# Probabilistic Reachability for Uncertain Stochastic Hybrid Systems via Gaussian Processes

Mariia Vasileva, Fedor Shmarov, Paolo Zuliani School of Computing Newcastle University Newcastle upon Tyne, UK {m.vasileva2, fedor.shmarov, paolo.zuliani}@newcastle.ac.uk

Abstract—Cyber-physical system models often feature stochastic behaviour that itself depends on uncertain parameters (e.g., transition rates). For these systems, verifying reachability amounts to computing a range of probabilities depending on how uncertainty is resolved. In general, this is a hard problem for which rigorous solutions suffer from the well-known curse of dimensionality. In this paper we focus on hybrid systems with random parameters whose distribution is subject to nondeterministic uncertainty. We show that for these systems the reachability probability is a smooth function of the nondeterministic parameters, and thus Gaussian processes can be used to approximate the reachability probability function itself very efficiently over its entire domain. Furthermore, we introduce a novel approach that exploits rigorous probability enclosures for training Gaussian processes. We apply our approaches to non-trivial hybrid systems case studies, and we empirically demonstrate their advantages with respect to standard statistical model checking.

# I. INTRODUCTION

Given a system model and a set of 'goal' states (indicating (un)wanted behaviour), the reachability problem is: "can the system evolution reach these states or not?" The corresponding problem for stochastic systems is known as probabilistic reachability, and it amounts to computing the probability that the system reaches a goal state. In this paper we focus on hybrid (discrete-continuous) systems [1], which are a very successful framework for modelling cyber-physical systems and biological systems (e.g., infusion pumps and cancer therapies). However, the formal analysis of hybrid systems, including reachability, is notoriously hard: It is well known that the satisfiability of general real formulas (e.g., involving trigonometric functions) is also an undecidable problem [2]. Furthermore, for all but the simplest hybrid systems (timed automata) checking reachability is undecidable [3]. (See [4] for an up to date survey on reachability for hybrid systems.) This motivates us to find efficient and accurate statistical techniques that can deal with realistic, nonlinear hybrid systems.

In this work, we address hybrid systems with random parameters whose distribution is subject to nondeterministic uncertainty (for example, to account for changing environmental conditions or individual patients variability). For these systems we aim at solving (approximately) the *bounded* probabilistic reachability problem, where 'bounded' means that we consider only a finite number of discrete steps and finite time in the

978-1-7281-9148-5/20/\$31.00 © 2020 IEEE

system evolution. This problem amounts to computing a range of reachability probabilities depending on how the nondeterministic uncertainty in the system is resolved. State-of-the-art statistical techniques [5], [6] struggle with nondeterministic systems, and cannot fully compute the bounded reachability probability function for the systems we consider.

In this paper, we motivate the use of, and apply Gaussian Processes (GPs) for approximating the bounded reachability probability function over the uncertain parameters domain. In particular, we show that said function is smooth – infinitely differentiable – with respect to the uncertain parameters, and therefore GPs can approximate the function arbitrarily well. A notable feature of GPs is that they offer an efficient analytical approximation over the *entire* domain of the function. In contrast, statistical model checking can only provide a pointwise approximation.

In our context, building a GP approximation requires evaluating the reachability probability function at a finite number of training points, which can be done via a Monte Carlo approach [6] that returns statistical estimates of the function value at each training point. In this paper, we propose a new training approach which instead uses rigorous (non-statistical) enclosures of the function [7]. Summarising, in this paper we make the following contributions:

- we show that the reachability probability function for our systems is a smooth function of the nondeterministic uncertain parameters, thereby justifying the use of GP approximations;
- we introduce a novel approach for GP training that combines rigorous enclosures computed at sampled training points;
- we compare GP approximation with statistical model checking (SMC) on three non-trivial hybrid systems models, and show that GP offer comparable accuracy to SMC while requiring much less computational effort.

**Related Work.** We focus on works that combine verification with GP-based methods, which are the main subject areas of the paper. In [8] it is shown that the satisfaction probability of temporal logic formulas over uncertain continuous-time Markov chains is a smooth function of the nondeterministic parameters, so GP approximation can be used. Online model learning using stochastic hybrid systems based on GP has been proposed, but without formal justification [9]. In [10], GPs are used to compute reachable sets of (non-hybrid) dynamic systems, but the authors directly assume smooth system dynamics. In [11], GPs are used for the design of robust systems by approximating and optimising the robustness score of the system (given by the quantitative semantics of a given Signal Temporal Logic formula). The basis of the GP training algorithm we use was initially proposed in [12] and then generalised in [13]. In [14], the authors consider GP training using multiple annotators per input point, although with binary values only. Statistical approaches are proposed in [5], [6] for nondeterministic stochastic hybrid systems, but can only compute the extrema of the reachability probability function, while we analytically approximate it over its entire domain. Very recent works develop a probably approximately correct (PAC) learning framework for building linear models of black-box systems [15] and PAC barrier certificates for hybrid systems safety [16]. In the latter case appropriate barrier certificate templates must be developed by the user, rendering the technique only semi-automated, and in both cases the system is assumed to be deterministic and all its inputs are sampled (from scenario distributions). This means the reachability probability function can only be evaluated pointwise, as in statistical model checking. Our approach is instead fully automated and allows dynamics that depends on random and nondeterministic parameters, and offers an analytical approximation of the reachability probability function over its full domain.

## II. BACKGROUND

**Hybrid systems.** Stochastic Parametric Hybrid Systems (SPHS) [6] combine discrete and continuous dynamic behaviour with continuous and discrete parameters whose values are set in the initial state and remain unchanged during the system evolution. The parameters can be nondeterministic and probabilistic, in which case a probability measure is associated to them. We define Stochastic *nondeterministic* Parametric Hybrid Systems (SnPHS) as a subclass of SPHS [6] in which only random parameters are allowed, and their probability measure depends on nondeterministic parameters. Again, both parameters remain fixed throughout the system evolution.

**Definition 1.** A Stochastic nondeterministic Parametric Hybrid System (*SnPHS*) is a tuple  $\langle Q, \Upsilon, X, P, Y, I, \Xi, \mathbf{jump}, \mathbf{goal} \rangle$ :

- $Q = \{q_0, \dots, q_m\}$  a set of modes (discrete components of the system),
- $\Upsilon = \{(q,q') : q,q' \in Q\}$  a set of transitions between modes,
- $X = [u_1, v_1] \times \cdots \times [u_n, v_n] \subset \mathbb{R}^n$  a domain of continuous variables,
- $P \subset \mathbb{R}^k$  is the (compact) nondeterministic parameter space, with associated probability density functions  $f_1(\cdot, \mathbf{p}), ..., f_r(\cdot, \mathbf{p})$  for r random parameters with  $\mathbf{p} \in P$ and with domain  $R = [a_1, b_1] \times \cdots \times [a_r, b_r] \subset \mathbb{R}^r$ ,
- $Y = \{\mathbf{flow}_q : q \in Q\}$  where  $\mathbf{flow}_q : X \times R \times [0, T] \to X$ is the continuous system dynamics,

- I = {init<sub>q</sub> : q ∈ Q} where init<sub>q</sub> : R → X computes the initial continuous state in mode q,
- $\Xi = \{ \mathbf{reset}_{(q,q')} : (q,q') \in \Upsilon \}$  where  $\mathbf{reset}_{(q,q')} : X \times R \times [0,T] \to X$  defines the continuous state at time t = 0 in mode q' after taking the transition from mode q.

and predicates (or relations)

- jump<sub>(q,q')</sub>(x) ≡ discrete transition (q,q') ∈ Υ occurs upon reaching the jump condition in state (x,q) ∈ X × R × [0,T] × Q,
- $\operatorname{goal}_q(\mathbf{x}) \equiv state \ \mathbf{x} \in X \times R \times [0,T]$  in mode q is a goal state.

