



HAL
open science

A Formal Approach for Tuning Stochastic Oscillators

Paolo Ballarini, Mahmoud Bentrion, Paul-Henry Cournède

► **To cite this version:**

Paolo Ballarini, Mahmoud Bentrion, Paul-Henry Cournède. A Formal Approach for Tuning Stochastic Oscillators. Computational Methods in Systems Biology, Sep 2023, Luxembourg, Luxembourg. pp.1-17, 10.1007/978-3-031-42697-1_1. hal-04570564

HAL Id: hal-04570564

<https://hal.science/hal-04570564>

Submitted on 14 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Formal Approach For Tuning Stochastic Oscillators

Paolo Ballarini¹, Mahmoud Bentrion¹, and Paul-Henry Cournède¹

MICS, CentraleSupélec, Université Paris-Saclay, France.

`paolo.ballarini@centralesupelec.fr`

`mahmoud.bentrion@centralesupelec.fr`

`paul-henry.cournede@centralesupelec.fr`

Abstract. Periodic recurrence is a prominent behavioural of many biological phenomena, including cell cycle and circadian rhythms. Although deterministic models are commonly used to represent the dynamics of periodic phenomena, it is known that they are little appropriate in the case of systems in which *stochastic noise* induced by small population numbers is actually responsible for periodicity. Within the stochastic modelling settings automata-based model checking approaches have proven an effective means for the analysis of oscillatory dynamics, the main idea being that of coupling a *period detector* automaton with a continuous-time Markov chain model of an alleged oscillator. In this paper we address a complementary aspect, i.e. that of assessing the dependency of oscillation related measure (period and amplitude) against the parameters of a stochastic oscillator. To this aim we introduce a framework which, by combining an Approximate Bayesian Computation scheme with a hybrid automata capable of quantifying how *distant* an instance of a stochastic oscillator is from matching a desired (average) period, leads us to identify regions of the parameter space in which oscillation with given period are highly likely. The method is demonstrated through a couple of case studies, including a model of the popular Repressilator circuit.

Keywords: Stochastic oscillators, Approximate Bayesian Computation, Parameter estimation, Hybrid Automata Stochastic Logic, Statistical Model-Checking.

1 Introduction

Oscillations are prominent dynamics at the core of many fundamental biological processes. They occur at different level and concern different time scales, ranging from ion channels regulated transmission of intercellular electrical signal driving the heartbeat (with period in the order of one second), to intracellular calcium oscillations triggering glycogen-to-glucose release in liver cells (with periods ranging from few seconds to few minutes [30]), to gene-expression regulated circadian cycle (with typical period of roughly 24 hours).

Mathematical modelling and computational methods have proved fundamental to gain a better understanding of the complexity of the mechanisms that

regulate oscillations [22]. Although continuous-deterministic models (i.e. ODEs) are more often considered in the literature of biological oscillators, with systems characterised by low population numbers (e.g., genetic circuits responsible for *circadian rhythms*), discrete-stochastic models are more appropriate [24]: when few molecules are involved the stochasticity of the system becomes important resulting in noisy periodic behavior. Understanding the effect that an oscillator’s parameters have on the quality of the oscillations is a relevant research problem.

Contribution. We introduce a methodology for calibrating stochastic models that exhibit a noisy periodic dynamics. Given a parametric stochastic oscillator the methodology allows to identify regions of the parameter space with a positive probability of matching a given target oscillation period. This entails integrating formal means for noisy periodicity analysis within a parameter inference approach, more specifically, integrating a *period distance meter* (formally encoded as a Hybrid Automaton) within an Approximate Bayesian Computation scheme. Given some period-related requirements (e.g. mean value and variance of the oscillation period) the trajectories sampled from the oscillator are classed through the distance period meter leading us to the approximation of the corresponding posterior distribution, that is, to the identification of the region of the parameter space that are more likely to comply with the considered period-related requirements.

Paper organisation. The paper is organised as follows: in Section 2 we overview the background material our approach relies upon, that includes an overview of HASL model checking and of ABC algorithms. In Section 3 we introduce our approach for calibrating stochastic oscillators w.r.t. to the oscillation period. The method is then demonstrated through experiments presented in Section 4. Conclusive remarks and future perspective are discussed in Section 5.

1.1 Related work

Temporal logic based analysis of stochastic oscillators. The analysis of oscillatory behaviour entails two complementary aspects: determining whether a model exhibits recurrent patterns (detection) and assessing relevant periodicity indicators (e.g. period and amplitude of oscillations). In the continuous-deterministic settings these can be achieved through a combination of mathematical approaches including structural analysis of the corresponding ODE system and stability analysis of its *steady-state* solutions [22]. As those approaches clearly do not apply in the discrete-stochastic settings researchers progressively started looking at alternatives such as the adaptation of model checking to the analysis of periodicity [1]. Detection of sustained oscillations through model checking requires identifying temporal logical formulae that single out infinite cyclic (non-constant) behaviour. Seminal ideas introduced for (non-probabilistic) *transition systems* models [18] and further developed in [8] included Computational Tree Logic (CTL) [19] *qualitative* specifications such as

$$EG(((X_i = k) \Rightarrow EF(X_i \neq k)) \wedge ((X_i \neq k) \Rightarrow EF(X_i = k))) \quad (1)$$

$$AG(((X_i = k) \Rightarrow EF(X_i \neq k)) \wedge ((X_i \neq k) \Rightarrow EF(X_i = k))) \quad (2)$$

which encode *infinite alternation* by demanding that for “at least one path” (1) or for “all paths” (2) of a given model a state condition $X_i = k^1$ can be met and then left (and, inversely, whenever it is not met $X_i \neq k$ it is possible to eventually meet it) infinitely often. In the stochastic, continuous time Markov chains (CTMCs), settings the Continuous Stochastic Logic (CSL) [2] counterpart of (2) leads to the following qualitative probabilistic formulae (introduced in [8]):

$$P_{=1}(((X_i = k) \Rightarrow P_{>0}(X_i \neq k)) \wedge ((X_i \neq k) \Rightarrow P_{>0}(X_i = k))) \quad (3)$$

which identifies those states of a CTMC for which the probability of outgoing paths that infinitely alternate between $X_i = k$ and $X_i \neq k$ states adds up to 1. If formulae such as (3) can be used to rule out CTMCs that do not oscillate sustainably (as they contain at least an absorbing state), they fall short w.r.t. effectively detecting sustained oscillators as they are satisfied by any *ergodic* CTMC regardless whether it actually exhibits sustained oscillations. The problem is that, in the stochastic settings, a logical characterisation such as (3) is too weak to allow for distinguishing between models that gather probability mass on actual oscillatory paths (of given amplitude and period) from those whose probability mass is concentrated on infinite noisy fluctuations which do not correspond with actual oscillations. Further developments included employing the probabilistic version of the linear time logic (LTL) [28] to characterise oscillations based on a notion of *noisy monotonicity* [5]: although an improvement w.r.t. to the limitation of the CSL based characterisation of periodicity this approach still fails to satisfactorily treat the oscillation detection problem.

