *if, for any location* $\ell \in \mathcal{L}$ *and for any pair* $(\ell, \Phi_1, \mathsf{C}_1, \ell_1), (\ell, \Phi_2, \mathsf{C}_2, \ell_2) \in \mathcal{E}$, *we have that* $\Phi_1 \wedge \Phi_2$ *is unsatisfiable.*

We use DTA to determine whether a trajectory $X : \mathbb{R}_{\geq 0} \to \mathcal{S}$ satisfies a timed property. More precisely, the DTA reads the trajectory X and traverses edges between locations on the basis of (1) the states visited by the trajectory as time passes and (2) the current values of the clocks. The values of the clocks increase

at the same rate as real-time. The DTA *must* leave its current location ℓ without letting time pass if there exists an edge $(\ell, \Phi, \mathsf{C}, \ell') \in \mathcal{E}$ such that the enabling condition Φ is currently satisfied (hence, the DTA can be regarded as having an "urgent" semantics in which an enabled edge must be taken as soon as possible): this satisfaction of the enabling condition of the guard may occur, for example, because the value of a state variable falls below some threshold, or the value of a clock reaches a particular value. Furthermore, an additional constraint on the trajectory is imposed by the invariant conditions: during a period in which the DTA is in a particular location ℓ, the invariant condition $\mathrm{Inv}(\ell)$ must be satisfied by the states visited by the trajectory and by the current value of the clocks during that period, otherwise the trajectory will be regarded as not satisfying the timed property. A set of clocks can be reset to 0 when an edge is taken. If the DTA, starting from the initial location, reaches a final location when reading the trajectory X, then we say that the trajectory is accepted by the DTA (which, intuitively, corresponds to the trajectory X satisfying the timed property represented by the DTA), otherwise it is rejected.

In the following, we describe formally the acceptance of trajectories by a DTA. Let $(\ell, \Phi, \mathsf{C}, \ell') \in \mathcal{E}$ be an edge of a DTA $\mathcal{A}$. Then we write $\mathsf{source}(\ell, \Phi, \mathsf{C}, \ell') = \ell$, $\mathsf{guard}(\ell, \Phi, \mathsf{C}, \ell') = \Phi$, $\mathsf{reset}(\ell, \Phi, \mathsf{C}, \ell') = \mathsf{C}$, and $\mathsf{target}(\ell, \Phi, \mathsf{C}, \ell') = \ell'$. Let $\ell \in \mathcal{L}$ be a location of $\mathcal{A}$, and let $y \in \mathcal{S}$ be state and $v \in \mathrm{Val}(\mathcal{C})$. We write $(y, v) \not\models \mathsf{Guards}(\ell)$ if and only if $(y, v) \not\models \mathsf{guard}(e)$ for all $e \in \mathcal{E}$ such that $\mathsf{source}(e) = \ell$. A pair $(\ell, v) \in \mathcal{L} \times \mathrm{Val}(\mathcal{C})$ is called a *configuration*. We write $(\ell, v) \xrightarrow{\gamma, e} (\ell', v')$ to denote the DTA-transition from configuration (ℓ, v) to configuration (ℓ', v') after $\gamma > 0$ time units have elapsed and by taking the edge e. The transition $(\ell, v) \xrightarrow{\gamma, e} (\ell', v')$ exists if (1) $\mathsf{source}(e) = \ell$, (2) $v' = (v + \gamma)[\mathsf{reset}(e){:=}0]$, and (3) $\mathsf{target}(e) = \ell'$. A path of $\mathcal{A}$ is a finite sequence of DTA-transitions $\pi = (\ell_0, v_0) \xrightarrow{\gamma_0, e_0} (\ell_1, v_1) \xrightarrow{\gamma_1, e_1} \cdots \xrightarrow{\gamma_{m-1}, e_{m-1}} (\ell_m, v_m)$. Let $\Lambda^\pi = \{\lambda_0^\pi, \lambda_1^\pi, \ldots, \lambda_m^\pi\}$ be the set of constants such that $\lambda_0^\pi = 0$ and $\lambda_i^\pi = \sum_{k=0}^{i-1} \gamma_k$ for all i such that $1 \leq i \leq m$.