Furthermore, SnPHS restrict SPHS [6] by disallowing nondeterministic jumps, *i.e.*, we require that every  $\mathbf{reset}_{q,q'} \in \Xi$ is a Boolean predicate and their true preimages are disjoint. We assume that the random parameters' domain R does not depend on the nondeterministic parameters p, but this is not a restriction in practice. Indeed, the domain of certain distributions (e.g., the uniform distribution) can depend on nondeterministic parameters. However, it is possible to apply a change of variable to make the domain independent of the nondeterministic parameters (e.g., given a < b and a random variable U uniformly distributed over an interval [0, 1], then Z = (b - a)U + a is uniformly distributed over [a, b]). The boundedness of R is also not a significant restriction - any probability density can be approximated arbitrarily well by a density with bounded support. The continuous dynamics Y is made of Lipschitz-continuous ordinary differential equations (ODEs), which have a unique solution for any initial condition in  $X \times R \times [0, T]$  according to the well-known Picard-Lindelöf theorem.

**Probabilistic Reachability.** Two possible approaches to circumvent the undecidability of reachability include computing rigorous enclosures [7] and verified confidence intervals [6]. Both approaches exploit  $\delta$ -complete decision procedures [17]. Let  $\phi$  be a sentence in the standard form:

$$\phi = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^m (\bigvee_{j=1}^{k_i} f_{ij}(x_1, ..., x_n) = 0)$$
(1)

where the  $f_{ij}(x_1, ..., x_n)$  are (compositions) of Type 2computable (*i.e.*, 'numerically implementable') real functions [18] and the  $I_i$  are bounded intervals. Given  $\delta \in \mathbb{Q}^+$  and a sentence  $\phi$  as (1), the  $\delta$ -decision problem is to decide correctly one of the following about  $\phi$ : unsatisfiable ( $\phi$  is false), or  $\delta$ satisfiable ( $\phi^{\delta}$  is true), where  $\phi^{\delta}$  is the  $\delta$ -weakening of  $\phi$ :

$$\phi^{\delta} = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^m (\bigvee_{j=1}^{k_i} |f_{ij}(x_1, ..., x_n)| \le \delta).$$

If the two cases overlap either answer can be returned. Standard bounded reachability questions over hybrid systems can be coded as sentences as above and  $\delta$ -decided by  $\delta$ -complete decision procedures [19], [20].

Moving towards probabilistic bounded reachability, the following definition is a specialisation of the corresponding notion for SPHS [21]. **Definition 2.** [21] Given a SnPHS and reachability depth  $l \in \mathbb{N}$ , *the* bounded reachability probability *function*  $\mathbf{Pr}: P \to [0, 1]$  *is:* 

$$\mathbf{Pr}(\mathbf{p}) = \int_G d\mu(\mathbf{p})$$

where  $\mu(\mathbf{p})$  is the probability measure given by the product of the probability densities  $f_1(\cdot, \mathbf{p}), \ldots, f_r(\cdot, \mathbf{p})$  and  $G = \{x \in R : \exists \pi \in \Pi(l) : \mathbf{reach}(\pi, x)\}$  is the goal set;  $\Pi(l)$  is the set of paths of length l in the SnPHS and  $\mathbf{reach}(\pi, x)$  is the formula in standard form (1) that defines l-step reachability for path  $\pi$  and random parameter x:

$$\begin{aligned} \mathbf{reach}(\pi, x) &\equiv \exists^{[0,T]} t_0, \cdots, \exists^{[0,T]} t_{l-1}, \exists^X \mathbf{x}_0, \cdots, \exists^X \mathbf{x}_{l-1} : \\ \left(\mathbf{x}_0 = \mathbf{flow}_{\pi[0]}(\mathbf{init}_{\pi[0]}(x), x, t_0)\right) \bigwedge_{i=0}^{l-2} \left[\mathbf{jump}_{(\pi[i], \pi[i+1])} \\ \left(\mathbf{x}_i, x, t_i\right) \wedge \left(\mathbf{x}_{i+1} = \mathbf{flow}_{\pi[i+1]}(\mathbf{reset}_{\pi[i], \pi[i+1]} \\ \left(\mathbf{x}_i, x, t_i\right), x, t_{i+1}\right)\right) \right] \wedge \mathbf{goal}_{\pi[l-1]}\left(\mathbf{x}_{l-1}, x, t_{l-1}\right). \end{aligned}$$

We recall that in a SnPHS the predicates flow, init, jump, reset and goal must involve Type 2 computable functions only.

Gaussian Processes. We assume some familiarity with Gaussian Processes, so here we only give a brief, high-level presentation of the topic: for a longer presentation see [22], and a thorough treatment can be found in [23]. Given a SnPHS depending on (nondeterministic) parameters  $\mathbf{p} \in P$ , our goal is to estimate the (unknown) reachability probability function, *i.e.*,  $\mathbf{Pr}(\mathbf{p})$  for every  $\mathbf{p} \in P$ . We use a Bayesian approach, which means that **Pr** will be approximated by the posterior distribution of a stochastic process over P, given a set of observations of **Pr** at points  $\mathbf{p}_1, \ldots, \mathbf{p}_N$  in *P*. In particular, we first need to define a prior distribution, which describes our "guess" about the unknown function, *i.e.*, **Pr**. Then it is necessary to determine the functional form of the likelihood. The final step is computing the posterior distribution given the observations, via the Bayes theorem and the likelihood function. We obtain the required estimate and confidence interval for  $\mathbf{Pr}$  for any point in P by evaluating the statistics of the induced posterior distribution.

For prior we choose a Gaussian Process (GP). Given an unknown *smooth* real function and a set of function evaluations at a *finite* set of training input points, a GP offers a statistical, analytical approximation for said function over its *entire* domain, with asymptotic guarantees of accuracy with respect to the number of training points. (Essentially, given N training points, a GP approximates the unknown function by a N-dimensional Gaussian distribution whose covariance matrix is obtained by integrating so-called basis functions evaluated at the training points [22].) We show in Theorem 1 that the **Pr** function is smooth, hence justifying our use of GPs for approximating it. We again emphasise that statistical model checking only gives point-wise approximations.

The observations of function **Pr** necessary to train a GP are obtained by checking probabilistic reachability over a finite set

of points in the uncertain parameter space P. Such points constitute the GP *training set*, and can be generated by a simple partition of P or more effectively by using low-discrepancy sequences, *e.g.*, quasi-Monte Carlo [24]. Pseudocode for GP training via sampling can be found in Algorithm 1.

Evaluating reachability in a SnPHS for a given  $\mathbf{p} \in P$ returns Boolean observations distributed as a Bernoulli with parameter  $\mathbf{Pr}(\mathbf{p})$ . For each  $\mathbf{p}$  in the training set we generate S observations, thus distributed as a  $Binomial(S, \mathbf{Pr}(\mathbf{p}))$ random variable. Closed-form solutions for GP posterior inference typically require a normal likelihood, while in our case we have Bernoulli's. To solve this problem we apply the approach presented in [8]. Specifically, one first needs to map probabilities to the full real line, which can be done via the inverse probit transformation:

$$\forall w \in [0,1], \zeta \in \mathbb{R} \quad \Psi(w) = \zeta \Leftrightarrow w = \int_{-\infty}^{\zeta} \mathcal{N}(0,1),$$

where  $\mathcal{N}(0, 1)$  is the standard Gaussian density (with mean zero and variance 1). Then, for each point  $\mathbf{p}_i$  in a training set  $\mathbf{p}_1, \ldots, \mathbf{p}_N \in P$ , our data  $\mathcal{O}_i$  consist of S binary outcomes sampled from a Bernoulli with parameter  $\mathbf{Pr}(\mathbf{p}_i)$  (see Algorithm 1). Thus, the joint probability of function  $\mathbf{Pr}(\mathbf{p})$  and outcomes  $\mathcal{O}$  is:

$$p(\mathcal{O}, \mathbf{Pr}(\mathbf{p})) = \mathrm{GP}(\Psi(\mathbf{Pr}(\mathbf{p}))) \prod_{i=1}^{N} \prod_{j=1}^{S} \mathrm{Bernoulli}(\mathcal{O}_{i,j} | \mathbf{Pr}(\mathbf{p}_i)).$$

Algorithm 1 Sampling Pr for GP training

- **Input:** H : SnPHS,  $l \in \mathbb{N}$  : reachability depth,  $\delta > 0$  : solver precision, N : number of training points, S : number of samples per training point.
- **Output:**  $X = x_1, \ldots, x_N$  training points,  $Y = y_1, \ldots, y_N$  probability values.

