System Design of Stochastic Models using Robustness of Temporal Properties

Ezio Bartocci[a], Luca Bortolussi[c,b,d], Laura Nenzi[e], Guido Sanguinetti[f,g]

[a] *Vienna University of Technology, Austria*
[b] *Saarland University, Germany*
[c] *DMG, University of Trieste, Italy*
[d] *CNR/ISTI, Pisa, Italy*
[e] *IMT, Lucca, Italy*
[f] *School of Informatics, University of Edinburgh, UK*
[g] *SynthSys, Centre for Synthetic and Systems Biology, University of Edinburgh. UK*

Abstract

Stochastic models such as Continuous-Time Markov Chains (CTMC) and Stochastic Hybrid Automata (SHA) are powerful formalisms to model and to reason about the dynamics of biological systems, due to their ability to capture the stochasticity inherent in biological processes. A classical question in formal modelling with clear relevance to biological modelling is the model checking problem, i.e. calculate the probability that a behaviour, expressed for instance in terms of a certain temporal logic formula, may occur in a given stochastic process. However, one may not only be interested in the notion of satisfiability, but also in the capacity of a system to maintain a particular emergent behaviour unaffected by the perturbations, caused e.g. from extrinsic noise, or by possible small changes in the model parameters. To address this issue, researchers from the verification community have recently proposed several notions of *robustness* for temporal logic providing suitable definitions of distance between a trajectory of a (deterministic) dynamical system and the boundaries of the set of trajectories satisfying the property of interest. The contributions of this paper are twofold. First, we extend the notion of robustness to stochastic systems, showing that this naturally leads to a distribution of robustness degrees. By discussing three examples, we show how to approximate the distribution of the robustness degree and the *average robustness*. Secondly, we show how to exploit this notion to address the *system design problem*, where the goal is to optimise some control parameters of a stochastic model in order to maximise robustness of the desired specifications.

1. Introduction

Biological systems at the single cell level are inherently stochastic. Molecules inside cells perform random movements (*random walk*) and the reactions among

them may occur when the probability of collision is high enough. The number of molecules of each species at each time point is therefore a random process: assuming instantaneous reactions, this process can be modelled as a Markovian (i.e. memoryless) discrete state, continuous time process. When the number of molecules of each species involved is large, so that many reactions happen in any small interval of time, stochastic effects can be neglected. However, if the concentration of the molecules (of at least some of the species) is low the stochasticity plays an important role and must be taken into account [1]. Stochastic models such as Continuous-Time Markov Chains (CTMC) [2] and Stochastic Hybrid Automata [3] are particularly powerful and suitable formalisms to model and to reason about biological systems defined as stochastic systems over time.

A classical question in formal modelling is to calculate the probability that a behavioural property, expressed in temporal logic, may occur in a given stochastic process, with specified parameters. *Probabilistic Model Checking* [4, 5] (PMC) is a well-established verification technique that provides a quantitative answer to such a question. The algorithm used to calculate this probability [6] produces the exact solution, as it operates directly on the structure of the Markov chain. Despite the success and the importance of PMC, this technique suffers some computational limitations, either due to state space explosion or to the difficulty (impossibility) in checking analytically formulae in specific logics, like Metric Temporal Logic (MTL) [5, 7]. Furthermore, PMC provides only a quantitative measure of the *satisfiability* (yes/no answer) of a temporal logic specification (i.e. the probability of the property being true).

However, especially when we deal with stochastic models, the notion of satisfiability may be not enough to determine the capacity of a system to maintain a particular emergent behaviour unaffected by the uncertainty of the perturbations due to its stochastic nature or by possible small changes in the model parameters. A similar issue also arises when considering the satisfiability of a property by deterministic dynamical systems which may be subject to extrinsic noise or uncertainty in the parameter. To address this question in the deterministic case, researchers from the verification community have proposed several notions of *temporal logic based robustness* [8, 9, 10], providing suitable definitions of distance between a trajectory of a system and the behavioural property of interest, expressed in terms of a temporal logic formula. These effectively endow the logic of interest with quantitative semantics, allowing us to capture not only whether a property is satisfied but also *how much* it is satisfied. A similar notion of robustness for stochastic models would clearly be desirable but, to our knowledge, has not been formalised yet.

The contributions of this paper are twofold. First, we provide a simulation-based method to define a notion of robust satisfiability in stochastic models. Simulation-based approaches, such as statistical model checking [11], can be used to estimate for a stochastic model the *robust satisfiability distribution* for a given temporal logic formula, with a guarantee of asymptotic correctness. This distribution is the key to understand how the behaviour specified by the logic temporal formula is affected by the stochasticity of the system. In particular, in this paper we consider two indicators of this distribution: the *average*

robustness and the *conditional average robustness* of a formula being true or false. We discuss how to compute the robust satisfiability distribution and its indicators on two biological examples. Secondly, we show how to exploit the average robustness to address the *system design problem*, where the goal is to optimise (few) control parameters of a stochastic model in order to maximise these indicators. The proposed approach takes advantage of Gaussian Process Upper Confidence Bound (GP-UCB) algorithm introduced in [12].

The paper is structured as follows: in Section 2 we introduce the background material. In Section 3 we discuss the robustness of stochastic models using the quantitative semantics of the Signal Temporal Logic (STL). In Section 4 we present some experimental results for the robustness of STL formulae for three stochastic models that we have chosen as our case studies: the Schlögl system, a Repressilator-like model of the circadian clock of *Ostreococcus Tauri*, and a Feed-Forward motif of a gene regulatory network. In Section 5 we show an application of the robust semantics to the system design problem. The related works and the final discussion are in Section 7.

2. Background

We now introduce some material needed in the rest of the paper. We start from a general definition of stochastic process, in order to fix the notation and introduce the space of trajectories of the system, which plays a central role in our approach. Then we instantiate this general framework in different ways, presenting Population Continuous Time Markov Chains, Stochastic Differential Equations and Stochastic Hybrid Systems, describing as these particular semantics may arise as models of biological systems. We then briefly introduce the Skorokhod metric, which endows the space of trajectories with a natural topology. At the end of the section, we introduce the Signal Temporal Logic, including its boolean and quantitative semantics.

2.1. Stochastic Processes

Stochastic processes are useful mathematical constructs to describe the random evolution of a system in time. The following definition formalises the intuitive concept of random evolution.

Definition 1. *Let $(\Omega, \mathcal{A}, P)$ be a probability space*[1] *and $E \subseteq \mathbb{R}^n$. A continuous time stochastic process with values in E is a collection of E-valued random variables $\mathbf{X}(t)$, indexed by $t \in [0, \infty)$ and defined on the same probability space $(\Omega, \mathcal{A}, P)$.*

In this paper, we will restrict our attention to the case where the sample space E is a subset of the m dimensional Euclidean space $\mathbb{R}^m$. We will also in

[1] Here Ω is the sample space, $\mathcal{A}$ is a sigma-algebra, and P a probability measure. See [13] for an introduction of measure and probability theory.

general assume that the process is *Markovian*, i.e. memoryless, although this assumption could be weakened to include processes for which efficient sampling algorithms exist (e.g. semi-Markov processes).

We will be particularly concerned with the trajectory-based view of stochastic processes, restricting to those processes whose trajectories are *cadlag* functions. A *cadlag function* $f : [0, \infty) \to E$ is a right continuous function having left limits for any $t \in [0, \infty)$, i.e. $f(t) = f(t^+)$ and $f(t^-)$ exists for all t. Call $\mathcal{D}([0, \infty), E)$ the space of cadlag functions with values in E. Stochastic processes considered in this paper, including Continuous-Time Markov Chains, solutions of Stochastic Differential Equations, Stochastic Hybrid Systems, can be seen as random variables on the space $\mathcal{D}([0, \infty), E)$.

In the rest of the paper, we will adopt the following *notational conventions*: by $\mathbf{x}$ we denote an element of $\mathcal{D}([0, \infty), E)$, with $\mathbf{x}(t)$ representing the value of the cadlag function at time t. Stochastic processes, when seen as a random variable over $\mathcal{D}([0, \infty), E)$, are denoted by $\mathbf{X}$, while $\mathbf{X}(t)$ denotes the E-valued random variable at time t. Points of E are instead indicated by $\mathbf{d}, \mathbf{d}'$.

2.2. Markov Population Models

Biochemical and genetic networks can be represented as populations of interacting molecules. As the interactions are the results of random collisions between particles, the evolution of the number of particles of each time will be a random variable. Similar reasoning can be applied in other scientific domains, ranging from population dynamics in ecology to epidemiology. We can define a class of models, called *population processes*, which intuitively describe a system in which agents or objects of different kinds (e.g. molecules, individuals in a social network), and possibly with different internal states (e.g. being phosphorylated, being infected or not) interact together. In the simplest case, population processes can be described by Continuous Time Markov Chains (CTMC) [2], one of the simplest classes of stochastic processes. These CTMC can be described by a simple formalism, known as Population CTMC (PCTMC) or Markov Population Model [14], inspired by biochemical reaction networks.

A PCTMC is a tuple $\mathcal{X} = (\mathbf{X}, \mathcal{E}, \mathcal{R})$. The state of the system is described by a vector $\mathbf{X} = (X_1, \ldots, X_n)$ of n integer-valued random variables X_i, each counting the number of entities of a given class or species, with domain $\mathcal{E} \subseteq \mathbb{N}^n$. The dynamics of the system, instead, is specified by a set of m *reactions* or *transition classes* $\mathcal{R} = \{\eta_1, \ldots, \eta_m\}$, which can be seen as description of events changing the state of the system. Each reaction η_l is a rule of the form

$$r_1 X_{i_1} + \ldots + r_k X_{i_k} \to s_1 X_{j_1} + \ldots + s_h X_{j_h},$$

where X_{i_a} is a reactant and X_{j_b} is a product (by abusing the notation, we use the same letter for species and variables of $\mathbf{X}$), and r_i, s_j are the stoichiometric coefficients, i.e. the amount of agents/entities consumed or produced by the reaction. Stoichiometric information of a reaction η_l can be condensed into an update vector $\mathbf{v}_l$, giving the net change in population variables due to η_l: $\mathbf{v}_l = \sum_{b \leq h} s_b \mathbf{1}_{j_b} - \sum_{a \leq k} r_a \mathbf{1}_{i_a}$, where $\mathbf{1}_j$ equals one in position j and zero

elsewhere. Additionally, each reaction η_l has an associated rate function $f_l(\mathbf{X})$ giving the rate of the transition as a function of the global state of the system. Formally, reactions η_l are tuples $\eta_l = (\mathbf{v}_l, f_l(\mathbf{X}))$.

From a set of reactions $\mathcal{R}$ and species $\mathbf{X}$, we can easily derive the formal representation of a CTMC in terms of its infinitesimal generator matrix, see for instance [14]. The state space of the CTMC is $\mathcal{E}$ (i.e. $\mathbb{N}^n$ or a proper subset, if any conservation law is in force). The infinitesimal generator matrix Q, instead, is defined by $Q_{\mathbf{d},\mathbf{d}'} = \sum_{\eta_l \in \mathcal{R} \mid \mathbf{d}+\mathbf{v}_l=\mathbf{d}'} f_l(\mathbf{d})$, for any $\mathbf{d} \neq \mathbf{d}'$, $\mathbf{d}, \mathbf{d}' \in \mathcal{E}$. Such CTMC can be simulated with standard algorithms, like SSA [15].

2.3. Fluid Approximation

In many systems, stochasticity is attenuated in particular regimes. For example, the evolution of a biochemical reaction network becomes essentially deterministic when the number of particles in each species is large, so that the observed behaviour is effectively the average of many microscopic events. In these cases, it is often useful to approximate the system behaviour as deterministic. From a Markov population model, we can easily construct an alternative semantics in terms of Ordinary Differential Equations (ODE), assuming variables $\mathbf{X}$ to be continuous and interpreting each rate as a flow, thus obtaining the vector field

$$F(\mathbf{X}) = \sum_{\eta_l \in \mathcal{R}} \mathbf{v}_l f_l(\mathbf{X}),$$

defining the ODE $d\mathbf{X}/dt = F(\mathbf{X})$.[2] This equation, known as *fluid approximation*, can be shown to be a first order approximation of the CTMC average dynamics. It is possible to prove that, with a suitable rescaling of the variables (dividing by the system size, which for biochemical reactions is just the volume), the dynamics of the CTMC for large populations converge to the solution of this ODE (see [14] for a detailed discussion of such limit results).

2.4. Stochastic Hybrid Automata

Fluid approximations are only justified when *all* chemical species in a biochemical network are present at high concentrations. This is often problematic when modelling genetic networks in a single cell: genes are normally present at very low copy numbers in cells, and their state can be usefully described as small finite state machines [17], i.e. as entities with a small number of internal states (e.g. free state, bound by a repressor molecule, etc). On the other hand, gene products (mRNAs and proteins) can have very high counts, so that modelling the genetic network as a PCTMC may incur significant computational costs. In these cases, a better strategy is to approximate only some variables as continuous, keeping discrete the others. This reflects in the dynamics: some reactions will be converted into flows (generally those modifying only continuous

[2] The use of $\mathbf{X}$ to denote a deterministic process is justifiable if we see the solution of the ODE as a degenerate random process, having delta-Dirac distributions at each time point, see for instance [16].

variables), while the others will remain stochastic discrete events. This gives rise to a model that can be expressed in terms of a class of Stochastic Hybrid Automata (SHA, [17]) known as Piecewise-Deterministic Markov Processes [16].

More specifically, the SHA so obtained have discrete modes identified by the value of discrete variables. In between discrete transitions, the system evolves following the solution of the differential equation, whose vector field is mode-dependent (via the value of discrete variables). Discrete jumps happen at exponentially distributed random times, at a non-constant rate that can depend on the continuous variables. After each jump, the value of the discrete variables can change. Also continuous variables can be updated, even if we do not consider this possibility in this paper, see [17] for further details. Similarly to the fluid approximation case, we can see SHA models as the limit of CTMC, taking to the limit only the populations corresponding to continuous variables (under a suitable scaling of rates, see [18] for further details).

SHA can also be defined by assuming a stochastic continuous dynamics within each mode [19, 20]. In this case, the system evolves in mode q by following a trajectory which is a solution of a stochastic differential equation of the form $d\mathbf{X} = F_q(\mathbf{X})dt + \sigma(\mathbf{X})d\mathbf{W}$, where $\sigma(\mathbf{X})$ is a mode and state dependent Lipschitz continuous $n \times r$ *diffusion matrix*, $F_q(\mathbf{X})$ is the mode dependent *drift*, and $d\mathbf{W}$ is an $r \times 1$ vector of uncorrelated Wiener processes (white noise). Solutions of such a SDE can be simulated using the standard Euler-Maruyama algorithm [21].