Definition 4. *Let $\mathcal{A}$ be a DTA. We say that $X : \mathbb{R}_{\geq 0} \to \mathcal{S}$ is accepted by the DTA if there exists a path $\pi = (\ell_0, v_0) \xrightarrow{\gamma_0, e_0} (\ell_1, v_1) \xrightarrow{\gamma_1, e_1} \cdots \xrightarrow{\gamma_{m-1}, e_{m-1}} (\ell_m, v_m)$ of $\mathcal{A}$ such that $\ell_0 = \ell_{\mathrm{init}}$, $v_0 = \mathbf{0}$, $\ell_m \in \mathcal{F}$ and, for all $0 \leq i < m$, the following conditions are satisfied:*

- *for all $0 \leq \gamma' < \gamma_i$, we have $(X(\lambda_i^\pi + \gamma'), v_i + \gamma') \models \mathrm{Inv}(\ell_i)$ and $(X(\lambda_i^\pi + \gamma'), v_i + \gamma') \not\models \mathsf{Guards}(\ell_i)$;*
- *$(X(\lambda_i^\pi + \gamma_i), v_i + \gamma_i) \models \mathsf{guard}(e_i)$.*

5 Assessing timed automata based properties by diffusion approximations

In this paper we limit our attention to illustrating the practical applicability of the approach. According to (9), there is a correspondence between the trajectories of the CTMC and those of the approximating jump diffusion process.

Moreover, the larger N is, the tighter the relation gets. It is natural hence to expect that over a certain threshold for N, which depends on the considered model, one can safely use trajectories of the diffusion process instead of trajectories of the CTMC to assess DTA-based temporal properties.

There is, however, a fundamental difference between a CTMC and a diffusion process. A diffusion process exhibits extreme oscillatory nature along its drift in any infinitesimal interval. This means that if a diffusion exceeds a given limit $\mathbf{L}$ for the first time then it goes below $\mathbf{L}$ with probability 1 afterwords in any infinitesimal interval. Consider now a diffusion process $X(t)$ and a DTA with three locations. The initial location is with invariant $X(t) \leq \mathbf{L}$ and has a transition enabled if $X(t) \geq \mathbf{L}$. The second location is with invariant $X(t) \geq \mathbf{L}$, it does not have an enabled transition associated with the situation $X(t) \leq \mathbf{L}$ and it has a transition enabled when $X(t) \geq 2\mathbf{L}$ that leads to the third location which is a final one. Due to the oscillatory nature of the diffusion process, with probability 0 a trajectory is accepted by the DTA. Note however that such a situation is coherent with what happens in a CTMC as N grows large. For large values of N, the trajectories of a CTMC are more and more similar to those of a diffusion. Consequently, the probability of the set of those trajectories of the CTMC that are accepted by the above described DTA tends to 0 as N tends to infinity. The characteristics of the diffusion process and that of the CTMCs with large N must be taken into account during the definition of the DTA in order to avoid results that are consequences of these characteristics and not the properties of the studied phenomenon. In practice, the problem is alleviated by using piecewise constant abstractions of the trajectories of the diffusion process.

We consider now the three kinds of approximations errors that occur during the analysis of CTMCs based on diffusion processes. First, the diffusion process is an approximation of the original CTMC. The goodness of this approximation was discussed in Section 3. Second, the analysis is carried out based on traces generated by approximate simulation. Indeed, exact simulation can be carried out only in special cases of diffusion (for example, in case of a Wiener process without drift) but in general the process is multidimensional and it

Fig. 1. Two versions of the same trajectory of a pure Wiener process with different time steps.

includes a state dependent drift and a state dependent noise that cannot be simulated exactly in general. Third, a diffusion process fluctuates in any infinitesimal interval which means that it is not possible to obtain a complete representation of a trace. Indeed, the temporal properties are assessed based on a constant piecewise approximation of an infinitely fluctuating trace.