1: 
$$n \leftarrow 0$$
;

2: for 
$$n < N$$
 do

3: 
$$v \leftarrow 0;$$

- 4:  $d \leftarrow 0;$
- 5:  $x_n \leftarrow QMC\_sample(P);$   $\triangleright$  sampling nondeterministic parameters via Quasi-Monte Carlo
- 6: for d < S do
- 7:  $rnd \leftarrow MC\_sample(R)$ ;  $\triangleright$  Monte Carlo sampling random parameters

8: 
$$d \leftarrow d+1;$$

9: ▷ formally evaluate reachability - see Section 4.2 in [25]

10: **switch evaluate** $(H, l, x_n, rnd, \delta)$  **do** 

11: **case unsat do**  $v \leftarrow v + 1$ ;  $\triangleright$  count decidable unsatisfiable reachability only, anything else is deemed satisfiable

```
12: end for
```

13:  $y_n \leftarrow (S-v)/S;$   $\triangleright$  estimate  $\mathbf{Pr}(x_n)$ 

14:  $n \leftarrow n+1;$ 

15: end for

Finally, the value of the target function at a new point  $\mathbf{p}^* \in P$  (*i.e.*,  $\mathbf{Pr}(\mathbf{p}^*)$ ) is approximated by the mean of the posterior distribution of the function at  $p^*$ , given the observations  $\mathbf{Pr}(\mathbf{p}_1), \ldots, \mathbf{Pr}(\mathbf{p}_N)$ . As observed in [8], this can be achieved efficiently via the Expectation-Propagation (EP) algorithm [12], [13]. From the posterior variance one can compute confidence intervals for  $Pr(p^*)$  with arbitrary coverage probability by modifying the standard GP regression technique as shown in Algorithm 2, which we dubbed GP-EP. (More details about the GP-EP approach can also be found in Appendix A and B.)

Algorithm 2 GP-EP regression (adapted from Algorithm 3.6 in [23])

- **Input:** k : covariance function,  $p^*$  : nondeterministic test point,  $c \in (0,1)$  : confidence (coverage probability),  $\dot{\omega}$ ,  $\dot{\eta}$ : natural site parameters (computed by Expectation-Propagation)
- **Output:**  $\bar{y}^*$ : mean  $\mathbf{Pr}(\mathbf{p}^*)$  value, Low: lower CI bound, Up : upper CI bound

- 1:  $L \leftarrow cholesky(I_n + \check{C}^{\frac{1}{2}}K\check{C}^{\frac{1}{2}});$ 2:  $\zeta \leftarrow \check{C}^{\frac{1}{2}}L^{\top} \setminus (L \setminus (\check{C}^{\frac{1}{2}}K\check{\omega}));$ 3:  $\bar{y}^* \leftarrow \mathbf{k}(\mathbf{p}^*)^{\top}(\check{\omega} \zeta); \qquad \triangleright cc$  $\triangleright$  compute mean value (Eqs. (3.60) and (3.71) [23])
- 4:  $\mathbf{t} \leftarrow L \setminus (\check{C}^{\frac{1}{2}} \mathbf{k}(\mathbf{p}^*));$ ▷ see Eq. (3.61) [23] 5:  $\mathcal{V}(y^*) \leftarrow k(\mathbf{p}^*, \mathbf{p}^*) - \mathbf{t}^\top \mathbf{t};$
- 6:  $Low, Up \leftarrow \text{CDF}(\bar{y}^* \mp \text{CDF}^{-1}(1 (1 c)/2)\mathcal{V}(y^*)); \triangleright$ compute confidence interval of coverage at least c (Eq. (3.72) [23])

# **III. PROBABILISTIC REACHABILITY SMOOTHNESS**

In this section we prove that the reachability probability function of a SnPHS is, under reasonable conditions, a smooth function of the uncertain parameters. This result is crucial for justifying the use of GP to approximate the reachability probability function.

**Theorem 1.** Let H be an SnPHS in which the random parameter densities are smooth, i.e.,  $f_i(x, \mathbf{p}) \in C^{\infty}(R \times P)$ for all  $1 \leq i \leq r$ . Then, the reachability probability function of H is a smooth function of the uncertain parameters, i.e.,  $\mathbf{Pr}(\mathbf{p}) \in C^{\infty}(P).$ 

*Proof.* We recall that  $P \subset \mathbb{R}^k$ ,  $R \subset \mathbb{R}^r$  and G are the uncertain parameters space, random parameters domain and goal set of the SnPHS H, respectively. We need to show that the function of Definition 2:

$$\mathbf{Pr}(\mathbf{p}) = \int_{G} d\mu(\mathbf{p}) = \int_{R} I_{G} d\mu(\mathbf{p})$$

admits derivatives of any order, where  $I_G$  is the indicator function over G. Since the random parameters are independent, we can rewrite the above as:

$$\mathbf{Pr}(\mathbf{p}) = \int_R I_G(\mathbf{x}) F(\mathbf{x}, \mathbf{p}) \, d\mathbf{x}$$

where  $\mathbf{x} \in \mathbb{R}^r$  and  $F(\mathbf{x}) = \prod_{i=1}^r f_i(x_i, \mathbf{p})$  is the product measure obtained from the random parameters' densities. For clarity of presentation, we assume that **p** is a single uncertain parameter, *i.e.*, k = 1. The extension to multiple uncertain parameters (k > 1) is easily obtained by considering each coordinate.

Since  $\int_{R} I_G(\mathbf{x}) d\mathbf{x} < \infty$  (recall R is a bounded set), by Lebesgue's criterion [26, Theorem 1, Sect. 11.1] the function  $I_G$  is continuous almost everywhere on R. Let  $D \subset R$  be the set of points at which  $I_G$  is discontinuous. Since D has measure zero we have that

$$\mathbf{Pr}(\mathbf{p}) = \int_{R \setminus D} I_G(\mathbf{x}) F(\mathbf{x}, \mathbf{p}) \, d\mathbf{x}$$
(2)

and by the hypothesis on the densities  $f_i$ 's the function  $I_G(\mathbf{x})F(\mathbf{x},\mathbf{p})$  is then continuous over  $R \setminus D \times P$  and has continuous partial derivative with respect to p. Therefore, by Eq. (2) and Leibniz's rule [26, Proposition 2, Sect. 17.5.1], we have that

$$\frac{d\mathbf{Pr}(\mathbf{p})}{d\mathbf{p}} = \frac{d\int_{R\setminus D} I_G(\mathbf{x})F(\mathbf{x},\mathbf{p}) d\mathbf{x}}{d\mathbf{p}} = \int_{R\setminus D} \frac{\partial I_G(\mathbf{x})F(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}} d\mathbf{x} = \int_{R\setminus D} I_G(\mathbf{x})\frac{\partial F(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}} d\mathbf{x}$$

and  $\frac{d\mathbf{Pr}(\mathbf{p})}{d\mathbf{p}}$  is a continuous function over *P* by [26, Proposition 1, Sect. 17.5.1]. The proof now simply proceeds by induction on the order of the derivative, and the smoothness hypothesis. 

In our case studies we shall use normally-distributed random parameters: it is easy to show that Gaussian densities satisfy the hypothesis of Theorem 1.

# **IV. GAUSSIAN PROCESS TRAINING WITH ENCLOSURES**

We now present a novel approach that uses rigorous enclosures for Pr, rather than point approximations, for training a GP. There is an algorithm [7] that computes such enclosures for the range of function Pr over its full domain. However, for our purposes we do not need to have so much information. Instead, we compute enclosures on a selection of *training* points only, chosen over the nondeterministic parameter's space (again, the training points may be sampled via a quasi-Monte Carlo algorithm for better coverage of the parameter space). Therefore, for any  $\mathbf{p} \in P$  we can compute enclosures, *i.e.*, absolutely precise upper and lower bounds, for Pr(p).