Automata-based analysis of stochastic oscillators. To work around the limited suitability of temporal logic approaches Spieler [31] proposed to employ a single-clock deterministic timed automata (DTA) as *noisy period detector*. The idea is that, through synchronisation with a CTMC model, such DTA is used to accept *noisy periodic* paths described as those that infinitely alternate between crossing a lower threshold L , and a higher threshold H (hence corresponding to fluctuations of minimal amplitude $H-L$) and by imposing that such fluctuations should happen with a period falling in a chosen interval $[t_p^{min}, t_p^{max}]$ (with H , L , t_p^{min} and t_p^{max} being parameters of the DTA). The issues that CSL detector formulae such as (3) suffer from are overcome, as by properly settings thresholds L and H the DTA rules out non-oscillating ergodic CTMC models. The detection procedure boils down to computing the probability measure of all CTMC paths that are accepted by the DTA which is achieved through numerical procedures (requiring the construction of the CTMC x DTA product process). Spieler’s original idea has then further evolved by resorting to the more expressive hybrid automata as detectors of periodicity [6, 7]. This allowed, on one hand, to account for more sophisticated oscillation related indicators such as the variance (other than the mean value) of the period of a stochastic oscillator and also to develop an alternative, *peaks detector*, automaton which, differently from Spieler’s *period detector*, does not depend on the chosen L and H thresholds.

¹ the population of species i is $k \in \mathbb{N}$

Parametric verification of stochastic models. Parametric verification of a probabilistic model is concerned with combining parameter estimation techniques with stochastic model checking, i.e. with studying how the stochastic model checking problem for a property Φ is affected by the parameters θ upon which a probabilistic model \mathcal{M}_θ depends. A number of approaches have been proposed in the literature such as [15], in which a bounded approximation of parameter space fulfilling a CSL [2] threshold formula is efficiently determined through an adaptation of *uniformisation*, or also [17, 23] and, more recently, a novel ABC-based method [27] that is based on observations even solves parameter inference and statistical parameter synthesis in one go.

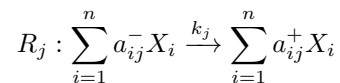
The framework we introduce in this paper, on the other hand, tackles the *estimation problem* in stochastic model checking and is in line with the works of Bor-tolussi *et al.* [14], where the so-called smoothed model checking (Smoothed MC) method to estimate the satisfaction probability function of parametric Markov population models is detailed. The goal here is to obtain a good estimate of the so-called satisfaction probability function of the considered property Φ w.r.t. to model’s parameter space Θ . In our case we rely on hybrid automata-based adaptations of Approximate Bayesian Computation (ABC) schemes [11, 12] to estimate the satisfaction probability function, similarly to [26] where ABC algorithms are combined with statistical model checking to approximate the satisfaction function of non-nested CSL reachability formulae.

2 Background

We briefly overview the notion of chemical reaction network (CRN) models, of Hybrid Automata Stochastic Logic (HASL) model checking and the Approximate Bayesian Computation (ABC) scheme for parameter estimation that Section 3 relies on.

Chemical Reaction Network and discrete-stochastic interpretation.

We consider models that describe the time evolution of biochemical species X_1, X_2, \dots which interact through a number of reactions R_1, R_2, \dots forming a so-called chemical reaction network (CRN) expressed as a system of chemical equations with the following form:



where a_{ij}^- (a_{ij}^+), are the stoichiometric coefficients of the *reactant* (*product*), species and k_j is the kinetic constant of the j -th reaction R_j of the CRN. We assume the discrete stochastic interpretation of CRNs models, i.e. species X_i represent molecules counting (rather than concentrations) hence models consist of countable states $x = (x_1, \dots, x_n) \in \mathbb{N}^n$ representing the combined populations count while the state transitions consist of time-delayed jumps governed by a probability distribution function. In case of Exponentially distributed jumps the underlying class of models is that of continuous-time Markov chains (CTMCs),

conversely in case of generically distributed jumps we refer to underlying class of models as of discrete event stochastic process (DESP). A path of a DESP model is a (possibly infinite) sequence of jumps denoted $\sigma = x^0 \xrightarrow[R^0]{t_0} x^1 \xrightarrow[R^1]{t_1} x^2 \xrightarrow[R^2]{t_2} \dots$ with x^i being the i -th state, $t_i \in \mathbb{R}_{>0}$ being the sojourn-time in the i -th state and R^i being the reaction whose occurrence issued the $(i+1)$ -th jump. Notice that paths of a DESP are *càdlàg* (i.e. step) functions of time (e.g. Figure 2). A DESP depend on a p -dimensional vector of parameters $\theta = [\theta_1, \dots, \theta_p] \in \Theta \subseteq \mathbb{R}^p$, which, in the context of this paper, affect the kinetic rate of the reaction channels. For $\theta \in \Theta \subseteq \mathbb{R}^p$ we denote \mathcal{M}_θ the corresponding DESP model. Given \mathcal{M}_θ we let $Path(\mathcal{M}_\theta)$ denote the set of paths of \mathcal{M}_θ . It is well known that a DESP model \mathcal{M}_θ induces a probability space over the set of events $2^{Path(\mathcal{M}_\theta)}$, where the probability of a set of paths $E \in 2^{Path(\mathcal{M}_\theta)}$ is given by the probability of their common finite prefix [3]. For \mathcal{M}_θ an n -dimensional DESP population model, we denote $dom(\mathcal{M}_\theta)$ its state space, and $dom_i(\mathcal{M}_\theta)$ the projection of $dom(\mathcal{M}_\theta)$ along the i^{th} dimension $1 \leq i \leq n$ of \mathcal{M}_θ . For σ a path of an n -dimensional \mathcal{M}_θ we denote σ_i its projection along the i -th dimension.

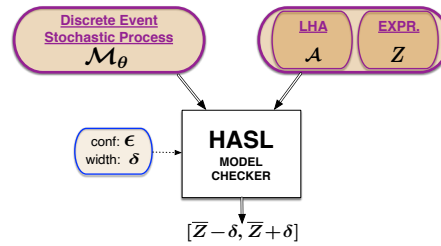


Fig. 1: HASL-SMC schema: sampled paths are filtered by a LHA and the accepted ones used for a confidence interval estimate of the target measure.