2.5. Topology of the space of trajectories

The space $\mathcal{D}([0,\infty), E)$ can be given the structure of a metric space by the *Skorokhod metric*. The Skorokhod metric is first defined on compact time intervals $[0, T]$ and then extended over the whole positive time axis $[0, \infty)$. Consider the uniform metric on the space $\mathcal{D}([0, T], E)$, i.e. $d_U(\mathbf{x}', \mathbf{x}) = \sup_{0\leq t\leq T} \|\mathbf{x}'(t) - \mathbf{x}(t)\|$. The uniform metric endows the space of cadlag functions with a topology, however it is easy to see that this is too restrictive for most purposes. Consider, for example, the cadlag function $\mathbf{x}$ defined to be 0 for $t < t_0$, $t_0 > 0$ and 1 for $t \geq t_0$, and the sequence of cadlag functions $\mathbf{x}_n$ defined to be 0 for $t < t_0 + \frac{1}{n}$ and 1 otherwise. Then, for every $t \neq 0$, it is possible to find an N s.t. $\mathbf{x}(t) = \mathbf{x}_n(t)$ $\forall n > N$, i.e. the sequence converges point-wise almost everywhere. However, under the uniform topology $d_U(\mathbf{x}, \mathbf{x}_n) = 1\ \forall n$, so the sequence does not converge as a sequence of functions. This is a general fact: if we have a sequence $\mathbf{x}_n$ of cadlag functions, then they will converge to $\mathbf{x}$ in the uniform norm if and only if the discontinuous jumps of $\mathbf{x}_n$ happen precisely at the same times as those of $\mathbf{x}$ (for $n \geq n_0$).

The idea behind the Skorokhod metric is to relax the uniform metric by allowing a small difference in these jump times by resynchronising them. Informally, if the uniform metric allows one to wiggle space a bit, the Skorokhod metric allows us also to wiggle time. To formalise this statement, let $\omega(t) : [0, T] \to [0, T]$ be a time-wiggle function, i.e. a strictly increasing continuous function. Call $\mathcal{I}_T$ the set of such functions. Then, the Skorokhod distance

between $\mathbf{x}, \mathbf{y} \in \mathcal{D}([0,T], E)$ is

$$d_T(\mathbf{x}, \mathbf{y}) = \inf_{\omega \in \mathcal{I}_T} \max\{ \sup_{t \in [0,T]} \|\omega(t) - t\|, \sup_{t \in [0,T]} \|\mathbf{x}(t) - \mathbf{y}(\omega(t))\| \}. \tag{2.1}$$

In our example above, one can simply choose the time wiggle function $\omega(t) = t + \frac{1}{n}$, so that the second term in the r.h.s. of equation (2.1) is always zero. The Skorokhod distance therefore evaluates to $\frac{1}{n}$, and the sequence is seen to converge. The metric d_T is extended to a metric on $\mathcal{D}([0,\infty), E)$ by discounting large times as follows:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{K \in \mathbb{N}} 2^{-K} \min\{1, d_K(\mathbf{x}, \mathbf{y})\}.$$

The Skorokhod metric defines a topology for which $\mathcal{D}([0,\infty), E)$ is complete and separable, i.e. it is a Polish space[3]. See [22] for a detailed introduction to the metric and its properties.

2.6. Signal Temporal Logic

Temporal logic [23] provides a very elegant framework to specify in a compact and formal way emergent behaviours in terms of *time-dependent* events. Among the myriads of temporal logic extensions available, Signal Temporal Logic [24] (STL) is very suitable to characterise behavioural patterns in real-valued time series generated during the simulation of a dynamical system. STL extends the dense-time semantics of Metric Interval Temporal Logic [25] (MITL), with a set of parametric numerical predicates playing the role of atomic propositions. STL provides two different semantics: a boolean semantics that returns yes/no depending if the observed trace satisfies or not the STL specification and a quantitative semantics that returns also a measure of the specification robustness for a given trace. Recently, Donzé et al. [26] proposed a very efficient monitoring algorithm for STL robustness, now implemented in the Breach [27] tool. The use of the sensitivity-based analysis guided by the measure of the robustness has been successfully applied in several domains ranging from analog circuits [28] to systems biology [29, 30], to study the parameter space and also to refine the uncertainty of the parameter sets. In the following, we recall the syntax and the quantitative semantics of STL [8] that will be used in the rest of the paper.

Definition 2 (STL syntax). *The syntax of the STL is given by*

$$\varphi := \top \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathbf{U}_{[a,b]} \, \varphi_2,$$

where $\top$ is a true formula, conjunction and negation are the standard boolean connectives, $[a,b]$ is a dense-time interval with $a < b$ and $\mathbf{U}_{[a,b]}$ is the until operator. The atomic predicate $\mu : \mathbb{R}^n \to \mathbb{B}$ is defined as $\mu(\mathbf{d}) := (y(\mathbf{d}) \geqslant 0)$, $\mathbf{d} \in$

[3] In fact, completeness requires one to work with an equivalent metric, but this is not relevant for the rest of the paper [22].

$\mathbb{R}^n$, *where* $y : \mathbb{R}^n \to \mathbb{R}$ *is sufficiently smooth real-valued function. The predicate* μ *can be lifted to an operation between signals*[4]*, transforming a real valued signal into a boolean one, i.e. to a mapping* $\mu : \mathcal{D}([0,\infty), \mathbb{R}^n) \to \mathcal{D}([0,\infty), \mathbb{B})$*, by* $\mu(\mathbf{x})(t) = (y(\mathbf{x}(t)) \geqslant 0)$*. In this context,* $\mu(\mathbf{x})(t)$ *is known as a* boolean signal*,* $\mathbf{x}(t) = (x_1(t), ..., x_n(t))$ *as the* primary signal*, and* $y(\mathbf{x}(t))$ *as the* secondary signal.

The (bounded) until operator $\varphi_1 \mathbf{U}_{[a,b]} \varphi_2$ requires φ_1 to hold from now until, in a time between a and b time units, φ_2 becomes true. The *eventually* operator $\mathbf{F}_{[a,b]}$ and the *always* operator $\mathbf{G}_{[a,b]}$ can be defined as usual: $\mathbf{F}_{[a,b]}\varphi := \top \mathbf{U}_{[a,b)}\varphi$, $\mathbf{G}_{[a,b]}\varphi := \neg \mathbf{F}_{[a,b]} \neg \varphi$. We introduce now the boolean and the quantitative semantics for STL as in [24, 8].

Definition 3 (STL Boolean Semantics). *The boolean satisfaction relation* $\models$ *for an STL formula* φ *on a temporal trace* $\mathbf{x}$ *is defined recursively by:*

$$
\begin{array}{lcl}
(\mathbf{x}, t) \models \top & & \\
(\mathbf{x}, t) \models \mu & \Leftrightarrow & \mu(\mathbf{x}(t)) \textit{ is true} \\
(\mathbf{x}, t) \models \neg\varphi & \Leftrightarrow & (\mathbf{x}, t) \not\models \varphi \\
(\mathbf{x}, t) \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & (\mathbf{x}, t) \models \varphi_1 \textit{ and } (\mathbf{x}, t) \models \varphi_2 \\
(\mathbf{x}, t) \models \varphi_1 \, \mathbf{U}_{[a,b]} \varphi_2 & \Leftrightarrow & \exists t' \in [t+a, t+b] \textit{ s.t. } (\mathbf{x}, t') \models \varphi_2 \textit{ and } \forall t'' \in [t, t'), (\mathbf{x}, t'') \models \varphi_1
\end{array}
$$

A trace $\mathbf{x}$ *satisfies* φ*, denoted by* $\mathbf{x} \models \varphi$*, if and only if* $(\mathbf{x}, 0) \models \varphi$*.*

Definition 4 (STL Quantitative Semantics). *The* quantitative satisfaction function ρ *returns a value* $\rho(\varphi, \mathbf{x}, t) \in \tilde{\mathbb{R}}$[5] *quantifying the robustness degree (or satisfaction degree) of the property* φ *by the signal* $\mathbf{x}$ *at time* t *with respect to perturbations. It is defined recursively as follows:*

$$
\begin{array}{lcl}
\rho(\top, \mathbf{x}, t) & = & +\infty \\
\rho(\mu, \mathbf{x}, t) & = & y(\mathbf{x}(t)) \qquad \textit{where } \mu \equiv y(\mathbf{x}(t)) \geqslant 0 \\
\rho(\neg\varphi, \mathbf{x}, t) & = & -\rho(\varphi, \mathbf{x}, t) \\
\rho(\varphi_1 \wedge \varphi_2, \mathbf{x}, t) & = & \min(\rho(\varphi_1, \mathbf{x}, t), \rho(\varphi_2, \mathbf{x}, t)) \\
\rho(\varphi_1 \, \mathbf{U}_{[a,b]} \varphi_2, \mathbf{x}, t) & = & \sup\limits_{t' \in t + [a,b]} (\min(\rho(\varphi_2, \mathbf{x}, t'), \inf\limits_{t'' \in [t, t')} (\rho(\varphi_1, \mathbf{x}, t''))))
\end{array}
$$

Moreover, we let $\rho(\varphi, \mathbf{x}) := \rho(\varphi, \mathbf{x}, 0)$*.*

The sign of $\rho(\varphi, \mathbf{x})$ provides the link with the standard boolean semantics of [24]: $\rho(\varphi, \mathbf{x}) > 0$ if and only if $\mathbf{x} \models \varphi$, while $\rho(\varphi, \mathbf{x}) < 0$ if and only if $\mathbf{x} \not\models \varphi$. The case $\rho(\varphi, \mathbf{x}) = 0$, instead, is a borderline case, and the truth of φ cannot be assessed from the robustness degree alone, see [8] for a deeper discussion of this

[4] In the context of STL, to be consistent with terminology in the literature, we use the term signal to refer to cadlag functions.

[5] $\tilde{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

issue. The absolute value of $\rho(\varphi, \mathbf{x})$, instead, can be interpreted as a measure of the robustness of the satisfaction with respect to noise in signal $\mathbf{x}$, measured in terms of the induced perturbation in the secondary signal. This means that if $\rho(\varphi, \mathbf{x}, t) = r$ then for every signal $\mathbf{x}'$ for which every secondary signal satisfies $\max_t |y_j(t) - y'_j(t)| < r$, we have that $\mathbf{x}(t) \models \varphi$ if and only if $\mathbf{x}'(t) \models \varphi$.

Remark 5. *We stress that the choice of the secondary signals $y : \mathbb{R}^n \to \mathbb{R}$ is an integral part of the definition of the STL formula, reflecting the intuition of the modeller and encoding the behaviour of interest. Different choices of secondary signals result in formulae expressing different behavioural properties, hence naturally in different robustness measures.*

The robustness degree of Definition 4 has to be interpreted as a weight of "how much" a given model (with fixed initial conditions and parameters) satisfies a STL formula. More precisely, its absolute value represents the max distance of the signal $\mathbf{x}$ under consideration from the set of trajectories satisfying/violating the formula [9].

In this sense, this measure is different from the more common sensitivity-based notions of robustness, like those discussed in [31], measuring the size of a region in the parameter space in which the system behaviour is roughly constant. However, sensitivity analysis and its related techniques can be applied as well in combination with the robustness degree of Definition 4 as a discriminative function for interesting behavioural patterns to investigate. The definition of *robustness* or *robust satisfaction* considered here was first proposed by Fainekos and Pappas in [9] and later extended by Donzé et al. in [8]. This is different from the *satisfaction/violation degree* or *quantitative satisfaction* defined by Rizk et al. in [10]. In the first case the *robustness* corresponds to the distance of a signal from a set of signals satisfying the same formula (the minimal perturbation value that can violate the specification), while in the second case it provides a distance between a formula and a set of formulas satisfying the same signal [10].

3. Robustness of Stochastic Models

Consider a STL formula φ, with predicates interpreted over the state variables of a stochastic process $\mathbf{X}(t)$ with values in $E \subseteq \mathbb{R}^n$. Recall that $\mathbf{X}(t)$ can be seen also as a random variable $\mathbf{X}$ on the space E-valued cadlag functions $\mathcal{D}([0, \infty), E)$, which in this section we denote by $\mathcal{D}$, assuming the domain E to be fixed. The boolean semantics of φ is readily extended to stochastic models as customary, by measuring the probability of the set of the trajectories that satisfy the formula

$$P(\varphi) = \mathbb{P}\{\mathbf{x} \in \mathcal{D} \mid \mathbf{x} \models \varphi\}.$$

The rationale behind such definition is that a stochastic process defines a probability distribution on the space of trajectories. In case of PCTMC, this is usually obtained by applying the cylindric construction [5], while for SHA or SDE more refined constructions are needed, see for instance [16, 22]. Furthermore, the set of trajectories that satisfy/falsify a formula is a measurable set, so

that we can safely talk about its probability. In order to extend this definition to the robustness score, it is convenient to think of the set of trajectories that satisfy φ as a measurable function $I_\varphi : \mathcal{D} \to \{0,1\}$, such that $I_\varphi(\mathbf{x}) = 1$ if and only if $\mathbf{x} \models \varphi$. Then, we can define the random variable $I_\varphi(\mathbf{X})$ on $\{0,1\}$ induced by the stochastic process $\mathbf{X}$ via I_φ as the Bernoulli random variable which is equal to 1 with probability $P(\varphi)$. We can equivalently write:

$$\mathbb{P}(I_\varphi(\mathbf{X}) = 1) = \mathbb{P}(\{\mathbf{x} \in \mathcal{D} \mid I_\varphi(\mathbf{x}) = 1\}) = \mathbb{P}(I_\varphi^{-1}(1))$$

We can extend the robustness degree to stochastic processes in the same way: given a trajectory $\mathbf{x}$, we can compute its robustness degree according to Definition 4 and interpret $\rho(\varphi, \mathbf{x}) = \rho(\varphi, \mathbf{x}, 0)$ as a functional $\mathtt{R}_\varphi$ from the trajectories in $\mathcal{D}$ to $\mathbb{R}$. To propagate the distribution of $\mathbf{X}$ to $\mathbb{R}$, we need the to show that $\mathtt{R}_\varphi$ is measurable (with respect to the Borel σ-algebra of the topology of $\mathcal{D}$ induced by the Skorokhod metric, see Section 2.1). This is shown in the following theorem, proved in Appendix A.1.

Theorem 6. *For any STL formula φ, the functional $\mathtt{R}_\varphi : \mathcal{D} \to \mathbb{R}$ is measurable.* □

In virtue of Theorem 6, $\mathtt{R}_\varphi$ induces a real-valued random variable $R_\varphi = \mathtt{R}_\varphi(\mathbf{X})$ with probability distribution given by

$$\mathbb{P}\left(\mathtt{R}_\varphi(\mathbf{X}) \in [\mathtt{a}, \mathtt{b}]\right) = \mathbb{P}\left(\mathbf{X} \in \{\mathbf{x} \in \mathcal{D} \mid \rho(\varphi, \mathbf{x}, 0) \in [a, b]\}\right)$$

Stated otherwise, if we apply the definition of robustness to a stochastic model, we obtain a distribution of robustness degrees. This distribution is much more informative than the standard satisfaction probability, because it tells us "how much" a formula is true. In this paper we are interested in some statistics of this distribution, specifically the *average robustness degree* $\mathbb{E}(R_\varphi)$ and the *conditional average robustness* of a formula being true $\mathbb{E}(R_\varphi \mid R_\varphi > 0)$ or false $\mathbb{E}(R_\varphi \mid R_\varphi < 0)$. $\mathbb{E}(R_\varphi)$ measures how strongly a formula is satisfied on average. The larger this number, the more strongly robust is the satisfaction of a formula. This number is generally correlated with the satisfaction probability. However, in some cases it is possible to observe a large average robustness even for a small probability of satisfaction. For this reason, the conditional averages $\mathbb{E}(R_\varphi \mid R_\varphi > 0)$ and $\mathbb{E}(R_\varphi \mid R_\varphi < 0)$ are better indicators of the intensity of satisfaction and violation of a STL formula w.r.t. a stochastic model.