We now simply train two GPs, one with the upper bounds and one with the lower bounds for  $Pr(p_i)$  on a training set  $\mathbf{p}_1, \ldots, \mathbf{p}_N \in P$ , and so we need to run two GP regressions, as well. The GP regression output will then provide estimated mean approximation and two confidence intervals (CIs): a lower CI  $[\underline{L}, \overline{L}]$  and an upper CI  $[\underline{U}, \overline{U}]$ , for the two function approximations. If both CIs are computed with coverage probability c, then it is obvious that  $[\underline{L}, \overline{U}]$  is a CI with coverage c for  $Pr(p_i)$ . The pseudocode is sketched in Algorithm 3.

This approach, dubbed GP-Enc, essentially replaces the (possibly large number of) Monte Carlo simulations needed for accurately evaluating  $\mathbf{Pr}(\mathbf{p})$  in the GP-EP approach with absolutely correct bounds for  $\mathbf{Pr}(\mathbf{p})$ . This could be a significant advantage when analysing rare-event probabilities, which are a well-known crux for Monte Carlo techniques because of the exceedingly large sample sizes required. However, the computational complexity of computing the enclosures grows exponentially in the number of random parameters of the systems and in the enclosure precision required (the GP-EP approach does not instead suffer from this curse of dimensionality). Therefore, an acceptable tradeoff between accuracy of the rigorous enclosures and computational effort should be determined when using GP-Enc for a given system.

Finally, the computational complexity of GP-Enc remains  $O(N^3)$  for N training points [23] (as for the GP-EP approach).

**Algorithm 3** GP-Enc regression (adapted from Algorithm 2.1 in [23])

**Input:** k : covariance function,  $L_{up}$ : list of upper bounds of probability enclosures,  $L_{low}$ : list of lower bounds of probability enclosure,  $\mathbf{p}^*$  : nondeterministic test point,  $c \in (0, 1)$  : confidence (coverage probability)

Output: CIs with coverage c.

- 1:  $S_{up} \leftarrow cholesky(K(L_{up}) + \sigma^2 I); \triangleright$  compute Cholesky decomposition (see [23, Appendix A.4])
- 2:  $\alpha_{up} \leftarrow S_{up}^{\top} \setminus (S_{up} \setminus (L_{up});$
- 3:  $\bar{y}_{up}^* \leftarrow \mathbf{k}_*^{\top} \alpha_{up}$ ;  $\triangleright$  compute mean value (Eq. (2.25) [23]) 4:  $\mathbf{t}_{up} \leftarrow S_{up} \setminus \mathbf{k}_*$ ;
- 5:  $\mathcal{V}_{up}(y_{up}^*) \leftarrow k(\mathbf{p}^*, \mathbf{p}^*) \mathbf{t}_{up}^\top \mathbf{t}_{up}; \qquad \triangleright \text{ compute variance}$ (Eq. (2.26) [23])
- 6:  $\overline{U} \leftarrow \text{CDF}(\bar{y}_{up}^* + \text{CDF}^{-1}(1 (1 c)/2)\mathcal{V}_{up}(y_{up}^*)); \triangleright$ compute CI upper bounds for  $L_{up}$  points
- 7:  $\langle$  repeat steps 1-5 for  $L_{low}$  points  $\rangle$
- 8:  $\underline{L} \leftarrow \text{CDF}(\overline{y}_{lo}^* \text{CDF}^{-1}(1 (1 c)/2)\mathcal{V}_{lo}(y_{lo}^*));$ compute CI lower bounds for  $L_{low}$  points

⊳

9: return  $[\underline{L}, \overline{U}]$ 

# V. EXPERIMENTS

In our experiments we apply to three models the GP-EP approach, the GP-Enc approach, and statistical model checking (SMC) using confidence interval (CI) estimation based on the standard Clopper-Pearson technique [27].

We used the ProbReach tool [7], which allows computing bounded reachability in stochastic parametric hybrid systems. ProbReach can use either dReal [28] or iSAT-ODE [19] for analysing (non-probabilistic) bounded reachability question. We implemented in ProbReach the GP-EP algorithm and the GP-Enc approach (available at https://github.com/dreal/ probreach). In our experiments we used ProbReach with dReal3 (version 3.16.08.01) with precision  $\delta = 0.001$  (except where noted) on a 32-core, 2.9GHz Ubuntu 16.04 machine.

# A. Models

*a)* **UVB Irradiation Therapy for Treating Psoriasis:** We use a simplified version of an Ultra-Violet B (UVB) irradiation therapy model [29] for the treatment of psoriasis, an immune system-mediated chronic skin condition which is characterised by overproduction of keratinocytes. The model consists of three types of keratinocytes: stem cells (SC), transit-amplifying cells (TA) and growth-arrested cells (GA) which can all be normal and psoriatic. The model's (nonlinear) ODEs are given below, where the psoriatic species are denoted by a d (disease) subscript

$$\begin{split} \frac{dSC}{dt} &= \gamma_1 \frac{\omega (1 - \frac{SC + \lambda SC_d}{SC_{max}})SC}{1 + (\omega - 1)(\frac{TA + TA_d}{P_{ta,h}})^n} - \beta_1 In_A SC - \\ &- \frac{k_{1s}\omega}{1 + (\omega - 1)(\frac{TA + TA_d}{P_{ta,h}})^n SC + k_1 TA}; \\ \frac{dTA}{dt} &= \frac{k_{1a,s}\omega SC}{1 + (\omega - 1)(\frac{TA + TA_d}{P_{ta,h}})^n} + \\ &+ \frac{2k_{1s\omega}}{1 + (\omega - 1)(\frac{TA + TA_d}{P_{ta,h}})^n + \gamma_2 GA - \beta_2 In_A TA - k_{2s} TA - k_1 TA}; \\ \frac{dGA}{dt} &= (k_{2a,s} + 2k_{2s}) TA - k_2 GA - k_3 GA - \beta_3 GA; \\ \frac{dSC_d}{dt} &= \gamma_{1d} (1 - \frac{SC + SC_d}{SC_{max,t}} SC_d - \beta_{1d} In_A SC_d - \\ &- k_{1sd} SC_d - \frac{k_p SC_d^2}{k_a^2 + SC_d^2} + k_{1d} TA_d); \\ \frac{dTA_d}{dt} &= k_{1a,sd} SC_d + 2k_{1sd} SC_d + \gamma_{2d} TA_d + k_{2d} GA_d - \\ &- \beta_{2d} In_A TA_d - k_{2sd} TA_d - k_{1d} TA_d; \\ \frac{dGA_d}{dt} &= (k_{2a,sd} + 2k_{2sd}) TA_d - k_{2d} GA_d - k_{3d} GA_d - \beta_{3d} GA_d. \\ \end{split}$$
The therapy involves a series of UVB irradiation episodes,

The therapy involves a series of UVB irradiation episodes, which are simulated in the model by increasing  $In_A$  times the apoptosis rate constants  $\beta_1$  and  $\beta_2$  for SCs and TA cells, respectively. The duration of each episode is 48 hours, followed by 8 hours of rest ( $In_A = 1$ ), before the next irradiation can be started.

The efficacy of the therapy depends on the apoptosis rate (modified by  $In_A$ ) and on the number of irradiation episodes. An insufficient number of treatments can lead to psoriasis relapse: the deterministic variant of this model predicts psoriasis relapse for less than seven treatments [29]. Our model has *one* random parameter  $In_A \sim N(\mu, 10,000)$ , which depends on the *uncertain* parameter  $\mu \in [55,000, 65,000]$  to model interpatient variability. We compute the probability of a relapse within a year (except were noted) following a seven-treatment therapy.

b) Pharmocokinetics Model for Anaesthesia Delivery: This case study considers a pharmacokinetics model for anaesthesia delivery which tracks how the drug concentration changes as it is being metabolised by the body [30]. The model features three species:  $c_p$  - concentration of the drug in the plasma,  $c_1$  ( $c_2$ ) - concentration of the drug in the fast (slow) peripheral compartment, and u - drug infusion concentration.

