IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 6, 2020, accepted September 26, 2020, date of publication October 2, 2020, date of current version October 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028476

Automated Synthesis of Safe Digital Controllers for Sampled-Data Stochastic Nonlinear Systems

FEDOR SHMAROV[1], SADEGH SOUDJANI[1], NICOLA PAOLETTI[2], EZIO BARTOCCI[3], SHAN LIN[4], SCOTT A. SMOLKA[5], AND PAOLO ZULIANI[1]

[1]School of Computing, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.
[2]Royal Holloway, University of London, Egham TW20 0EX, U.K.
[3]Faculty of Informatics, TU Wien, 1040 Vienna, Austria
[4]Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA
[5]Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA

Corresponding author: Sadegh Soudjani (sadegh.soudjani@ncl.ac.uk)

The work of Ezio Bartocci was supported by the Vienna Science Fund and Technology (WWTF) through the ProbInG Project under Grant ICT19-018. The work of Scott A. Smolka was supported by the NSF under Grant DCL-2040599, Grant CCF-1918225, and Grant CPS-1446832. The work of Paolo Zuliani was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) through the Portabolomics Project under Grant EP/N031962/1.

ABSTRACT We present a new method for the automated synthesis of digital controllers with formal safety guarantees for systems with nonlinear dynamics, noisy output measurements, and stochastic disturbances. Our method derives digital controllers such that the corresponding closed-loop system, modeled as a *sampled-data stochastic control system*, satisfies a safety specification with probability above a given threshold. Our technique uses a fast solver and an optimization method to search for candidate controllers, which are then formally evaluated in closed-loop with the system in question by a verified solver. Unstable candidate controllers are discarded by efficiently checking a sufficient condition for Lyapunov stability of sampled-data nonlinear systems. We evaluate our technique on three case studies: an artificial pancreas model, a powertrain control model, and a quadruple-tank process.

INDEX TERMS Formal controller synthesis, parameter synthesis, probabilistic guarantees, safety verification, sampled-data nonlinear systems, satisfiability modulo theories, statistical model checking, stochastic systems.

I. INTRODUCTION

Due to its superior flexibility, scalability, and lower cost, digital control is used in many cyber-physical and embedded systems applications, ranging from aircraft autopilots to biomedical devices. The synthesis of digital controllers for linear systems is well-studied [34], but its extension to nonlinear and stochastic systems has proven much more challenging [3], [17], [24]. The digital-control problem is further complicated by, *e.g.*, time discretization and signal quantization. Yet another issue is the lack of automated synthesis techniques with provable guarantees, especially for properties beyond stability (*e.g.*, safety) for nonlinear stochastic systems.

In this article, we introduce a new method for the *synthesis of probabilistically safe digital controllers* for a large class of stochastic nonlinear systems, viz. *sampled-data stochastic control systems* (SDSSs). In an SDSS, the plant is modeled by a set of nonlinear differential equations subject to random input disturbances, and the digital controller samples the noisy plant output, generating the control input with a fixed frequency. This class of systems is general enough to describe, for instance, numerical solutions of stochastic differential equations [39]. By a probabilistically safe system, we mean a system that has a small probability of entering a forbidden (unsafe) subset of its state space. Note that we are not concerned with safety from external attacks.

Controllers are usually designed to ensure stability of the closed-loop system [24]. Given a possibly nonlinear SDSS S and a linearization S' of S, previous work [32] provides *sufficient* conditions on S' that ensure the stability of S. Unfortunately, it is difficult to verify these conditions algorithmically. Lawrence [26] provides a necessary and sufficient condition for *exponential stability* of SDSS based on its linearized version.

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Mercaldo.

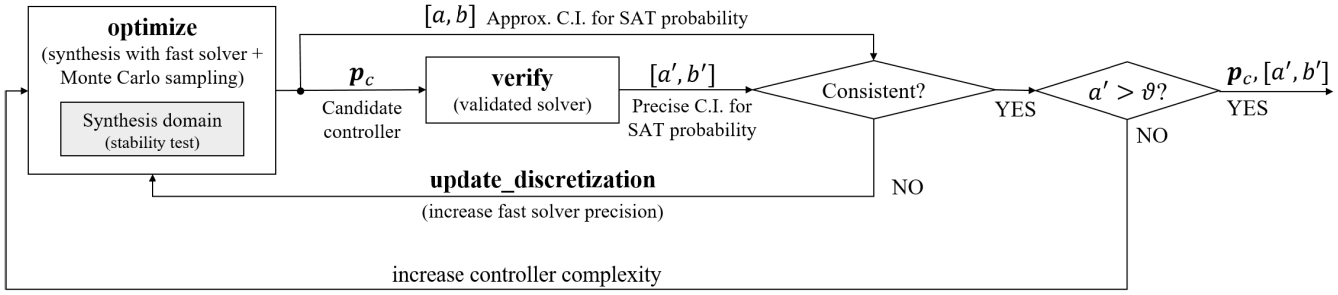


FIGURE 1. Overview of our synthesis approach for safe digital controllers.

In this article, we focus on *Lyapunov stability* of SDSSs and use the condition provided in [26] as a sufficient criterion for Lyapunov stability. This condition is easy to verify, and we use it to restrict the controller synthesis domain. Note that even if the SDSS is stable, it is not necessarily safe since during the transient, the system might reach an unsafe state. The synthesis approach that we propose derives controllers that render the closed-loop SDSS provably probabilistically safe and Lyapunov-stable.

Given an invariant ϕ (*i.e.*, a correctness specification), and a nonlinear plant with stochastic disturbances and noisy outputs, our method synthesizes a digital controller such that the corresponding closed-loop system satisfies ϕ with probability above a given threshold ϑ. An overview of the synthesis algorithm (presented in detail in Section V) is given in Figure 1. It works by alternating between two steps: generation of a candidate controller $\mathbf{p}_c$, and its verification.

The candidate controller $\mathbf{p}_c$ is generated via the **optimize** procedure (see Algorithm 2), which maximizes a Monte Carlo estimate of the satisfaction probability by simulating a discrete-time approximation of the system with a non-validated ordinary differential equation (ODE) solver. Such a candidate is therefore sub-optimal, but can be generated very rapidly. To rule out unstable candidate controllers, we use an easy-to-check condition that is sufficient to guarantee Lyapunov stability of the closed-loop SDSS. Along with $\mathbf{p}_c$, procedure **optimize** returns an approximate confidence interval (CI) $[a, b]$ for the satisfaction probability.

Next, in the verification step (procedure **verify**), we use a validated solver based on SMT (Satisfiability Modulo Theories) to compute a numerically and statistically valid CI $[a', b']$ for the satisfaction probability of $\mathbf{p}_c$. If the deviation between the approximate CI $[a, b]$ and the precise CI $[a', b']$ is too large, indicating that the candidate generated by **optimize** is not sufficiently accurate, we increase the precision of the non-validated, fast solver (procedure **update_discretization**). If instead a' is not above the threshold ϑ, we expand the search space for candidates by increasing the controller complexity.

In summary, the contributions of this article are:

- We present a novel algorithm for the synthesis of digital controllers with stability and probabilistic safety guarantees.
- We utilize Lyapunov's indirect method to derive an easy-to-check condition for Lyapunov stability of nonlinear systems in closed-loop with digital controllers.
- We restrict the synthesis domain based on the stability test and use a verified solver to check the safety of sampled trajectories of the system.
- We conduct an extensive experimental evaluation of the algorithm based on three significant case studies.

Related Work: Advanced controller synthesis techniques for nonlinear systems are studied extensively in the literature. Recent examples of such techniques include [6], [7], [30], [31], [41]. The paper [30] represents the nonlinear plant as a Takagi-Sugeno fuzzy system and designs a static output feedback tracking controller to achieve dissipative tracking performance subject to quantization effects. The paper [6] assumes a similar representation of the nonlinear plant and studies networked nonlinear systems with multi-path data packet dropouts. State estimation of Markovian coupled networks with nonlinear dynamics is studied in [41]. The analyses of these papers are conducted for discrete-time dynamics while our work studies nonlinear systems with continuous-time dynamics. The paper [7] considers a class of switched stochastic nonlinear systems and provides an output-feedback tracking control scheme based on fuzzy observers and fuzzy logic approximations of the unknown nonlinear functions. The paper [31] studies adaptive neural tracking control of nonlinear systems with multiple inputs and outputs.

The problem of controller synthesis under safety requirements has been investigated mainly in the context of Model Predictive Control (MPC) [4], the goal of which is to find the control input that optimizes the predicted performance of the closed-loop system up to a finite horizon. The works of [14], [23], [25], [29], [35], [36], [40], [50] consider safety requirements expressed as temporal logic formulas, and synthesize MPC controllers that optimize the robust satisfaction of the formulas [10] (i.e., a quantitative measure of satisfaction). MPC is an online method that often requires solving at runtime computationally expensive optimization problems. In contrast, our approach performs controller synthesis at design time.

Another approach to the stochastic synthesis problem with safety requirements is to directly maximize the safety

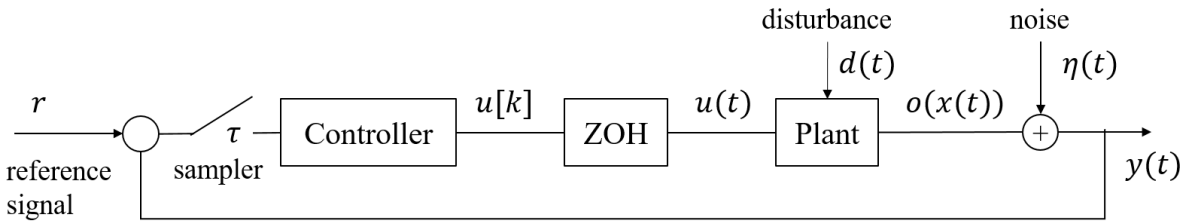


FIGURE 2. Architecture of a sampled-data stochastic control system.

probability under all classes of policies [48]. The related computational techniques generally rely on abstraction methods [18], [47]. A set-based computation approach to controller synthesis with safety guarantees is proposed in [8]. Reach-avoid specification on linear systems is studied in [13]. It is shown in [11] that the reach-avoid probability can be formulated as a solution of partial differential equations for continuous-time systems. Another research line related to our work involves randomized methods [5], [12] that find a solution for a chanced-constrained optimization. Our work tackles an orthogonal aspect of the optimization, which is evaluating sampled constraints that depend on the solutions of ODEs. This task is generally undecidable and requires verified solvers as used in our approach.

Due to its alternation between candidate controller generation and candidate verification, our synthesis method shares features with Counterexample-Guided Inductive Synthesis (CEGIS) [45]. The main difference with our method is that in CEGIS, the candidate-generation step actively uses the counterexamples found by the verifier by attempting to generalize the counterexamples to larger regions of the search domain. This aspect, however, is not relevant to our method, where there is no notion of counterexample. CEGIS-based approaches to controller synthesis include [1], [37], but, unlike our method, these approaches do not support nonlinear and stochastic dynamics.

The closest paper to our own is [42], where the authors propose a method to synthesize continuous-time PID controllers for satisfying bounded reachability properties. That method works under the assumption that the system can measure the output of the plant continuously and without sensing noise. This assumption, however, is unrealistic for the majority of embedded systems whose operations are governed by a discrete-time clock and where sensor noise is unavoidable.

This article is organized as follows. Section II provides a formal definition of sampled-data stochastic systems (SDSSs), while Section III defines digital controllers for SDSSs and describes the problem of stability of a closed-loop system. Section IV discusses synthesis of digital controllers for SDSSs that satisfy a given safety property. Section V introduces our digital controller synthesis algorithm, and Section VI discusses the challenges of implementing digital controllers on finite-precision hardware. Section VII presents our case studies (artificial pancreas, powertrain system and quadruple-tank process) while Section VIII offers our concluding remarks and directions for future work.

II. SAMPLED-DATA STOCHASTIC SYSTEMS

We consider *sampled-data stochastic control systems* (SDSSs), a rich class of control systems where the plant is specified as a nonlinear system subject to random disturbances. The controller periodically samples the plant output, which is subject to random noise, and generates a control input which depends on the history of past plant outputs and control inputs and which is kept constant during the sampling period with a zero-order hold; see Figure 2. The controller is characterized by a number of unknown parameters, which are the target of our synthesis algorithm.