These indices express how large is the robustness degree in average, given that the formula is satisfied or violated, respectively. The conditional averages of robustness are related to the average robustness degree by the equation

$$\mathbb{E}(R_\varphi) = P(\varphi)\mathbb{E}(R_\varphi \mid R_\varphi > 0) + (1 - P(\varphi))\mathbb{E}(R_\varphi \mid R_\varphi < 0)$$

which holds provided $\mathbb{P}(R_\varphi = 0)$ is zero.

One goal of this paper is to investigate to what extent these three synthetic indices are good descriptors of the robustness distribution, and how they can be exploited to do parameter synthesis for stochastic models.

3.1. Continuity of $\mathtt{R}_\varphi$

An interesting question about the robustness degree $\mathtt{R}_\varphi$ of a formula φ is whether it is continuous as a functional on the space of trajectories $\mathcal{D} = \mathcal{D}([0,\infty), E)$, with respect to the topology induced by the Skorokhod metric, see Section 2.1. It turns out that $\mathtt{R}_\varphi$ is not continuous on $\mathcal{D}$, because $\mathcal{D}$ contains trajectories with discontinuous jumps, and the notion of metric convergence in $\mathcal{D}$ allows one to align close jumps (in time) between two trajectories. On the other hand, in the definition of the robustness degree ρ, there is no such a flexibility on the time bounds of the formula. This discrepancy results in the lack of continuity. The interested reader can find a counterexample in Appendix A.

Continuity, however, is a desirable feature, as it guarantees that small perturbations in the system trajectories will result in small perturbations in the robustness degree. Hence, a more precise characterisation of the continuity properties of $\mathtt{R}_\varphi$ on subspaces of $\mathcal{D}$ would be valuable, yet non-trivial. It is quite easy to assess that $\mathtt{R}_\varphi$ is continuous on the subspace $\mathcal{C} \subset \mathcal{D}$ of *continuous* trajectories, as in $\mathcal{C}$ the metric on $\mathcal{D}$ reduces to the standard supremum norm, for which $\mathtt{R}_\varphi$ is known to be continuous [26]. We conjecture that $\mathtt{R}_\varphi$ will be continuous also for most of the trajectories with jumps. The argument is as follows. Consider a simple formula $\varphi = \mathbf{F}_{[T_1,T_2]} X \geq 0$. Then problems may arise for all those trajectories for which two discontinuous jumps happen at exactly T_2, T_1, or $T_2 - T_1$ time instants apart. We conjecture, although we still do not have a formal proof, that the functional $\mathtt{R}_\varphi$ is continuous on all other trajectories. This soon implies that $\mathtt{R}_\varphi$ is almost surely continuous with respect to any probability measure on $\mathcal{D}$ induced by a Continuous Time Markov Chain (or by other nicely behaved stochastic processes, like Feller processes [32], which include Stochastic Hybrid Automata without forced jumps [16]). By standard arguments about weak convergence of probability measures on $\mathcal{D}$ [22], this will guarantee that small perturbations of any CTMC model will result in small perturbations of the distribution R_φ of the robustness degree.

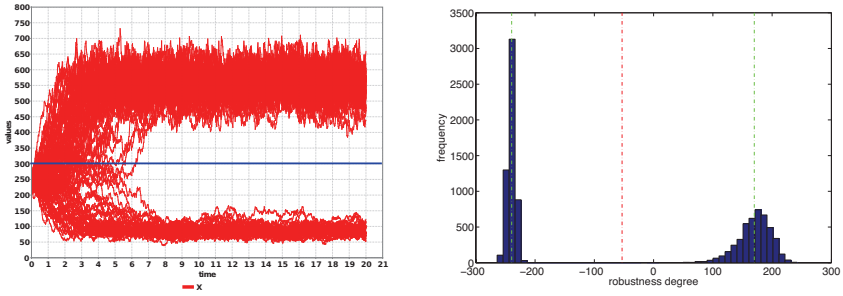
The alternative would be to modify the definition of the robustness degree ρ to enforce continuity of $\mathtt{R}_\varphi$. This would require a notion of space-time robustness. In [8], the authors consider a definition which is based on the localisation of zeros (of the atomic predicates or of the robustness function). Unfortunately, this degree can be computed easily for piecewise linear continuous signals, but it is undecidable in general, even for continuous functions [33, 34]). A possible alternative can be that of "blurring" the boundaries of the time intervals by a proper use of integrals. Investigating such a direction, however, is out of the scope of this paper.

4. Case Studies

In this section, we investigate experimentally the notion of robust semantics of STL formulae for stochastic models. We will consider three systems: the Schlögl system [36], a simple set of biochemical reactions exhibiting a bistable behaviour, the Incoherent type 1 Feed-forward loops (I1-FFL) [37], a frequent

Reaction	rate constant	init pop
$A + 2X \to 3X$	$k_1 = 3 \cdot 10^{-7}$	$X(0) = 247$
$3X \to A + 2X$	$k_2 = 1 \cdot 10^{-4}$	$A(0) = 10^5$
$B \to X$	$k_3 = 1 \cdot 10^{-3}$	$B(0) = 2 \cdot 10^5$
$X \to B$	$k_4 = 3.5$	

Table 1: Biochemical reactions of the Schlögl model. Parameters are taken from [35].



(a) CTMC simulation

(b) Distribution of the robustness degree

Figure 1: Simulation of the Schlögl model (100 runs), for parameters as in Table 1. The blue straight line is the value $X = 300$ (left). The distribution of robustness degree for the STL formula 4.2 with $T_1 = 10$ and $T_2 = 15$ time units (10000 runs). The average robustness is -53.15 (vertical red line), the conditional averages of robustness are 169.89 and -239.52 (vertical green lines), and satisfaction probability is 0.4552 (right).

motif in gene regulatory systems, and the Repressilator [38, 39, 40], a synthetic biological clock implemented as a gene regulatory network. More specifically, we consider CTMC models of the Schlög system and the I1-FFL, and a hybrid model of the Repressilator, to illustrate the general applicability of the stochastic robust semantics introduced in Section 3.

4.1. Schlögl system

The Schlögl model is a simple biochemical network with four reactions, listed in Table 1. The rates of the reactions are computed according to the mass action principle for stochastic models [15]. Species A and B are considered to be present in large quantities, hence they are assumed to be constant and the system will be represented by only one variable, the concentration of the species X. The characteristic of this system is to have, for certain parameter values, like the one shown in Table 1, a bistable behaviour. More specifically, the reaction rate ODE system has two stable steady states, and for this model the trajectories of the stochastic system starting from a fixed initial state, $X(0) = x_0$, can end up in one attractor or the other. The probability of choosing one stable state or the other depends on the position of x_0 relative to the basin of attraction of the

two equilibria. If we start close to its boundary, the bistable behaviour becomes evident, see Figure 1(a). This model have been implemented and executed in MATLAB®.

We now consider the property of eventually ending up in one basin of attraction, using the following STL formula to express it

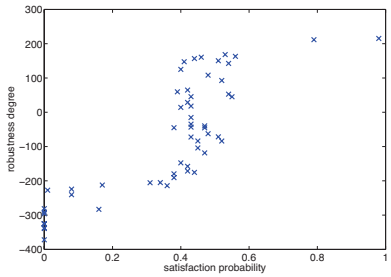
$$\varphi : \mathbf{F}_{[0,T_1]}\mathbf{G}_{[0,T_2]}(X - k_t \geq 0), \quad k_t = 300. \tag{4.2}$$

The exact meaning of the formula is: *after at most T_1 time units, the concentration of the species X stabilises to a value which remains above $k_t = 300$ for at least T_2 time units.* From the atomic predicate $\mu(X) = X - k_t \geq 0$, we can derive the secondary signal $y(x(t)) = x(t) - k_t$, where $x(t)$ is the primary signal corresponding to a trajectory of the system (the variation in the concentration of X over the time).

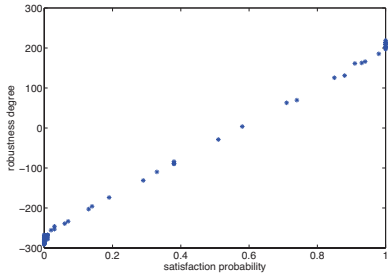
As Figure 1(a) shows, if the model is in the large equilibrium, then this property will be true, and false in the other case. If we estimate the probability of the formula statistically, for model parameters as in Table 1 and formula parameters $T_1 = 10$ and $T_2 = 15$, then we obtain the value $p = 0.4583$ (10000 runs, error ± 0.02 at 95% confidence level and execution time of 99.49 seconds). This raw number cannot be used to retrieve any information on the bistability of the system. Indeed, a system stabilising just above $x(t) = k_t = 300$, and such that roughly 55% of its trajectories cross such threshold "frequently", can satisfy the same formula with the same probability. The bimodal behaviour of the system becomes evident, instead, if we look at the distribution of the robustness degree of the formula, see Figure 1(b). The figure shows also the conditional robustness averages of the formula φ in (4.2) being true or false that are 169.89 and -239.52, respectively. These two indicators estimate how robustly the system remains in the basin of attraction of each steady state. Hence, the robustness degree carries additional amount of information w.r.t. the satisfaction probability of a STL formula. We stress that we are not comparing the robustness degree with the probability distribution of the CTMC $X(t)$: both the satisfaction probability of φ and its robustness are (unidimensional) quantities derived from $X(t)$, which are easier to compute and visualise. In Figure 2, we show two examples of the relationship between the average robustness and the satisfaction probability. We varied the threshold level k_t in the formula (Figure 2(a)), and the rate constant k_3 (Figure 2(b)), and then we plot the satisfaction probability versus the average robustness degree, estimating them statistically from 10000 runs for each parameter combination. As we can see these two quantities seem to be correlated. By varying the threshold, the Pearson's correlation coefficient between satisfaction probability and robustness degree is 0.8386, while the dependency, by visual examination of the data, seems to follow a sigmoid shaped curve. In the second case, instead, the correlation between satisfaction probability and average robustness degree is 0.9718, with an evident linear trend.

4.2. Incoherent type 1 Feed-forward loops

The second example we discuss is a small frequent motif in genetic regulatory networks [37], known as the feed forward loop (FFL). FFL are composed by

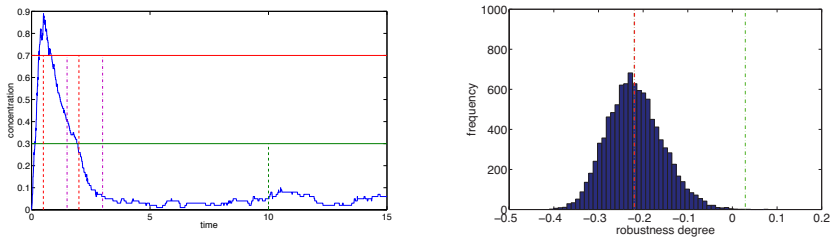


(a) Varying the threshold k_t.



(b) Varying the parameter k_3.

Figure 2: Satisfaction probability versus average robustness degree for varying (left) the threshold k_t in the STL formula (4.2) and (right) the parameter k_3. k_3 was varied between 100 and 300 in steps of 10 units, while the threshold k_t was varied between 50 and 600 in steps of 10.



(a) Stochastic simulation

(b) Robustness distribution

Figure 3: Simulation of the I1FFL model. Trajectory of X_C, for parameters $K_{AB} = 1$, $K_{AC} = 1$, $K_{BC} = 0.17$, $\alpha_C = 1$, $\beta_C = 1$, $\alpha_B = 1$, $\beta_B = 1$ and n = 2, $X_A = 1$ and initial value $x_B(0) = 0$, $x_C(0) = 0$. The range of variables is expressed as a concentration (left). The distribution of the robustness degree for the formula 4.3 with $\theta_{high} = 0.7$, $T_r = 0.5$, $h = 1$, $T_s = 1.5$, $T_{off} = 10$, $T = 15$, and $\theta_{low} = 0.3$ for 10000 runs (right). The simulator for the model has been implemented in Java.

two genes, B and C, in which B regulates C and both are regulated by a third transcription factor, the product of gene A. The regulation is acyclic: there are no feedback loops. However, different roles played by A and B as regulators of the expression of B and C give rise to different behaviours, which are used within the cell to modulate the response to external (or internal) stimuli, changing the expression of A. In this paper, in particular, we discuss the *incoherent type-1 FFL* (I1FFL), according to the nomenclature of [37]. I1FFL is characterised by a topology with two parallel but competitive paths: A activates the production of both B and C, while B is a repressor of C. We consider the case of an "AND" logic gate in C, corresponding to the situation in which we have production of C if A is above its concentration threshold and B is not above its concentration threshold (i.e. B repression is not active).

The dynamics of this network can be understood in terms of an input/output relationship, where C is the output signal. In the presence of an external input signal, which corresponds to a high concentration of A, the production of B and C is activated in parallel. B takes some time to accumulate and to cross the threshold to activate C repression. This effect translates in an initially high production of C followed by a decrease and a consequent stabilisation to a low steady state. This results in pulse-like response to the input signal, as can be seen in Figure 3(a), left.

The network will be described by a Markov population process, in which activation and repression are modelled as Hill functions, while degradation is described in the standard mass action style [15]. The concentration X_A is considered to be constant, as an input signal. More precisely, we have the following reactions:

B **production:** $\emptyset \to B$, at rate $\beta_B \frac{X_A^n}{K_{AB}^n + X_A^n}$;

B **degradation:** $B \to \emptyset$, at rate $\alpha_B X_B$;

C **production:** $\emptyset \to C$, at rate $\beta_C \frac{X_A^n}{K_{AC}^n + X_A^n} \frac{1}{K_{BC}^n + X_B^n}$;

C **degradation:** $C \to \emptyset$, at rate $\alpha_C X_C$;

In order to capture the pulse-like behaviour, we use the STL formula

$$\varphi_{pulse} = \mathbf{F}_{[T_r, T_r+h]} \mathbf{G}_{[0,T_s]} (X_C \geq \theta_{high}) \wedge \mathbf{G}_{[T_{off}, T]} (X_C \leq \theta_{low}). \tag{4.3}$$

The formula φ_{pulse} requires the output signal to be above an high threshold θ_{high} at a certain time $t \in [T_r, T_r + h]$ from the introduction of the input signal, and remains high for at least T_s time units. Furthermore, the formula imposes that the pulse has terminated after T_{off} units of time, so that the concentration of X_C stabilises to a low value (less than θ_{low}) for at least T time units.

In Figure 3(b), we show the distribution of the robustness degree of the formula (see caption for parameters) with 10000 simulations. The average robustness degree is of -0.2198 and is almost equal to the negative average robustness, -0.2204. Indeed, the formula is "almost always" false, as confirmed by the satisfaction probability degree that is 0.0014. Figure 3(a) illustrates on the left that the negative robustness on a simulated trajectory is caused by the pulse peak, which does not last enough time. In the next section we will show how to tackle the system design problem of this stochastic model by using the robustness degree to guide the parameter synthesis for which the pulse will become larger and longer.