Let us illustrate this third source of approximation error in some detail. In Figure 1 we plotted two versions of the same trajectory with two different time steps (δ). Consider now a DTA which accepts only those traces along which
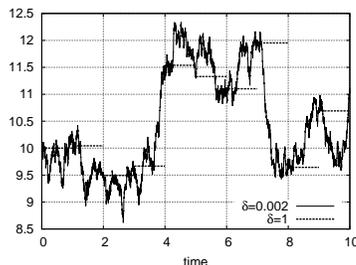
name	description	init. values
A	Axin2 protein	0
A_m	Axin2 mRNA	0
G	GSK3 protein	$50 \cdot N$
L	LRP5/6 coreceptor	$20 \cdot N$
B	free β-catenin	0
AL	Axin2-LRP5/6 complex	$50 \cdot N$
GA	GSK3-Axin2 complex	0
C	GSK3-Axin-β-catenin complex	0

Table 1. Species of the Wnt pathway.

$$R_1 : C \xrightarrow{k_1} B + GA \qquad R_7 : Am \xrightarrow{k_7} A + Am$$
$$R_2 : C \xrightarrow{k_2} GA \qquad R_8 : A + L \xrightarrow{k_8} AL$$
$$R_3 : B + GA \xrightarrow{k_3} C \qquad R_9 : AL \xrightarrow{k_9} A + L$$
$$R_4 : A + G \xrightarrow{k_4} GA \qquad R_{10} : 2B \xrightarrow{k_{10}} 2B + Am$$
$$R_5 : GA \xrightarrow{k_5} G + A \qquad R_{11} : Am \xrightarrow{k_{11}} \varnothing$$
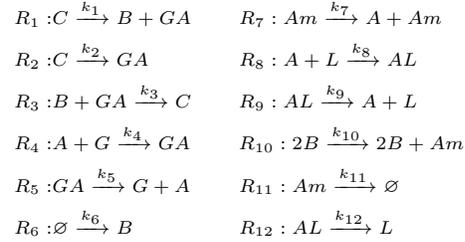$$R_6 : \varnothing \xrightarrow{k_6} B \qquad R_{12} : AL \xrightarrow{k_{12}} L$$

Table 2. Reactions of the Wnt pathway.

the process never exceeds level 12 in the time interval $[0, 10]$. Clearly, using the piecewise constant abstraction of the trace, the trace with $\delta = 1$ is accepted while the other is rejected.

The previous example indicates that the choice of the time step during the generation of the traces is of fundamental importance to achieve good approximation of the original behavior. The same problem, i.e., not knowing the fluctuation between two consecutive time points, is present to a somewhat lesser extent also when jump diffusion processes are used to obtain approximations of more classical measures, like transient probabilities. In that case, it is of crucial importance to find with sufficient precision the time instants when the process reaches the boundary, i.e., the time instants when the change from pure diffusion process to jump diffusion process has to be made. When assessing temporal properties described by DTA, the problem appears also inside the state space around the thresholds present in the automaton. The choice of the time step was discussed to some extent in [3, 7]. In theory, it is possible to add intermediate points given a trace but this can be done only in very special cases, like the one used before, i.e., the pure Wiener process.

All the three kinds of approximation error decrease as the indexing parameter N increases. Numerical experiments suggest that in the situation when it is reasonable to use the diffusion approximation, i.e., when the CTMC is too large for the analysis but there are still important stochastic behaviors in the system, the approximation errors are in an acceptable range.

6 Experimental results

The experimental results described in this section were carried out using a prototype implementation integrated in the GreatSPN suite [2], for the SDE part, and with the COSMOS statistical model checker [6] (which uses a generalisation of the DTA formalism [5]), for the CTMC part.