The model's ODEs are:

$$\frac{dc_p(t)}{dt} = -(k_{10} + k_{12} + k_{13})c_p(t) + k_{12}c_1(t) + k_{13}c_2(t) + \frac{u(t)}{V_1}$$
$$\frac{dc_1(t)}{dt} = k_{21}c_p(t) - k_{21}c_1(t)$$
$$\frac{dc_2(t)}{dt} = k_{31}c_p(t) - k_{31}c_2(t)$$
$$\frac{du(t)}{dt} = p\cos(\frac{2t\pi}{T}).$$

We assume that the drug delivery is continuous, except that every T = 15 minutes (starting at time 0) the drug infusion rate is subject to random errors. We compute the probability of reaching the unsafe state:

$$(c_p(t) \ge 6) \lor (c_p(t) \le 1) \lor (c_1(t) \ge 10) \lor (c_1(t) \le 0) \lor \lor (c_2(t) \ge 10) \lor (c_2(t) \le 0)$$

in one jump within 60 minutes. The model features *one* random parameter (reset value of u)  $u_0 \sim N(\mu, 1,000)$ , which depends on the *uncertain* parameter  $\mu \in [6,000, 8,000]$  that models the discrepancy of the infusion device from its nominal behaviour (due to, *e.g.*, aging and wear of the mechanical components).

c) **Deceleration Model:** This model describes a car deceleration scenario [21]. In the first mode the car accelerates from 0 to 27.78 m/s (0 to 100 km/h). During this period its velocity changes according to  $\frac{dv(t)}{dt} = \alpha \exp(-\alpha t + \beta) - c_d v^2(t)$ , where  $\alpha = 0.05776$  and  $\beta \sim \mathcal{N}(\mu, \sigma)$  are coefficients modelling the acceleration characteristics of the car, which depend on the nondeterministic parameters  $\mu \in [3.9, 4.1]$ ,  $\sigma \in [0, 0.2]$  and  $c_d = 3.028 \cdot 10^{-4} m^{-1}$  is the drag coefficient.

When the target velocity 27.78 m/s is achieved, it takes  $t_{react} = 1.2$  seconds for the driver to react and to begin decelerating. There is no acceleration of the car in the "reaction" mode, and its velocity is controlled by the equation  $\frac{dv(t)}{dt} = -c_d v^2(t)$ . In the final (braking) mode the car's deceleration is governed by the equation  $\frac{dv(t)}{dt} = \mu a_d - c_d v^2(t)$  where  $a_d = -4 m/s^{-2}$  is the car's braking force, and  $\mu = 1$  is the coefficient modelling the road characteristics, such as slope, friction, etc. The distance s(t) covered by the car is controlled by  $\frac{ds(t)}{dt} = v(t)$ . We calculate the probability of the car stopping within 400 meters in the braking mode.

# B. Results

*a)* Accuracy and cost of GP-EP vs. SMC: We compare the accuracy of GP-EP with SMC using the average CI interval size and root mean squared error (RMSE) of our estimates across all input points.

As it can be seen in Table I, GP-EP offers not only shorter intervals but also smaller standard deviation for both the tested models over a different number of points and samples with 0.99 confidence. (The same trend holds for 0.99999 confidence – see Table II in the Appendix). For example, for the Psoriasis model with n = 20 training points and S = 200 samples, GP-EP has 0.0727 average interval size while SMC has as much as 0.4014 (see Table I). Also, in Figure 1 it is clearly visible that GP-EP provides smaller CIs than SMC. It is also important to note that GP-EP presents a much smoother mean curve in comparison to SMC, and that even for small sample sizes (S = 20) the GP-EP CIs intersect the rigorous enclosures (computed by ProbReach's formal approach) in most, but not all, cases. Increasing S to 100 results in all the GP-EP CIs intersecting the rigorous enclosures – see Figure 1 (b) and Figure 2 (b).

The comparison of RMSE  $\pm$  standard deviation (see Table IV) shows that for the tested models GPs are more accurate than SMC. In Table IV, the true probability values used to compute the RMSE were calculated via 10,000 SMC simulations. (We also note that the RMSE for both GP and SMC approaches do not depend on the confidence level because that is used only to construct the CI bounds, so for all confidence levels the result of RMSE depends only on the number of points, samples, and parameters of the chosen model.)

As described in Section II, GP-EP has a significant computational advantage after the training process, because any subsequent test input can be calculated using a simple regression process, which is relatively fast and does not require further sampling. Table V reports the total CPU time in seconds needed to construct CIs by GP-EP and SMC for the first nrandomly chosen points + second n randomly chosen points. As it was noted above, the GP testing process is relatively fast, however, it takes time to recompute all the covariances according to the new testing points, while for SMC almost all the time is spent on sampling. As the Psoriasis and Anaesthesia models are non-trivial to simulate. Table V shows a significant GP-EP advantage for all points and sample sizes over SMC. It is however conceivable that with 'lighter' models SMC could be faster than GP-EP. In fact, the Deceleration model is faster to simulate, and it can be seen from Table V that SMC is generally faster than GP-EP (except for  $S \ge 100$ ).

b) Accuracy and Cost of GP-Enc: We compare two versions of the GP-Enc approach (with enclosures precision equal to 0.1 and 0.001, respectively) with the GP-EP approach with different number of samples per every training point.

As it can be seen in Table III, where the average CI size comparison is presented, GP-Enc gives better (tighter) intervals than GP-EP only when precision  $\epsilon = 0.001$  is used for GP-Enc training, *irrespective* of the sample size S used by GP-EP. This advantage holds for all the models. The advantage of the GP-Enc approach also potentially depends on the models' complexity and probability function placing (far or close to the 0-1 borders).

Figure 3 shows these results in a graphical way. In particular, Figure 3(a) shows that the GP-EP CIs are smaller than GP-Enc's when using precision 0.1 for the training part in GP-Enc. However, when precision 0.001 is used (Figure 3(b)), the GP-Enc CIs become smaller (*i.e.*, more precise) than GP-EP's.

Finally, with respect to computational cost, Table VI reports the total CPU time difference in seconds between the GP-EP and GP-Enc approaches. It can be seen that for the

Model		S=20		S=	100	S=200		
	n	SMC	GP-EP	SMC	GP-EP	SMC	GP-EP	
	20	0.5471±0.0179	0.0871±0.0162	0.4627±0.0165	0.0731±0.0157	0.4014±0.0162	<b>0.0727</b> ±0.0156	
Psoriasis	100	$0.4827 \pm 0.0162$	0.0673±0.0158	$0.3854 \pm 0.0157$	0.0565±0.0155	$0.3365 \pm 0.0156$	<b>0.0525</b> ±0.0154	
	200	$0.4013 \pm 0.0157$	0.0495±0.0156	$0.3264 \pm 0.0153$	0.0384±0.0153	$0.3245 \pm 0.0153$	<b>0.0371</b> ±0.0152	
Anaesthesia	20	$0.4008 \pm 0.1188$	0.0687±0.0389	0.2891±0.0543	0.0613±0.0291	0.1334±0.0389	0.0518±0.0286	
	100	$0.3932 \pm 0.1203$	0.0326±0.0216	$0.2291 \pm 0.0523$	0.0295±0.0139	$0.1286 \pm 0.0337$	<b>0.0238</b> ±0.0138	
	200	$0.3622 \pm 0.1141$	<b>0.0286</b> ±0.0164	$0.1891 \pm 0.0467$	<b>0.0211</b> ±0.0119	$0.1124 \pm 0.0311$	<b>0.0205</b> ±0.0097	
Deceleration	20	$0.3635 \pm 0.1232$	0.0879±0.0742	0.1723±0.1138	0.0842±0.0683	0.1271±0.1139	0.0831±0.0628	
	100	$0.3551 \pm 0.1236$	0.0488±0.0715	$0.1621 \pm 0.1125$	<b>0.0469</b> ±0.0625	$0.1106 \pm 0.1116$	<b>0.0454</b> ±0.0605	
	200	$0.3553 \pm 0.1221$	<b>0.0396</b> ±0.0667	$0.1559 \pm 0.1120$	0.0385±0.0502	$0.1042 \pm 0.0928$	<b>0.0356</b> ±0.0539	

#### Table I

Average CI size  $\pm$  standard deviation for SMC vs. GP-EP, obtained via ProbReach, with solver  $\delta$  precision equal to  $10^{-3}$  and 0.99 confidence level for 10 independent runs of the experiment, n - number of training points in the uncertain parameter domain and S - number of samples per point. Min between SMC and GP-EP results reported in **bold**.