4.3. Repressilator

The last case study is a genuine stochastic hybrid model of the Repressilator [38], a synthetic genetic clock composed of three genes expressing three transcription factors repressing each other in a cyclical fashion (see Figure 4). Genetic networks like the Repressilator can also be found in actual biological systems. Here we consider the putative regulatory network of the Circadian Clock in *Ostreococcus Tauri*, an unicellular alga that is widely studied as a model organism. The model and parameters are taken from [19], where the authors start from experimental data and learn a stochastic hybrid model of the circadian clock network in *O. Tauri*, which is known to involve only two genes, expressing transcription factors TOC1 and CCA1. They conjecture the existence of a third regulatory protein X, similarly to the mechanism recently discovered for the circadian clock in *Arabidopsis Taliana*, forming a repression cycle as in Figure 4.

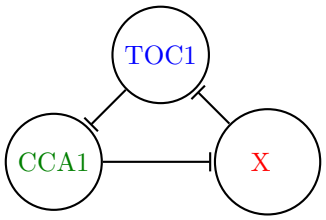
This system is modelled by a SHA with three continuous variables, X_{TOC1}, X_{CCA1}, and X_x, and with eight discrete modes, corresponding to all possible combinations of active and inactive states of each involved gene. The dynamics of gene repression is modelled as a telegraph process, i.e. as a two states Markov model, with a phenomenological rate of repression of gene i (binding of the protein to the gene) equal to

$$f_{bind,i}(\mathbf{X}) = k_{p_i} \exp(k_{e_i} X_j),$$

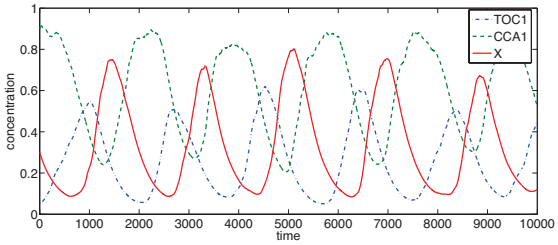
where X_j is the repressor of gene i. The unbinding rate, instead, equals a constant: $f_{unbind,i}(\mathbf{X}) = k_{m_i}$. The steady state distribution of the probability of gene i being active is a sigmoid (saturating) function of the repressor concentration. Production and degradation of protein i, instead, are modelled by stochastic differential equations of the form

$$dX_i = (A_i\mu_i + b_i - \lambda_i X_i)dt + \sigma dW,$$

where μ_i denotes the state of gene i (with $\mu_i = 1$ denoting the repressed state and $\mu_i = 0$ the active state), b_i is the basal production rate and $A_i < 0$ reduces it in case of repression, and λ_i is the degradation rate. The form of this model is particularly efficient to perform statistical inference of parameters in presence of observed data. Note also that the stochastic hybrid model we consider here is different from previous hybrid models of the repressilator [39, 40], in that it assumes a different form for the rate of gene repression, and in that it models protein production and degradation by stochastic differential equations rather than ODEs. In Figure 4(b), we show a simulation of the model, for the parameter values taken from [19], which exhibit sustained oscillations with a more or less stable period.



(a) Repressilator-like gene network of the O. Tauri circadian clock [19]



(b) Hybrid stochastic simulation

Figure 4: The repressilator-like model of the O.Tauri circadian clock (left) is a cyclic negative-feedback loop composed of three repressor genes: TOC_1, CCA_1, and an unknown gene X. Oscillatory behaviour of the model (right), for model parameters taken from [19].

In order to specify the presence of oscillations, we use the STL formula

$$\psi = \mathbf{G}_{[0,T]}((X_i > k_{high}) \rightarrow \mathbf{F}_{[T_1,T_1+h]}(X_i < k_{low})) \wedge \mathbf{G}_{[0,T]}((X_i < k_{low}) \rightarrow \mathbf{F}_{[T_2,T_2+h]}(X_i > k_{high})) \wedge \mathbf{F}_{[0,T]}(X_i > k_{high}), \tag{4.4}$$

expressing the fact that high values of X_i alternate to low values, with a period between T_1+T_2 and T_1+T_2+2h, where i is TOC1, CCA1 or X. In particular, we require that a high value of X_i ($X_i > k_{high}$) is followed within time $[T_1, T_1 + h]$ by a low value of X_i, which is subsequently followed by a high value in a time between $[T_2, T_2+h]$. The last part of the formula requires that the model indeed reaches a high value of X_i in order to trigger the chain of implications. The parameter h gives a value of the period stability, the higher the h, the more irregular is the period. As said before, X_i can be the concentration of one of the three proteins of the clock. In the next discussion, we focus on the unknown protein X.

Again, the robustness degree provides us a measure of the satisfaction/violation of the formula. An example is shown in Figure 5(a) left, where we can see that the formula, for the parameters in the caption, tends to be false (the average robustness is negative and the satisfaction probability is 0.256). This, of course, may depend on the choice of the parameters, which do not properly capture the amplitude, the period, and the shape of the oscillations. For instance, a negative robustness value of δ can be obtained if, from a point in which $X_i < k_{low}$, the system remains below $k_{high} - \delta$ for a whole (half) period of oscillation (which is constrained to be in $[T_2, T_2 + h]$). We will see in Section 5 how these parameters can be chosen in a principled way, taking inspiration from requirement mining [41, 42]. In the same figure, we can also see the conditional average robustness, whose values are both close to the average robustness, suggesting that the low satisfaction probability is not very robust. In Figure 5(b), we plot the average robustness against the satisfaction probability, varying the property parameter T_1, showing once again the correlation between the two quantities. The dependency seems to be linear, with a Pearson correlation coefficient is 0.9841.

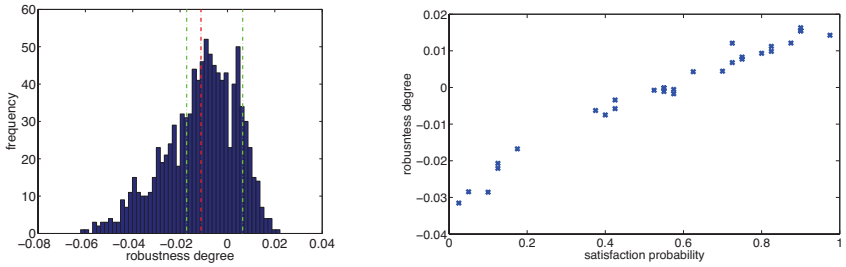
5. System Design

We now discuss an application of the robust semantics to the system design problem. First we summarize the problem, then we illustrate the method to tackle it, describing in detail the *Gaussian Processes* and the *GP-UCB algorithm*. Finally, we present some experimental results.

The problem we want to tackle is the following:

> Given a population (hybrid) model, depending on a set of parameters $\boldsymbol{\theta} \in \boldsymbol{K}$, and a specification φ given by a STL formula, find the parameter combination $\boldsymbol{\theta}^*$ such that the system satisfies φ as *robustly* as possible (in expectation).

We will tackle this problem by:



(a) Robustness distribution (b) Satisfaction probability vs robustness degree

Figure 5: Robustness distribution for Formula 4.4 parameters $k_{low} = 0.14$, $k_{high} = 0.5$, $T_1 = 800$, $T_2 = 700$, $h = 350$, $T = 7000$ and 1000 runs. The average robustness (red line) is -0.0151, conditional averages of robustness are -0,0183 and 0,0064 (green lines), and the estimated satisfaction probability is 0.256 (left). Satisfaction probability versus average robustness. T_1 was varied between 800 and 1100 in steps of 10 units (right).

- rephrasing it as a (non-linear, non-convex) optimisation problem;
- evaluating the function to optimise using statistical model checking at a few parameter values with a fixed number of runs, usually 100;
- solving the optimisation problem using an optimisation strategy for reinforcement learning, based on statistical emulation and Gaussian processes regression (Gaussian Process - Upper Confidence Bound optimisation, GP-UCB [12]).

In certain cases, such as bistable systems, a large average robustness may not be the appropriate objective; in fact, for highly unbalanced robustness scores, a formula can have a high average robustness without having a high probability of being true. Therefore, one would like to modify the design problem to incorporate an additional constraint that the satisfaction probability p of the formula be bounded below by a fixed q. This considerably complicates the problem: we are not aware of provably convergent non-convex constrained optimisation algorithms. Nevertheless, the problem can be approximately solved using penalty terms to encode for probability constraints. More specifically, assuming we want to enforce the satisfaction probability to be at least q, we add a penalty term of the form $\alpha(q - p)$, if $p < q$, and 0 otherwise, where $\alpha < 0$ controls the penalty intensity. A sufficiently high value of α (which can be chosen manually with a few trial runs) will ensure that the optimisation will satisfy the probability constraint.

5.1. Gaussian Processes - Upper Confidence Bound Optimisation

Gaussian Processes. The key ingredient for the design problem is an efficient estimation of the unknown objective function, i.e. the average robustness as a function of the process kinetic parameters. Our simulation-based method provides us with an efficient tool to estimate approximately the robustness value

for a specific set of parameters; the question is therefore how to best estimate (and optimise) an unknown function from observations of its value at a finite set of input points. Function approximation is a central task in machine learning and statistics, where it is usually referred to as *regression*. The general regression task can be formulated as follows [43]: given a set of input-output pairs (θ_i, y_i), $i = 1, \ldots, N$ (*training data*), with $\theta_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, determine a function $f\colon \mathbb{R}^d \to \mathbb{R}$ s.t. $f(\theta_i)$ is optimally close to the target values y_i (usually in terms of minimising a suitable loss function). Several methods exist for addressing this task; in this paper we adopt a Bayesian perspective: we specify a prior distribution over a suitable function space, and condition on the observed values to obtain a posterior estimation of the function value at all possible input points.

Probability distributions over spaces of functions are complex infinite dimensional objects; we can glean an intuition about their nature by considering a generative process in terms of *basis functions*. Let $\Phi = \{\varphi_1(\theta), \ldots, \varphi_N(\theta)\}$ be a set of *fixed* functions of the input variable θ, and define a function

$$f(\theta) = \sum_{j=1}^{N} \alpha_j \varphi_j(\theta) \quad \alpha_j \sim \mathcal{N}(0, 1)$$

by taking a linear combination of the basis functions with Gaussian distributed random coefficients. The value of the function f at any point is clearly a Gaussian distributed random variable (as it is a linear combination of Gaussian random variables); thus, the procedure outlined above is a generative construction providing a probability distribution over functions. Nevertheless, the expressivity of this probability distribution is severely limited by the finite set of basis functions chosen; this problem can be obviated by choosing an infinite set of basis functions (and scaling appropriately the coefficients). In this way, we arrive at a *Gaussian Process* (GP), a popular stochastic process which is often employed in Bayesian non-parametric regression [44].

GPs are flexible non-parametric distributions over spaces of functions which can be used as prior distributions in a Bayesian framework, where the input-output pairs represent noisy observations of the unknown function. This enables a natural quantification of the uncertainty of the estimated function at every new input value; this uncertainty will play a central role in the optimal design strategy we propose in Section 5. The basis function construction described above is useful to grasp the essence of GPs, but not particularly suited for practical computation. One can instead equivalently define a Gaussian process in terms of its marginal distributions as follows [44]:

Definition 7. *A Gaussian Process over a (portion of) $\mathbb{R}^d$ is a collection of random variables indexed by $\theta \in \mathbb{R}^d$ such that every finite dimensional marginal distribution is multivariate normal. Furthermore, there exist two functions $\mu\colon \mathbb{R}^d \to \mathbb{R}$ (mean function) and $K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (covariance function) such that the mean and covariance of the finite dimensional normal marginals is given by evaluating the mean and covariance functions at each point and each pair of points respectively.*

We denote a sample from a GP with mean function μ and covariance function K as

$$f \sim \mathcal{GP}(\mu, K).$$

A common choice, which we will adhere to, is to use zero mean GPs with *radial basis function* (RBF) covariance

$$K(\theta_1, \theta_2) = A \exp\left[-\frac{\|\theta_1 - \theta_2\|^2}{\lambda}\right]$$

where A and λ are hyper-parameters controlling the prior variance and auto-correlation distance of the process respectively. For further details about the definition of GPs and the RBF covariance we refer to [44].

Gaussian Process regression. Definition 7 characterises a GP in terms of its finite dimensional marginals; this is extremely useful because it gives an analytically tractable route to estimate a posterior distribution over the function value at a new point θ_{new}. Denote again as $(\theta_i, y_i), \quad i = 1, .., N$ our observations, and let $p(y_i|f(\theta_i))$ denote observation error model. Then the basic rules of probability yield

$$\begin{aligned} p(f(\theta_{new})|y_1, \dots, y_N) &= \int \prod_{i=1}^{N} df(\theta_i) p(f(\theta_{new}), f(\theta_1), \dots, f(\theta_N)|y_1, \dots, y_N) = \\ &= \int \prod_{i=1}^{N} df(\theta_i) p(f(\theta_{new})|f(\theta_1), \dots, f(\theta_N)) p(f(\theta_1), \dots, f(\theta_N)|y_1, \dots, y_N) \end{aligned} \tag{5.5}$$

where in the last equality we have used the fact that the function value at the new point depends on the observations only through the function values at the training points (not on the actual observations).

A case of particular interest arises when the observation error model $p(y_i|f(\theta_i)) = \mathcal{N}(0, \sigma^2)$ is Gaussian itself. Then the integrand in equation (5.5) is Gaussian and can be computed analytically. Denote as K the matrix obtained evaluating the covariance matrix at all pairs of training inputs, as $\mathbf{k}_*$ the vector obtained evaluating the covariance matrix at the new input and at all training input, and as $k_{**} = K(\theta_{new}, \theta_{new})$ the prior variance at the new input point. Using the formula for inverting block matrices, one can easily obtain the following closed form expression for the predicted mean and variance of the function at the new input point

$$\begin{aligned} E[f(\theta_{new})] &= \mathbf{k}_*^T \left(K + \sigma^2 I\right)^{-1} \mathbf{y} \\ var(f(\theta_{new}) &= k_{**} - \mathbf{k}_*^T \left(K + \sigma^2 I\right)^{-1} \mathbf{k}_*. \end{aligned} \tag{5.6}$$

Few facts are remarkable about these equations: first of all, the expected value at a new point is a linear combination of the previous observations. Secondly,

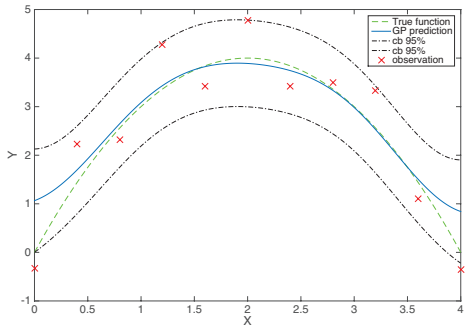


Figure 6: An example of Gaussian Process regression. The green dashed curve is the true functions, sampled noisily ($\sigma^2 = 0.8$) in few input points (red crosses). The solid blue curve is the GP-prediction function, obtained with a Gaussian kernel with $\lambda = 0.8$ and $A = 1$. The dashed-dotted black lines are the 95% confidence bounds of the predictive distribution.

variance is always reduced by adding new observations. Finally, and most importantly, these equations provide an analytic functional form for the predictive mean and variance at a new input, in terms of a linear combination of *basis functions* centred at the training points. A visual example of Gaussian process regression can be found in Figure 6.