Definition 1 (Sampled-data Stochastic Control System): An SDSS is described in the following state-space notation:

$$\begin{aligned} &\frac{d}{dt}x(t) = f(x(t), u(t), d(t)),\ x(0) = x_0 \\ &y(t_k) = o(x(t_k)) + \eta(t_k),\ t_k = k \cdot \tau,\ k \in \mathbb{Z}^{\geq 0} \\ &u(t) = h(y(t_0), \ldots, y(t_k), u(t_0), \ldots, u(t_k), \mathbf{p}), \forall t \in [t_k, t_{k+1}), \end{aligned} \tag{1}$$

where

- $x(t) \in \mathbb{R}^n$ is the state of the plant at time t;
- x_0 is the initial state at time $t = 0$;
- $d(t) \in \mathbb{R}^q$ is the disturbance at time t;
- $y(t) \in \mathbb{R}^o$ is the plant output at time t, which is a function of the state with additive noise η;
- $h(\cdot)$ is the digital controller (defined in Section III);
- $u(t) \in \mathbb{R}^m$ is the control input at time t, updated at every sampling period $\tau > 0$ by the digital controller h;
- $\mathbf{p} \in \mathbb{P} \subset \mathbb{R}^p$ is the vector of unknown controller parameters, where $\mathbb{P}$ is a hyperbox (*i.e.*, a product of closed intervals);
- $f(\cdot)$ is the vector field governing the dynamics of the plant (we assume $f \in C^1$, hence Lipschitz-continuous);
- $o : \mathbb{R}^n \to \mathbb{R}^o$ is the output map and is in C^1.

Assumption 1: The additive measurement noises $\eta(t_k)$ are i.i.d. random variables with probability density function f_η. The disturbance $d(\cdot)$ is random with realizations that are piecewise-continuous with a finite number of discontinuities for a given time horizon T. We assume $d(\cdot)$ can be defined

in terms of a finite number of random parameters $\delta_1, \ldots, \delta_{\mathsf{d}}$ having a joint density function f_δ.

Note that these assumptions on $d(\cdot)$ are reasonably mild and allow us to define very general classes of systems, which subsume, for instance, *numerical* solutions of stochastic differential equations [39]. Such numerical solutions indeed rely on computing the value of the Wiener process at discrete time points, which makes it a special case of our disturbances.

We denote with Θ_T the joint distribution of the measurement noises and the disturbances up to time T. The density f_θ of $\theta \sim \Theta_T$ can be written as

$$f_\theta\,(g_1, \ldots, g_{\mathsf{d}}, e_0, \ldots, e_{k'}) = f_\delta(g_1, \ldots, g_{\mathsf{d}}) \prod_{i=0}^{k'} f_\eta(e_i) \qquad (2)$$

with $k' = \lfloor T/\tau \rfloor$. We use Θ_T with density function f_θ as the underlying probability space for assigning probability to state trajectories of the SDSS.

III. DIGITAL CONTROLLERS

The operation of a digital controller is succinctly indicated in Equation (1). These computations are generally performed using current and past output samples and past input samples.

Definition 2 (Digital Controller for SDSS): Given an SDSS, we denote $y[k] = y(t_k)$ and $u[k] = u(t_k)$ and define the *tracking error* as

$$e[k] = r - y[k], \quad k \in \mathbb{Z}^{\geq 0} \qquad (3)$$

where r is a constant *reference signal*. The *output of the controller* is $u[\cdot] \in \mathbb{R}^m$ defined by

$$u[k] = -\sum_{i=1}^{L} a_i u[k-i] + \sum_{i=0}^{L} b_i e[k-i], \qquad (4)$$

where $u[j] = e[j] = 0$ for $j < 0$,[1] L is the *controller degree*, and $a_i \in \mathbb{R}^m \times \mathbb{R}^m$ and $b_i \in \mathbb{R}^m \times \mathbb{R}^{\mathsf{o}}$ are coefficient matrices of the controllers.

Controller design amounts to finding a degree L and coefficient matrices $\{a_i\}_{i=1}^L$, $\{b_i\}_{i=0}^L$ that ensure the desired behavior of the closed-loop system. The vector of parameters $\mathbf{p}$ defined in (1) is recovered by setting $\mathbf{p}$ as a vector containing all the entries of the coefficient matrices a_i, b_i. In the following we use an alternative description of the controller using the state-space representation

$$\begin{aligned} z[k+1] &= G_c z[k] + H_c e[k], \quad z[0] = z_0 \\ u[k] &= C_c z[k] + D_c e[k], \quad k \in \mathbb{Z}^{\geq 0}, \end{aligned} \qquad (5)$$

where $z[k] \in \mathbb{R}^{\mathsf{z}}$ is the state of the controller and matrices (G_c, H_c, C_c, D_c) need to be designed. The above two representations are equivalent. Given a controller in the form of (4), one can transform it to the representation (5), for instance by taking states as memories that store previous values of inputs and outputs. Given matrices (G_c, H_c, C_c, D_c), one can easily compute the coefficient matrices $\{a_i\}_{i=1}^L$, $\{b_i\}_{i=0}^L$ in (4) using matrix multiplication [34].

[1] Note that if the controller has been previously deployed, i.e., it starts from a non-empty history, then $u[j]$, $e[j]$ may be nonzero for $j < 0$.

A. STABILITY OF THE CLOSED-LOOP SYSTEM

In the remainder of this section, we consider a version of the SDSS (1) controlled by (5) without any external inputs, *i.e.*, when d and η are identically zero. We call this the closed-loop SDSS and define its augmented state via

$$x_a(t) = \begin{bmatrix} x(t) \\ z[k] \end{bmatrix}, \quad \forall t \in [t_k, t_{k+1}),\ k \in \mathbb{Z}^{\geq 0}. \qquad (6)$$

An essential requirement in the design of any controller is stability of the closed-loop system with respect to an equilibrium point.

Definition 3 (Equilibrium Point): Any vector $x_{ae} \in \mathbb{R}^n \times \mathbb{R}^{\mathsf{z}}$ with $x_{ae} = [x_e^T, z_e^T]^T$ is called an *equilibrium point* of the closed-loop SDSS if $z_e = G_c z_e$, $o(x_e) = r$, and $f(x_e, u_e, 0) = 0$ with $u_e := C_c z_e$.

The above definition implies that if the SDSS starts at equilibrium $x(0) = x_e$, the controller starts at $z[0] = z_e$, and the input is kept constant at $u(t) = u_e$ for all $t \geq 0$, then the augmented state remains constant $x_a(t) = x_{ae}$ for all $t \geq 0$ and the output is the reference signal $y(t) = r$.

Different notions of stability exist in the literature [46] that require different limiting behavior from the trajectories of the system. We focus on local Lyapunov stability, which is the very first notion that addresses the behavior of the closed-loop system with respect to the effect of initial states without considering external inputs d and η. The precise definition is stated next. Since only local stability is discussed in this article, stability in the rest of this article means *local* stability.

Definition 4: The equilibrium point x_{ae} of the system (1) controlled by (5) and without any external inputs is called *Lyapunov stable* if for every $\epsilon > 0$ there exists a $\delta > 0$ such that for all augmented initial state $x_a(0)$ with $\|x_a(0) - x_{ae}\| \leq \delta$, we have $\|x_a(t) - x_{ae}\| \leq \epsilon$ for all $t \geq 0$.

The following theorem is partly due to Lawrence [26], [27].

Theorem 1: Suppose f and o are continuously differentiable. The equilibrium point $x_{ae} = [x_e^T, z_e^T]^T$ of the system (1) is Lyapunov stable if all the eigenvalues of the matrix

$$\hat{G} := \begin{bmatrix} G - HD_cC & -HC_c \\ -H_cC & G_c \end{bmatrix}, \qquad (7)$$

are inside the unit circle, where $G := e^{A\tau}$, $H := \int_0^\tau e^{A\lambda} B d\lambda$ with

$$A := \frac{\partial f}{\partial x}(x_e, u_e, 0),\ B := \frac{\partial f}{\partial u}(x_e, u_e, 0),\ C := \frac{\partial o}{\partial x}(x_e). \qquad (8)$$

Moreover, x_{ae} is Lyapunov unstable if at least one of the eigenvalues of the matrix (7) is outside of the unit circle.

Note that the matrix (7) is obtained by first linearizing the nonlinear dynamics of the augmented state (6) around x_{ae} and then discretizing it in time.

Proof: It is shown by Lawrence [26], [27] that the equilibrium point x_{ae} is *exponentially stable* if and only if all eigenvalues of the matrix $\hat{G}$ in (7) are inside the unit circle. Since exponential stability implies Lyapunov stability, the first part of the theorem holds, *i.e.*, x_{ae} is Lyapunov stable if all eigenvalues of $\hat{G}$ are inside the unit circle. The second

part of the theorem cannot be derived from [26], [27] since exponential stability is a stronger notion than Lyapunov stability. For the sake of completeness, the proof of the second part is presented in the appendix and is based on Lyapunov's *indirect* method. ■

We acknowledge that similar theorems exist in the literature separately for both cases of linearization of systems and time-discretization of systems. However, we have been unable to find a proof of Lyapunov instability of sampled-data systems based on their linearized time-discretized versions. Therefore, a proof of this is provided in the appendix.

Stability test. The characteristic polynomial $P(s)$ of matrix $\hat{G}$ in (7) is

$$P(s) = \det(sI - \hat{G}) = 0,$$

which is a polynomial whose coefficients depend on the choice of parameters $\mathbf{p}$ for the digital controller (1) in either of the representations (4)-(5). As we have shown in Theorem 1, if this polynomial has a root s outside unit circle, *i.e.*, $\|s\| > 1$, then the closed-loop SDSS is Lyapunov unstable, so we can eliminate that controller from the synthesis domain. If all the roots are inside the unit circle, the closed-loop SDSS is Lyapunov stable, so we keep that controller for further analysis. Note that if the polynomial does not have any roots outside unit circle but have roots on the unit circle, we still discard the controller since it lies at the boundary of the stability domain and its stability cannot be decided using the above general test.

Remark 1: The assumption of a constant reference signal r made in this article can be generalized by having a 'reference model' that has known dynamics and a bounded input (see e.g., [30]). The output of the reference model gives a time-varying reference signal for tracking purposes. In this case, the dynamics of the reference model can be included in our analysis and other stronger notions of stability can be utilized. Here we assume constant reference signal to provide a succinct presentation of our results.

IV. DIGITAL CONTROLLER SYNTHESIS

In this section, we define the digital controller synthesis problem for SDSS using the same notation of Definition 1. Recall that we consider controller parameters $\mathbf{p} \in \mathbb{P}$ (where $\mathbb{P}$ is a hyperbox of appropriate dimension) and that Θ_T in (2) is the joint distribution of the underlying probability space for trajectories of the SDSS up to time T. For clarity, let us denote with $x(\mathbf{p}, \theta, t)$ the state of the SDSS at time t under controller parameters $\mathbf{p} \in \mathbb{P}$ and stochastic variables $\theta \sim \Theta_T$. We consider time-bounded safety properties of the form $\mathrm{G}_{[0,T]}\psi$, where $\psi : \mathbb{R}^n \to \mathbb{B}$ is a predicate that assigns either **true** or **false** to the SDSS state vector. In particular, ψ describes the *invariant* that the SDSS must satisfy at every time point, and is described as a quantifier-free first-order-logic formula over the theory of nonlinear real arithmetic. The semantics of the safety properties is defined with respect to parameters $\mathbf{p} \in \mathbb{P}$ and stochastic variables $\theta \sim \Theta_T$ as follows:

$$(\mathbf{p}, \theta) \models \mathrm{G}_{[0,T]}\psi \iff \forall t \in [0, T] \;\; \psi(x(\mathbf{p}, \theta, t)). \quad (9)$$

Boolean combinations of safety properties follow the usual interpretation. We remark that our focus is on safety properties, and hence we do not consider the full range of temporal logic operators.[2]

For a time-bounded safety property $\mathrm{G}_{[0,T]}\psi$, we denote by $Pr(\mathbf{p}, \psi)$ the probability with respect to Θ_T that the property is satisfied by the SDSS instantiated with parameters $\mathbf{p}$, i.e.,

$$Pr(\mathbf{p}, \psi) = \int f_\theta(z) \cdot \mathbf{1}\left((\mathbf{p}, z) \models \mathrm{G}_{[0,T]}\psi\right) dz, \quad (10)$$

where $\mathbf{1}$ is the indicator function and f_θ is the density function of $\theta \sim \Theta_T$.

We seek to synthesize controller parameters $\mathbf{p}$ for the SDSS such that the safety probability $Pr(\mathbf{p}, \psi)$ is above a given threshold ϑ. In particular, we restrict the search space for $\mathbf{p}$ by assuming a maximum controller degree L (see Definition 2).

Definition 5 (Digital Controller Synthesis): Given an SDSS, a time-bounded safety property $\mathrm{G}_{[0,T]}\psi$, a maximum controller degree L, and a probability threshold $\vartheta \in (0, 1)$, the digital controller synthesis problem is finding the degree l^*

$$l^* = \min\{\, l \mid C(l) \neq \emptyset, l \leq L\}$$

and a controller parameter $\mathbf{p}^* \in C(l^*)$ where

$$C(l) := \{\mathbf{p} \in \mathbb{P}_l \mid Pr(\mathbf{p}, \psi) > \vartheta\}$$

and $\mathbb{P}_l$ is the parameter space of controllers of degree l. We define the feasible set of this controller synthesis problem by $\mathbf{C} := \bigcup_{l \leq L} C(l)$. If $\mathbf{C} = \emptyset$ we say that the problem is *infeasible*.