Case study: A model of the Wnt pathway. We consider a model of the Wnt/β-catenin pathway, an intracellular signalling pathway involved in neuroinflammation, a key mechanism in numerous brain diseases [14]. Such model [15] accounts

kinetic rates							
k_1	7	k_4	$0.2/N$	k_7	0.7	k_{10}	$0.7/N$
k_2	200	k_5	1.2	k_8	$10/N$	k_{11}	0.025
k_3	$0.1/N$	k_6	$0.4 \cdot N$	k_9	0.08	k_{12}	0.1

initial population	
G	$N \cdot 50$
L	$N \cdot 20$
AL	$N \cdot 50$

Table 3. Kinetic rate constants and initial populations for the DDMC model of the Wnt pathway (both dependent on index N).
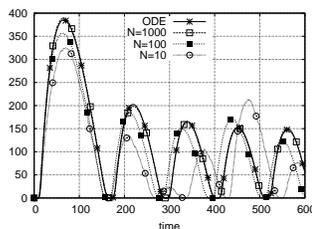

Fig. 2. Sample paths of A (Axin2 protein) for various values of N and the behavior obtained by the ODE description.

for 8 biochemical species (Table 1) regulated through 12 reactions (Table 2). It consists of three main actors: the β-catenin (denoted B) and Axin2 proteins (A), forming a negative feedback loop, and the Wnt protein (here subsumed by the LRP5-6 membrane receptor, i.e., L), representing the extracellular signal. The behavior can be summarised as follows. With scarcity of extracellular Wnt molecules (low L), a degradation complex (C, a trimer resulting by β-catenin binding to previously formed GSK3-Axin2 dimer, i.e., GA) causes the phosphorylation and subsequent destruction of β-catenin located in the cell's cytosol. On the other hand with an abundant Wnt signal (high L), the degradation complex is deactivated (as Axin is degraded through reversibly binding with receptor L, i.e., forming the AL complex) resulting in an accumulation of β-catenin which in turn activates (through transcription of the Axin2 messenger RNA, i.e., A_m) the expression Axin2, and therefore determining its own destruction (i.e., negative feedback loop).

In [15] the model is given in ODE form and it is shown to exhibit sustained oscillations (Figure 2) for specific parameter settings. Here we consider a sequence of DDMCs indexed by N (here proportional to the volume) and of the parameters of the ODE model in [15][5]. Table 3 depicts the kinetic rate constants and the initial populations of the Wnt-pathway DDMC [6] whereas Figure 2 compares species A's projection of a sample path of the CTMC for various values of N with the deterministic trajectory of the corresponding ODEs (notice that for readability the CTMC paths have been normalised, i.e., the molecule count of A is divided by N). Furthermore observe that ODEs exhibit sustained oscillations, which, after the second period, have almost constant amplitude, and that, for increasing values of N, CTMC trajectories approximated quite accurately the ODE's. The choice of N when analyzing a real scenario depends on the consid-

---

[5] I.e., for $N = 1$ we assumed the *discrete* initial populations and reaction intensities being equal to the *continuous* ones as given in [15], note that this is in agreement with a cell volume $V = 10^9/n_A$ where n_A is the Avogadro number given that species concentrations and kinetic rate constants of the ODE model are expressed in nM.

[6] Notice that zero-order and second-order reactions' rates are dependent on N because for these conversion from continuous to discrete rates depends on cell's volume.
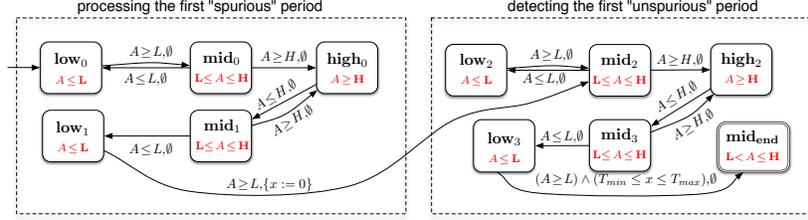
Fig. 3. DTA to study the oscillation period of Axin2 proteins.

ered volume; in wet-lab experiments typically molecular value is usually greater than 500.