In practice, the input-output pairs in a regression task are often different features of experimentally observed data points. In this paper, the output points correspond to true functional evaluations of an unknown (and analytically intractable) function of the inputs. In this case, the regression task is often given the special name of *emulation* in the statistics literature: the true (but unknown) function is assumed to be a draw from a GP, and the functional evaluations are used as observations to obtain a posterior estimate of the unknown function. This approach was initially introduced in order to perform sensitivity analysis for deterministic computer models in [45]; in that case, the function evaluations could be assumed to be noiseless (apart from numerical errors that were considered negligible in that paper). In our case, the function linking model parameters to average robustness cannot be computed, and we can only obtain a sampling approximation through a Statistical Model Checking procedure. This means that our function evaluations will be noisy; by virtue of the Central limit theorem we can assume that, provided sufficient samples were used for the SMC estimates, the noise in the observed robustness estimates will be approximately Gaussian.[6] This therefore enables us to obtain an analytical estimate of the posterior process [44]. Furthermore, the SMC samples also allow us to estimate

[6]Note that here we approximate as a GP the average robustness score (or any other fitness score) as a function of parameters. We are not imposing any (Gaussian) approximation of the process itself.

the (sample) variance in the average robustness at every sampled parameter value; this information can also be included leading us to a heteroscedastic (i.e. with non identical noise) regression problem (which is however still analytically tractable).

Gaussian Processes Optimization: the GP-UCB algorithm. As we have seen, GP emulation provides a convenient way to explore approximately the average robustness of a stochastic process for different values of the model parameters. One could then be tempted to also use the emulated robustness profile for model design, i.e. find the optimum of the emulated function. This strategy, while appealing in its simplicity, is vulnerable to local optima: the emulated function is estimated based on relatively few function evaluation, so that, while the emulator typically provides a good approximation of the true function near the sampled points, regions of parameter space far from the sampled points may contain the true maximum undetected. Using the language of reinforcement learning, maximising the emulated function would privilege exploitation (i.e. using currently available information) at the expense of exploration. Obviously, given sufficient computational power, one may consider sampling many parameter points so as to have sufficient coverage of the whole region of interest; this strategy is however bound to fail in even moderate dimensions due to the curse of dimensionality.

An elegant solution to the above conundrum can be obtained by also considering the *uncertainty* of the emulated function (which is also computed analytically in GP regression): intuitively, one should explore regions where the maximum *could plausibly be*, i.e. regions in parameter space where there is substantial posterior probability mass for the function to take a high value. We formalise these ideas in a recursive search rule, the so called Gaussian Process Upper Confidence Bound (GP-UCB) algorithm: assume we have computed the average robustness at N parameter values (so that we have N input output pairs). Let $\mu_N(\boldsymbol{\theta})$ and $\nu_N(\boldsymbol{\theta})$ be the mean and variance of the GP emulator at a given point in input space $\boldsymbol{\theta}$ (recall that the marginal at any point will be Gaussian)[7]. We select the parameter value $\boldsymbol{\theta}_{N+1}$ for the next function evaluation according to the following rule

$$\boldsymbol{\theta}_{N+1} = \text{argmax}_{\boldsymbol{\theta}} \left(\mu_N(\boldsymbol{\theta}) + \beta_{N+1}\nu_N(\boldsymbol{\theta})\right) \tag{5.7}$$

where β_{N+1} is a parameter. Thus, the next point for exploration does not maximise the emulated function, but an upper confidence bound at a certain confidence level specified by the parameter β_{N+1} (the quantile can be obtained by applying the inverse probit transform to the parameter). [12] proved that this algorithm converges to the global maximum of the unknown function with high probability (which can be adjusted by varying the algorithm's parameters). The primary difficulty in applying GP-UCB is that, in order to be able to

[7]We now denote the input as $\boldsymbol{\theta}$ to emphasise that they are the parameters of a stochastic process

apply the rule, the emulated function must be computed at a large number of points; while this is obviously not as onerous as evaluating the true unknown function (as the emulator is known analytically), it may still be problematic for high dimensional parameter spaces. Nevertheless, the algorithm can be applied effectively for moderate sized parameter spaces (of the order of 10 parameters), and modular construction may be used to extend to higher dimensional systems [46, 47].

5.2. Experimental Results

Here we present some experimental results. We considered three different design problems on the three case studies illustrated in Section 4. The optimisation algorithm and the experiments are implemented in `Matlab` exploiting the Breach toolbox [27] for the verification and a dedicated `Java` implementation for the SMC. All the experiments were run on a Macbook Pro, OS X 10.9.5, Intel Core i5 processor with 2.6 GHz, 8GB 1600 MHz memory.

Schlögl system. We set up the experiment as follows. We ask to maximise the robustness degree of the formula 4.2 optimising the parameter k_3. We varied k_3 uniformly in $[50, 1000]$, fixing all other parameters to the values of Table 1. We ran the GP-UCB optimisation algorithm by first estimating the robustness degree and the satisfaction probability, using statistical model checking, for 15 points sampled randomly and uniformly from the parameter space with 100 runs for each sample, and then using the GP-UCB strategy to estimate the maximum of the upper bound function in a grid of 200 points. If in this grid a point is found with a larger value than those of the observation points, we compute the robustness also for this new point, and add it to the observations (thus changing the GP approximation). Termination happens when no improvement can be made after three grid resampling. Further integration of local maximisation can further improve the method.
In the experiment, repeated 10 times, we used a GP with radial basis kernel [43], with length scale fixed to 0.5 (after standardisation of the parameter range to $[-1, 1]$). The amplitude of the kernel was adaptively set to 60% of the difference between the max and the mean value of the robustness for the initial observations. The observation noise was experimentally fixed to 1, by monitoring the average standard deviation at different random parameter combinations.
The median of the results are shown in Table 2. As we can see, the result of the optimisation suggests that the more robust system satisfying the specification (i.e. remaining as much as possible above the threshold 300 for a sufficiently long amount of time) is the one obtained for $k_3 = 997, 78$. In Table 2 we report also the optimisation time, 24.99 seconds, which is almost entirely spent in evaluating the likelihood (24.34 seconds), i.e. in running the simulations of the system. In Figure 7(a), we plot the emulation of the robustness function obtained in the last iteration of one of the 10 experiments. We can see also from here that the optimum value of k_3 (the value that maximise the emulation function and the robustness degree) is 1000. The result is confirmed by the computation of the robustness distribution via SMC, Figure 7(b): for $k_3 = 1000$, the system

k_3 **optimisation**	Median	Range
Parameter	1000	[939.08, 1000]
Average robustness	350.6	[344.98, 353.74]
Probability satisfaction	1	1
n Rob. fcn evaluations	17	[16, 20]
n simulation runs	1700	[1600, 2000]
Rob. fcn eval. time (sec.)	22.565	[20.19,29.43]
Optim. time (sec.)	23.156	[20.58, 30.82]

Table 2: Statistics of the results of ten experiments to optimise the parameter k_3 in the range $[50, 1000]$. We report the median and the range of the optimum parameter, the average robustness, the probability satisfaction, the number of robustness function evaluations (n Rob. fcn evaluations), the total number of simulation runs, and the time, in seconds, of the robustness function evaluations (Rob. fcn eval. time) and the optimisation (Optim. time). The number of runs for each evaluation, i.e. each SMC, is 100.

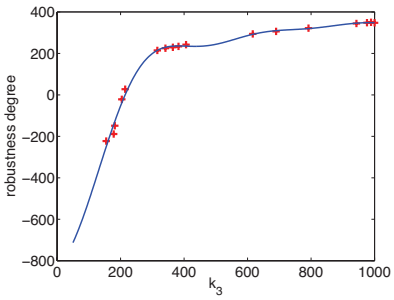
becomes monostable, and X stably remains above 550 units (corresponding to an average robustness score above 250).

Type 1 Incoherent Feed Forward Loop. In the I1FFL example, we try to optimise model parameters to force a specific shape to the pulse, namely a duration of 1.5 time units with an amplitude larger than 0.7. This is obtained in the STL framework by assigning the following parameters to the formula (4.3): $\theta_{high} = 0.7$, $T_r = 0.5$, $h = 1$, $T_s = 1.5$,$T_{off} = 10$, $T = 15$ and $\theta_{low} = 0.3$.

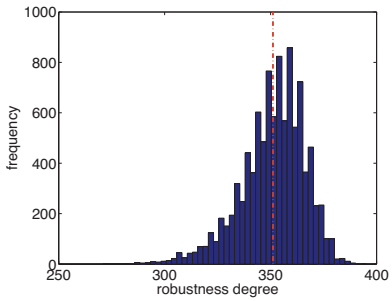
We considered two scenarios, assuming we can regulate the repression and degradation rates of the regulation of protein C. In the first case, we optimise only the degradation α_C, while in the second case, we optimise simultaneously both K_{BC} and α_C. For each scenario, we use the robustness score of the STL formula 4.3.

The setting of the algorithm are similar to the ones for the Schlögl model, except for the hyper-parameters of the kernel and the observations noise. In this case, the hyper-parameters of the kernel have been identified relying on a model selection criterion, i.e. optimising the robustness function as computed from an initial batch of observations, cf. [44] for more details. We improve also the treatment of the observation noise by using an heteroscedastic noise model. The noise of the robustness function for each explored point of the parameter space is estimated by bootstrapping. Furthermore, in this set of experiments, we use 10 initial random samples for each parameter.

We ran the optimisation algorithm 10 times. The results are reported in Figure 5.2 and Table 3. As we can see, the algorithm returns a precise value in one dimension, while it tends to be more erratic when searching the two dimensional space. This is typically a sign of uncertainty in the identification of parameters, meaning that there is some sort of dependency between the parameters we are exploring, resulting in a flat maximum or in a ridge of points more or less with the same robustness. This is confirmed in Figure 9, where we plot the emulated function at the end of one optimisation. We can see

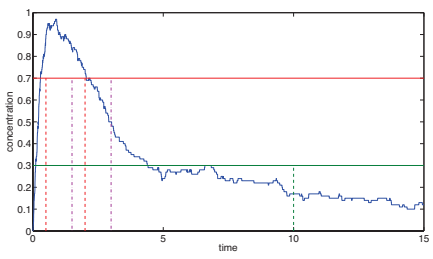


(a) The estimated robustness function

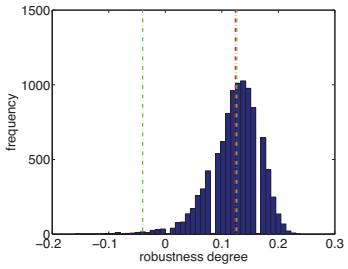


(b) Robustness distribution after the optimisation

Figure 7: The emulated robustness function in the optimisation of k_3 (left). The distribution of the robustness score for $k_3 = 1000$ and 10000 runs; the average robustness (red line) is 351.1 and the satisfaction probability is 1.



(a) Stochastic simulation



(b) Robustness distribution

Figure 8: Simulation of the I1FFL model, for parameters $kBC = 0.1732$, $\alpha_C = 0.3555$, $K_{AB} = 1$, $K_{AC} = 1$, $\beta_C = 1$, $\alpha_B = 1$, $\beta_B = 1$ and n = 2, and initial value $x_A(0) = 1$, $x_B(0) = 0$, $x_C(0) = 0$. The range of variables is expressed as a concentration (left). The robustness distribution of the formula 4.3 with $\theta_{high} = 0.7$, $T_r = 0.5$, $h = 1$, $T_s = 1.5$, $T_{off} = 10$, $T = 15$, and $\theta_{low} = 0.3$ for 10000 simulations (right).

that in almost all the region identified by the parameter range of Table 3 for α_C and k_{BC}, the robustness degree is similar (the dark red region); furthermore, the variability is more evident for the α parameter than the threshold concentration K_{BC}. Finally, we can note from Table 3 that also in this case most of the computational cost is spent in the evaluation of the robustness function, i.e. in simulation of the model and statistical model checking.

Repressilator. In this final scenario, we consider a different optimisation problem, in which we keep model parameters constant and we try to optimise the parameters of the formula to make the robustness score as large as possible. This is a version of the requirement mining problem [41], which can be seen as a sort of dual problem to system design, in which the goal is to learn the

α_C, K_{BC} **optimisation**	α_C		α_C, K_{BC}	
	Median	Range	Median	Range
Parameter	0.3759	[0.3703, 0.4038]	0.3812, 0.1742	[0.3183, 0.4806], [0.1556, 0.1899]
Average robustness	0.1239	[0.1187, 0.1358]	0.1270	[0.1186, 0.1352]
Probability satisfaction	0.942	[0.899, 0.999]	0.973	[0.897, 0.989]
n rob. fcn evaluations	22	[17, 23]	58	[55, 62]
n simulation runs	2200	[1700, 2300]	580	[5500, 5800]
Rob. fcn eval. time (sec.)	28.38	[21.22, 30.83]	81.69	[77.59, 89.77]
Optim. time (sec.)	30.42	[22.17, 33.36]	108.98	[99.73, 129.91]

Table 3: Statistics of the results of 10 experiments to optimise the parameter α_C in the range $[0.035, 3.5]$ and simultaneously both α_C in the same range and the parameter K_{BC} in the range [0.017,1.7]. We report the median and the range of the optimal parameter, the average robustness, the probability satisfaction, the number of robustness function evaluations (n rob. fcn evaluations), the total number of simulation runs, and the time, in seconds, of the robustness function evaluations (Rob. fcn eval. time) and the optimisation (Optim. time). The number of runs for each function evaluation, i.e. the number of simulations for SMC, is 100.

emergent behaviour of the model in terms of the most robustly satisfied formula (of fixed structure). Furthermore, the parametrisation of a formula is usually an underestimated problem, as the satisfaction/robustness heavily depends on these parameters. This problem has been partially tacked e.g. in [10, 41] for deterministic models, but never for stochastic ones, to authors' knowledge.

In particular, we consider Formula (4.4), and optimise the temporal delays T_1 in the range $[100, 1700]$ and T_2 in the range $[100, 1000]$. This can be seen as an attempt to learn the best bounds on the oscillatory period, through the filter of the logical specification of oscillations of Formula (4.4). Inspecting the structure of the formula, we can observe that it is the conjunction of two temporal properties, one containing the parameter T_1 ($\varphi_1 = \mathbf{G}_{[0,T]}((X_i >$

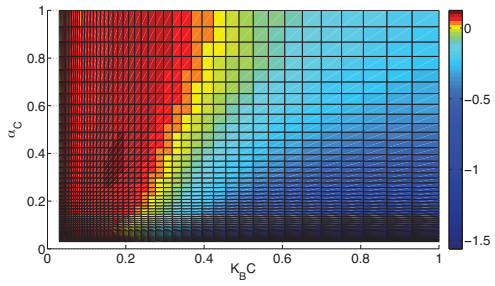


Figure 9: Part of the emulated robustness function in the optimisation of α_C and K_{BC} for the I1FFL example. The colour corresponds to the value of the average robustness in agreement with the legend on the right of the plot.