In general, the above synthesis problem is very hard to solve exactly due to the presence of nonlinearities, ordinary differential equations (ODEs) introduced by the SDSS dynamics, and multi-dimensional integration for computing probabilities. In fact, a decision version of the digital controller synthesis problem (*i.e.*, given $l \leq L$ decide whether $C(l)$ is nonempty), is easily shown to be undecidable. While Satisfiability Modulo Theory (SMT) approaches, *e.g.*, [15], can now in principle handle nonlinear arithmetics via a sound numerical relaxation, their computational complexity is exponential in the number of variables. In particular, our stochastic optimization problem is high-dimensional and currently infeasible for fully formal approaches, but it can be tackled, as done in [44], by replacing the exact probability $Pr(\mathbf{p}, \psi)$ with a statistical estimate, thereby obtaining a Monte Carlo version of Definition 5.

From a statistical viewpoint, the satisfaction of ψ is a Bernoulli random variable with parameter $Pr(\mathbf{p}, \psi)$, which is the true probability that the SDSS with parameter $\mathbf{p}$ satisfies ψ. Let $\boldsymbol{\theta}_K = (\theta_1, \ldots, \theta_K)$ be a finite-dimensional random vector in which each θ_i is independent and identically

[2]Time-bounded reachability $\mathrm{F}_{[0,T]}\psi$ can be expressed as $\neg\mathrm{G}_{[0,T]}\neg\psi$.

distributed as Θ_T. A *sample* of Θ_T is a (non-random) vector $\mathbf{z}_K = (z_1, \ldots, z_K)$ where each observation z_i is a realization of θ_i.

The *Monte Carlo estimator* $\widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$ of $Pr(\mathbf{p}, \psi)$ is the proportion of times θ_i satisfies the safety property:

$$\widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{1}\left((\mathbf{p}, \theta_i) \models \mathrm{G}_{[0,T]}\psi\right). \quad (11)$$

Note that the estimator $\widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$ is a random variable as it depends on the random vector $\boldsymbol{\theta}_K$. Importantly, by the properties of the mean, $\widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$ is an unbiased estimator, meaning that its expected value $\mathbb{E}\left[\widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)\right]$ is equal to the true probability $Pr(\mathbf{p}, \psi)$. Given a sample $\mathbf{z}_K$, i.e., a realization of $\boldsymbol{\theta}_K$, the (non-random) quantity $\widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K)$ denotes the *Monte-Carlo estimate* of $Pr(\mathbf{p}, \psi)$, i.e., a concrete realization of the estimator $\widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$.

To solve the Monte Carlo synthesis problem, we need to decide if $Pr(\mathbf{p}, \psi) > \vartheta$ on the basis of a concrete sample $\mathbf{z}_K$ and estimate $\widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K)$. This corresponds to a statistical hypothesis testing problem, where one tests the null hypothesis $H_0 : Pr(\mathbf{p}, \psi) \leq \vartheta$ against the alternative hypothesis $H_a : Pr(\mathbf{p}, \psi) > \vartheta$. The test rejects H_0 in favor of H_a if it is unlikely that the estimate $\widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K)$ would have been observed if H_0 was true. For any test, there is some probability of committing type-1 errors, i.e., of wrongly rejecting H_0, and type-2 errors, i.e., of failing to reject H_0 even though H_0 is false. This probabilities are induced by sampling, i.e., by the vector of random parameters $\boldsymbol{\theta}_K$. Any test procedure guarantees that the probability of type-1 errors equals an arbitrarily small *significance level* $\alpha \in (0, 1)$ [28]. The *confidence* $c = 1-\alpha$ is the probability that H_0 is correctly rejected.

Below we describe our statistical test by the predicate $\gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha\right)$, which is true iff, on the basis of the estimate $\widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K)$, the test rejects hypothesis $Pr(\mathbf{p}, \psi) \leq \vartheta$ in favor of $Pr(\mathbf{p}, \psi) > \vartheta$ at significance level α. Therefore, in the Monte Carlo version of the synthesis problem, we synthesize controller parameters $\mathbf{p}$ for the SDSS such that the test predicate is true for some given α.

Definition 6 (Digital Controller Synthesis – Monte Carlo): Given an SDSS, a time-bounded safety property $\mathrm{G}_{[0,T]}\psi$, a maximum controller degree L, a probability threshold $\vartheta \in (0, 1)$, a sample $\mathbf{z}_K = (z_1, \ldots, z_K)$ of Θ_T, and a significance level $\alpha \in (0, 1)$, the Monte Carlo digital controller synthesis problem is finding

$$\hat{l}^* = \min\{\, l \mid \hat{C}(l) \neq \emptyset, l \leq L\}$$

and a controller parameter $\hat{\mathbf{p}}^* \in \hat{C}(\hat{l}^*)$ where

$$\hat{C}(l) := \left\{\mathbf{p} \in \mathbb{P}_l \mid \gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha\right)\right\}.$$

For a sample $\mathbf{z}_K$, we define the feasible set of this controller synthesis problem by $\hat{\mathbf{C}} := \bigcup_{l \leq L} \hat{C}(l)$. If $\hat{C} = \emptyset$, then we say that the problem is *infeasible* for $\mathbf{z}_K$.

The following proposition provides a way to decide the test predicate $\gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha\right)$ by using a confidence interval (CI). Given a confidence level $c \in (0, 1)$ and a random i.i.d. vector $\boldsymbol{\theta}_K$, an interval $[a(\boldsymbol{\theta}_K), b(\boldsymbol{\theta}_K)]$ is a c-CI for $Pr(\mathbf{p}, \psi)$ if the probability w.r.t. $\boldsymbol{\theta}_K$ that the interval contains $Pr(\mathbf{p}, \psi)$ is c. Note that the interval endpoints are random as they depend on $\boldsymbol{\theta}_K$. For a concrete sample $\mathbf{z}_K$, we denote by $[a(\mathbf{z}_K), b(\mathbf{z}_K)]$ the corresponding (non-random) interval estimate based on $\mathbf{z}_K$.

Proposition 1: For significance level $\alpha \in (0, 1)$ and sample $\mathbf{z}_K$ of Θ_T, let $[a(\mathbf{z}_K), b(\mathbf{z}_K)]$ be a $(1 - \alpha)$-confidence interval estimate for $Pr(\mathbf{p}, \psi)$. Then, for any $\vartheta \in (0, 1)$,

$$a(\mathbf{z}_K) > \vartheta \Rightarrow \gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha'\right),$$

for some $\alpha' \leq \alpha$.

Proof: The proof is based on standard results in statistical hypothesis testing and is presented in the appendix. ∎

That is, if the lower end of the $(1-\alpha)$-CI estimate for $Pr(\mathbf{p}, \psi)$ is above ϑ, then we can reject the hypothesis $Pr(\mathbf{p}, \psi) \leq \vartheta$ in favor of $Pr(\mathbf{p}, \psi) > \vartheta$ at level at most α.

We remark that solving the Monte Carlo synthesis problem of Definition 12 is simpler than solving the problem of Definition 11, because deciding whether $\mathbf{p} \in \hat{C}(l)$ for a parameter $\mathbf{p}$ only requires performing an hypothesis test, which boils down to computing a CI estimate. The cost of this procedure reduces to that of evaluating the property $(\mathbf{p}, z_i) \models \mathrm{G}_{[0,T]}\psi$ for each individual observation z_i, plus a constant-time operation to derive the interval. On the other hand, the problem of Definition 11 involves precise integration of $(\mathbf{p}, z) \models \mathrm{G}_{[0,T]}\psi$ over the probability distribution of $\theta \sim \Theta_T$ as written in (10), which is, clearly, more computationally expensive.

However, even with this simplification, a decision version of the Monte Carlo digital controller synthesis problem (*i.e.*, deciding whether $\hat{C}(l) = \emptyset$ for some sample $\mathbf{z}_K$) remains undecidable when plants with nonlinear ODEs are involved. That is because evaluating $(\mathbf{p}, z_i) \models \mathrm{G}_{[0,T]}\psi$ amounts to solving reachability, which is well known to be an undecidable problem for general nonlinear systems. Hence, as explained in the next Section, one can only solve the Monte Carlo controller synthesis problem *approximately*, in the sense that we might not be always able to return the optimal controller $\hat{l}^*$.

We now discuss the relation between the feasible set of the synthesis problem of Definition 5, $\mathbf{C}$, and that of the Monte Carlo version of Definition 6, $\hat{\mathbf{C}}$.

Proposition 2: Let $\boldsymbol{\theta}_K = (\theta_1, \ldots, \theta_K)$ be the vector representing a random sample of the stochastic SDSS parameters, i.e., where each θ_i is i.i.d. as Θ_T. Then the following statements hold:

i. $\mathrm{Prob}_{\boldsymbol{\theta}_K}\{\hat{\mathbf{C}} \subseteq \mathbf{C}\} = 1 - \alpha$, where α is the significance level chosen in Definition 6 for the test predicate γ.
ii. $\mathrm{Prob}_{\boldsymbol{\theta}_K}\{\mathbf{C} \subseteq \hat{\mathbf{C}}\} = 1 - \beta$, where β is the probability of type-2 errors for γ.

Proof: i. follows from noticing that $\mathbf{p} \in \hat{\mathbf{C}} \Rightarrow \mathbf{p} \notin \mathbf{C}$ can be rewritten as $\gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \boldsymbol{\theta}_K), \alpha\right) \Rightarrow Pr(\mathbf{p}, \psi) \leq \vartheta$, which is a type-1 error. By the same argument, ii. holds because $\mathbf{p} \in \mathbf{C} \Rightarrow \mathbf{p} \notin \hat{\mathbf{C}}$ corresponds to a type-2 error. ∎

Unlike α, the error probability β cannot be controlled *a priori* and depends on the true probability $Pr(\mathbf{p}, \psi)$. However, it is easy to show that β can be reduced by increasing the sample size K or the level α .[3]

V. SYNTHESIS ALGORITHM

In this section, we present an algorithm for approximately solving the Monte Carlo controller synthesis problem of Definition 12. Our synthesis algorithm starts from controllers with degree $l = 0$ and iteratively increases l until the constraint $\hat{C}(l) \neq \emptyset$ is satisfied or l reaches a maximum value.

The synthesis algorithm, summarized in Algorithm 1, consists of two nested loops. The inner loop (lines 4–10) comprises two stages: optimization and verification. Procedure **optimize** (line 5) aims at finding parameters $\mathbf{p}$ that maximize the empirical probability that the closed-loop system with a discrete-time approximation of the plant satisfies the safety specification; **optimize** also returns an approximate confidence interval (CI) $[a, b]$ for such probability. This CI is approximate since it is derived using an approximation of the true closed-loop system. Then, procedure **verify** (line 6) checks the candidate controller $\mathbf{p}$ in closed-loop with the *original* (continuous-time) plant model and computes a precise CI $[a', b']$ for $Pr(\mathbf{p}, \psi)$ (11). We use the continuous-time plant only in **verify** because of the high computational complexity of validated ODE solving compared to solving its discrete-time approximation. The interval returned by **verify** is compared with the current best precise interval $[a^*, b^*]$, which is then updated accordingly (line 7).

Note that we use the lower bound of the CI as the optimization objective, because, by Proposition 1, this being above ϑ ensures that the safety probability of the synthesized controller is above ϑ with confidence $1 - \alpha$.[4]

The procedures **optimize** and **verify** are iterated until the approximate CI (for the discrete-time plant) $[a, b]$ and the verified CI (for the continuous-time plant) $[a', b']$ overlap to a sufficient length, or $a^* > \vartheta$ (line 10), or a maximum number of iterations is reached. The first condition implements an heuristics aimed at terminating early when the discrete-time plant and the continuous-time plant essentially behave the same, and thus a finer discretization is not needed. If $a^* > \vartheta$ holds, then the current best parameter $\mathbf{p}^*$ is a witness for $\hat{C}(l) \neq \emptyset$. Therefore, $\mathbf{p}^*$ approximately solves the digital controller synthesis problem of Definition 12. (The approximation lies in the fact that we cannot guarantee that the synthesized controller has minimum degree.) Thus, the algorithm terminates returning $\mathbf{p}^*$. Otherwise, if $a^* \leq \vartheta$, we increase the controller degree l (line 11) up to a maximum degree L and proceed with another iteration of the outer loop.

[3]Assuming to use the precise binomial test for proportions [9], $\beta \leq F_{B(K,Pr(\mathbf{p},\psi))}\left(F^{-1}_{B(K,\vartheta)}(1-\alpha)-1\right)$, where $F_{B(n,p)}$ is the cumulative distribution function of the binomial distribution with parameters n and p, and $F^{-1}_{B(n,p)}$ is its inverse.

[4]On the contrary, using the midpoint of the interval does not provide any statistical guarantee about the satisfaction of our specification.