2.1 HASL model checking

The procedure for tuning stochastic oscillators we introduce in the remainder relies on the HASL statistical model checking (SMC) framework. We quickly overview the basic principles of HASL-SMC referring the reader to the literature [4] for more details. HASL-SMC is a framework for assessing properties of stochastic model expressed by means of a linear hybrid automaton (LHA). It consists of an iterative procedure which takes 3 inputs (Figure 1): a parametric DESP model \mathcal{M}_θ , a LHA \mathcal{A} , and a target expression denoted Z and outputs a confidence interval estimation of the mean value \hat{Z} of the target measure. At each iteration the procedure samples a path $\sigma \in Path(\mathcal{M}_\theta \times \mathcal{A})$ of the product process $\mathcal{M}_\theta \times \mathcal{A}$ (whose formal semantics is given in [4]), that means that a path $\sigma \in Path(\mathcal{M}_\theta)$ is sampled and synchronised *on-the-fly* with \mathcal{A} leading to either acceptance or rejection of σ . In the synchronisation process, relevant statistics

of the simulated path are computed and are used to define the target measure Z of interest (see [10]). This let HASL-SMC be a very expressive formalism that allows for sophisticated signal-processing-like analysis of a model.

Synchronisation of a model with a LHA. The semantics of the synchronised process $\mathcal{M} \times \mathcal{A}$ naturally yields a stochastic simulation procedure which is implemented by the HASL model checker. For the sake of space here we only provide an intuitive (informal) description of such synchronisation based on the example given in Figure 2. An LHA consists of a set of locations L (with at least one initial and one final location) a set of real-valued variables V whose value may evolve, according to a flow function which map each location to a (real-valued) rate of change to the variables, during the sojourn in a location. Locations change through transitions denoted $l \xrightarrow{\gamma, E', U} l'$ where γ is an enabling guard (an inequality built on top of variables in V), E' is either set of events names (i.e. the transition is *synchronously* traversed on occurrence of any reaction in E' occurring in the path being sampled) or \sharp (i.e. the transition is *autonomously* traversed without synchronisation) and U are the variables' update. A state of $\mathcal{M}_\theta \times \mathcal{A}$ is a 3-tuple (s, l, ν) , with s the current state of \mathcal{M}_θ , l the current location of \mathcal{A} and $\nu \in \mathbb{R}^{|V|}$ the current values of \mathcal{A} 's variables. Therefore if $\sigma : s \xrightarrow[e_1]{t_1} s_1 \xrightarrow[e_2]{t_2} s_2 \dots$ is a path of \mathcal{M}_θ the corresponding path $\sigma \times \mathcal{A} \in \text{Path}(\mathcal{M}_\theta \times \mathcal{A})$ may be $\sigma \times \mathcal{A} : (s, l, \nu) \xrightarrow[e_1]{t_1} (s_1, l_1, \nu_1) \xrightarrow[\sharp]{t_1^*} (s_1, l_2, \nu_2) \xrightarrow[e_2]{t_2} (s_2, l_3, \nu_3) \dots$ where, the sequence of transitions e_1 and e_2 observed on σ is interleaved with an autonomous transition (denoted \sharp) in the product process: i.e. from state (s_1, l_1, ν_1) the product process jumps to state (s_1, l_2, ν_2) (notice that state of \mathcal{M}_θ does not change) before continuing mimicking σ .

Example 1. Figure 2 depicts an example of synchronisation between a path of a toy 3-species, 3-reactions CRN model (left), with a 2-locations LHA (right) designed for assessing properties of it. The LHA locations are l_0 (initial) and l_1 (final) while its variables are $V = \{t, x_1, n_2\}$ with t a clock variable, x_1 a real valued variable (for measuring the average population of A) and n_2 an integer variable (for counting the number of occurrences of the R_1 reaction). While in l_0 the value of variables evolves according to their flow, which is constant and equal to 1 for clock t , while is given by the current population of A for x_1 (therefore x_1 measures the integral of population A along the synchronising path). The synchronisation of a path σ with \mathcal{A} works as follows: \mathcal{A} stays in the initial location l_0 up until at $t = 4$ the autonomous transition the synchronisation with σ ends as soon as the *autonomous* transition $l_0 \xrightarrow[\sharp, t=4, \{x_1/=4\}]{}$ l_1 becomes enabled (guard $t = 4$ gets true) hence is fired (by definition autonomous transitions have priority over *synchronised* transitions). As long as $t < 4$ the LHA is in l_0 where it synchronises with the occurrences of the reactions of the CRN model: on occurrence of R_1 the $\mathcal{M} \times \mathcal{A}$ transition $l_0 \xrightarrow[\{R_1\}, t < 4, \{n_2++\}]{}$ l_0 (*synchronised* on event set $\{R_1\}$) is fired hence increasing the counter n_2 , whereas on occurrence of any other reaction transition $l_0 \xrightarrow[ALL \setminus \{R_1\}, t < 4, \emptyset]{}$ l_0 (*synchronised* on event

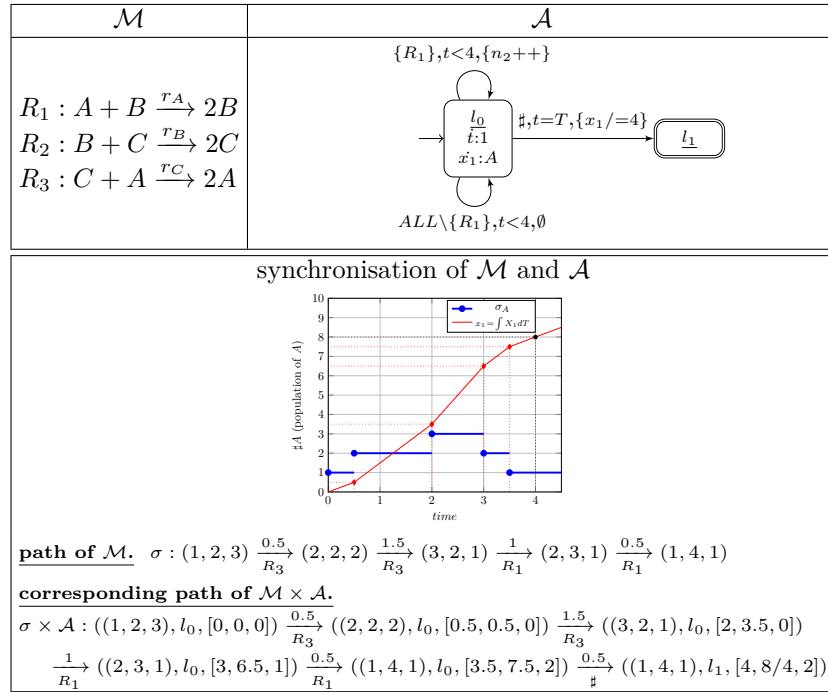


Fig. 2: Example of synchronisation of a CRN model with a LHA.

set $ALL \setminus \{R_1\}$, where ALL denotes all reactions of the CRN) fires without updating any variable. Finally on ending the synchronisation with σ variable x_1 is update to $x_1/4$ which corresponds to average population of A observed over the time interval $[0, 4]$. Such a LHA can therefore be used (through iterated synchronisation with a sufficiently large number of trajectories) for estimating the confidence interval of random variables such as the “average population of A ” as well as the “number of R_1 occurrences” observed over time interval $[0, 4]$. An example of synchronisation between a path σ consisting of 2 occurrences of R_3 (at $t = 0.5$, $t = 2.0$ respectively) followed by 2 occurrences of R_1 (at $t = 3.0$ and $t = 3.5$ respectively) and for which $(A_0, B_0, C_0) = (1, 2, 3)$ is assumed as the initial state is depicted at bottom of Figure 2. The synchronised path shows the combined evolution of the model’s state, the LHA location and the value of LHA variables. Notice that when synchronisation ends (l_1 is reached) variable x_1 is assigned with $8/4$ which indeed is the mean population of A along σ until $t = 4$.