T_1, T_2 **optimisation**	T_1		T_2	
	Median	Range	Median	Range
Parameter	903.65	[895.89, 947.56]	675.28.	[658.73, 688.11]
Average robustness	13.72	[11.45, 15.15]	0.0113	[0.0112, 0.0115]
Probability satisfaction	0.882	[0.874, 0.883]	0.689	[0.685, 0.695]
n rob. fcn evaluations	23	[22, 23]	20	[17, 22]
n simulation runs	2300	[2200, 2300]	2000	[1700, 22]
Rob. fcn eval. time (sec.)	2534	[2381, 2543]	2189	[1848, 2727]
Optim. time (sec.)	2537	[2385, 2546]	2189	[1849, 2729]

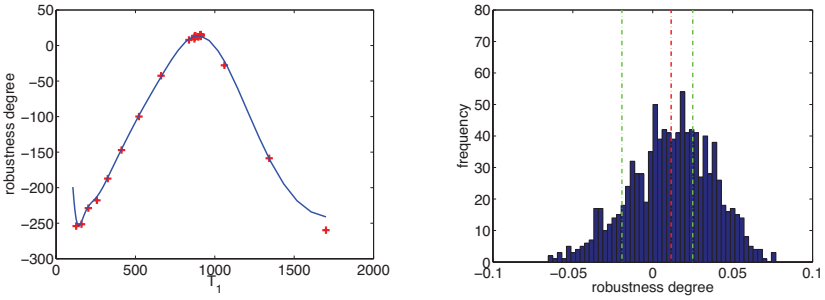
Table 4: Statistics of the results of 10 experiments to optimise the parameter T_1 in the range $[100, 1700]$ and of 10 experiments to optimise T_2 in the range $[100, 1000]$. We report the median and the range of the optimal parameter, the average robustness, the probability satisfaction, the number of robustness function evaluations (n rob. fcn evaluations), the total number of simulation runs, and the time, in seconds, of the robustness function evaluations (Rob. fcn eval. time) and the optimisation (Optim. time). The number of runs for each evaluation, i.e. each SMC, is 100.

$k_{high}) \rightarrow \mathbf{F}_{[T_1,T_1+h]}(X_i < k_{low})))$ and the other one containing the parameter T_2 ($\varphi_2 = \mathbf{G}_{[0,T]}((X_i < k_{low}) \rightarrow \mathbf{F}_{[T_2,T_2+h]}(X_i > k_{high}))$). We can exploit this structure and perform the optimisation of φ_1 and φ_2 separately, as model parameters are fixed and the total robustness degree is just the minimum of those of φ_1 and φ_2. In these formulas, we also keep T and h constant. T is related to the length of the signal we are observing, while h governs the length of the error we allow on the half period. In particular, observe that maximising h is meaningless: it is easy to show that the (average) robustness score will increase monotonically with h, so that the optimisation will always pick the upper bound. Hence, we fix $h = 350$. We decide to study the oscillation of the unknown gene X, i.e we set $i = X$.

The results of the optimisation of the two parameters are reported in Table 4, while in Figure 10(a) we show the emulated robustness function of T_1. The optimisation algorithm was set as for the I1FFL case, save for the number of initial observations, set to 12. The parameters of the model are fixed to those shown in the caption of Figure 4. As we can see from the robustness distribution in Figure 10 right, we find parameters increasing the average robustness score of the formula to a positive value. From the emulated robustness function of parameter T_1 (Figure 10 left), we can note that T_1 is identified precisely: the robustness function has a strict maximum. Finally, we observe that if we had taken the search domains of T_1 or T_2 to be much larger, we would have found additional maxima of similar height, corresponding to values of T_1 increased by one period, two periods, and so on.

6. Related Work

The *system design problem* consists in tuning properly the uncertain parameters of a biological model in order to reproduce the behavioural properties observed



(a) The estimated robustness function

(b) Robustness distribution after the optimisation

Figure 10: The emulated robustness function in the optimisation of T_1 (left). The distribution of the robustness score for $T_1 = 903.65$ and $T_2 = 675.28$. Average robustness is 0.0113 and estimated satisfaction probability is 0.689. (right)

in the experimental data. Temporal logic [23] is a very intuitive specification language to express formally the behavioural property emerging in a complex biological system. The success of this simple, but powerfully expressive formalism has led the computer aided verification community to introduce several important extensions of temporal logic, such as Metric Interval Temporal Logic [25], Signal Temporal Logic (STL) [24] and Temporal Frequency Logic [48], to deal with dense-time, real-valued signals and time-frequency patterns, respectively. In the last years, there was a great effort to develop new and efficient techniques to guide the system design of biological models with uncertain parameters, using the verification techniques available for the aforementioned behavioural specification languages.

For example, Batt et al. in [49] showed that the behaviour of a genetic regulatory network can be approximated with a piecewise multi-affine system. This class of models exhibits useful convexity properties which allows to compute a conservative finite-state automaton abstraction where the states represent the reachable sets in the form of hyper-rectangles in which the original state-space is partitioned and the transitions among the states characterise an over-approximation of the flows among the reachable sets. In a model with uncertain parameters, several different finite-state automata can be derived starting from different parameter sets. The model checking of a temporal logic formula guides the selection of the parameters sets for which the conservative abstraction and so the model will violate the problem of interest. Recently, other authors have extended this approach [50, 51], by introducing an optimal approximation algorithm, to biological models with generic nonlinear differential equations such as the cardiac cell excitability [50] and the bone remodelling [51] case studies. However, this approach cannot be applied to stochastic models and the use of an over-approximation abstracts away important timing relations, resulting in the selection of very coarse parameter sets.

An alternative approach is to use under-approximation techniques such as simulation or sampling. Following this direction, in the last years, there was a great scientific effort to enrich the classical qualitative semantics of temporal logic or *satisfiability* (yes/no answer for the formula satisfaction of a trajectory) with more powerful and useful notions of quantitative semantics [52, 9, 8, 10, 53, 54, 55] (or *robustness degree*), providing a real value measuring the level of satisfaction or violation for a trajectory of the property of interest. Several tools, such as BIOCHAM [56], S-TaLiRo [53] and Breach [27], are now available to perform robustness analysis on the time series collected in wet-lab experiments or produced by simulation-based techniques. The robustness degree have been successfully employed in the analysis of ODE-based biological models, to tune the parameters discriminating the behaviours observed experimentally. In [30], Donzé et al. proposed a multi-step analysis, where they adopt STL to express dynamical properties and they use robustness and sensitivity analysis to sample efficiently the parameter space, searching for feasible regions in which the model exhibit a particular behaviour. In [47] the authors proposed a new approach, based on robustness degree, for the design of a synthetic biological circuit whose behaviour is specified in terms of signal temporal logic (STL) formulae. However, in all the aforementioned cases, stocasticity is not taken into account. It is worth mentioning that all these simlation based techniques are based on the (approximate) solution of reachability problems for non-linear ODE systems, which can be tackled with Bayesian optimisation techniques presented in this paper [57].

With regard to the stochastic models, the satisfiability analysis has been considered as a discriminating criterion to tune the parameters in the design process using both simulation-based statistical approximated methods [6, 58] and symbolic methods [59, 60]. Lanotte et. al [59] showed that for parametric probabilistic transition systems (or discrete time Markov chains) the problem of finding (symbolically) an instance of parameter values for a reachability property to be satisfied is equivalent to the problem of finding the roots of a general polynomial and so it is generally undecidable if proper restrictions are not considered. In [61], the authors proposed a combined approach of a model checker together with a genetic algorithm to guide the parameter-estimation process by reducing the distance between the desired behaviour and the actual behaviour. The work [62] concerns with the parameter-synthesis problem, using symbolic methods, for parametric continuos time markov chains and time-bounded properties. Also in this case the problem results to be generally undecidable and the authors proposed [62] an approximation method that provide a solution in most cases. Another related work in this sense is that of [63], where authors compute exactly upper and lower bounds on the satisfaction probability within a given region of the parameter space. In [64], instead, the authors propose a method to statistically estimate the satisfaction probability as a function of parameters based on Gaussian Process classification. All these approaches are designed to work with specific classes of stochastic models and to the best of our knowledge, we are not aware of approaches using the robustness degree.

7. Conclusion

Discussion. In this paper we investigated a notion of robustness of behaviours of stochastic models, extending the robustness degree of STL formulae in a probabilistic setting. Discussing three case studies, a bistable model, the Repressilator, and a feed-forward-loop, we showed that the distribution of the robustness degree of a formula provides valuable information that is not captured by the satisfaction probability alone. Furthermore, its average can be used to enforce robust behaviours by optimising it.

Such optimisation is carried out using state-of-the-art optimisation algorithms coming from reinforcement learning, which emulate the true function from just few samples, and perform very well in a simulation based scenario. Remarkably, the proposed approach to evaluate robustness and to system design can be applied both to CTMC and to SHA models. We also briefly considered the problem of learning the most effective parameters of a given formula maximising the robustness score similarly to [65, 66].

This is a first step towards the ambitious goal of finding machine learning procedures to learn the emergent behaviours (described as temporal logic formulae) from models and from experimental data. Many problems need to be faced to achieve this goal, like how to learn formula structure, how to avoid overfitting (with respect to both formula structure and parameters), how to deal with the curse of dimensionality afflicting GP-UCB and other optimisation algorithms.

Future Work. The present work uses advanced machine learning concepts to address core problems in formal modelling; this is a relatively new line of work [67, 58, 57, 64, 68, 65, 66, 69, 70, 55] which opens significant new avenues for further research. From the practical point of view, more extensive testing and an efficient and robust implementation (exploiting some of the possible parallelisms e.g. in SMC) will be important for the tool to be adopted. From the theoretical perspective, we plan to use multi-objective optimisation to find good parametrisation for conflicting objectives. Another interesting direction is to combine the design problem with the inference problem, which has recently been addressed for a number of continuous time stochastic systems [19]; this would open the possibility of addressing the control problem for such systems, simultaneously inferring the state of the system and designing the optimal input to lead it to a desired state.

Acknowledgements. We thank Andrea Ocone for having kindly shared with us the matlab code of the *O. Tauri* circadian clock model. Work partially supported by the EU-FET project QUANTICOL (nr. 600708), the FRA-UniTS, the ERC under grant MLCS 306999, the Austrian FFG project HARMONIA (nr. 845631), the e-COST project ARVI (ICT COST Action IC1402), and by the German Research Council (DFG) as part of the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University and the Transregional Collaborative Research Center SFB/TR 14 AVACS.

References

[1] A. Elowitz, M.B.and Levine, E. Siggia, P. Swain, Stochastic gene expression in a single cell, Science 297 (5584) (2002) 1183.

[2] R. Durett, Essentials of stochastic processes, Springer, 2012. doi:10.1007/978-1-4614-3615-7.

[3] M. L. Bujorianu, J. Lygeros, M. C. Bujorianu, Bisimulation for General Stochastic Hybrid Systems, in: Proc. of HSCC 2005: the 8th International Workshop on Hybrid Systems: Computation and Control, Vol. 3414 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 198–214. doi:10.1007/978-3-540-31954-2_13.

[4] C. Baier, E. Clarke, V. Hartonas-Garmhausen, M. Kwiatkowska, M. Ryan, Symbolic Model Checking for Probabilistic Processes, in: Proc. of ICALP '97: the 24th International Colloquium on Automata, Languages and Programming, Bologna, Italy, July 7-11, Vol. 1256 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1997, pp. 430–440. doi:10.1007/3-540-63165-8_199.

[5] C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen, Model-checking Algorithms for Continuous-Time Markov Chains, IEEE Trans. Softw. Eng. 29 (6) (2003) 524–541. doi:10.1109/TSE.2003.1205180.

[6] M. Kwiatkowska, G. Norman, D. Parker, Probabilistic symbolic model checking with PRISM: a hybrid approach, Int. J. Softw. Tools Technol. Transf. 6 (2) (2004) 128–142. doi:10.1007/s10009-004-0140-2.

[7] T. Chen, M. Diciolla, M. Kwiatkowska, A. Mereacre, Time-Bounded verification of CTMCs against Real-Time Specifications, in: Proc. of FORMATS 2011: the 9th International Conference on Formal Modeling and Analysis of Timed Systems, Aalborg, Denmark, September 21–23, Vol. 6919 of Lecture Notes in Computer Science, Springer-Verlag, 2011, pp. 26–42. doi:10.1007/978-3-642-24310-3_4.

[8] A. Donzé, O. Maler, Robust satisfaction of temporal logic over real-valued signals, in: Proc. of FORMATS 2010: the 8th International Conference on Formal Modeling and Analysis of Timed Systems, Klosterneuburg, Austria, September 8–10, Vol. 6246 of Lecture Notes in Computer Science, Springer-Verlag, 2010, pp. 92–106. doi:10.1007/978-3-642-15297-9_9.

[9] G. E. Fainekos, G. J. Pappas, Robustness of temporal logic specifications for continuous-time signals, Theor. Comput. Sci. 410 (42) (2009) 4262–4291. doi:10.1016/j.tcs.2009.06.021.

[10] A. Rizk, G. Batt, F. Fages, S. Soliman, On a Continuous Degree of Satisfaction of Temporal Logic Formulae with Applications to Systems Biology, in: Proc. of CMSB 2008: the 6th International Conference on Computational

Methods in Systems Biology, Rostock, Germany, October 12–15, Vol. 5307 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 251–268. doi:10.1007/978-3-540-88562-7_19.

[11] H. L. S. Younes, M. Z. Kwiatkowska, G. Norman, D. Parker, Numerical vs. statistical probabilistic model checking: An empirical study, in: Proc. of 2004: the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Barcelona, Spain, March 29 - April 2, Vol. 2988, Springer-Verlag, 2004, pp. 46–60. doi:10.1007/978-3-540-24730-2_4.

[12] N. Srinivas, A. Krause, S. M. Kakade, M. W. Seeger, Information-theoretic regret bounds for gaussian process optimization in the bandit setting, IEEE Transactions on Information Theory 58 (5) (2012) 3250–3265. doi:10.1109/TIT.2011.2182033.

[13] P. Billingsley, Probability and measure, Wiley, 2012.

[14] L. Bortolussi, J. Hillston, D. Latella, M. Massink, Continuous Approximation of Collective Systems Behaviour: a Tutorial, Performance Evaluation 70 (5) (2013) 317–349. doi:10.1016/j.peva.2013.01.001.

[15] D. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. of Physical Chemistry 81 (25). doi:10.1021/j100540a008.

[16] M. Davis, Markov Models and Optimization, Chapman & Hall, 1993.

[17] L. Bortolussi, A. Policriti, Hybrid) automata and (stochastic) programs. The hybrid automata lattice of a stochastic program, J. Log. Comput. 23 (4) (2013) 761–798. doi:10.1093/logcom/exr045.

[18] L. Bortolussi, Limit Behavior of the Hybrid Approximation of Stochastic Process Algebras, in: Proc. of ASMTA 2010: the 17th International Conferenceon Analytical and Stochastic Modeling Techniques and Applications, Cardiff, UK, June 14-16, Vol. 6148 of Lecture Notes in Computer Science, Springer-Verlag, 2010, pp. 367–381. doi:10.1007/978-3-642-13568-2_26.

[19] A. Ocone, A. J. Millar, G. Sanguinetti, Hybrid regulatory models: a statistically tractable approach to model regulatory network dynamics, Bioinformatics 29 (7) (2013) 910–916. doi:10.1093/bioinformatics/btt06.

[20] M. Opper, A. Ruttor, G. Sanguinetti, Approximate inference in continuous time Gaussian-Jump processes, in: Proc. of NIPS 2010: the 4th Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 6–9 December, 2010, pp. 1831–1839.

[21] B. K. ksendal, Stochastic Differential Equations: An Introduction with Applications., Springer., 2003.