In the inner loop, line 9 improves the approximation of the closed-loop system used in **optimize**. This can be any adjustment to the ODE solver complexity (*e.g.*, increasing the Taylor series order). In our case, it corresponds to increasing the number of time points used for ODE integration. We next explain both **optimize** and **verify** in more detail.

Algorithm 1 Main Synthesis Algorithm

Input: S – SDSS, $L \geq 0$ – maximum controller degree, $\mathbb{P} = [c_0, d_0] \times \cdots \times [c_{2L}, d_{2L}]$ – parameters domain, ϑ – probability threshold, $\mathbf{m}$ – initial solver discretization, $\varepsilon \in (0, 1)$ – factor for tuning $\mathbf{m}$ (interval overlap), ξ – confidence interval size, c – confidence value.

Output: $\{\mathbf{p}^*, [a^*, b^*]\}$ - best performing controller.

1: $[a^*, b^*] := [0, 0]$; $l := 0$
2: **repeat**
3: $\quad \mathbb{P}_l := [c_0, d_0] \times \cdots \times [c_{2l}, d_{2l}]$
4: $\quad$ **repeat**
5: $\quad\quad \{\mathbf{p}, [a, b]\} :=$ **optimize**$(S, \mathbb{P}_l, \mathbf{m}, \xi, c)$
6: $\quad\quad [a', b'] :=$ **verify**$(S, \mathbf{p}, \xi, c)$
7: $\quad\quad$ **if** $a' > a^*$ **then** $\mathbf{p}^* := \mathbf{p}$; $[a^*, b^*] := [a', b']$
8: $\quad\quad$ **end if**
9: $\quad\quad \mathbf{m} :=$ **update_discretization**$(\mathbf{m})$
10: $\quad$ **until** $|[a, b] \cap [a', b']| \geq \varepsilon(b - a)$ or $a^* > \vartheta$ or *MAX_ATTEMPTS*
11: $\quad l = l + 1$;
12: **until** $l > L$ or $a^* > \vartheta$
13: **return** $\{\mathbf{p}^*, [a^*, b^*]\}$

a: PROCEDURE OPTIMIZE

The procedure, described in Algorithm 2, implements a modified cross-entropy (CE) optimization algorithm [38] which works as follows: 1) a set of controller parameters is drawn randomly from a normal distribution (whose mean is initially set to the center of the parameter domain); 2) a CI for the safety probability is computed for each sampled parameter, and only the best ones (the 10% with highest safety probabilities) are used for updating the mean and the variance of the normal distribution used for sampling controller parameters. Multiple iterations of steps 1) and 2) allow the CE distribution to approach parameter values that maximize the safety probability. CE is an effective black-box optimization algorithm, which has been shown to converge to locally optimal solutions.[5]

After sampling a controller $\mathbf{p}$, we first apply the stability check of Theorem 1 (line 7). If $\mathbf{p}$ does not pass the test, *i.e.*, it is necessarily unstable, it is rejected. Otherwise, we compute a CI for $Pr(\mathbf{p}, \psi)$. For this purpose, we consider a discrete-time version of the plant, where the time interval between the controller sampling points (defined by τ – see Definition 1) is discretized uniformly using $\mathbf{m}$ time points. Then the model ODEs are evaluated by an approximate ODE

[5]Making it more advanced than plain random sampling. Our approach, however, can be easily adapted to other black-box optimization algorithms.

Algorithm 2 $\{\mathbf{p}, [a, b]\} :=$ **optimize**$(S, \mathbb{P}_l, \mathbf{m}, \xi, c)$

Input: $S, \mathbb{P}_l, \mathbf{m}, \xi, c$
Output: $\mathbf{p}, [a, b]$

▷ Modified Cross-Entropy (CE) algorithm
1: $\mathbf{p}^* := \perp$; $[a^*, b^*] := [0, 0]$
2: **repeat**
3: $Q := \{(\mathbf{p}^*, [a^*, b^*])\}$ ▷ queue of samples
4: **repeat**
5: $\mathbf{p}$:= sample parameters from CE distribution
6: **if** ($\mathbf{p}$ passes stability check) **then**
7: ▷ Theorem 1
8: $[a, b]$:= confidence interval, with size ξ and confidence c, for probability of satisfying ψ with plant discretization $\mathbf{m}$ and controller $\mathbf{p}$
9: **else** $[a, b] := [0, 0]$
10: **end if**
11: $Q := Q \cup \{(\mathbf{p}, [a, b])\}$ ▷ add sample to Q
12: **until** *MAX_SAMPLES*
13: $(\mathbf{p}^*, [a^*, b^*]) := \arg\max \{a \mid (\mathbf{p}, [a, b]) \in Q\}$
14: update CE distribution using $tail(Q)$
15: ▷ using the best *MAX_SAMPLES* $- 1$ samples
16: $Q := \emptyset$ ▷ empty queue
17: **until** *MAX_ITERATIONS*
18: **return** $\{(\mathbf{p}^*, [a^*, b^*])\}$

solver (based on the first term of the Taylor series expansion) only at the points of the obtained grid $0 = t_1 < \cdots < t_{\mathbf{m}} = \tau$. (The safety property is also checked at these time points only.) If there is a discrepancy between the CIs produced by procedures **optimize** and **verify**, the discretization granularity $\mathbf{m}$ is increased, and procedure **optimize** is executed again.

To compute the CI, we use sequential Bayesian estimation for efficiency reasons [44], but other standard statistical techniques may also be employed (*e.g.*, the Chernoff-Hoeffding bound). After an adequate number of controller parameters are sampled and evaluated, the best performing sample is chosen (line 13), and the CE distribution is updated accordingly (line 14, see [38] for more details). This is repeated until a maximum number of iterations is reached.

b: PROCEDURE VERIFY

We use the ProbReach tool [43] to compute a CI for $Pr(\mathbf{p}, \psi)$ (line 6 of Algorithm 1). This step is necessary since the candidate controller has been obtained using an approximate, discrete-time solver for simulating the continuous plant dynamics, while ProbReach uses instead an SMT solver [16] to handle the plant dynamics in a sound manner. In particular, ProbReach allows to derive a *numerically* and *statistically* valid confidence interval.

Procedure **verify** consists of two steps. The first step builds a hybrid system (the model format accepted by ProbReach) representing the closed-loop system under the candidate controller. In the second step, **verify** invokes ProbReach with three parameters: the hybrid system, and the required minimum size ξ and confidence c of the confidence interval to compute. We remark that the size of the confidence interval cannot be guaranteed in general [44] because of the undecidability of reasoning about nonlinear arithmetic. The confidence interval returned by ProbReach via **verify** can be fully trusted from both the statistical and numerical viewpoints: while the interval size might be larger than ξ, the confidence is guaranteed to be at least c, as the sampled controllers are evaluated by SMT and verified numerical techniques, and the confidence interval is computed exactly without relying on approximation techniques. We stress that, by relying on verified integration techniques, we have guaranteed bounds on the ODE solution regardless of the sampling period, which means that we have a provably correct method to assess safety of each deterministic state trajectory explored.

Theorem 2: Let S be an SDSS for which the synthesis problem of Definition 5 is feasible for a given $\vartheta \in (0, 1)$ and minimum controller degree l^*. Suppose that Algorithm 1, with parameters $S, \vartheta, L \geq l^*$ and $c \in (0, 1)$, returns a controller $\mathbf{p}$ of degree l, and an interval $[a, b]$ such that $a > \vartheta$. Then, with some probability $c' \geq c$ w.r.t. sampling, $l^* \leq l$ and $\mathbf{p}$ is in the feasible set of the controller synthesis problem of Definition 5 for S, ϑ, and L.

Proof: It suffices to note that if Algorithm 1 terminates by finding a controller $\mathbf{p}$ of degree l and an associated interval $[a, b]$ with $a > \vartheta$, then by Proposition 1, the test predicate $\gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha'\right)$ holds for some $\alpha' \leq 1 - c$, where $\mathbf{z}_K$ is the sample of the stochastic SDSS variables used to compute the CI. This in turn implies that $\mathbf{p}$ is in $\mathbf{C}$ with probability (w.r.t. sampling) at least c, for the same argument made in the proof of Proposition 2. Now note that $l < l^*$ corresponds to saying that $\mathbf{p}$ is in the feasible set of the Monte Carlo synthesis problem of Definition 6 with $L = l$ and that $\mathbf{p}$ is not in the feasible set of the original synthesis problem of Definition 5 with $L = l$. By the same argument as above, this happens with probability at most $1 - c$, and hence $l \geq l^*$ with probability at least c. ■

Remark 2: Note that synthesizing a controller for the SDSS that makes the closed-loop system stable and guarantees satisfaction of the specification is a difficult task. Our approach proposes an interplay between these two requirements. An easy-to-check sufficient condition for Lyapunov stability is used in the first stage of Algorithm 1 to keep only a subset of those controllers that make the closed-loop system stable. Then, only the controllers that satisfy this condition are checked against the second requirement. This makes our method very efficient, as it excludes from the synthesis a consistent portion of irrelevant controller parameters, i.e., those that failed our stability check. Finally, our algorithm ensures safety and Lyapunov stability of the SDSS with the synthesized controller.

Remark 3: We can estimate the amount of samples required to obtain a controller of degree l in the worst case scenario. Assuming that *MAX_ATTEMPTS* in Algorithm 1 and *MAX_ITERATIONS* and *MAX_SAMPLES* in Algorithm 2

are constant, it will take $(l+1) \cdot MAX\_ATTEMPTS \cdot (MAX\_ITERATIONS \cdot MAX\_SAMPLES \cdot n + m)$ samples to synthesize a controller of degree l, where n is the number of sample evaluations required to compute a confidence interval (line 7 of Algorithm 2) for the given controller parameters, and m is the number of sample evaluations required by procedure **verify** to compute a confidence interval. Thus, with the assumptions above the number of samples grows linearly with the controller degree. However, the amount of time required to evaluate every sample inside the optimization algorithm and procedure **verify** varies, since the solution of the model's ODEs depends on the sampled values.

VI. FINITE-PRECISION CONTROLLER IMPLEMENTATION

In practice, digital controllers are implemented by some finite-precision hardware, which could result in unexpected, erroneous behavior. We show that the safety guarantees of the controllers found by our approach are valid under finite-precision implementations. The ProbReach tool (which we use to compute CIs for the satisfaction probability of the closed-loop system) overapproximates the system dynamics by exploiting interval methods, in which interval bounds are variables of a fixed type (*e.g.*, `double`). Now, any digital controller (as per Definition 2) operating over a finite time horizon is essentially a finite sum of basic arithmetic operations. Given a float type $\mathcal{T}$, the behavior of an implementation of the controller using variables of type $\mathcal{T}$ is overapproximated by the interval version of that controller with interval bounds of type $\mathcal{T}$. Thus, it follows that the probability of *avoiding* a 'bad' region by the system in closed loop with the interval controller is a *lower bound* for the probability with any floating-point implementation of the controller (with the same precision $\mathcal{T}$).

Theorem 3: Let $\widehat{Pr}^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$ and $\widehat{Pr}^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$ be the MC estimator (11) computed using a controller implemented by floating-point operations of type $\mathcal{T}$ and via interval arithmetic and interval bounds of type $\mathcal{T}$, respectively. Then

$$\text{Prob}_{\boldsymbol{\theta}_K}\{\widehat{Pr}^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) > \vartheta\} \geq \text{Prob}_{\boldsymbol{\theta}_K}\{\widehat{Pr}^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) > \vartheta\}$$

where $\boldsymbol{\theta}_K = (\theta_1, \ldots, \theta_K)$ are K i.i.d. random variables distributed as Θ_T.

Proof: Let $x^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \theta, t)$ be the state trajectory of the closed-loop system under a floating-point implementation of a controller using type $\mathcal{T}$, and let $x^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \theta, t)$ be the *set* of state trajectories obtained via interval arithmetic and interval bounds of type $\mathcal{T}$ in ProbReach. By (9) and (11) we write:

$$\widehat{Pr}^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{1}\left\{\forall t \in [0,T] \;\; \psi(x^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \theta_i, t))\right\},$$

$$\widehat{Pr}^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{1}\left\{\forall t \in [0,T],\; \forall x \in x^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \theta_i, t)\, \psi(x)\right\}$$

with $\mathbf{1}\{\cdot\}$ being the indicator function. Since x_{ia} is an interval arithmetic extension with endpoints in $\mathcal{T}$ of the trajectory and x_{fp} is the trajectory implemented with floating-point type $\mathcal{T}$, then $\forall t \in [0,T]\; x^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \theta, t) \in x^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \theta, t)$ which implies that

$$\mathbf{1}\left\{\forall t \in [0,T] \;\; \psi(x^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \theta_i, t))\right\} \geq \mathbf{1}\left\{\forall t \in [0,T],\; \forall x \in x^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \theta_i, t) \;\; \psi(x)\right\},$$

and thus $\widehat{Pr}^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) \geq \widehat{Pr}^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K)$ for all $\mathbf{p}$, ψ and $\boldsymbol{\theta}_K$. In particular,

$$\{\boldsymbol{\theta}_K : \widehat{Pr}^{\mathcal{T}}_{\text{fp}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) > \vartheta\} \supseteq \{\boldsymbol{\theta}_K : \widehat{Pr}^{\mathcal{T}}_{\text{ia}}(\mathbf{p}, \psi, \boldsymbol{\theta}_K) > \vartheta\},$$

which implies the thesis. ■

In our implementation $\mathcal{T}$ is `double`, hence the LHS of the confidence intervals computed by ProbReach is effectively a lower bound for the LHS of the confidence intervals obtained by using any double-precision controller implementation (under the same sample set, obviously). This means that our synthesis approach already takes into account any error due to floating-point approximation in a deployed controller.