2.2 Approximate Bayesian Computation

The framework for tuning of stochastic oscillators we present in the remainder relies on the integration of HASL-based measurements within the class of *Bayesian*

inference methods known as Approximate Bayesian Computation (ABC) [25,29]. Generally speaking *statistical inference* is interested with inferring properties of an underlying distribution of probability (in our case f_θ) based on some data observed through an experiment y_{exp} . Bayesian inference methods, on the other hand, rely on the Bayesian interpretation of probability, therefore starting from some *prior distribution* $\pi(\cdot)$, which expresses an *initial belief* on the distribution to be estimated over the parameters domain Θ , they allow for computing the *posterior distribution* $\pi(\theta|y_{exp})$, that is, the target probability distribution over Θ based on the observed data y_{exp} . Formally, in Bayesian statistics, the posterior distribution is defined by:

$$\pi(\theta|y_{exp}) = \frac{p(y_{exp}|\theta)\pi(\theta)}{\int_{\theta'} p(y_{exp}|\theta')\pi(\theta') d\theta'}$$

where $p(\cdot|\theta)$ denotes the *likelihood function*, that is, the function that measures how probable y is to be observed given the model's parameters θ . An inherent drawback of Bayesian statistics is in that, by definition, the posterior distribution relies on the accessibility to the likelihood function $p(y_{exp}|\theta)$ which, particularly for complex models, may be too expensive to compute or even intractable. ABC algorithms have been introduced to tackle this issue, i.e. as a *likelihood-free* alternative to classical Bayesian methods (we refer to [25,29] for exhaustive surveys of ABC or rejection-sampling methods). The basic idea behind the ABC method is to obtain an estimate, denoted $\pi_{ABC,\epsilon}$, of the posterior distribution $\pi(\theta|y_{exp})$ through an iterative procedure through which, at each iteration, we draw a parameter vector θ from a prior, i.e. $\theta \sim \pi(\cdot)$, we simulate the model, and we keep the parameter vector if the corresponding simulation is close enough to the observations according to a threshold ϵ . These selected parameters are samples from $\pi_{ABC,\epsilon}$ and approximates the posterior distribution: the smaller the ϵ , the better the approximation. The chosen value of ϵ is crucial for the performance of ABC algorithm: a small ϵ is needed to achieve a good approximation, however this may result in high rejection rate leading to cumbersome computations. To overcome this issue, more elaborate algorithms were proposed, like ABC-SMC algorithms [9,20].

3 ABC-HASL method for tuning oscillators

We introduce an approach for exploring the parameters space of stochastic oscillators so that a given oscillatory criteria, e.g., the mean duration of the oscillation period, is met. The approach is based on the Automata-ABC procedure described in Section 3.3. The overall idea is to provide one with the characterisation of some linear hybrid automaton capable of assessing oscillation related measures and to plug in the Automata-ABC scheme so that it can effectively be applied to the analysis the effect the model's parameters have on the oscillations. We start off with an overview of preliminary notions necessary for understanding the functioning of the automata for oscillation related measures.

3.1 Characterising of noisy periodicity

The mathematical notion of periodic function (i.e. a function $f : \mathbb{R}^+ \rightarrow \mathbb{R}$ for which $\exists t_p \in \mathbb{R}^+$ such that $\forall t \in \mathbb{R}^+, f(t) = f(t + t_p)$, with t_p being the period) is of little use in the context of stochastic models as paths of a stochastic oscillator are noisy by nature (e.g. Figure 3-left) hence will have (unless in degenerative cases) zero probability of matching such strict notion of periodicity.

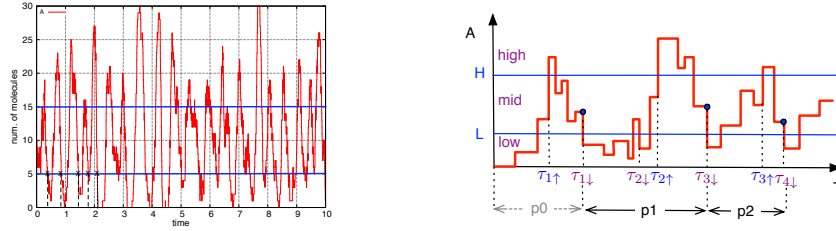


Fig. 3: A *noisy periodic* trajectory (left) and the corresponding *partition dependent* characterisation of noisy period realisations (right).

Therefore to take into account the noisy nature of stochastic oscillators we resort to a less strict notion of *noisy periodicity* [31].

Definition 1 (noisy periodic trajectory). For \mathcal{M}_θ an n -dimensional DESP population model let $L, H \in \mathbb{N}$, $L < H$, be two levels establishing the partition $\text{dom}_i(\mathcal{M}_\theta) = \text{low} \cup \text{mid} \cup \text{high}$ with $\text{low} = [0, L)$, $\text{mid} = [L, H)$ and $\text{high} = [H, \infty)$. A path $\sigma \in \text{Path}(\mathcal{M}_\theta)$ is said *noisy periodic w.r.t the i^{th} dimension*, and the considered L, H induced partition of $\text{dom}_i(\mathcal{M}_\theta)$ if the projection σ_i visits the intervals *low*, *mid* and *high* infinitely often.