Model		S=20		S=.	100	S=200		
	n	SMC	GP	SMC	GP	SMC	GP	
	20	$0.7867 \pm 0.0246$	<b>0.1486</b> ±0.0212	0.5633±0.0214	0.1135±0.0197	0.4726±0.0211	0.1034±0.0196	
Psoriasis	100	$0.7694 \pm 0.0214$	<b>0.1122</b> ±0.0189	$0.5464 \pm 0.0193$	<b>0.1115</b> ±0.0185	$0.4964 \pm 0.0191$	<b>0.1113</b> ±0.0184	
	200	$0.7602 \pm 0.0158$	<b>0.0914</b> ±0.0147	$0.5325 \pm 0.0152$	<b>0.0909</b> ±0.0144	$0.4867 \pm 0.0151$	0.0908±0.0143	
	20	0.6375±0.1246	<b>0.1176</b> ±0.0637	0.3428±0.0951	<b>0.1106</b> ±0.0486	$0.2236 \pm 0.0668$	<b>0.1083</b> ±0.0482	
Anaesthesia	100	$0.6304 \pm 0.1213$	0.0564±0.0364	$0.3246 \pm 0.0903$	0.0426±0.0276	$0.2175 \pm 0.0651$	0.0341±0.0235	
	200	$0.6296 \pm 0.1172$	<b>0.0735</b> ±0.0348	$0.3133 \pm 0.0885$	0.0361±0.0171	$0.2134 \pm 0.0651$	<b>0.0322</b> ±0.0155	
Deceleration	20	0.5883±0.1392	<b>0.1395</b> ±0.1047	0.3329±0.1152	<b>0.1366</b> ±0.0952	0.2598±0.0921	0.1278±0.0883	
	100	$0.5802 \pm 0.1337$	0.0921±0.0963	$0.3265 \pm 0.1097$	0.0885±0.0726	$0.2544 \pm 0.0889$	0.0858±0.0691	
	200	$0.5813 \pm 0.1366$	<b>0.0782</b> ±0.0846	$0.3188 \pm 0.1095$	0.0651±0.0657	$0.2116 \pm 0.0831$	<b>0.0519</b> ±0.0634	

Table II

Average CI size  $\pm$  standard deviation for SMC vs. GP-EP, obtained via ProbReach, with solver  $\delta$  precision equal to  $10^{-3}$  and 0.99999 confidence level for 10 independent runs of the experiment, n - number of training points and S - number of samples per training point. Min between SMC and GP-EP results reported in **bold**.

Model	CP-Fnc	GP-EP							
	OI -Elic	S=20	S=50	S=200	S=1,000	S=3,000			
Anaesthesia	$\epsilon = 0.1$	-0.2313±0.0156	$-0.2355 \pm 0.0154$	$-0.2482 \pm 0.0151$	-0.2515±0.0147	-0.2583±0.0144			
	$\epsilon = 0.001$	<b>0.0387</b> ±0.0095	<b>0.0345</b> ±0.0094	<b>0.0264</b> ±0.0088	<b>0.0192</b> ±0.0087	<b>0.0119</b> ±0.0085			
Deoriosis*	$\epsilon = 0.1$	$-0.2736 \pm 0.0235$	$-0.2794 \pm 0.0201$	-0.2835±0.0193	$-0.2868 \pm 0.0182$	-0.2994±0.0173			
r sollasis	$\epsilon = 0.001$	<b>0.0784</b> ±0.0117	<b>0.0711</b> ±0.0104	<b>0.0675</b> ±0.0098	<b>0.0632</b> ±0.0097	<b>0.0595</b> ±0.0095			
Deceleration	$\epsilon = 0.1$	-0.1247±0.0137	$-0.1285 \pm 0.0132$	-0.1311±0.0132	$-0.1375 \pm 0.0128$	$-0.1494 \pm 0.0127$			
	$\epsilon = 0.001$	<b>0.0617</b> ±0.0104	<b>0.0587</b> ±0.0101	<b>0.0511</b> ±0.0092	<b>0.0468</b> ±0.0093	<b>0.0398</b> ±0.0091			

#### Table III

Average CI size ( $\pm$  standard deviation) difference between the GP-EP and GP-Enc approaches (GP-EP – GP-Enc), obtained for 20 training points with 0.99 confidence for 10 independent runs of the experiment,  $\epsilon$  - training enclosure precision (GP-Enc only), S - number of samples per training point (GP-EP only); \* = psoriasis relapse checked within half a year. Results in **bold** show advantage for GP-Enc.

Model	п	S=20		S=.	100	S=200			
		SMC	GP-EP	SMC	GP-EP	SMC	GP-EP		
	20	$0.0756 \pm 0.017$	0.0593±0.016	$0.0584 \pm 0.014$	0.0485±0.012	$0.0388 {\pm} 0.011$	0.0347±0.011		
Psoriasis	100	$0.0566 \pm 0.015$	0.0416±0.015	$0.0496 \pm 0.013$	0.0295±0.011	$0.0352 {\pm} 0.012$	0.0208±0.011		
	200	$0.0393 \pm 0.012$	0.0317±0.012	$0.0279 \pm 0.011$	<b>0.0218</b> ±0.009	$0.0218 {\pm} 0.009$	<b>0.0159</b> ±0.006		
	20	$0.0663 \pm 0.042$	<b>0.0573</b> ±0.038	$0.0473 \pm 0.027$	0.0375±0.023	$0.0311 \pm 0.021$	0.0288±0.017		
Anaesthesia	100	$0.0573 \pm 0.035$	<b>0.0419</b> ±0.024	$0.0357 \pm 0.017$	0.0285±0.016	$0.0239 \pm 0.016$	<b>0.0184</b> ±0.014		
	200	$0.0512 \pm 0.028$	0.0375±0.021	$0.0326 \pm 0.016$	<b>0.0273</b> ±0.014	$0.0225 {\pm} 0.013$	<b>0.0153</b> ±0.011		
Deceleration	20	$0.0652 \pm 0.027$	0.0478±0.025	$0.0421 \pm 0.018$	0.0316±0.011	$0.0359 \pm 0.014$	0.0264±0.010		
	100	$0.0515 \pm 0.025$	0.0402±0.017	$0.0377 \pm 0.011$	<b>0.0263</b> ±0.008	$0.0321 \pm 0.011$	0.0207±0.007		
	200	$0.0463 \pm 0.017$	0.0325±0.012	$0.0312 \pm 0.009$	<b>0.0293</b> ±0.008	$0.0258 {\pm} 0.009$	<b>0.0167</b> ±0.006		

Table IV

Root-Mean-Square Error  $\pm$  standard deviation for SMC vs. GP-EP, obtained via ProbReach, with solver  $\delta$  precision equal to  $10^{-3}$ and 0.99 confidence level for 10 independent runs of the experiment, n - number of training points and S - number of samples per point. Min between SMC and GP-EP results reported in **bold**.



Figure 1. Model: Psoriasis. Formal enclosures, GP-EP and SMC CI (0.99 confidence) comparison with respect to uncertain parameter  $\mu$  for 20 training points and a) 20 samples and b) 100 samples per training point; 10 independent runs of the experiment.