VII. CASE STUDIES AND EVALUATION

We evaluate our approach on three case studies: a model of insulin control for Type 1 diabetes (T1D) [19], also known as the artificial pancreas, a model of a powertrain control system [20], and a quadruple-tank process. For all case studies we use the following input parameters for Algorithm 1: $\xi = 0.05, c = 0.99$ (obtained with 99%-confidence intervals) and $\varepsilon = 0.5$. The parameter domains were selected after a preliminary evaluation of our algorithm on larger domains, to avoid parameter regions yielding numerous unstable and unsafe controllers. The experiments were performed on a 32-core Intel 2.90GHz system running Ubuntu.

The considered case studies were also evaluated using MATLAB toolboxes for PID tuning. All the models were linearized around the steady state value and discretized with the controller sampling rate. The SISO systems (*i.e.*, artificial pancreas and powertrain) were evaluated using the PID Tuner toolbox (`pidTuner` function), and the quadruple-tank model was evaluated by the Robust Control toolbox (`hinfstruct` function). The obtained parameter values were then checked for safety using **verify** procedure.

Digital PID controllers While our algorithm can synthesize any digital controller as per Definition 2, we here exemplify its use via proportional-integral-derivative (PID) controllers, one of the most popular control techniques. A PID controller output is the weighted sum of three terms: the error itself weighted with K_P, its rate of change weighted with K_D, and accumulated error weighted with K_I. The input/output equation of a digital PID controller is

$$u(k) = u(k-1) + K_P\,[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)]. \quad (12)$$

Essentially, the controller needs to store the previous value of the input and the previous two values of the error. In the following case studies, we focus on the synthesis of controllers in the PID form, hence we consider a maximum degree $L = 2$.

TABLE 1. Controller synthesis for the artificial pancreas system. l – controller degree, K_P, K_I, K_D – controller gains, $[a^*, b^*]$ – 99%-confidence interval for safety probability, $\#_o(\#_o^0)$ – number of points used by the non-verified ODE solver at the end (beginning) of iteration, $\#_c(\#_{un})$ – number of candidates (unstable) sampled by the optimization algorithm, CPU(opt) – total (only optimize procedure) runtime in minutes.

l	$10^3 \times K_P$	$10^7 \times K_I$	$10 \times K_D$	$[a^*, b^*]$	$\#_o(\#_o^0)$	$\#_c(\#_{un})$	CPU(opt)
0	-5.006	-	-	[0.938,0.988]	8(1)	165 (0)	1,130 (586)
1	-5.4	-2.179	-	[0.939,0.989]	8(8)	217 (0)	1,534 (873)
2	-5.716	-1.88	-2.002	[0.942,0.992]	8 (8)	301 (0)	2,100 (1,312)

A. ARTIFICIAL PANCREAS

The artificial pancreas (AP) is a system for the automated delivery of insulin, which is required to keep blood glucose (BG) levels of diabetic patients within safe ranges, typically between 4-11 mmol/L. A *continuous glucose monitor* (CGM) sends BG measurements to a control algorithm that computes the adequate insulin input. PID control is one of the main techniques used for this purpose [49], and is also found in commercial devices [22].

Meals are the major disturbance in insulin control, which make fully closed-loop control challenging. Our approach is therefore well suited to solve this problem because it can synthesize controllers attaining arbitrary safety probability by minimizing the impact of such disturbances. To model insulin and glucose dynamics, we employ the nonlinear model of Hovorka *et al.* [19], considered as one of the most faithful models. The plant has 9 state variables describing insulin and glucose concentration in different physiological compartments. We evaluate the system for a time bound of 24 hours.

In our SDSS model, we consider three meals (breakfast, lunch and dinner) with random timing and random amount, expressed by the following normally-distributed parameters: the amount of carbohydrates (CHO) of each meal in grams, $D_{G_0} \sim \mathcal{N}(50, 100)$, $D_{G_1} \sim \mathcal{N}(70, 100)$ and $D_{G_2} \sim \mathcal{N}(60, 100)$, and the waiting times between meals, $T_1 \sim \mathcal{N}(300, 100)$ and $T_2 \sim \mathcal{N}(300, 100)$. The corresponding disturbance input is given by:

$$d(t) = \{D_{G_0} \text{ if } t \in [0, T_1);\ D_{G_1} \text{ if } t \in [T_1, T_2);\\ D_{G_2} \text{ if } t \in [T_2, 1440];\ 0 \text{ otherwise}\}.$$

The system output $y(t)$ is the CGM measurement (performed every 5 minutes), given by the equation $y(t) = C(t) + \eta(t)$, where C is the state variable for interstitial glucose and $\eta(t)$ is white Gaussian sensor noise with standard deviation 0.25.

The control input $u(t)$ is the insulin infusion rate computed by the PID controller. The tracking error is defined as $e(t) = r - y(t)$ with the constant reference signal $r = 6.11$ mmol/L. The total infusion rate is given by $u(t) + u_b$ where u_b (≈ 0.05548) is the basal insulin, i.e., a low and continuous dose to regulate glucose outside meals. The value of u_b is chosen to guarantee a steady-state BG value equals to r in absence of meals. This steady state is used as the initial state of the system.

c: SAFETY PROPERTY

Insulin control seeks to prevent *hyperglycemia* (BG above 11 mmol/L) and *hypoglycemia* (BG below 4 mmol/L). Hypoglycemia happens when the controller overshoots the insulin dose, and has more severe health effects than hyperglycemia, which is tolerated to a small extent after meals. For this reason we consider a safe BG range of [4, 16] mmol/L, which strictly avoids hypoglycemia and allows for some post-meal hyperglycemia tolerance. In addition, we want that the glucose level stays close to the reference signal towards the end of the 24 hours (1,440 minutes). Our invariant is given by:

$$G \in [4, 16] \wedge (t \in [1, 410, 1, 440] \rightarrow G \in [r - 0.25, r + 0.25]) \tag{13}$$

where G is the state variable for the BG concentration and $r = 6.11$ is the (constant) reference signal.

In the synthesis algorithm, we use a probability threshold of $\vartheta = 0.95$ (we want to satisfy the above invariant with probability at least 95%).

d: SYNTHESIS RESULTS

Table 1 shows the PID controllers synthesized at each iteration of the algorithm. The domain of controller parameters was chosen as follows: $K_P \in [-10^{-2}, 10^{-3}]$, $K_I \in [-10^{-5}, 10^{-6}]$ and $K_D \in [-1, 10^{-1}]$. Even though none of the synthesized controllers achieves the probability threshold $\vartheta = 0.95$, the degree-2 controller (PID) is very close to satisfying the property, with a 99%-confidence interval of [0.94242, 0.99242]. Also, note that no unstable controller was explored during the synthesis – see column $\#_c(\#_{un})$ in Table 1.

To better understand the performance of the controllers, we analyze their behavior on 1,000 Monte Carlo executions of the system. The results, reported in Figure 3, evidence that hyper- and hypo-glycemia episodes are never sustained.

e: RESULTS VALIDATION

The controller parameter values produced by MATLAB PID tuner and the resulting confidence intervals computed by the **verify** procedure are given in Table 2. It can be seen that the obtained safety probabilities are very low, and the simulations (see Figure 4) evidence that the controller fails to quickly in a timely manner from hypoglycemia and to reach the reference point by the end of the 24-hour period. In contrast, our controller prevents any hypoglycemia episode and successfully drives the system to the reference glucose after the last meal disturbance.

B. POWERTRAIN SYSTEM

We consider the automotive air-fuel control system adapted from the powertrain control benchmark in [20]. The plant

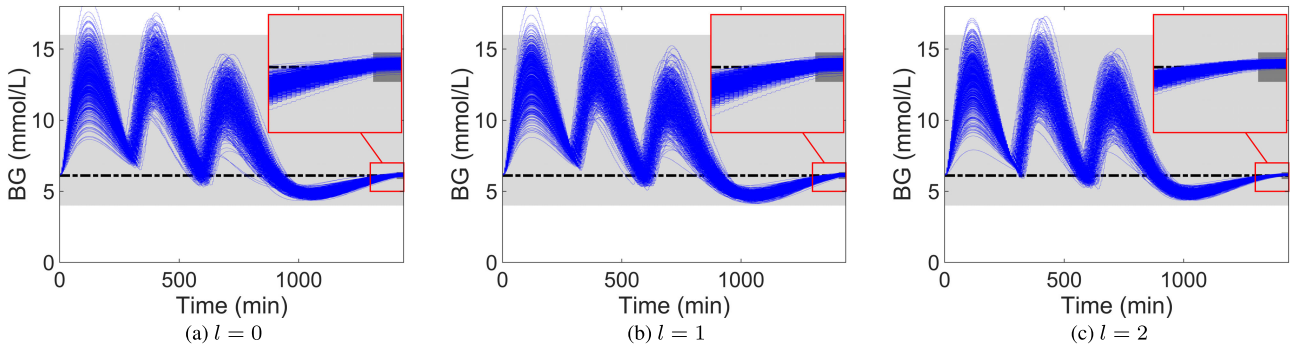


FIGURE 3. Evaluation of synthesized controllers (degrees $l = 0, 1, 2$) on 1,000 simulations of the AP system. Blue lines: BG profiles; light gray areas - healthy BG interval ($G \in [4, 16]$); dark gray areas - tighter interval [5.86, 6.36] where the BG level should remain for the last 30 minutes (see (13)); dashed black lines: reference BG level (6.11).

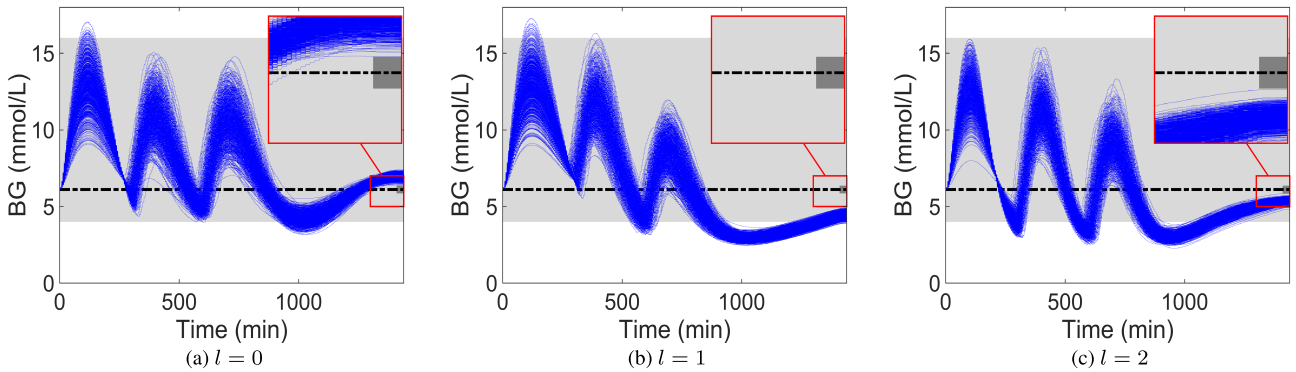


FIGURE 4. Evaluation of controllers obtained with MATLAB's PID tuner toolbox (degrees $l = 0, 1, 2$) on 1,000 simulations of the AP system. Blue lines: BG profiles; light gray areas - healthy BG range ($G \in [4, 16]$); dark gray areas - tighter interval [5.86, 6.36] where the BG level should remain for the last 30 minutes (see (13)); dashed black lines: reference BG level (6.11).

TABLE 2. Controller synthesis results obtained by MATLAB's PID tuner toolbox for the artificial pancreas system. l – controller degree, K_P, K_I, K_D – controller gains, $[a^*, b^*]$ – 99%-confidence interval for safety probability obtained using the verify procedure, CPU – total runtime in minutes for computing the confidence intervals.

l	$10^3 \times K_P$	$10^7 \times K_I$	$10 \times K_D$	$[a^*, b^*]$	CPU
0	-10.4	-	-	[0,0.0267794]	226
1	-5.534	-156.7	-	[0,0.0267953]	127
2	-11.51	-312.5	-10.59	[0,0.0267794]	48

model consists of a system of three nonlinear ODEs describing the dynamics of the engine in relation to the throttle air dynamics, intake manifold and air-fuel path.