H/L-crossing points. Given a noisy periodic trace σ_A we denote $\tau_{j\downarrow}$ ($\tau_{j\uparrow}$), the instant of time when σ_A enters for the j -th time the *low* (*high*) region. $T_{\downarrow} = \cup_j \tau_{j\downarrow}$ (resp. $T_{\uparrow} = \cup_j \tau_{j\uparrow}$) is the set of all *low-crossing points* (reps. *high-crossing points*). Observe that T_{\downarrow} and T_{\uparrow} reciprocally induce a partition on each other. Specifically $T_{\downarrow} = \cup_k T_{k\downarrow}$ where $T_{k\downarrow}$ is the subset of T_{\downarrow} containing the k -th sequence of contiguous *low-crossing points* not interleaved by any *high-crossing point*. Formally $T_{k\downarrow} = \{\tau_{i\downarrow}, \dots, \tau_{(i+h)\downarrow} \mid \exists k', \tau_{(i-1)\downarrow} < \tau_{k'\uparrow} < \tau_{i\downarrow}, \tau_{(i+h)\downarrow} < \tau_{(k'+1)\uparrow}\}$. Similarly T_{\uparrow} is partitioned $T_{\uparrow} = \cup_k T_{k\uparrow}$ where $T_{k\uparrow}$ is the subset of T_{\uparrow} containing the k -th sequence of contiguous *high-crossing points* not interleaved by any *low-crossing point*. For path σ_A in Figure 3 (right) we have that $T_{\downarrow} = T_{1\downarrow} \cup T_{2\downarrow} \cup T_{3\downarrow} \dots$ with $T_{1\downarrow} = \{\tau_{1\downarrow}, \tau_{2\downarrow}\}$, $T_{2\downarrow} = \{\tau_{3\downarrow}\}$, $T_{3\downarrow} = \{\tau_{4\downarrow}\}$, while $T_{\uparrow} = T_{1\uparrow} \cup T_{2\uparrow} \cup T_{3\uparrow} \dots$ with $T_{1\uparrow} = \{\tau_{1\uparrow}\}$, $T_{2\uparrow} = \{\tau_{2\uparrow}\}$, $T_{3\uparrow} = \{\tau_{3\uparrow}\}$. Based on H/L crossing points we formalise the notion of *period realisation* for a noisy period path.

Definition 2 (k^{th} noisy period realisation). For σ_A a noisy periodic trajectory with crossing point times $T_{\downarrow} = \cup_{k \geq 1} T_{k\downarrow}$, respectively $T_{\uparrow} = \cup_{k \geq 1} T_{k\uparrow}$, the realisation of the k^{th} noisy period, denoted t_{p_k} , is defined as $t_{p_k} = \min(T_{(k+1)\downarrow}) - \min(T_{k\downarrow})^2$.

Figure 3 (right) shows an example of *period realisations*: the first two period realisations, denoted $p1$ and $p2$, are delimited by the *mid-to-low* crossing points corresponding to the first entering of the *low* region which follows a previous sojourn in the *high* region and their duration (as per Definition 2) is $t_{p_1} = \tau_{3\downarrow} - \tau_{1\downarrow}$ respectively $t_{p_2} = \tau_{4\downarrow} - \tau_{3\downarrow}$. Notice that the time interval denoted as $p0$ does not represent a complete period realisation as there's no guarantee that $t = 0$ corresponds with the actual entering into the *low* region. Definition 2 correctly does not account for the first *spurious* period $p0$. Relying on the notion of period realisation we characterise the *period average* and *period variance* of a noisy periodic trace. Observe that the *period variance* allows us to analyse the regularity of the observed oscillator, that is, a “regular” (“irregular”) oscillator is one whose traces exhibit little (large) period variance.

Definition 3 (period average). For σ_A a noisy periodic trajectory the period average of the first $n \in \mathbb{N}$ period realisations, denoted $\bar{t}_p(n)$, is defined as $\bar{t}_p(n) = \frac{1}{n} \sum_{k=1}^n t_{p_k}$, where t_{p_k} is the k -th period realisation.

Observe that for a sustained oscillator, the average value of the noisy-period, in the long run, corresponds to the limit $\bar{t}_p = \lim_{n \rightarrow \infty} \bar{t}_p(n)$.

Definition 4 (period variance). For σ_A a noisy periodic trajectory the period variance of the first $n \in \mathbb{N}$ period realisations, denoted $s_{\bar{t}_p}^2(n)$, is defined as $s_{\bar{t}_p}^2(n) = \frac{1}{n-1} \sum_{k=1}^n (t_{p_k} - \bar{t}_p(n))^2$, where t_{p_k} is the k -th period realisation and $\bar{t}_p(n)$ is the period average for the first n period realisations.

Based on the period average and variance we now introduce a notion of distance of noisy periodic path from a target mean period value. We will employ such distance in the HASL-based adaptation of the ABC method for inferring the parameters of an oscillator.

Definition 5 (distance from target period). For σ_A a noisy periodic trajectory and $\bar{t}_p^{(obs)} \in \mathbb{R}_{>0}$ a target mean period duration we define the distance of σ_A from $\bar{t}_p^{(obs)}$ w.r.t. the first $n \in \mathbb{N}$ period realisations as

$$\text{dist}(\sigma_A, n, \bar{t}_p^{(obs)}) = \text{dist}(\bar{t}_p(n), s_{\bar{t}_p}^2(n), \bar{t}_p^{(obs)}) = \min\left(\frac{|\bar{t}_p(n) - \bar{t}_p^{(obs)}|}{\bar{t}_p^{(obs)}}, \frac{\sqrt{s_{\bar{t}_p}^2(n)}}{\bar{t}_p^{(obs)}}\right) \quad (4)$$

where $\bar{t}_p(n)$ ($s_{\bar{t}_p}^2(n)$) denotes the mean value (the variance) of the first n periods detected along σ_A (as per Def. 2 and Def. 4).

² t_{p_k} could alternatively be defined as $t_{p_k} = \min(T_{(k+1)\uparrow}) - \min(T_{k\uparrow})$, that is, w.r.t. crossing into the *high* region, rather than into the *low* region. It is straightforward to show that both definitions are semantically equivalent, i.e., the average value of t_{p_k} measured along a trace is equivalent with both definitions.

Notice that distance (4) establishes a form of multi-criteria selection of parameters as both the mean value and the variance of the detected periods are constrained. For example with a 10% tolerance (i.e. $\epsilon = 0.1$ in ABC terms) only the parameters $\theta \in \Theta$ that issue a relative error (w.r.t. the target mean period) not above 0.1 are selected.