Figure 2. Model: Anaesthesia. Formal enclosures, GP-EP and SMC 0.99 CI comparison with respect to one uncertain parameter  $\mu$  for 20 training points and a) 20 samples and b) 100 samples per training point, for 10 independent runs of the experiment.

smaller number of samples (20 and 50) the GP-EP approach outperforms the GP-Enc approach. The picture changes with the increase of the number of samples (S) per training point, so that for S = 1,000 and S = 3,000 the GP-Enc approach significantly outperforms the GP-EP approach (standard deviations are not reported due to their being smaller than 0.0001). We also note that despite the Psoriasis model being more computationally demanding than the other two models, the GP-EP approach shows significantly smaller CPU time increase for all number of samples in comparison with the GP-Enc approach. However, when increasing the precision of the enclosure computation of the GP-Enc approach and for models with probability values far from the bounds, the CPU time difference will increase in favour of GP-EP. At the same time with the increase in the number of samples for the GP-EP approach, the GP-Enc approach still can show better results. For example, for the Anaesthesia model the CPU computation time difference between the GP-Enc approach with 0.1 precision and GP-EP approaches 100 samples is 114, while for GP-EP approaches for 3,000 samples it is 25,836 (see Table VI).

Madal	n	S=20		S=50		S=100		S=200	
WIGUEI		SMC	GP-EP	SMC	GP-EP	SMC	GP-EP	SMC	GP-EP
	20+20	4,628	2,396	10,027	5,577	20,168	11,473	39,208	19,808
Psoriasis	100+100	23,879	11,973	53,190	27,890	118,350	57,398	172,908	98,145
	200+200	46,284	24,371	114,326	56,150	237,364	115,065	391,879	198,075
	20+20	648	338	946	598	2,753	1,386	4,853	2,649
Anaesthesia	100+100	3,306	1,626	5,374	2,684	13,998	6,959	25,863	13,250
	200+200	7,764	3,871	12,854	7,453	28,278	14,139	43,634	26,438
Deceleration	20+20	31	43	112	75	128	117	263	174
	100+100	142	357	253	418	625	563	1353	996
	200+200	315	982	746	1,450	1,250	1,896	2,568	2,473

TOTAL CPU TIME (SEC) NEEDED TO CONSTRUCT A CI BY GP-EP (ONE TRAINING AND TWO TESTING) AND BY SMC FOR THE FIRST *n* RANDOMLY CHOSEN POINTS + SECOND RANDOMLY *n* CHOSEN POINTS, *S* - NUMBER OF SAMPLES PER POINT; 10 INDEPENDENT RUNS OF THE EXPERIMENT. MIN BETWEEN SMC AND GP RESULTS REPORTED IN **bold**.

Table V



Figure 3. Model: Anaesthesia. GP-Enc (with two precisions) and GP-EP approaches 0.99 confidence CIs comparison with respect to uncertain parameter  $\mu$  for 20 training points and 100 samples per training point (GP-EP only).

# VI. CONCLUSIONS

This paper offers contributions on two different levels to the verification of uncertain stochastic hybrid systems. Our main theoretical result is a proof that the reachability probability function for the considered systems is, under mild conditions, a smooth function of the uncertain parameters of the model. Hence, Gaussian process (GP) techniques can be used to obtain an efficient analytical approximation of the function.

Furthermore, we have introduced an approach (GP-Enc) that leverages probability enclosures for training GPs instead of Monte Carlo simulations. We have empirically compared GP-Enc, statistical model checking and GP with expectation-propagation (GP-EP) on three non-trivial hybrid systems models, and have shown that GPs are usable in practice even for systems with complex dynamics. Summarising, our experiments have shown that:

 for small sample sizes, GP-EP can return confidence intervals an order of magnitude smaller than statistical model checking;

- GP-EP can be twice as fast than statistical model checking when evaluating multiple points of the reachability probability function;
- the GP-Enc approach can be much more efficient than GP-EP when very narrow confidence intervals are required.

We conclude that for analysing probabilistic reachability in uncertain stochastic hybrid systems, GPs offer an all-around, efficient technique which in many cases can replace statistical model checking. In the future we plan to investigate GP approaches for uncertain rare events, where the reachability probability function is extremely close to 0. This is an important practical problem, which in the GP setting is made interesting by the high computational costs for computing probability enclosures in GP-Enc and the large sample sizes required by statistical model checking and GP-EP. As such, efficient solutions are needed.

Model	GP-Enc		GP-EP						
WIUdei	€ <b>=0.1</b>	€ <b>=0.001</b>	S=20	S=50	S=100	S=200	S=1,000	S=3,000	
Anaesthesia	1,534	5,067	321	596	1,564	2,646	11,365	28,579	
Psoriasis*	5,285	23,671	476	753	1865	2,834	14,033	33,843	
Deceleration	264	841	29	57	88	149	516	1,246	

Table VI

CPU TIME (SEC) FOR TRAINING GP-ENC AND GP-EP,  $\epsilon$  - TRAINING ENCLOSURE PRECISION (GP-ENC ONLY), FOR 20 TRAINING POINTS FOR 10 INDEPENDENT RUNS OF THE EXPERIMENT, **S** - NUMBER OF SAMPLES PER TRAINING POINT (GP-EP ONLY); \* = PSORIASIS RELAPSE CHECKED WITHIN HALF A YEAR.

#### **ACKNOWLEDGEMENTS**

M.V. was supported by a Newcastle University DTA Award; P.Z. was partially supported by EPSRC Portabolomics (EP/N031962/1).

#### REFERENCES

- O. Maler, Z. Manna, and A. Pnueli, "From timed to hybrid systems," in *Real-Time: Theory in Practice*, J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg, Eds. Springer, 1992, pp. 447–484.
- [2] A. Tarski, "A decision method for elementary algebra and geometry," *Manuscript, RAND Corp.*, 1948.
- [3] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid Systems*, ser. LNCS, vol. 736, 1992, pp. 209–229.
- [4] M. Fränzle, M. Chen, and P. Kröger, "In memory of Oded Maler: Automatic reachability analysis of hybrid-state automata," ACM SIGLOG News, vol. 6, no. 1, pp. 19–39, 2019.
- [5] C. Ellen, S. Gerwinn, and M. Fränzle, "Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains," *International Journal on Software Tools for Technology Transfer*, vol. 17, no. 4, pp. 485–504, 2015.
- [6] F. Shmarov and P. Zuliani, "Probabilistic hybrid systems verification via SMT and Monte Carlo techniques," in *HVC*, ser. LNCS, vol. 10028, 2016, pp. 152–168.
- [7] —, "ProbReach: Verified probabilistic δ-reachability for stochastic hybrid systems," in *HSCC*. ACM, 2015, pp. 134–139.
- [8] L. Bortolussi, D. Milios, and G. Sanguinetti, "Smoothed model checking for uncertain continuous-time Markov chains," *Inf. Comput.*, vol. 247, pp. 235–253, 2016.
- [9] H. Abdel-Aziz and X. Koutsoukos, "Online model learning of buildings using stochastic hybrid systems based on Gaussian processes," *Journal* of Control Science and Engineering, 2017, Article ID 3035892.
- [10] L. Bortolussi and G. Sanguinetti, "A statistical approach for computing reachability of non-linear and stochastic dynamical systems," in *QEST*, G. Norman and W. Sanders, Eds. Springer, 2014, pp. 41–56.
- [11] E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti, "System design of stochastic models using robustness of temporal properties," *Theor. Comput. Sci.*, vol. 587, pp. 3–25, 2015.
- [12] M. Opper and O. Winther, "Gaussian processes for classification: Meanfield algorithms," *Neural Computation*, vol. 12, no. 11, pp. 2655–2684, 2000.
- [13] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in UAI, 2001, pp. 362–369.
- [14] F. Rodrigues, F. C. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *ICML*, 2014, pp. 433–441.
- [15] B. Xue, Y. Liu, L. Ma, X. Zhang, M. Sun, and X. Xie, "Safe inputs generation for black-box systems," in *ICECCS*, 2019, *to appear*.
- [16] B. Xue, M. Fränzle, H. Zhao, N. Zhan, and A. Easwaran, "Probably approximate safety verification of hybrid dynamical systems," in *ICFEM*, 2019, *to appear.*
- [17] S. Gao, J. Avigad, and E. M. Clarke, "Delta-decidability over the reals," in LICS, 2012, pp. 305–314.
- [18] K.-I. Ko, Complexity Theory of Real Functions. Birkhäuser, 1991.