Automaton. Inspired by Mikeev *at al.* [20] we propose a DTA (Figure 3) to measure the duration of the period exhibited by the population of the Axin2 protein. The rationale for *noisy period* detection [20] is to split the domain of the observed species, i.e., A, in three subintervals: **low** (i.e., $A \leq \mathbf{L}$), **mid** (i.e., $L < A \leq \mathbf{H}$) and **high** (i.e., $A \geq \mathbf{H}$)[7]. A *noisy period realisation* [4] corresponds to the time interval occurring between two successive entries to the **low** region of the state-space interleaved by a visit to the **high** region. The single clock DTA in Figure 3 is indeed designed to detect the first noisy period realisation of species A. It consists of two parts: the first one processes the initial **low₀**-**high₀**-**low₁** traversal (representing a spurious period), at the end of which (**low₁**-**mid₂** transition) the clock x is reset to start timing the realisation of the first non-spurious period whose termination corresponds with the **low₃**-**mid_end** edge. Note that ignoring the first spurious period (through the first part of the DTA) is necessary since to detect a complete period we need to identify the actual starting point (i.e., the first **low**-**mid** crossing that follows a visit to **high**) which we cannot do from the initial state because the system starts at $A = 0$. Furthermore note that trajectories are accepted on condition that the observed duration of the first period is within $T_{min} \leq x \leq T_{max}$ which, by choosing different values for T_{min} and T_{max}, allows us to assess the probability density of the period duration (Figure 4). Observe that for any element $N > 1$ of the Wnt-DDMC sequence the probability of non-sustainably oscillating paths (i.e., paths non-perpetually traversing the **low**-**mid**-**high** regions) is negligible, therefore, given that L and H are properly chosen (so to be above, resp. below, the average height of minimal, resp. maximal peaks of oscillations), the DTA accepts all trajectories of the model.

Experiments. We compare the proposed approach by assessing of DTA-based oscillation-period properties on both a few CTMCs (of the Wnt-DDMC se-

---

[7] **L** and **H**, where $\mathbf{L} < \mathbf{H}$, are two thresholds chosen so that the minimal, resp. maximal, peaks of oscillation are most likely to fall below **L**, resp. above **H**.

			SDE		$\frac{\textbf{Jump}}{\textbf{Tot.}}$	COSMOS		
N	**L**	**H**	**Time**	**Average period**		**Time**	**Average period**	**Speedup**
100	50	1×10^4	180h.	[103.174, 103.344]	0.90	30h.	[112.628, 113.133]	0.17
500	50	5×10^4	88h.	[111.021, 111.115]	0.76	167h.	[112.654, 112.784]	1.9
1,000	500	1×10^5	54h.	[111.977, 112.089]	0.60	344h.	[112.381, 112.499]	6,37
2,000	500	1.5×10^5	39h.	[112.408, 112.474]	0.45	705h.	[112.504, 112.578]	18.08
5,000	500	5×10^5	28h.	[112.693, 112.725]	0.26	1763h.	[112.708, 112.749]	62.96

Table 4. Comparing SDE and COSMOS results considering 10000 traces.

quence) and on their SDE approximation. The experiments were executed on a server with 48 core AMD Opteron(tm) Processor 6176 by considering five CTMCs of the Wnt-DDMC sequence corresponding to the following values of N, i.e., $N \in \{100, 500, 1000, 2000, 5000\}$, and while the SDE results were computed with the GSPN prototype the CTMCs results were computed with COSMOS [6] which, to the best of our knowledge, is one of the most efficient statistical model checkers. We have run two families of experiments. The first one is devoted to assessing the density function of the oscillation period (Figure 4) and employs the DTA of Figure 3. The second one is devoted to comparing both runtime and accuracy of the two approaches w.r.t. estimating the duration of the oscillation period (Table 4) and employs a slightly modified DTA[8].