3.2 An automaton for the distance from a target period

We introduce a LHA named $\mathcal{A}_{per}^{\bar{t}_p^{(obs)}}$ (Figure 4) for assessing the distance (as per Definition 5) between the mean period measured on paths of DESP oscillator \mathcal{M}_θ and a target period duration $\bar{t}_p^{(obs)}$. The automaton consists of three main

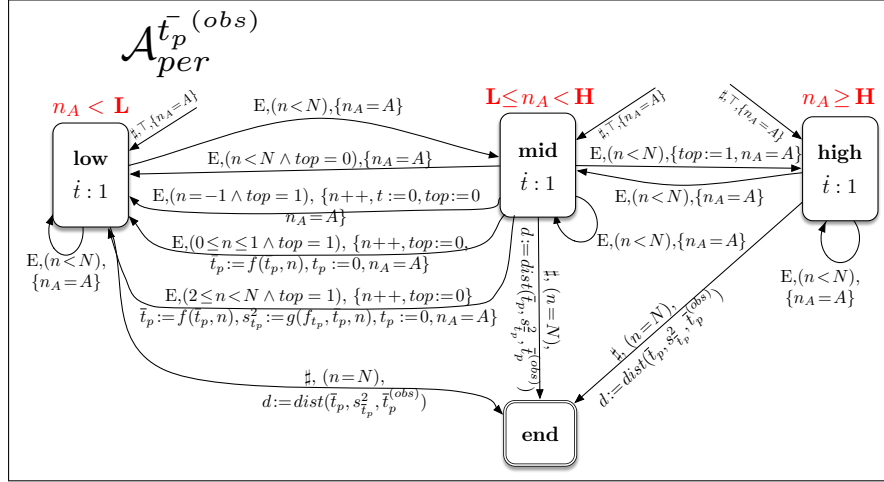


Fig. 4: \mathcal{A}_{per} : an LHA for selecting noisy periodic traces (with respect to an observed species A) related to partition $low = (-\infty, L]$, $mid = (L, H)$ and $high = [H, +\infty)$.

locations **low**, **mid** and **high** (corresponding to the regions of the partition of A 's domain induced by thresholds $L < H$). Its functioning is as follows: processing starts in either of the 3 state (which are all initial) depending on the initial state of the oscillator (population of oscillating species A is stored in n_A which is initialised through autonomous transitions before unfolding of path σ begins). Detection of one period realisation (as of Definition 2) correspond with the completion of a loop from **low** to **high** and back to **low** locations. The analysis of the simulated trajectory ends by entering location **end** as soon as the N -th period has been detected. Table 1 report about some of the variables (see [10] for the complete list) that $\mathcal{A}_{per}^{\bar{t}_p^{(obs)}}$ uses to store relevant statistics of the simulated path. Variable d is updated with the computed distance (as per Definition 4) on detection of the N -th period.

name	domain	update definition	description
n	\mathbb{N}	<i>increment</i>	counter of detected periods
t_p	$\mathbb{R}_{\geq 0}$	<i>reset</i>	duration last period
\bar{t}_p	$\mathbb{R}_{\geq 0}$	$f(\bar{t}_p, t_p, n) = \frac{t_p n \cdot n + t_p}{n+1}$	period mean
$s_{t_p}^2$	$\mathbb{R}_{\geq 0}$	$g(s_{t_p}^2, \bar{t}_p, t_p, n) = \frac{n-1}{n-2} \cdot s_{t_p}^2 + \frac{(t_p - \bar{t}_p)^2}{n-1}$	period variance
d_p	$\mathbb{R}_{\geq 0}$	$\min\left(\frac{\bar{t}_p - \bar{t}_p^{(obs)}}{\bar{t}_p^{(obs)}}, \frac{\sqrt{s_{t_p}^2}}{\bar{t}_p^{(obs)}}\right)$	distance from target period

Table 1: Variables of the $\mathcal{A}_{per}^{\bar{t}_p^{(obs)}}$ automaton.

3.3 HASL-ABC method for tuning oscillators

To calibrate the period of noisy oscillators we adapt the HASL-based version of the ABC method [12] so that it operates with \mathcal{A}_{per} automaton.

Algorithm 1: Rejection sampling HASL-ABC Algorithm

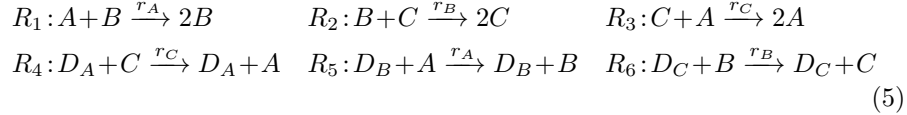
Require: $(\mathcal{M}_\theta)_{\theta \in \Theta}$ parametric DESP, π prior,
 N : number of particles, ϵ : tolerance level, \mathcal{A}_{per} distance period LHA
Ensure: $(\theta^{(i)})_{1 \leq i \leq N}$ drawn from π_{ABC}^ϵ
for $i = 1 : N$ **do**
 repeat
 $\theta' \sim \pi$
 $d' \sim (d_p, \mathcal{A}_{per}) \times \mathcal{M}_{\theta'}$
 until $d' \leq \epsilon$
 $\theta^{(i)} \leftarrow \theta'$
end for

Algorithm 1 describes the general HASL-based version of the ABC method. Further from the usual arguments of the ABC scheme (i.e. a parametric model \mathcal{M}_θ , a prior distribution over its parameters, the target number of particles N and the level of tolerance ϵ) it also takes as input an HASL automaton \mathcal{A} and an expression Y representing a distance measure computed over the paths of the product process $\mathcal{M}_\theta \times \mathcal{A}_{per}$. The output is the N values $\theta^{(i)}$ of accepted parameters which represent samples from the posterior distribution. By using automaton $\mathcal{A}_{per}^{\bar{t}_p^{(obs)}}$ and the distance from the target period $\bar{t}_p^{(obs)}$ as expression $Y \equiv last(d)$ (with d as in Table 1) the algorithm computes an estimation of the posterior w.r.t. to target period. In order to improve the convergence we also developed a sequential version of the HASL-based ABC (see [13]).

4 Case studies

We demonstrate the HASL-ABC procedure for tuning stochastic oscillator on two examples of stochastic oscillators.

A synthetic 3-ways oscillator. The CRN in (5) represents a model of synthetic sustained oscillator called *doping 3-way oscillator* [16].



It consists of 3 main species A , B and C forming a positive feedback loop (through reactions R_1, R_2, R_3) plus 3 corresponding invariant (*doping*) species D_A, D_B, D_C whose goal is to avoid extinction of the main species (through reactions R_4, R_5, R_6). It can be shown that the total population is invariant and that the model yields sustained noisy oscillation for the 3 main species, whose period and amplitude depend on the model’s parameters r_A, r_B and r_C (as well as on the total population). Figure 5 (left) depicts a simulated trajectory showing the oscillatory character of species A together with the period realisations induced by a given partition ($L = 300$ and $H = 360$). The corresponding location changes of the synchronised automaton \mathcal{A}_{per} as the trajectory is unfolded are depicted in different colors.

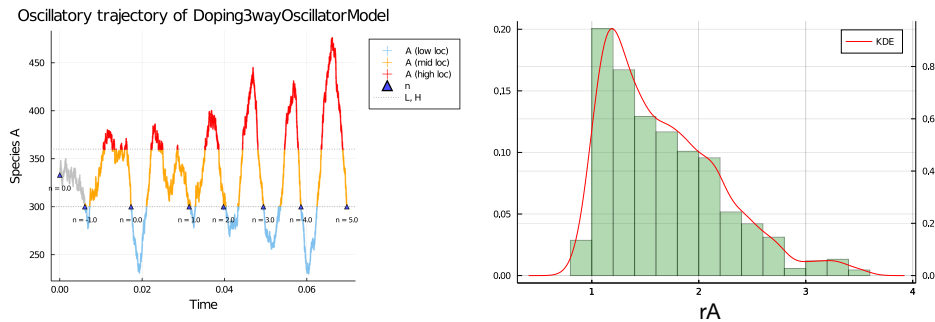


Fig. 5: The noisy periodic character of species A (left) and the posterior distribution for a single-parameter estimation experiment of the 3-ways oscillator.