[22] P. Billingsley, Convergence of probability measures, Wiley, 1999.

[23] A. Pnueli, The temporal logic of programs, IEEE Annual Symposium on Foundations of Computer Science (1977) 46–57doi:10.1109/SFCS.1977.32.

[24] O. Maler, D. Nickovic, Monitoring Temporal Properties of Continuous Signals, in: Proc. of FORMATS-FTRTFT 2004: the Joint International Conferences on Formal Modeling and Analysis of Timed Systmes and Formal Techniques in Real-Time and Fault -Tolerant Systems, Grenoble, France, September 22-24, Vol. 3253 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 152–166. doi:10.1007/978-3-540-30206-3_12.

[25] R. Alur, T. Feder, T. A. Henzinger, The Benefits of Relaxing Punctuality, J. ACM 43 (1) (1996) 116–146. doi:10.1145/227595.227602.

[26] A. Donzé, T. Ferrer, O. Maler, Efficient Robust Monitoring for STL, in: Proc. of CAV 2013: the 25th International Conference on Computer Aided Verification, Saint Petersburg, Russia, July 13-19, Vol. 8044 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 264–279. doi:10.1007/978-3-642-39799-8_19.

[27] A. Donzé, Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems, in: Proc. of CAV 2010: the 22nd International Conference on Computer Aided Verification, Edinburgh, UK, Vol. 6174 of Lecture Notes in Computer Science, Springer-Verlag, 2010, pp. 167–170. doi:10.1007/978-3-642-14295-6_17.

[28] K. D. Jones, V. Konrad, D. Nickovic, Analog property checkers: a DDR2 case study, Formal Methods in System Design 36 (2) (2010) 114–130. doi:10.1007/s10703-009-0085-x.

[29] A. Donzé, G. Clermont, C. Langmead, Parameter Synthesis in Nonlinear Dynamical Systems: Application to Systems Biology, Journal of Computational Biology 17 (3) (2010) 325–336. doi:10.1007/978-3-642-02008-7_11.

[30] A. Donzé, E. Fanchon, L. M. Gattepaille, O. Maler, P. Tracqui, Robustness analysis and behavior discrimination in enzymatic reaction networks, PLoS One 6 (9) (2011) e24246. doi:10.1371/journal.pone.0024246.

[31] M. Komorowski, M. J. Costa, D. A. Rand, M. P. Stumpf, Sensitivity, robustness, and identifiability in stochastic chemical kinetics models, PNAS USA 108 (21) (2011) 8645–8650. doi:10.1073/pnas.1015814108.

[32] O. Kallenberg, Foundations of Modern Probability, Springer, 2010.

[33] P. Collins, Computability and representations of the zero set, Electronic Notes in Theoretical Computer Science 221 (2008) 37–43. doi:10.1016/j.entcs.2008.12.005.

[34] D. Richardson, How to recognize zero, Journal of Symbolic Computation 24 (6) (1997) 627–645.

[35] A. Degasperi, S. Gilmore, Sensitivity Analysis of Stochastic Models of Bistable Biochemical Reactions, in: Formal Methods for Computational Systems Biology, Vol. 5016 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 1–20. doi:10.1007/978-3-540-68894-5_1.

[36] R. Gunawan, Y. Cao, L. Petzold, F. Doyle III, Sensitivity analysis of discrete stochastic systems, Biophysical Journal 88 (4) (2005) 2530. doi:10.1529/biophysj.104.053405.

[37] U. Alon, An introduction to systems biology: design principles of biological circuits, Chapman & Hall/CRC, 2007.

[38] M. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators, Nature 403 (2000) 335–338. doi:10.1038/35002125.

[39] L. Bortolussi, A. Policriti, Hybrid Approximation of Stochastic Process Algebras for Systems Biology World Congress, in: Proc. of IFAC WC 2008: The 17th World Congress of the International Federation of Automatic Control, COEX, Korea, South, Vol. 17, 2008, pp. 12599–12606. doi:10.3182/20080706-5-KR-1001.02132.

[40] L. Bortolussi, A. Policriti, Hybrid Dynamics of Stochastic Programs, Theoretical Computer Science 411 (20) (2010) 2052–2077. doi:10.1016/j.tcs.2010.02.008.

[41] X. Jin, A. Donzé, J. V. Deshmukh, S. A. Seshia, Mining Requirements from Closed-loop Control Models, in: Proc. of HSCC 2013: the 16th International Conference on Hybrid Systems: Computation and Control, ACM, 2013, pp. 43–52. doi:10.1145/2461328.2461337.

[42] R. Grosu, S. Smolka, F. Corradini, A. Wasilewska, E. Entcheva, E. Bartocci, Learning and detecting emergent behavior in networks of cardiac myocytes, Communications of the ACM 52 (3) (2009) 97–105.

[43] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[44] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[45] M. Kennedy, A. O'Hagan, Bayesian Calibration of Computer Models, Journal of the Royal Stat. Soc. Ser. B 63 (3) (2001) 425–464. doi:10.1111/1467-9868.00294.

[46] A. Georgoulas, A. Clark, A. Ocone, S. Gilmore, G. Sanguinetti, A subsystems approach for parameter estimation of ode models of hybrid systems, in: Proc. of HSB 2012: the First International Workshop on Hybrid Systems and Biology, Vol. 92 of EPTCS, 2012, pp. 30–41. doi:10.4204/EPTCS.92.3.

[47] E. Bartocci, L. Bortolussi, L. Nenzi, A Temporal Logic Approach to Modular Design of Synthetic Biological Circuits, in: Proc. of CMSB 2013: the 11th International Conference on Computational Methods in Systems Biology, IST Austria, Klosterneuburg, Austria, September 23-25, Vol. 8130 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 164–178. doi:10.1007/978-3-642-39176-7.

[48] A. Donzé, O. Maler, E. Bartocci, D. Nickovic, R. Grosu, S. A. Smolka, On Temporal Logic and Signal Processing, in: S. Chakraborty, M. Mukund (Eds.), Proc. of ATVA 2012: 10th International Symposium on Automated Technology for Verification and Analysis, Thiruvananthapuram, India, October 3-6, Vol. 7561 of Lecture Notes in Computer Science, Springer-Verlag, 2012, pp. 92–106. doi:10.1007/978-3-642-33386-6_9.

[49] G. Batt, B. Yordanov, R. Weiss, C. Belta, Robustness analysis and tuning of synthetic gene networks, Bioinformatics 23 (18) (2007) 2415–2422. doi:10.1093/bioinformatics/btm362.

[50] R. Grosu, G. Batt, F. Fenton, J. Glimm, C. Le Guernic, S. Smolka, E. Bartocci, From Cardiac Cells to Genetic Regulatory Networks, in: Proc. of CAV 2011: the 23rd International Conference on Computer Aided Verification, Vol. 6806 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, pp. 396–411. doi:10.1007/978-3-642-22110-1_31.

[51] E. Bartocci, P. Liò, E. Merelli, N. Paoletti, Multiple Verification in Complex Biological Systems: The Bone Remodelling Case Study, T. Comp. Sys. Biology 14 (2012) 53–76. doi:10.1007/978-3-642-35524-0_3.

[52] G. Fainekos, G. Pappas, Robust Sampling for MITL Specifications, in: Proc. of FORMATS 2007: the 5th International Conference on Formal Modeling and Analysis of Timed Systems, Vol. 8044 of Lecture Notes in Computer Science, Springer-Verlag, 2007, pp. 264–279. doi:10.1007/978-3-540-75454-1_12.

[53] Y. Annapureddy, C. Liu, G. Fainekos, S. Sankaranarayanan, S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems, in: Proc. of TACAS 2011: the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Vol. 6605 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, pp. 254–257. doi:10.1007/978-3-642-19835-9_21.

[54] E. Aydin Gol, E. Bartocci, C. Belta, A formal methods approach to pattern synthesis in reaction diffusion systems, in: Proc. of CDC 2014: the 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 108–113.

[55] I. Haghighi, A. Jones, J. Z. Kong, E. Bartocci, R. Grosu, C. Belta, Spatel: A novel spatial-temporal logic and its applications to networked systems, in: In Proc of HSCC 2015: the 18th International Conference on Hybrid Systems: Computation and Control, ACM, 2015, p. to appear.

[56] L. Calzone, F. Fages, S. Soliman, BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge, Bioinformatics 22 (2006) 1805–1807. doi:10.1093/bioinformatics/btl172.

[57] L. Bortolussi, G. Sanguinetti, A statistical approach for computing reachability of non-linear and stochastic dynamical systems, in: Proc. of QEST 2014: the 11th International Conference on Quantitative Evaluation of Systems, Vol. 8657 of Lecture Notes in Computer Science, Springer, 2014, pp. 41–56.

[58] L. Bortolussi, G. Sanguinetti, Learning and Designing Stochastic Processes from Logical Constraints, in: Proc. of QEST 2013: the 10th International Conference on Quantitative Evaluation of Systems, Buenos Aires, Argentina, August 27-30, Vol. 8054 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 89–105. doi:10.1007/978-3-642-40196-1.

[59] R. Lanotte, A. Maggiolo-Schettini, A. Troina, Parametric Probabilistic Transition Systems for System Design and Analysis, Form. Asp. Comput. 19 (2007) 93–109. doi:10.1007/s00165-006-0015-2.

[60] E. Bartocci, R. Grosu, P. Katsaros, C. Ramakrishnan, S. A. Smolka, Model Repair for Probabilistic Systems, in: Proc. of TACAS 2011: the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Vol. 6605 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, pp. 326–340. doi:10.1007/978-3-642-19835-9_30.

[61] R. Donaldson, D. Gilbert, A Model Checking Approach to the Parameter Estimation of Biochemical Pathways, in: Proc. of CMSB 2008: the 6th International Conference on Computational Systems Biology, Rostock, Germany, October 12-15, Vol. 5307 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 269–287. doi:10.1007/978-3-540-88562-7_20.

[62] T. Han, J. Katoen, A. Mereacre, Approximate Parameter Synthesis for Probabilistic Time-Bounded Reachability, in: Proc. of RTSS 2008: the 29th IEEE Real-Time Systems Symposium, 2008, pp. 173–182. doi:10.1109/RTSS.2008.19.

[63] L. Brim, M. Ceska, S. Drazan, D. Šafránek, Exploring Parameter Space of Stochastic Biochemical Systems using Quantitative Model Checking, in: Proc. of CAV 2013: the 25th International Conference on Computer Aided Verification, Saint Petersburg, Russia, July 13-19, Vol. 8044 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 107–123. doi:10.1007/978-3-642-39799-8_7.

[64] L. Bortolussi, D. Milios, G. Sanguinetti, Smoothed model checking for uncertain CTMC, in: arXiv preprint 1402.1450, 2014.

[65] E. Bartocci, L. Bortolussi, S. Sanguinetti, Data-driven statistical learning of temporal logic properties, in: Proc. of FORMATS 2014: the 12th International Conference on Formal Modeling and Analysis of Timed Systems, Vol. 8711 of LNCS, 2014, pp. 23–37. doi:10.1007/978-3-319-10512-3.

[66] S. Bufo, E. Bartocci, G. Sanguinetti, M. Borelli, U. Lucangelo, L. Bortolussi, Temporal logic based monitoring of assisted ventilation in intensive care patients, in: B. Steffen, T. Margaria (Eds.), Proc. of ISoLA 2014: 6th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, Vol. 8803 of LNCS, 2014, pp. 391–403. doi:10.1007/978-3-662-45231-8.

[67] L. Calzone, N. Chabrier-Rivier, F. Fages, S. Soliman, Machine learning biochemical networks from temporal logic properties, Transactions on Computational Systems Biology VI 4220 (2006) 68–94. doi:http://dx.doi.org/10.1007/11880646_4.

[68] Z. Kong, A. Jones, A. Medina Ayala, E. Aydin Gol, C. Belta, Temporal logic inference for classification and prediction from data, in: Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control, ACM, 2014, pp. 273–282.

[69] A. Georgoulas, J. Hillston, D. Milios, G. Sanguinetti, Probabilistic programming process algebra, in: Proc. of QEST: the 1th International Conference on Quantitative Evaluation of Systems, Vol. 8657 of LNCS, Springer, 2014, pp. 249–264. doi:10.1007/978-3-319-10696-0_21.

[70] A. Legay, S. Sedwards, Statistical abstraction boosts design and test efficiency of evolving critical systems, in: Proc. of ISoLA, Part I: 6th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation Technologies for Mastering Change, Vol. 8802 of LNCS, Springer, 2014, pp. 4–25. doi:10.1007/978-3-662-45234-9_2.

Appendix A. Proofs

In this appendix, we present the proof of Theorem 6, about the measurability of the robustness score.

Appendix A.1. Proof of Theorem 6

We first note here that a sequence of functions $\mathbf{x}^n \in \mathcal{D}([0,T],E)$ converges to $\mathbf{x} \in \mathcal{D}([0,T],E)$ if and only if there is a sequence of time-wiggle functions $\omega^n \in \mathcal{I}_T$ satisfying $\sup_{t\in[0,T]} \|\omega^n(t) - t\| \to 0$ and $\sup_{t\in[0,T]} \|\mathbf{x}^n(t) - \mathbf{x}(\omega^n(t))\| \to 0$.

We also recall an important property of cadlag functions, which essentially allows us to approximate any cadlag function with a step function.

Lemma 8. *Let $\mathbf{x} \in \mathcal{D}([0,T],E)$. For each $\varepsilon > 0$, there exists a finite grid $\mathbb{T} = \{0 = t_0 < t_1 < t_2 < \ldots < t_k = T\}$ such that for each $i = 0,\ldots,k-1$*

$$\sup_{t,s\in[t_i,t_{i+1})} \|\mathbf{x}(t) - \mathbf{x}(s)\| < \varepsilon$$

Proof: See [22]. □

In order to prove theorem 6, we need some preliminary results about measurability of the functions involved in the definition of the robustness score ρ. As we always deal with time bounded signals, we start by considering ρ, for a given formula φ, as a transducer of real-valued signals, i.e. as a functional from $\mathcal{D}([0,T],E)$ to itself. To be more precise, we need to take into account that a STL formula looks T_φ time units into the future[8], hence $\hat{\mathtt{R}}_\varphi : \mathcal{D}([0,T],E) \to \mathcal{D}([0,T-T_\varphi],E)$ (where $\hat{\mathtt{R}}$ denotes the functional between cadlag function spaces associated with the robustness score ρ).