The system has two external inputs, captured by the disturbance vector $\mathbf{d}(t) = [\omega(t)\ \theta_{in}(t)]^T$, where ω (rad/s) is the engine speed ($\omega \sim \mathcal{N}(105, 4)$), and θ_{in} (degrees) is the throttle angle. $\theta_{in}(t)$ is defined as a pulse train wave with random amplitude $a \sim \mathcal{N}(30.6, 25)$ and period $\zeta = 4$:

$$\theta_{in}(t) = a\, I\{t \in [0, \zeta/2)\} + 8.8\, I\{t \in [\zeta/2, \zeta]\}.$$

The noisy plant output is $y(t) = \lambda(t) + \eta(t)$, where $\lambda(t)$ is the air/fuel ratio, and $\eta(t) \sim \mathcal{N}(0, 0.0625)$.

The engine is controlled by a PID controller that seeks to maintain a constant air/fuel ratio equals to the stoichiometric value $\bar{\lambda} = 14.7$, which is when the engine performs optimally. The tracking error is thus given by $e(t) = y(t) - \bar{\lambda}$. The control signal $u(t)$ determines the amount of fuel entering the system.

f: SAFETY PROPERTY

We consider the following invariant

$$|\mu(t)| < 1 \wedge (t \in [\zeta/8, \zeta/2] \cup [5\zeta/8, \zeta]) \rightarrow |\mu(t)| < 0.05)$$

where $\mu(t) = (\lambda(t) - \bar{\lambda})/\bar{\lambda}$ is the relative error from the setpoint. The first conjunct states that the air/fuel ratio should constantly be within $\pm 100\%$ of the ideal ratio $\bar{\lambda}$. The second conjunct states that whenever the input throttle angle θ_{in} rises (at time $t = 0$) or falls ($t = \zeta/2$), the plant should settle within time $\zeta/8$ and remain in the settling region ($\pm 5\%$ around $\bar{\lambda}$) until the next rise or fall (happening after time $\zeta/2$). We set the probability threshold to $\vartheta = 0.96$.

g: SYNTHESIS RESULTS

Table 3 shows the PID controllers synthesized at each iteration of the algorithm. The domain of controller parameters was chosen as follows: $K_P \in [-0.1, 0.5]$, $K_I \in [-0.05, 0.2]$ and $K_D \in [-0.05, 0.05]$. With our algorithm, we could synthesize a degree-2 controller (PID) satisfying the threshold. The optimal degree-1 controller has similar performance

TABLE 3. Controller synthesis for the powertrain system. See caption of Table 1.

l	K_P	K_I	$K_D \times 10^3$	$[a^*, b^*]$	$\#_o(\#_o^0)$	$\#_c(\#_{un})$	CPU(opt)
0	0.2713	-	-	[0.783,0.834]	128(64)	74(7)	1068(944)
1	0.2004	0.0537	-	[0.954,1]	128(128)	134(22)	1838(1690)
2	0.2082	0.0759	-4.9551	[0.963,1]	128(128)	214(72)	2337(2165)

(both yield confidence intervals with RHS equals to 1), albeit below the threshold.

Compared to the AP case study, we observe that the powertrain model requires generating (and verifying) fewer candidate parameters, although dozens of them turned out to generate unstable controllers – see column $\#_c(\#_{un})$ in Table 3. At the same time the dynamics of the powertrain system appear more challenging to control as the model requires more ODE integration steps (see column $\#_o(\#_o^0)$ of Tables 3 and 1).

h: RESULTS VALIDATION

The controller parameter values produced by MATLAB and the resulting confidence intervals computed by the **verify** procedure are given in Table 4. It can be seen that MATLAB produced a better controller of degree 0 ([0.925064,0.975064] vs. [0.783,0.834]). However, the controllers of higher degree demonstrate very low safety probability, while our approach produces much safer controllers.

TABLE 4. Controller synthesis results obtained by MATLAB's PID tuner toolbox for the fuel control system. l – controller degree, K_P, K_I, K_D – controller gains, $[a^*, b^*]$ – 99%-confidence interval for safety probability obtained using the verify procedure, CPU – total runtime in minutes for computing the confidence intervals.

l	K_P	K_I	$10^3 \times K_D$	$[a^*, b^*]$	CPU
0	0.2371	-	-	[0.925064,0.975064]	53
1	0.0007582	0.01516	-	[0,0.0267921]	10
2	0.001516	0.01516	379.1	[0,0.0267794]	10

C. QUADRUPLE-TANK PROCESS

We consider a quadruple-tank process adapted from [21], which consists of four interconnected water tanks. The process is illustrated in Figure 5. This model is an example of a multiple-input and multiple-output (MIMO) system with *multivariable right half-plane zeros* [17] (such zeros bring performance limitations in control problems). We extended the deterministic model of [21] to include uncertainties in the valve settings and random disturbances in the process that removes water from the tanks.

The process is controlled in a decentralized fashion, by which two digital controllers are designed for the input-output pairs (u_1, y_1) and (u_2, y_2), where u_1 and u_2 are the input voltages for the pumps, and y_1 and y_2 are the water level measurements obtained as $y_1 = 0.5 \cdot h_1$ and $y_2 = 0.5 \cdot h_2$, where h_1 and h_2 are the water levels in tanks 1 and 2, respectively. In this case study we assume that the pumps can only add water to the tanks (and cannot pump it out).

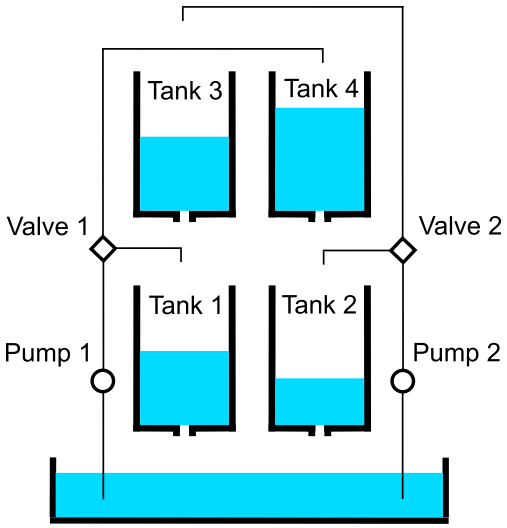


FIGURE 5. The diagram of the quadruple-tank model.

We consider a scenario where at time 0 and then twice after every minute we remove a random amount of water from tanks 1 and 2, reducing the corresponding water levels by a random value $\sim \mathcal{U}(0, 3)$. Every time such a disturbance happens, the valves parameters are randomly reset to $\gamma_1 \sim \mathcal{N}(0.7, 0.223)$ and $\gamma_2 \sim \mathcal{N}(0.6, 0.223)$. The system is subject to a measurement noise modeled as a white Gaussian noise with variance 0.33.

i: SAFETY PROPERTY

After each disturbance, we require that the system reaches the desired water levels in tanks 1 and 2 (within 1 centimeter above or below the corresponding set points $r_1 = 12.4$ and $r_2 = 12.7$) within 5 seconds, and that the water levels stay close to the setpoints for the remaining 55 seconds, before the next disturbance occurs. Also, all four water levels h_1, h_2, h_3, h_4 must always stay in the interval $[0, 20]$ and the input voltages u_1, u_2 for both pumps must be in the range $[0, 24]$.

j: SYNTHESIS RESULTS

The domain of controller parameters was chosen as $K_{P_1} \in [-1, 20]$, $K_{I_1} \in [-1, 10]$, $K_{D_1} \in [-1, 10]$, $K_{P_2} \in [-1, 20]$, $K_{I_2} \in [-1, 10]$, $K_{D_2} \in [-1, 10]$. The controller synthesis results are presented in Table 5, which show that we can obtain a confidence interval of up to $[0.94, 0.99]$ for the safety probability by using two PI controllers (see third row of Table 5). Note that the performance of the controller is not improved by including the derivative terms K_{D_1}, K_{D_2} (see last two rows). This can be attributed to the optimization

TABLE 5. Controller synthesis for the quadruple-tank system. $K_{P_1}, K_{I_1}, K_{D_1}, K_{P_2}, K_{I_2}, K_{D_2}$ – controller gains, $[a^*, b^*]$ – confidence interval for safety probability (with $c = 0.99$), $\#_c(\#_{un})$ – number of candidates (unstable) sampled by the optimization algorithm, CPU(opt) – total (only optimize procedure) runtime in minutes, * – the number of points in discretization was increased from 1 to 2.

K_{P_1}	K_{I_1}	K_{D_1}	K_{P_2}	K_{I_2}	K_{D_2}	$[a^*, b^*]$	$\#_c(\#_{un})$	CPU(opt)
10.916	-	-	13.085	-	-	[0.85,0.90]	58(1)	164(30)
8.463	0.650	-	12.749	-	-	[0.89,0.94]	146(1)	252(73)
6.555	1.233	-	10.057	1.359	-	[0.94,0.99]	251(1)	370(113)
6.576	1.144	1.737	9.019	1.048	-	[0.93,0.98]	507(2)*	717(246)
6.422	1.057	-0.075	6.760	1.724	3.973	[0.90,0.95]	654(2)*	917(346)

algorithm which works by sampling a finite number of controller parameters and thus, might fail to explore parameter regions with better safety probability.

k: RESULTS VALIDATION

For a MIMO system with 2 inputs u_i and 2 outputs e_i there are 4 PID controllers for all possible combinations of inputs and outputs. In matrix form this can be written as:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} PID_{11} & PID_{12} \\ PID_{21} & PID_{22} \end{bmatrix} \times \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}. \tag{14}$$

We remark that in our original case study we simplified the considered controller by assuming that $PID_{12} = 0$ and $PID_{21} = 0$. The following parameters were produced for the quadruple-tank model by MATLAB PID tuner:

$$K_P = \begin{bmatrix} 23.8422 & 0.3915 \\ 0.1842 & 31.6801 \end{bmatrix},$$
$$K_I = \begin{bmatrix} 0.0012 & -0.0039 \\ -0.0156 & 0.0078 \end{bmatrix},$$
$$K_D = \begin{bmatrix} -24.8422 & 0.404 \\ 0.186 & -32.0047 \end{bmatrix}. \tag{15}$$

The corresponding confidence interval [0,0.0267794] was obtained by our **verify** procedure in 3 minutes.

VIII. CONCLUSION

The synthesis of digital controllers for cyber-physical systems with nonlinear and stochastic dynamics is a challenging problem. For such systems, no automated methods currently exist for deriving controllers with rigorous and quantitative safety guarantees. In this article, we have presented a solution to this problem based on two key contributions: an easy-to-check condition for Lyapunov stability of the system; and a two-stage synthesis algorithm that alternates between a fast candidate-generation phase (based on Monte-Carlo sampling and approximate ODE solving), and a verification phase where we derive numerically and statistically valid confidence intervals for the safety probability of the closed-loop system. We applied our method to three nonlinear systems (artificial pancreas, powertrain, and quadruple-tank process) and synthesized controllers that are provably stable and safe. In future, we plan to extend our technique to plants modeled as switched or hybrid systems and to controllers with fixed-point precision. Furthermore, we plan to investigate the use of inductive methods (*e.g.*, barrier certificates) to speed up the safety verification of candidate controllers.

APPENDIX A
PROOF OF STATEMENTS

Outline of the proof of instability in Theorem 1. According to Definition 4, we should show that the following statement holds in order to get Lyapunov instability:

$$\exists \epsilon > 0 \cdot \forall \delta > 0 \cdot \exists x_1(0) \wedge z_1[0] \text{ with } \|(x_1(0), z_1[0])\| \leq \delta \\ \wedge \exists t \geq 0 \text{ with } \|(x_1(t), z_1[k])\| > \epsilon,$$

where $x_1 := x - x_e$ and $z_1 := z - z_e$. Our proof is constructive. We give an $\epsilon > 0$ and properly choose initial conditions $x_1(0), z_1[0]$ that can be made small arbitrarily. The overall aim of the proof is to construct a quadratic function of the augmented state and utilize a few intermediate inequalities to show that this function is monotonically increasing as long as $\|(x_1(t), z_1[k])\| \leq \epsilon$. This contradiction ensures existence of a $t \geq 0$ with $\|(x_1(t), z_1[k])\| > \epsilon$.

Intuition behind the intermediate steps. We first shift the variables around the equilibrium and write down their dynamics in (16) to simplify the notation. We then show in Lemma 1 that the nonlinearity in the dynamics has at most a local linear growth. Then we study in Lemma 2 the deviations in the continuous state inside the sampling intervals from the beginning of the interval. The bound on these deviations is used in Lemma 3 to find a bound for the nonlinear terms over the whole sampling interval. Finally, this bound is used to prove that the constructed function is monotonically increasing.