Experiment 1. This is a 1-dimensional experiment in which we considered $\mathbf{s}_0 = (A_0, B_0, C_0, (D_A)_0, (D_B)_0, (D_C)_0) = (333, 333, 333, 10, 10, 10)$ as initial state, we fixed the rate constants $r_B = r_C = 1.0$, and estimated the posterior distribution for r_A considering a uniform $\mathcal{U}(0, 10)$ prior and a target mean period $\bar{t}_p^{(obs)} = 0.01$. For the automaton \mathcal{A}_{per} the noisy-period dependent partition we considered is $L = 300$ and $H = 360$ while for each trajectory we observed $N = 4$ periods. For the ABC algorithm we used $N = 1000$ particles and considered a 20% tolerance ($\epsilon = 0.2$). Figure 5 (right) shows the resulting automaton-ABC posterior (histogram and Kernel density estimation). We observe that 1) the posterior support being included in the prior’s $[0.0, 4.0] \subset [0, 10.0]$, we have reduced the parameter space to a subset where it is probable to obtain trajectories

with a mean period of 0.01 (relative to a 20% tolerance) and 2) the posterior has only one mode, which is quite sensible, as having fixed r_B and r_C , one would expect the mean period duration being directly linked to the kinetics of reaction R_1 , which is only parametrised by r_A . **Experiment 2.** This is a 3-dimensional version of the previous experiment in which we considered the same uniform prior for the 3 parameters $r_A, r_B, r_C \sim \mathcal{U}(0, 10)$. Figure 6 (left) shows the correlation plot matrix of the resulting automaton-ABC posterior, where each one-dimensional histogram on the diagonal of plot matrix represents the marginal distribution of each parameter, whereas the two-dimensional marginal distributions are given in the upper triangular part of the matrix (e.g., plot on position (1,3) refers to the marginal distribution $p(r_A, r_C | \bar{t}_p^{(obs)})$) and the scatter plots of the two-dimensional marginal distributions are in the lower triangular matrix. Based on the diagonal plots we observe that most of the parameters that results in a period close to the target one ($\bar{t}_p^{(obs)} = 20$) are within the support $[0, 4] \times [0, 3] \times [0, 4]$ and, furthermore the particles form a 3D parabolic shape since the three two-dimensional projections have a parabolic shape, according to plots in the upper triangular part of Figure 6. Also, one can notice that for each two-dimensional histogram, the area near the point (1,1) is a high probability area, which is consistent with the previous one-dimensional experiment.

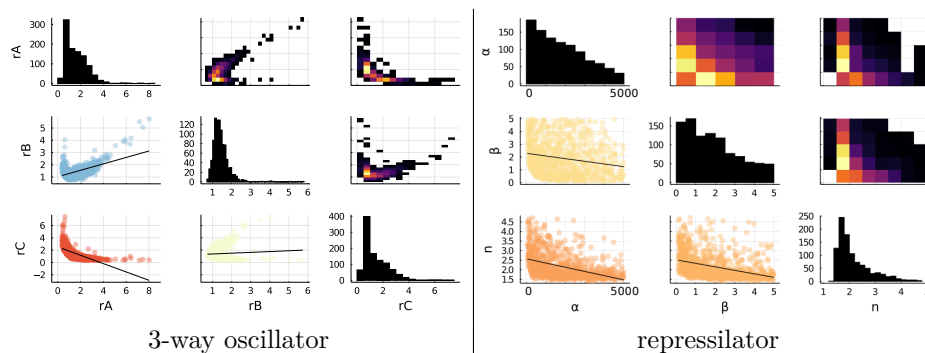
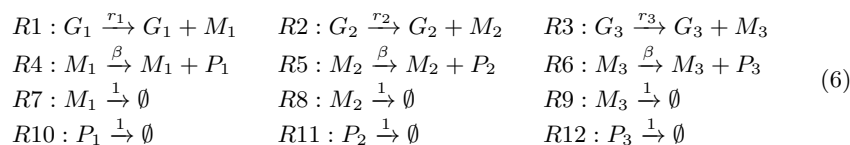


Fig. 6: Correlation plots of posterior distribution obtained through *Aper*-ABC for the 3D experiment of 3-way oscillator (left) and repressilator (right).

Repressilator. We consider an infinite-state model (6) of a synthetic genetic network known as Repressilator developed to reproduce oscillatory behaviours within a cell [21]. It consists of 3 proteins P_1, P_2, P_3 forming a negative feedback loop, with P_1 repressing P_2 's transcription gene G_2 , P_2 repressing P_3 's transcription and so on.



Following [21] we assumed mass-action dynamics with a common rate constant β for translation reactions (R_4, R_5, R_6) and 1 for species degradation (R_7 to R_{12}). Conversely transcription reactions (R_1, R_2, R_3) are assumed to follow a Hill function dynamics given by the follow parameter definitions $r_1 = \frac{\alpha}{1+[P_3]^n} + \alpha_0$, $r_2 = \frac{\alpha}{1+[P_1]^n} + \alpha_0$ and $r_3 = \frac{\alpha}{1+[P_2]^n} + \alpha_0$, where n is the Hill coefficient, α is related to transcription growth and α_0 is the parameter related to the minimum level of transcription growth. The parameter space is 4-dimensional with $\theta = (\alpha, \beta, n, \alpha_0) \in \mathbb{R}^4$. Each parameter affects the resulting oscillation

Experiment 1. This is a 3-dimensional experiment in which we considered $\mathbf{s}_0 = ((M_1)_0, (M_2)_0, (M_3)_0, (P_1)_0, (P_2)_0, (P_3)_0) = (0, 0, 0, 5, 0, 15)$ as initial state, we fixed $\alpha_0 = 0$, and estimated the posterior distribution for the remaining parameters considering the following priors: $\alpha \sim \mathcal{U}(50, 5000)$, $\beta \sim \mathcal{U}(0.1, 5.0)$, $n \sim \mathcal{U}(0.5, 5.0)$. The target mean period was $\bar{t}_p^{(obs)} = 20$, while the noisy-period dependent partition was set at $L = 50$ and $H = 200$. For the ABC algorithm we used $N = 1000$ particles and considered a 10% tolerance ($\epsilon = 0.1$). Figure 6 (right) shows the correlation plot of the resulting automaton-ABC posterior. We observe that with this setting the Repressilator oscillations are most sensitive to parameter n as its marginal posterior is much narrower than that of α and β (i.e. varying n induces more instability than α and β). More experiments, including a 4 dimensional one, are illustrated in [10]