Lemma 9. *Consider the space $\mathcal{D}([0,T],E)$ of cadlag functions. The following functionals are measurable, with respect to the Borel σ-algebra induced by the Skorokhod topology (or the product σ-algebra).*

- *a) Forward time shifts: $\delta_a : \mathcal{D}([0,T],E) \to \mathcal{D}([0,T-\hat{T}],E)$, defined by $\delta_a(\mathbf{x})(t) = \mathbf{x}(t+a)$, for $\hat{T} > a$. Similarly for backward time shifts.*
- *b) Pointwise maximum (and minimum): $\max : \mathcal{D}([0,T],E) \times \mathcal{D}([0,T],E) \to \mathcal{D}([0,T],E)$.*
- *c) Maximum (and minimum) over a dense interval in the future: $SUP_{[T_1,T_2]} : \mathcal{D}([0,T),E) \to \mathcal{D}([0,T-T_2),E)$, defined[9] as $SUP_{[T_1,T_2]}(\mathbf{x})(t) = \sup_{\tau \in t \oplus [T_1,T_2]} x(\tau)$, for $0 \le T_1 < T_2 < T$ fixed.*

Proof: To prove points a) and b), we rely on the fact that the Borel σ-algebra in $\mathcal{D}([0,T],E)$ is generated by the following collection of sets (forming a π-system): $\mathcal{A}_T = \{\pi^{-1}_{t_1,\dots,t_k}(\mathbf{h}) \mid k \in \mathbb{N}, \mathbf{h} \in E^k, 0 \le t_1 < \dots < t_k \le T\}$, where $\pi_{t_1,\dots,t_k}$ is the projection on E^k, which is measurable [22].

a) We just need to prove that $\delta_a^{-1}(\pi^{-1}_{t_1,\dots,t_k}(\mathbf{h}))$ is measurable for each $\pi^{-1}_{t_1,\dots,t_k}(\mathbf{h}) \in \mathcal{A}_{T-\hat{T}}$. But $\delta_a^{-1}(\pi^{-1}_{t_1,\dots,t_k}(\mathbf{h})) = \pi^{-1}_{t_1+a,\dots,t_k+a}(\mathbf{h})$.

b) Denote with $\mathbf{b}$ a boolean tuple of length k and with $\bar{\mathbf{b}}$ its element-wise boolean complement. Define $\pi^{-1}_{t_1,\dots,t_k,\mathbf{b}}(\mathbf{h})$ to be the set $\bigcap_{i:\mathbf{b}[i]\ true} \pi^{-1}_{t_i}(\{h_i\}) \cap \bigcap_{i:\mathbf{b}[i]\ false} \pi^{-1}_{t_i}((-\infty,h_i))$ which is measurable (by measurability of finite dimensional projections). It holds that

$$\max{}^{-1}(\pi^{-1}_{t_1,\dots,t_k}(\mathbf{h})) = \bigcup_{\mathbf{b}} \pi^{-1}_{t_1,\dots,t_k,\mathbf{b}}(\mathbf{h}) \times \pi^{-1}_{t_1,\dots,t_k,\bar{\mathbf{b}}}(\mathbf{h}),$$

where the union is taken over all possible boolean tuples of length k. Hence, the set $\max^{-1}(\pi^{-1}_{t_1,\dots,t_k}(\mathbf{h}))$ is measurable in the product σ-algebra.

c) We prove this only for the maximum, as the result for the minimum follows similarly.

[8] T_φ is defined recursively by $T_\mu = 0$, $T_{\varphi_1 \wedge \varphi_2} = \max\{T_{\varphi_1}, T_{\varphi_2}\}$, $T_{\neg\varphi} = T_\varphi$, and $T_{\varphi_1 \mathcal{U}_{[T_1,T_2]} \varphi_2} = \max\{T_{\varphi_1}, T_{\varphi_2}\} + T_2$.

[9] Recall that $\oplus$ is the Minkowski sum of two sets, $A \oplus B = \{a + b \mid a \in A, b \in B\}$

For each n, define the finite grid $\mathbb{T}_n = \{T_1, T_1 + \delta_n, \ldots, T_2\}$, where $\delta_n = \frac{T_2 - T_1}{n}$. Furthermore, let

$$SUP^n_{[T_1,T_2]}(\mathbf{x})(t) = \max_{\tau \in t \oplus \mathbb{T}_n} x(\tau).$$

Fix $\mathbf{x} \in \mathcal{D}([0,T),E)$, and call $g^n(t) = SUP^n_{[T_1,T_2]}(\mathbf{x})(t)$ and $g(t) = SUP_{[T_1,T_2]}(\mathbf{x})(t)$. We will prove that $g^n \to g$ in the Skorokhod metrics. By additionally showing the measurability of $SUP^n_{[T_1,T_2]}$ for each n, we can rely on a standard result for measurable functions, i.e. that the pointwise limit of a sequence of measurable functions is measurable [13], to prove the measurability of $SUP_{[T_1,T_2]}$.

Measurability of $SUP^n_{[T_1,T_2]}$. Call $\mathbf{x}_i(t) = \mathbf{x}(t + i \cdot \delta_n)$ and observe that $SUP^n_{[T_1,T_2]}(\mathbf{x})(t) = \max\{\mathbf{x}_0(t), \ldots, \mathbf{x}_n(t)\}$. The measurability of $SUP^n_{[T_1,T_2]}$ follows from points a) and b) above (extending point b to the maximum of n functions by induction is straightforward).

Convergence of g^n to g. Fix $\varepsilon > 0$ sufficiently small and use Lemma 8 for $\mathbf{x}$ on $[0,T]$ to find a grid $\mathbb{T}_{\mathbf{x},\varepsilon} = \{t_0, t_1, \ldots, t_h\}$ in $[0,T]$ satisfying the condition in the Lemma for the given ε. Fix a closed interval $I \subseteq [0,T]$ and consider any finite set $\hat{\mathbb{T}}_I$ that contains one point $\hat{t} \in [t_j, t_{j+1}) \cap I$ for each $[t_j, t_{j+1}) \cap I \neq \emptyset$. By splitting the supremum in each $[t_j, t_{j+1})$, Lemma 8 implies that

$$\sup_{\tau \in I} \mathbf{x}(\tau) \leq \max_{\hat{t} \in \hat{\mathbb{T}}} \mathbf{x}(\hat{t}) + \varepsilon. \tag{A.1}$$

Now let $\delta_{\mathbb{T}_{\mathbf{x},\varepsilon}} = \min_{t_j \in \mathbb{T}_{\mathbf{x},\varepsilon}} (t_{j+1} - t_j)$ be the smallest step size of $\mathbb{T}_{\mathbf{x},\varepsilon}$ and choose n_0 such that $\delta_{n_0} = \frac{T_2 - T_1}{n_0} < \delta_{\mathbb{T}_{\mathbf{x},\varepsilon}}/2$. It then follows that $t \oplus \mathbb{T}_n$ contains at least one point in each of the intervals $[t_j, t_{j+1})$ of $\mathbb{T}_{\mathbf{x},\varepsilon}$, hence (A.1) implies (uniformly in t) that

$$g(t) - \varepsilon < g^n(t) < g(t).$$

Concluding, we found n_0 such that, for all $n > n_0$, $d_T(g^n, g) < \varepsilon$, which implies the convergence of g^n to g in $\mathcal{D}([0,T],E)$. □

Lemma 10. *Let φ a STL formula. The functional $\hat{\mathtt{R}}_\varphi$ associated with it, $\hat{\mathtt{R}}_\varphi : \mathcal{D}([0,T],E) \to \mathcal{D}([0,T-T_\varphi],E)$, is measurable.*

Proof: We proceed by structural induction on the formula φ.

Atomic predicate μ. Let μ be defined by the function $y(\mathbf{x}[t])$, required to be at least continuous. As the pointwise extension to $\mathcal{D}([0,T],E)$ of a continuous function is a continuous functional, $\hat{\mathtt{R}}_\mu$ is measurable.

Negation $\neg\varphi$. $\hat{\mathtt{R}}_{\neg\varphi} = -\hat{\mathtt{R}}_\varphi$ is measurable by inductive hypothesis (and continuity of the function $-x$).

Conjunction $\varphi_1 \wedge \varphi_2$. $\hat{\mathtt{R}}_{\varphi_1 \wedge \varphi_2} = \min\{\hat{\mathtt{R}}_{\varphi_1}, \hat{\mathtt{R}}_{\varphi_2}\}$, which is measurable in virtue of Lemma 9 b) and of the fact that measurability is preserved when composing measurable functions.

Eventually $\mathcal{F}_{[T_1,T_2]}\varphi$. $\hat{\mathtt{R}}_{\mathcal{F}_{[T_1,T_2]}\varphi} = SUP_{[T_1,T_2]}(\hat{\mathtt{R}}_{\varphi})$ is measurable in virtue of the measurability of $\hat{\mathtt{R}}_{\varphi}$ (structural induction) and of Lemma 9 c).

Globally $\mathcal{G}_{[T_1,T_2]}\varphi$. $\hat{\mathtt{R}}_{\mathcal{G}_{[T_1,T_2]}\varphi} = INF_{[T_1,T_2]}(\hat{\mathtt{R}}_{\varphi})$ is also measurable for the same reason above.

Until $\varphi_1 \mathcal{U}_{[T_1,T_2]} \varphi_2$. By definition, $\hat{\mathtt{R}}_{\varphi}(\mathbf{x})(t) = \hat{\mathtt{R}}_{\varphi_1 \mathcal{U}_{[T_1,T_2]}\varphi_2}(\mathbf{x})(t) = \sup_{t' \in t \oplus [T_1,T_2]}\{\min\{\hat{\mathtt{R}}_{\varphi_2}(\mathbf{x})(t'), \inf_{\tau \in [t,t']} \hat{\mathtt{R}}_{\varphi_2}$ Measurability follows from a technical argument similar to the one of Lemma 9 c), combined with the measurability of $\hat{\mathtt{R}}_{\varphi_1}(\mathbf{x})$ and $\hat{\mathtt{R}}_{\varphi_2}(\mathbf{x})$ by inductive hypothesis. More specifically, we need to construct a sequence of convergent approximations $R^n(\mathbf{x})(t)$ of $\hat{\mathtt{R}}_{\varphi_1 \mathcal{U}_{[T_1,T_2]}\varphi_2}(\mathbf{x})(t)$, computed on a discrete grid that shifts with t and t'. The discrete grids, like in Lemma 9 c), have to be independent from the specific $\mathbf{x}$, while the convergence must be uniform in t, but it can depend on x (we need to prove only pointwise convergence).

More specifically, let $\mathbb{T}^n = \{T_1, T_1 + \delta_n, \ldots, T_2\}$ for $\delta_n = (T_2 - T_1)/n$, and $\mathbb{T}_1^n = \{0, \delta_n^1, \ldots, T_2\}$, for $\delta_n^1 = T_2/n_1 \leq \delta_n$. Then the finite approximation of $\hat{\mathtt{R}}_{\varphi}(\mathbf{x})(t)$ is $\mathtt{R}^{\mathtt{n}}(\mathbf{x})(\mathtt{t}) = \max_{\mathtt{t}' \in \mathtt{A_n}}\{\min\{\hat{\mathtt{R}}_{\varphi_2}(\mathbf{x})(\mathtt{t}'), \min_{\tau \in \mathtt{B_n}} \hat{\mathtt{R}}_{\varphi_2}(\mathbf{x})(\tau)\}\}$, with $A_n = t \oplus T^n$ and $B_n = (t \oplus \mathbb{T}_1^n \cup t \ominus \mathbb{T}_1^n) \cap [t, t']$. The definition of B_n keeps into account the fact that both t and t' can vary, and it is needed to ensure that there is an n_0 such that, for $n \geq n_0$ we find for any t, t' points of B_n in a fixed but arbitrary finite partition of $[0, T]$. The measurability of $\mathtt{R}^{\mathtt{n}}$ as a functional follows from similar arguments than those in Lemma 9 c).

Now, fix $\varepsilon > 0$ and an element $\mathbf{x}$ in $\mathcal{D}([0,T], E)$. Construct a finite partitioning $\mathbb{T}_{\mathbf{x},\varepsilon}$ of $[0,T]$ such that Lemma 8 is satisfied both for $\hat{\mathtt{R}}_{\varphi_1}(\mathbf{x})$ and $\hat{\mathtt{R}}_{\varphi_2}(\mathbf{x})$ for $\varepsilon/2$. Then it is easy to check that, if n is such that δ_n is smaller than half the step $\delta_{\mathbb{T}_{\mathbf{x},\varepsilon}} = \min_{t_j \in \mathbb{T}_{\mathbf{x},\varepsilon}}(t_{j+1} - t_j)$ of $\mathbb{T}_{\mathbf{x},\varepsilon}$, then A_n and B_n contain points of each interval of $\mathbb{T}_{\mathbf{x},\varepsilon}$, so that the minimum approximates the infimum with an error bounded by $\varepsilon/2$ for each t and t', and the maximum cumulates another $\varepsilon/2$ approximation error with respect to the supremum. It follows that the distance between $\mathtt{R}^{\mathtt{n}}(\mathbf{x})(\mathtt{t})$ and $\hat{\mathtt{R}}_{\varphi}(\mathbf{x})(t)$ is no more than ε, uniformly in t, showing the convergence of the approximation in the Skorokhod metric. □

We can finally prove Theorem 6, which we report here for convenience. Recall that $\mathtt{R}_{\varphi} : \mathcal{D}([\mathtt{0}, \mathtt{T}], \mathtt{E}) \to \mathbb{R}$ is defined by $\mathtt{R}_{\varphi}(\mathbf{x}) = \rho(\varphi, \mathbf{x}, \mathtt{0})$. Given a formula φ, let $\mathcal{D} = \mathcal{D}([0,T], E)$ for some $T > T_{\varphi}$.

Theorem 6. For any STL formula φ, the functional $\mathtt{R}_{\varphi} : \mathcal{D} \to \mathbb{R}$ is measurable.

Proof: The theorem follows form the fact that $\mathtt{R}_{\varphi} = \pi_{\mathtt{0}} \circ \hat{\mathtt{R}}_{\varphi}$, i.e. it is the composition of the measurable functional $\hat{\mathtt{R}}_{\varphi}$ (Lemma 10) with the measurable

projection π_0. $\square$

Proposition 11. *Let φ be a STL formula. The functional $\hat{\mathtt{R}}_\varphi : \mathcal{D}([0,T],E) \to \mathcal{D}([0,T-T_\varphi],E)$ is not continuous with respect to the Skorokhod topology.*

Proof: We provide a counterexample to continuity, by exibiting a formula φ and a trajectory $\mathbf{x} \in \mathcal{D}([0,T],E)$ such that $\hat{\mathtt{R}}_\varphi$ is not continuous in $\mathbf{x}$. Fix $\varphi = \mathcal{F}_{[1,2]} X \geq 0$, where X is the only system variable. The trajectory $\mathbf{x}(t)$ we consider is equal to 1 for $t \in [0,1) \cup [2,T)$, and equal to 0 for $t =\in [1,2)$. Let now $\mathbf{x}^n(t)$ be equal to 1 for $t \in [0,1) \cup [2+\varepsilon_n, T)$ and to 0 for $t =\in [1, 2+\varepsilon_n)$, where $\varepsilon_n > 0$ is a sequence such that $\varepsilon_n \to 0$. We have that $\hat{\mathtt{R}}_\varphi(\mathbf{x}) = \mathbf{y}$, with $\mathbf{y}$ the constant function 1, while $\hat{\mathtt{R}}_\varphi(\mathbf{x}^n) = \mathbf{y}^n$, where $\mathbf{y}^n(t) = 0$ for $t \in [0, \varepsilon_n)$ and $\mathbf{y}^n(t) = 1$ for $t \geq \varepsilon_n$. It is easy to check that $\mathbf{x}^n \to \mathbf{x}$ in the Skorokhod topology, using the sequence of time wiggle functions ω^n such that $\omega^n(2) = 2 + \varepsilon_n$ and ω^n linear elsewhere. On the other hand $\mathbf{y}^n$ does not converge to $\mathbf{y}$, which proves the claim, as continuous functions send convergent sequences to convergent sequences. $\square$