Table 4 compares, as a function of N, the execution times and the confidence intervals for the mean duration of the first non-spurious oscillation period with confidence level set to 0.99 in case of generating 10000 traces. Columns two and three depict the value of the **L**, resp. **H**, parameter of the DTA; the fourth and fifth (resp. seventh and eighth), columns show the runtime and estimated confidence-interval computed through our SDE prototype (resp. COSMOS); the sixth column shows the proportion of the number of jumps occurred because of hitting the border during the simulation of the SDE; finally the ninth column gives the speed up obtained by our SDE approach. In Figure 4 the probability density functions (pdf) of the length of the first non-spurious period are plotted for N equal to 100, 500, 1000 and 2000. The pdf for N equal to 5000 (not shown for the lack of space) confirms the trend toward a closer correspondence between the SDE and COSMOS results.

Discussion. As expected, the SDE approach becomes more convenient, in terms of runtime and precision, as N increases.[9] In particular, for $N = 100$ the precision is strongly affected by the SDE approximation error, moreover the SDE execution time is greater than COSMOS since for each SDE trace the process hits the

---

[8] I.e., we use the DTA in Figure 3 but without the $T_{min} \leq x \leq T_{max}$ conjunct on the edge from **low$_3$-mid$_{\textbf{end}}$**, which allows us to obtain the value of the clock x at the moment of reaching the final location of this modified DTA: this value gives the length of the first non-spurious oscillation period.

[9] Observe that the dimension of integration step is dynamically computed through a heuristic function which provides a good trade-off between speed-up and precision of the solution
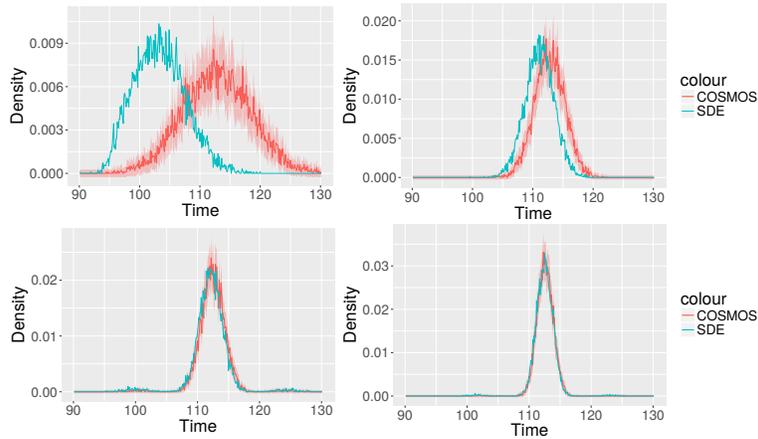
Fig. 4. Probability mass functions of the length of the first non-spurious oscillation period with bin length equal to 0.1 and with N equal to 100 (top left), 500 (top right), 1000 (bottom left), 2000 (bottom right).

boundaries, on average, along 90% of a trace length (col. 6, Table 4). With $N = 1000$ the SDE based analysis is about twice faster than that based on the CTMC (col. 9, Table 4) while the precision is acceptable: indeed the SDE approach is able to reproduce the multimodal behavior of the pdf generated by COSMOS (Figure 4 bottom-left plot, i.e., $N = 1000$). Such trend is confirmed by experiments with $N = 2000$, (speedup $\sim$6x, even closer approximation) and with $N = 5000$ (exhibiting a 63x speedup obtained with the SDE approach).

7 Conclusions

In this paper we presented a framework that allows for assessing temporal properties, described in terms of DTA, of DDMCs through their jump diffusion approximation. The applicability of the approach was illustrated through a case study regarding a biological oscillator. As future work we aim to study the theoretical limits of assessing DTA-based temporal properties of diffusion processes. Furthermore, the approach can be extended to hybrid jump diffusion processes, which are obtained by partial fluidification of DDMCs, that are useful to study systems in which not all population counts are high and thus fluidification of all state variables would lead to large approximation errors.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

2. E. G. Amparore, M. Beccuti, and S. Donatelli. (Stochastic) model checking in GreatSPN. In *Proc. PETRI NETS 2014*, volume 8489 of *LNCS*, pages 354–363. Springer, 2014.