5 Conclusion

We introduced a methodology that given a parametric, discrete-state stochastic oscillator model, allows for inferring regions of the parameter space that exhibit a positive probability to match a desired mean oscillation period. Such framework relies on a formal characterisation of noisy periodicity which is assessed through a meter encoded by a hybrid automaton. Parameter inference is then obtained by plugging of such an automaton-meter within a ABC scheme. The added value of such a rather cumbersome formalism is in terms of automation, generality and separation of concerns. The period meter automaton, being completely configurable, can straightforwardly be generated automatically, therefore avoiding an annoying overhead to the end user. This combined with the fact that the framework inherently takes care of synchronising the model with the automaton results in a highly configurable and generic approach, one in which different oscillation tuning criteria can easily be taken into account as long as they can be encoded into a corresponding meter automaton. Notice that alternative approaches that are not based on formal methods, such as e.g., those based on auto-correlation analysis, although effective, are not easily adaptable as they require the implementation of a customised, hard-coded procedure where periodicity indicators are obtained by offline analysis of trajectories sampled from the model.

Future developments include the integration of oscillation-amplitude amongst the criteria for tuning oscillators through a *peak detector* automaton [6].

References

1. Alexandr Andreychenko, Thilo Krüger, and David Spieler. Analyzing oscillatory behavior with formal methods. In Anne Remke and Mariëlle Stoelinga, editors, *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems - International Autumn School, ROCKS 2012, Vahrn, Italy, October 22-26, 2012, Advanced Lectures*, volume 8453 of *Lecture Notes in Computer Science*, pages 1–25. Springer, 2012.
2. Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. Verifying continuous time Markov chains. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification*, pages 269–276, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
3. Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-Checking Algorithms for Continuous-Time Markov Chains. *Software Engineering, IEEE Transactions on*, 29:524– 541, 07 2003.
4. 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. *Performance Evaluation*, 90:53 – 77, 2015.
5. P. Ballarini and M.L. Guerriero. Query-based verification of qualitative trends and oscillations in biochemical systems. *Theoretical Computer Science*, 411(20):2019 – 2036, 2010.
6. Paolo Ballarini. Analysing oscillatory trends of discrete-state stochastic processes through hasl statistical model checking. *International Journal on Software Tools for Technology Transfer*, pages 1–22, 2015.
7. Paolo Ballarini and Marie DufLOT. Applications of an expressive statistical model checking approach to the analysis of genetic circuits. *Theor. Comput. Sci.*, 599:4–33, 2015.
8. Paolo Ballarini, Radu Mardare, and Ivan Mura. Analysing Biochemical Oscillation through Probabilistic Model Checking. *Electronic Notes in Theoretical Computer Science*, 2009.
9. Mark A. Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P. Robert. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.
10. Mahmoud Bentrîou. *Statistical inference and verification of Chemical Reaction Networks*. PhD thesis, École doctorale Interfaces, University Paris Saclay, 2021.
11. Mahmoud Bentrîou, Paolo Ballarini, and Paul-Henry Cournède. Reachability design through approximate bayesian computation. In *International Conference on Computational Methods in Systems Biology*, pages 207–223. Springer, 2019.
12. Mahmoud Bentrîou, Paolo Ballarini, and Paul-Henry Cournède. Automaton-abc: A statistical method to estimate the probability of spatio-temporal properties for parametric markov population models. *Theor. Comput. Sci.*, 893:191–219, 2021.
13. Mahmoud Bentrîou, Stefanella Boatto, Gautier Viaud, Catherine Bonnet, and Paul-Henry Cournède. Assimilation de données par filtrage particulaire régularisé dans un modèle d’épidémiologie. pages 1–6, 2008.
14. Luca Bortolussi, Dimitrios Milios, and Guido Sanguinetti. Smoothed model checking for uncertain Continuous-Time Markov Chains. *Inf. Comput.*, 247:235–253, 2016.
15. Luboš Brim, Milan Češka, Sven Dražan, and David Šafránek. Exploring parameter space of stochastic biochemical systems using quantitative model checking. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification*, pages 107–123, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

16. Luca Cardelli. *Artificial Biochemistry*, pages 429–462. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
17. Milan Češka, Frits Dannenberg, Marta Kwiatkowska, and Nicola Paoletti. Precise parameter synthesis for stochastic biochemical systems. In Pedro Mendes, Joseph O. Dada, and Kieran Smallbone, editors, *Computational Methods in Systems Biology*, pages 86–98, Cham, 2014. Springer International Publishing.
18. Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, François Fages, and Vincent Schächter. Modeling and querying biomolecular interaction networks. *Theor. Comput. Sci.*, 325:25–44, 2004.
19. Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. In John R. Wright, Larry Landweber, Alan J. Demers, and Tim Teitelbaum, editors, *Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 1983*, pages 117–126. ACM Press, 1983.
20. Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.
21. M. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(335), 2000.
22. Albert Goldbeter. Computational approaches to cellular rhythms. *Nature*, 420:238–245, 2002.
23. T. Han, J. Katoen, and A. Mereacre. Approximate parameter synthesis for probabilistic time-bounded reachability. In *2008 Real-Time Systems Symposium*, pages 173–182, 2008.
24. Thomas PJ Lindner B. MacLaurin J, Fellous JM. Stochastic oscillators in biology: introduction to the special issue. *Biol Cybern.*, (116(2)):119–120., 2022.
25. Jean-Michel Marin, Pierre Pudlo, Christian P. Robert, and Robin J. Ryder. Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, Nov 2012.
26. Gareth W. Molyneux and Alessandro Abate. $\text{Abc}(\text{smc})^2$: Simultaneous inference and model checking of chemical reaction networks. In Alessandro Abate, Tatjana Petrov, and Verena Wolf, editors, *Computational Methods in Systems Biology - 18th International Conference, CMSB 2020, Konstanz, Germany, September 23-25, 2020, Proceedings*, volume 12314 of *Lecture Notes in Computer Science*, pages 255–279. Springer, 2020.
27. Gareth W. Molyneux, Viraj B. Wijesuriya, and Alessandro Abate. Bayesian verification of chemical reaction networks, 2020.
28. Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57. IEEE Computer Society, 1977.
29. Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. Chapman and Hall/CRC, 2018.
30. J. Sneyd, K. Tsaneva-Atanasova, V. Reznikov, Y. Bai, M. J. Sanderson, and D. I. Yule. A method for determining the dependence of calcium oscillations on inositol trisphosphate oscillations. *Proceedings of the National Academy of Sciences*, 103(6):1675–1680, 2006.
31. D. Spieler. Characterizing oscillatory and noisy periodic behavior in markov population models. In *Proc. QEST'13*, 2013.