Let us shift the variables around their equilibrium values and define $x_1 := x - x_e$, $z_1 := z - z_e$, and $u_1 := u - u_e$. The dynamics of x_1, z_1, u_1 are

$$\begin{aligned} \frac{d}{dt} x_1(t) &= \bar{f}(x_1(t), u_1[k]), \quad \forall t \in [t_k, t_{k+1}) \\ z_1[k+1] &= G_c z_1[k] - H_c \bar{o}(x_1[k]) \\ u_1[k] &= C_c z_1[k] - D_c \bar{o}(x_1[k]), \quad k \in \mathbb{Z}^{\geq 0}, \end{aligned} \tag{16}$$

where $\bar{f}(x_1, u_1) := f(x_1 + x_e, u_1 + u_e, 0)$ and $\bar{o}(x_1) := o(x_1 + x_e) - o(x_e)$. Note that we have used the notation $x_1[k] = x_1(k\tau)$ for all k. Thus $x_1 = 0, z_1 = 0$ is an equilibrium point for (16). The matrices A, B, C in (8) can be written as $A = \frac{\partial \bar{f}}{\partial x_1}(0,0)$, $B = \frac{\partial \bar{f}}{\partial u_1}(0,0)$, and $C = \frac{\partial \bar{o}}{\partial x_1}(0)$.

The following three lemmas are relatively standard for analyzing the local behavior of the system at equilibrium. Similar inequalities are utilized in the literature for stability of dynamical systems (e.g., [33]). We provide their proof for the sake of completeness.

Lemma 1: Define $g(x_1, u_1) := \bar{f}(x_1, u_1) - Ax_1 - Bu_1$ and $l(x_1) := \bar{o}(x_1) - Cx_1$. Then, for any $\gamma > 0$ there exists an $\mathsf{r}(\gamma) > 0$ such that

$$\|g(x_1, u_1)\| \leq \gamma\|x_1\| + \gamma\|u_1\| \text{ and } \|l(x_1)\| \leq \gamma\|x_1\|, \quad (17)$$

for all $x_1 \in \mathbb{R}^n$, $u_1 \in \mathbb{R}^m$ with $\|(x_1, u_1)\| \leq \mathsf{r}(\gamma)$.
Note that functions g, l are the nonlinear terms describing the deviation between nonlinear functions f, o and their linearized versions. The next lemma establishes a bound on $x_1(t)$ for any $t \in [k\tau, k\tau + \tau]$, as a function of $x_1[k]$ and $z_1[k]$.

Proof of Lemma 1. Functions g, l are the nonlinear terms describing the deviation between nonlinear functions f, o and their linearized versions. Since f and o are continuously differentiable, g, l are also continuously differentiable with their values and their first derivatives equal to zero. Thus, we have

$$\lim_{\|(x_1,u_1)\|\to 0} \frac{\|g(x_1, u_1)\|}{\|(x_1, u_1)\|} = 0 \quad \text{and} \quad \lim_{\|x\|\to 0} \frac{\|l(x_1)\|}{\|x_1\|} = 0.$$

The definition of limit implies for any $\gamma > 0$, there are $\mathsf{r}_1(\gamma) > 0$ and $\mathsf{r}_2(\gamma) > 0$ such that

$$\frac{\|g(x_1, u_1)\|}{\|(x_1, u_1)\|} \leq \gamma \quad \text{and} \quad \frac{\|l(x_1)\|}{\|x_1\|} \leq \gamma \quad (18)$$

for all $\|(x_1, u_1)\| \leq \mathsf{r}_1(\gamma)$ and $\|x_1\| \leq \mathsf{r}_2(\gamma)$. Set $\mathsf{r}(\gamma) = \min\{\mathsf{r}_1(\gamma), \mathsf{r}_2(\gamma)\}$ and take any $x_1 \in \mathbb{R}^n$ and $u_1 \in \mathbb{R}^m$ such that $\|(x_1, u_1)\| \leq \mathsf{r}(\gamma)$. This means

$$\|(x_1, u_1)\| \leq \mathsf{r}(\gamma) \leq \mathsf{r}_1(\gamma) \quad \text{and}$$
$$\|x_1\| \leq \|(x_1, u_1)\| \leq \mathsf{r}(\gamma) \leq \mathsf{r}_2(\gamma),$$

thus both inequalities in (18) hold. We can use triangle inequality on (18) as follows:

$$\|g(x_1, u_1)\| \leq \gamma\|(x_1, u_1)\| \leq \gamma(\|x_1\| + \|u_1\|) \text{ and}$$
$$\|l(x_1)\| \leq \gamma\|x_1\|,$$

which shows that the inequalities in (17) hold with the chosen $\mathsf{r}(\gamma)$. ■

Lemma 2: Under dynamics (16), there exist two functions h_1, h_2 such that for a given $t \in [k\tau, (k+1)\tau]$ and any $\gamma > 0$, we have

$$\|x_1(t)\| \leq h_1(t - k\tau, \gamma)\|x_1[k]\| + h_2(t - k\tau, \gamma)\|z_1[k]\| \quad (19)$$

if $\|(x_1(t_1), u_1[k])\| \leq \mathsf{r}(\gamma)$ for all $t_1 \in [k\tau, t]$, with $\mathsf{r}(\gamma)$ satisfying property (17). Functions h_1, h_2 are continuous and nonnegative with $h_1(0, \gamma) = 1$ and $h_2(0, \gamma) = 0$.

Proof of Lemma 2. We simplify the dynamics of the closed-loop SDSS as

$$\begin{cases} \frac{d}{dt}x_1(t) = Ax_1(t) + BC_c z_1[k] - BD_c Cx_1[k] \\ \qquad\qquad + g(x_1(t), u_1[k]), \\ z_1[k+1] = G_c z_1[k] - H_c Cx_1[k] - H_c l(x_1[k]). \end{cases} \quad (20)$$

Define the function $W(t) := \|x_1(t)\|^2$,

$$\begin{aligned} \frac{d}{dt}W(t) &= 2\,[Ax_1(t) + BC_c z_1[k] - BD_c Cx_1[k] \\ &\quad + g(x_1(t), u_1[k])]^T x_1(t) \\ &\leq 2\|x_1(t)\|\,[\|A\|\|x_1(t)\| + \|BC_c\|\|z_1[k]\| \\ &\quad + \|BD_c C\|\|x_1[k]\| + \gamma\|x_1(t)\| + \gamma\|u_1[k]\|]\,. \end{aligned}$$

Replace $W(t) =: \sigma^2(t)$ and divide both sides by $\sigma(t)$ to get

$$\frac{d}{dt}\sigma(t) \leq L\sigma(t) + L_1\sigma[k] + L_2\|z_1[k]\| \quad (21)$$

with $L := \|A\| + \gamma$, $L_1 := \|BD_c C\| + \gamma\|D_c C\| + \gamma^2\|D_c\|$, and $L_2 := \|BC_c\| + \gamma\|C_c\|$. Now we apply Grönwall's inequality [2] to (21) over the interval $[k\tau, t]$, which states that $\sigma(t)$ is upper bounded by the solution of the differential equation obtained from replacing inequality in (21) with equality.

$$\Rightarrow \sigma(t) \leq h_1(t - k\tau, \gamma)\sigma[k] + h_2(t - k\tau, \gamma)\|z_1[k]\|,$$

with functions

$$h_1(t, \gamma) := e^{Lt} + (e^{Lt} - 1)L_1/L, \;\; h_2(t, \gamma) := (e^{Lt} - 1)L_2/L, \quad (22)$$

where L, L_1, L_2 depend on γ as defined above. ■

The upper bound (19) enables us to study the effect of the nonlinear terms $g(\cdot)$ and $l(\cdot)$ in the sampled version of the dynamics, which can be written as

$$\begin{aligned} x_1[k+1] &= (\text{G-H}D_c C)x_1[k] - HC_c z_1[k] + \hat{g}[k], \\ z_1[k+1] &= G_c z_1[k] - H_c Cx_1[k] + \hat{l}[k], \end{aligned} \quad (23)$$

with G, H defined in Theorem 1, $\hat{l}[k] := -H_c l(x_1[k])$, and

$$\hat{g}[k] := \int_0^\tau e^{A(\tau-\lambda)} g(x_1(k\tau + \lambda), u_1[k])d\lambda. \quad (24)$$

Next, we derive a bound for $\hat{g}[\cdot]$ in terms of x_1 and z_1.

Lemma 3: For any $\gamma > 0$, there exist continuous functions $\hat{h}_1, \hat{h}_2$ such that the following inequality holds for $\hat{g}[\cdot]$ defined in (24),

$$\|\hat{g}[k]\| \leq \gamma\hat{h}_1(\tau)\|x_1[k]\| + \gamma\hat{h}_2(\tau)\|z_1[k]\| \quad (25)$$

if $\|(x_1(t), u_1[k])\| \leq \mathsf{r}(\gamma)$ for all $t \in [k\tau, (k+1)\tau]$, with $\mathsf{r}(\gamma)$ satisfying property (17). Functions $\hat{h}_1, \hat{h}_2$ are nonnegative with $\hat{h}_1(0) = 0$ and $\hat{h}_2(0) = 0$.

Proof of Lemma 3. Using the assumption of $\|(x_1(t), u_1[k])\| \leq \mathsf{r}(\gamma)$ for all $t \in [k\tau, (k+1)\tau]$, we get

$$\begin{aligned} \|\hat{g}[k]\| &\leq \int_0^\tau \|e^{A(\tau-\lambda)}\|\|g(x_1(k\tau + \lambda), u_1[k])\|d\lambda \\ &\leq \gamma\int_0^\tau \|e^{A(\tau-\lambda)}\|(\|x_1(k\tau + \lambda)\| + \|u_1[k]\|)d\lambda. \end{aligned}$$

Then we employ Lemma 2 to get

$$\begin{aligned} \|\hat{g}[k]\| &\leq \gamma\|x_1[k]\| \\ &\quad \times \int_0^\tau \|e^{A(\tau-\lambda)}\|(h_1(\lambda, \gamma) + \|D_c C\| + \gamma\|D_c\|)d\lambda \\ &\quad + \gamma\|z_1[k]\|\int_0^\tau \|e^{A(\tau-\lambda)}\|(h_2(\lambda, \gamma) + \|C_c\|)d\lambda. \end{aligned}$$

Next we use the fact that for the matrix A, there are constants α and Γ such that $\|e^{At}\|_2 \leq \Gamma e^{\alpha t}$ for all $t \geq 0$ and get

$$\begin{aligned}\hat{h}_1(\tau) &= \int_0^\tau \Gamma e^{\alpha(\tau-\lambda)}(h_1(\lambda,\gamma) + \|D_c C\| + \gamma\|D_c\|)d\lambda \\ &= \Gamma\left[1+\frac{L_2}{L}\right]\left[\frac{e^{L\tau}-e^{\alpha\tau}}{L-\alpha}\right] \\ &\quad + \Gamma\left[\|D_c C\| + \gamma\|D_c\| - \frac{L_2}{L}\right]\frac{e^{\alpha\tau}-1}{\alpha}.\end{aligned}$$

Similarly, the second integral is bounded by

$$\begin{aligned}\hat{h}_2(\tau) &= \int_0^\tau \Gamma e^{\alpha(\tau-\lambda)}(h_2(\lambda,\gamma) + \|C_c\|)d\lambda \\ &= \frac{\Gamma L_1}{L}\left[\frac{e^{L\tau}-e^{\alpha\tau}}{L-\alpha}\right] + \Gamma\left[\|C_c\| - \frac{L_1}{L}\right]\left[\frac{e^{\alpha\tau}-1}{\alpha}\right].\end{aligned}$$

■

We need the following well-known proposition on stability of a matrix in the construction of the quadratic function.

Proposition 3 ([24]): All eigenvalues of a square matrix G are inside the unit circle if and only if there exist positive definite matrices M, Q that satisfy $G^T MG - M = -Q$.

Construction of a monotonically increasing function. Suppose $\hat{G}$ in (7) has at least one eigenvalue outside the unit circle. We cluster the eigenvalues of $\hat{G}$ into a group of eigenvalues outside the unit circle and a group of eigenvalues on or inside the unit circle. Then there is a nonsingular matrix T such that

$$T\hat{G}T^{-1} = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix},$$

where G_1 contains all of the eigenvalues of $\hat{G}$ from the first group and G_2 has the remaining eigenvalues. The matrix T can be found for instance by transforming $\hat{G}$ into its real Jordan form. Now define $\mu > 0$ by

$$1 + 2\mu = \min_i |\lambda_i(G_1)|.$$

Then both $G_2/(1+\mu)$ and $(1+\mu)G_1^{-1}$ have their eigenvalues inside the unit circle. According to Proposition 3, there are positive definite matrices M_1, M_2 and Q_1, Q_2 such that the following matrix equalities hold

$$\begin{aligned}(1+\mu)^2 G_1^{T^{-1}} M_1 G_1^{-1} - M_1 &= -Q_1, \\ G_2^T M_2 G_2/(1+\mu)^2 - M_2 &= -Q_2.\end{aligned} \tag{26}$$

Define the quadratic function

$$V[k] := x_s[k]^T T^T \begin{bmatrix} M_1 & 0 \\ 0 & -M_2 \end{bmatrix} T x_s[k], \tag{27}$$

with $x_s[k] = x_s(k\tau)$ being the sampled shifted augmented state

$$x_s(t) := \begin{bmatrix} x_1(t) \\ z_1[k] \end{bmatrix}, \quad \forall t \in [t_k, t_{k+1}),\ k \in \mathbb{Z}^{\geq 0}.$$

Note that $V[\cdot]$ is positive only on a subset of the augmented state space.