- [19] A. Eggers, M. Fränzle, and C. Herde, "SAT modulo ODE: A direct SAT approach to hybrid systems," in ATVA, ser. LNCS, vol. 5311, 2008, pp. 171–185.
- [20] S. Gao, S. Kong, and E. M. Clarke, "Satisfiability modulo ODEs," in *FMCAD*, 2013, pp. 105–112.
- [21] F. Shmarov and P. Zuliani, "SMT-based reasoning for uncertain hybrid domains," in AAAI-16 Workhop on Planning for Hybrid Systems, 2016, pp. 624–630.
- [22] D. J. C. MacKay, "Introduction to Gaussian Processes," in *Neural Networks and Machine Learning*, C. M. Bishop, Ed. Springer-Verlag, 1998.
- [23] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.
- [24] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," USSR Comput. Math. and Math. Phys., vol. 7, no. 4, pp. 86 – 112, 1967.
- [25] F. Shmarov, "Probabilistic bounded reachability for stochastic hybrid systems," Ph.D. dissertation, Newcastle University, UK, 2018, http://hdl.handle.net/10443/4046.
- [26] V. A. Zorich, Mathematical Analysis II, 2nd ed. Springer, 2016.
- [27] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934.
- [28] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *CADE-24*, ser. LNCS, vol. 7898, 2013, pp. 208–214.
- [29] H. Zhang, W. Hou, L. Henrot, S. Schnebert, M. Dumas, C. Heusèle, and J. Yang, "Modelling epidermis homoeostasis and psoriasis pathogenesis," *Journal of The Royal Society Interface*, vol. 12, no. 103, 2015.
- [30] V. Gan, G. A. Dumont, and I. Mitchell, "Benchmark problem: A PK/PD model and safety constraints for anesthesia delivery," in ARCH@CPSWeek, 2014, pp. 1–8.

# APPENDIX

# A. Gaussian Processes

A Gaussian Process (GP) is a collection of random variables, any finite number of which have a multidimensional Gaussian distribution. A GP is fully described by its mean and covariance functions. Given an unknown function  $f(\mathbf{x})$ :  $\mathbb{R}^D \to \mathbb{R}$  which we wish to model via GP, the mean function  $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$  describes the expected value of the function, while the covariance function  $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - f(\mathbf{x}))]$  $m(\mathbf{x})(f(\mathbf{x}') - m(\mathbf{x}'))$  represents the expected correlation between the observations. We can then model f by  $f(\mathbf{x}) \sim$  $GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ . It is important to note that the choice of mean and covariance functions is crucial for the determination of the type of functions which are possible to be sampled from our GP prior. Without loss of generality, the mean function is chosen to be identically zero. For the covariance function, we use the sauared exponential covariance (kernel) function. which is known to define GP samples from smooth functions. It is defined as:

$$k(\mathbf{x}, \mathbf{x}') = \gamma \exp\left(-\sum_{i=1}^{D} \frac{(x_i - x'_i)^2}{\lambda_i^2}\right)$$

where the hyperparameters  $\gamma$  and  $\lambda$  control roughness and variance of the sampled functions. Given N input points  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$  and a random function  $f \sim GP(0, k)$ , by definition of GP we have that

$$(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_N)) \sim \mathcal{N}(0,\mathbf{K}(\mathbf{X},\mathbf{X}))$$

where **K** is the matrix obtained by applying the covariance function k to each pair of points in **X**.

The GP posterior distribution admits a rather simple form. After evaluating our model over a training set  $\mathbf{X}$ , we collect a test set  $\mathbf{X}_*$ .  $\mathbf{X}_*$ . set defines the points where we want to obtain the GP prediction. It can be shown that the joint distribution for an unknown  $\mathbf{y}_* \in \mathbf{X}_*$  using a known  $\mathbf{y} \in \mathbf{X}$  is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} = \mathcal{N} \left( 0, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right).$$

We can now compute mean and covariance of the conditional Gaussian distribution for the posterior distribution as:

$$\mathbb{E}[\mathbf{y}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*] = \mathbf{K}_*^T \boldsymbol{\beta},$$
  
Var $[\mathbf{y}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*] = \mathbf{K}_{**} - \mathbf{K}_*^T (\mathbf{K} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{K}_*$ 

where  $\mathbf{K}_{*} = k(\mathbf{X}, \mathbf{X}_{*}), \mathbf{K}_{**} = k(\mathbf{X}_{*}, \mathbf{X}_{*}), \mathbf{K} = k(\mathbf{X}, \mathbf{X}), \beta = (\mathbf{K} + \sigma_{w}^{2}\mathbf{I})^{-1}\mathbf{y}.$ 

The posterior prediction can be used to perform classification by computing the class membership probability for a new point, given a set of training points  $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ and class labels  $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ . GP classification exploits the mapping of the full real line to the [0, 1] interval using the probit regression function [23, Section 3.1]. Unfortunately, for our purposes it is impossible to use GP classification because we deal with Bernoulli's instead of Gaussians. To solve this problem we use the Expectation-Propagation (EP) algorithm [23, Section 3.6], which can deal with Bernoulli distributions.

The hyperparameters of the covariance function can be computed by maximising the marginal log-likelihood of the posterior probability with respect to the hyperparameters only [23, Section 5.4.1 and 5.5.2].

### **B.** Expectation Propagation

EP is an approximate inference algorithm that unifies two techniques: assumed density filtering and loopy belief propagation [12], [13]. In this algorithm a target density  $f(\theta)$  is approximated by a density from some specified parametric family  $q(\theta)$ . It assumes that our target density  $f(\theta)$  has a proper factorization with the proportion  $f(\theta) \propto \prod_{i=0}^{h} f_i(\theta)$ . The target f is the posterior density  $p(\theta|y)$  in the case of Bayesian interference. Hence, we can assign one factor as the prior and other factors as the likelihood for one data point. EP iteratively approximates  $f(\theta)$  with a density  $q(\theta)$ which takes the same factorization  $q(\theta) \propto \prod_{i=0}^{h} q_i(\theta)$ . The approximation which associates the factors  $f_i(\theta)$  with the approximation  $q_i(\theta)$  is usually called *sites* approximation. At each iteration of the algorithm, and for i = 1, ..., h, we take the current approximating function  $q(\theta)$  and replace  $q_i(\theta)$  by the corresponding factor  $f_i(\theta)$  from our target distribution. Now we can define the *cavity* distribution as:  $q_{-i}(\theta) \propto \frac{q(\theta)}{q_i(\theta)}$ , with the *tilted* distribution equals to  $q_{i}(\theta) \propto f_i(\theta)q_{-i}(\theta)$ .

In general EP firstly constructs an approximation  $q^{new}(\theta)$  for the tilted distribution  $q_{\setminus i}(\theta)$  and then updates approximation to the target density's  $f_i(\theta)$ , which can be computed as  $q_i^{new}(\theta) \propto q^{new}(\theta)/q_{-i}(\theta)$ . From the definition  $q_i^{new}(\theta)$  can be estimated via the Kullback–Leibler divergence [23]:

$$q_i^{new}(\theta) = \arg\min D(f_i(\theta)q_{-i}(\theta)||q_i(\theta)q_{-i}(\theta)),$$

where D(||) corresponds to the Kullback–Leibler divergence measure. It is important to note that other minimization methods can be also rather efficiently used for this purpose [13]. At a very simple level the EP algorithm works as follows:

- 1) Initialization of the initial site approximation  $q_i(\theta)$ .
- 2) Repeat for i = 1, ..., h until all site approximations  $q_i(\theta)$  convergence:
  - Compute cavity parameters approximation  $q_{-i}(\theta) \propto q(\theta)/q_i(\theta)$ ;
  - Update site parameters approximation  $q_i(\theta)$  and re-compute the posterior parameters so that  $q_i(\theta)q_{-i}(\theta)$  approximates as  $f_i(\theta)q_{-i}(\theta)$ .
- 3) Return natural site parameters.

More details on the EP algorithm can be found in [23, Section 3.6] and [8].