3. A. Angius, G. Balbo, M. Beccuti, E. Bibbona, A. Horvath, and R. Sirovich. Approximate analysis of biological systems by hybrid switching jump diffusion. *Theor. Comput. Sci.*, 587:49 – 72, 2015.

4. P. Ballarini. Analysing oscillatory trends of discrete-state stochastic processes through HASL statistical model checking. *STTT*, 17(4):505–526, 2015.

5. P. Ballarini, B. Barbot, M. Duflot, S. Haddad, and N. Pekergin. HASL: A new approach for performance evaluation and model checking from concepts to experimentation. *Perform. Evaluation*, 90:53–77, 2015.

6. P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin. COSMOS: A statistical model checker for the hybrid automata stochastic logic. In *Proc. QEST 2011*, pages 143–144. IEEE Computer Society, 2011.

7. M. Beccuti, E. Bibbona, A. Horváth, R. Sirovich, A. Angius, and G. Balbo. Analysis of Petri net models through stochastic differential equations. In *Proc. PETRI NETS 2014*, volume 8489 of *LNCS*, pages 273–293. Springer, 2014.

8. E. Bibbona and R. Sirovich. Strong approximation of density dependent Markov chains on bounded domains by jump diffusion processes. Technical report, Università di Torino, 2017.

9. L. Bortolussi and J. Hillston. Model checking single agent behaviours by fluid approximation. *Inf. Comput.*, 242:183–226, 2015.

10. L. Bortolussi and R. Lanciani. Model checking Markov population models by Central Limit Approximation. In *Proc. QEST 2013*, volume 8054 of *LNCS*, pages 123–138. Springer, 2013.

11. L. Bortolussi and R. Lanciani. Fluid model checking of timed properties. In *Proc. FORMATS 2015*, volume 9268 of *LNCS*, pages 172–188. Springer, 2015.

12. T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Model checking of continuous-time Markov chains against timed automata specifications. *Log. Meth. in Comput. Sci.*, 7(1), 2011.

13. S. Donatelli, S. Haddad, and J. Sproston. Model checking timed and stochastic properties with CSL^{TA}. *IEEE T. Software Eng.*, 35(2):224–240, 2009.

14. P. Gressens, J. V. Steenwinckel, A. Schang, S. Sigaut, V. Degos, S. Lebon, L. Schwendimann, T. Le Charpentier, H. Hagberg, N. Soussi, and B. Fleiss. Microglial Wnt signaling inhibition promotes microglia activation and oligodendrocyte maturation blockade. *J. Neurochem.*, 134, 2015.

15. P. B. Jensen, L. Pedersen, S. Krishna, and M. H. Jensen. A Wnt oscillator model for somitogenesis. *Biophys. J.*, 98(6):943–950, 2010.

16. A. Kolesnichenko, P. de Boer, A. Remke, and B. R. Haverkort. A logic for model-checking mean-field models. In *Proc. DSN 2013*, pages 1–12. IEEE Computer Society, 2013.

17. T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *J. Appl. Probab.*, 1(7):49–58, 1970.

18. T. G. Kurtz. Limit theorems and diffusion approximations for density dependent Markov chains. In *Stochastic Systems: Modeling, Identification and Optimization, I*, pages 67–78. Springer, 1976.

19. T. G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stoc. Proc. Appl.*, 6(3):223–240, 1978.

20. L. Mikeev, M. R. Neuhäußer, D. Spieler, and V. Wolf. On-the-fly verification and optimization of DTA-properties for large Markov chains. *Form. Method. Syst. Des.*, 43(2):313–337, 2013.