Lemma 4: The function $V[\cdot]$ in (27) satisfies the inequality

$$V[k+1] \geq (1+\mu)^2\, V[k] + (c_0 - c_1\gamma - c_2\gamma^2)\|Tx_s[k]\|^2,$$

for any $\gamma > 0$ as long as $\|(x_1(t), u_1[k])\| \leq \mathsf{r}(\gamma)$. The constants c_0, c_1, c_2 are

$$\begin{aligned} c_0 &:= \min_{i,j}\{\lambda_i(\bar{Q}_1), \lambda_j(\bar{Q}_2)\}, \\ c_1 &:= 2\varkappa\left(\|M_1 G_1\| + \|M_2 G_2\|\right), \\ c_2 &:= 2\varkappa^2 \lambda_{max}(M_2), \end{aligned}$$

where $\bar{Q}_1 := G_1^T Q_1\, G_1$, $\bar{Q}_2 := (1+\mu)^2\, Q_2$, and

$$\varkappa := \|T\|\|T^{-1}\|\sqrt{\hat{h}_1(\tau)^2 + \hat{h}_2(\tau)^2 + 1},$$

with $\hat{h}_1(\tau)$ and $\hat{h}_2(\tau)$ defined in (25).

Proof of Lemma 4. Define a transformation of x_s under T by v as

$$v := Tx_s = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} T_1 x_s \\ T_2 x_s \end{bmatrix},$$

where the partitions of v and T are compatible with the dimensions of G_1 and G_2. Using (23), we get the dynamics of $v[\cdot]$ as

$$v[k+1] = T\hat{G}T^{-1}v[k] + T\begin{bmatrix}\hat{g}[k] \\ \hat{l}[k]\end{bmatrix} = \begin{bmatrix} G_1 v_1[k] + \hat{g}_1[k] \\ G_2 v_2[k] + \hat{g}_2[k]\end{bmatrix}.$$

We have

$$\begin{aligned} &V[k+1] \\ &= v_1[k+1]^T M_1\, v_1[k+1] - v_2[k+1]^T M_2\, v_2[k+1] \\ &= v_1[k]^T (G_1^T M_1\, G_1) v_1[k] + 2\hat{g}_1[k]^T M_1 (G_1\, v_1[k] + \hat{g}_1[k]) \\ &\quad - v_2[k]^T (G_2^T M_2\, G_2) v_2[k] - 2\hat{g}_2[k]^T M_2 (G_2\, v_2[k] + \hat{g}_2[k]) \end{aligned}$$

According to (26) and definitions of $\bar{Q}_1, \bar{Q}_2$, we have

$$\begin{aligned} G_1^T M_1\, G_1 &= (1+\mu)^2\, M_1 + \bar{Q}_1 \\ G_2^T M_2\, G_2 &= (1+\mu)^2\, M_2 - \bar{Q}_2. \end{aligned}$$

Then,

$$\begin{aligned} V[k+1] = (1+\mu)^2\, v_1^T M_1\, v_1 + v_1^T \bar{Q}_1\, v_1 + \mathcal{A}_1 \\ - (1+\mu)^2\, v_2^T M_2\, v_2 + v_2^T \bar{Q}_2\, v_2 + \mathcal{A}_2. \end{aligned} \tag{28}$$

The terms in $\mathcal{A}_1, \mathcal{A}_2$ can be bounded using definition of $\hat{g}_1, \hat{g}_2$ as a function of $\hat{g}$ and Lemma 3. For $\mathcal{A}_2$ we have

$$\begin{aligned} \mathcal{A}_2 &:= 2\hat{g}_1[k]^T M_1 \hat{g}_1[k] - 2\hat{g}_2[k]^T M_2 \hat{g}_2[k] \\ &\geq -2\lambda_{max}(M_2)\|\hat{g}_2[k]\|^2 \\ &\geq -2\lambda_{max}(M_2)\|T\|^2\left(\|\hat{g}[k]\|^2 + |\hat{l}[k]\|^2\right) \\ &\geq -2\lambda_{max}(M_2)\gamma^2\varkappa^2\|v\|^2 \\ &= -c_2\gamma^2\|v\|^2. \end{aligned}$$

Similarly, we use the Cauchy-Schwartz inequality for $\mathcal{A}_1$ as

$$\begin{aligned} \mathcal{A}_1 &:= 2\hat{g}_1[k]^T M_1 G_1\, v_1[k] - 2\hat{g}_2[k]^T M_2 G_2\, v_2[k] \\ &\geq -2\|\hat{g}_1[k]\|\|M_1 G_1\|\|v_1[k]\| - 2\|\hat{g}_2[k]\|\|M_2 G_2\|\|v_2[k]\| \\ &\geq -2\|T\|\left\|[\hat{g}[k], \hat{l}[k]]^T\right\|\|v[k]\|(\|M_1 G_1\| + \|M_2 G_2\|) \end{aligned}$$

$$\geq -2\gamma\varkappa\|v\|^2(\|M_1G_1\| + \|M_2G_2\|)$$
$$= -c_1\gamma\|v\|^2.$$

Then, equality (28) implies that

$$V[k+1] \geq (1+\mu)^2\, V[k] + v_1^T\bar{Q}_1\, v_1 + v_2^T\bar{Q}_2\, v_2 - c_1\gamma\|v\|^2 - c_2\gamma^2\|v\|^2 \geq (1+\mu)^2\, V[k] + (c_0 - c_1\gamma - c_2\gamma^2)\|v\|^2$$

with c_0 being the least eigenvalue of $\bar{Q}_1$ and $\bar{Q}_2$. ■

Proof of instability in Theorem 1. For any $\delta > 0$, take an initial condition $x_s[0] = [x_1^T[0], z_1^T[0]]^T$ such that $\|x_s[0]\| \leq \delta$ and

$$V[0] = x_s[0]^T(T_1^T M_1\, T_1 - T_2^T M_2\, T_2)x_s[0] > 0.$$

This is always possible by setting $T_2\, x_s = 0$ and $T_2\, x_s$ to be an eigenvector of M_1, then scaling x_s down to have norm less than δ.

Take $0 < \gamma_0 \leq 1$ sufficiently small such that $c_0 - c_1\gamma_0 - c_2\gamma_0^2 \geq 0$. Note that this is always possible since $\hat{h}_1, \hat{h}_2$, and thus c_1, c_2, are bounded by the interval $\gamma \in (0, 1]$.
Select the associated radius $r_0 = \mathsf{r}(\gamma_0)$ according to Lemma 1. Then we have $V[k+1] \geq (1+\mu)^2\, V[k]$ as long as $\|(x_1(t), u_1[k])\| \leq r_0$. Take

$$\epsilon := \frac{r_0}{1 + \|C_c\| + \|D_c\| + \|D_cC\|}.$$

We claim that the trajectory starting from $x_s[0]$ will always leave the ball with radius ϵ. Suppose this is not true, *i.e.*, $\|x_s(t)\| \leq \epsilon$ for all $t \geq 0$. Then $\|(x_1(t), u_1[k])\| \leq r_0$ for all $t \in [k\tau, k\tau+\tau]$ and $k \in \mathbb{Z}^{\geq 0}$ and $V[k+1] \geq (1+\mu)^2\, V[k] \geq (1+\mu)^{2k} V(0)$. Then $\lim_{k\to\infty} V[k] = \infty$, which contradicts the boundedness of $x_s(t)$. ■

Proof of Proposition 1. We prove that, for significance level $\alpha \in (0, 1)$, sample $\mathbf{z}_K$ of Θ_T, $\vartheta \in (0, 1)$, $a(\mathbf{z}_K) > \vartheta \implies \gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha'\right)$, where $\alpha' \leq \alpha$, $[a(\mathbf{z}_K), b(\mathbf{z}_K)]$ is a $(1-\alpha)$-confidence interval estimate for $Pr(\mathbf{p}, \psi)$.

By standard results in statistical hypothesis testing, a $(1-\alpha)$-confidence interval $[a(\mathbf{z}_K), b(\mathbf{z}_K)]$ for $Pr(\mathbf{p}, \psi)$ contains all (and only) the values ϑ' such that we fail to reject the null hypothesis in the two-sided test $Pr(\mathbf{p}, \psi) = \vartheta'$ vs $Pr(\mathbf{p}, \psi) \neq \vartheta'$ at level α. This means that if ϑ is outside the confidence interval, then the hypothesis $Pr(\mathbf{p}, \psi) = \vartheta$ is rejected in favor of $Pr(\mathbf{p}, \psi) \neq \vartheta$ at level α. In particular, if ϑ is below the lower end of the interval, then $Pr(\mathbf{p}, \psi) = \vartheta$ is rejected in favor of hypothesis $Pr(\mathbf{p}, \psi) > \vartheta$ at level between $\alpha/2$ and α.[6] Finally, if our estimate is extreme enough to reject $Pr(\mathbf{p}, \psi) = \vartheta$, it will also reject the weaker hypothesis $Pr(\mathbf{p}, \psi) \leq \vartheta$. Hence, the test predicate $\gamma\left(Pr(\mathbf{p}, \psi) > \vartheta, \widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K), \alpha'\right)$, with $\alpha' \leq \alpha$, holds. ■

[6]If the confidence interval $[a(\mathbf{z}_K), b(\mathbf{z}_K)]$ is always symmetric about the estimate $\widehat{Pr}(\mathbf{p}, \psi, \mathbf{z}_K)$, then $Pr(\mathbf{p}, \psi) > \vartheta$ is accepted exactly at level $\alpha/2$. However, binomial confidence intervals for proportions can be asymmetric when they are close to 1 or 0.

APPENDIX B
MODELS OF CASE STUDIES

Gluco-regulatory ODE model.

The model consists of three subsystems:

- *Glucose Subsystem*: it tracks the masses of glucose (in mmol) in the accessible (Q_1) and non-accessible (Q_2) compartments, G (mmol/L) represents the glucose concentration in plasma, EGP_0 (mmol/min) is the endogenous glucose production rate and $U_G(t)$ (mmol/min) is the glucose absorption rate from the gut.
- *Gut absorption*: this subsystem uses a chain of two compartments, G_1 and G_2 (mmol), to model the absorption dynamics of ingested food, given by the disturbance $D_G(t)$. A_g is the CHO bio-availability. t_{maxG} (min) is the time of maximum appearance rate of glucose.
- *Interstitial glucose*: C is the subcutaneous glucose concentration (mmol/L) detected by the CGM sensor and has a delayed response w.r.t. the blood concentration G.
- *Insulin Subsystem*: it represents absorption of subcutaneously administered insulin. It is defined by a two-compartment chain, S_1 and S_2 measured in U (units of insulin), where $u(t)$ (U/min) is the administration of insulin computed by the PID controller, u_b (U/min) is the basal insulin infusion rate and I (U/L) indicates the insulin concentration in plasma.
- *Insulin Action Subsystem*: it models the action of insulin on glucose distribution/transport, x_1, glucose disposal, x_2, and endogenous glucose production, x_3 (unitless).

The model parameters are given in Table 6.

$$\frac{dQ_1}{dt} = -F_{01} - x_1\, Q_1 + k_{12}Q_2 - F_R + EGP_0(1 - x_3) + U_G$$
$$\frac{dQ_2}{dt} = x_1\, Q_1 - (k_{12} + x_2)Q_2$$
$$\frac{dS_1}{dt} = u + u_b - \frac{S_1}{t_{maxI}},\ \frac{dS_2}{dt} = \frac{S_1 - S_2}{t_{maxI}}$$
$$\frac{dI}{dt} = \frac{S_2}{t_{maxI} \cdot V_I} - k_eI,\ \frac{dx_i}{dt} = -k_{a_i} \cdot x_i + k_{b_i} \cdot I,\ i = 1, 2, 3$$
$$U_G(t) = 5.55\, A_G t \frac{e^{-t/t_{maxG}}}{t_{maxG}^2} d(t), \quad G(t) = \frac{Q_1(t)}{V_G}$$
$$\frac{dC}{dt} = k_{int}(G - C).$$

TABLE 6. Parameter values for the glucose-insulin regulatory model. w (kg) is the body weight.

par	value	par	value	par	value
w	100	k_e	0.138	k_{12}	0.066
k_{a_1}	0.006	k_{a_2}	0.06	k_{a_3}	0.03
k_{b_1}	0.0034	k_{b_2}	0.056	k_{b_3}	0.024
t_{maxI}	55	V_I	$0.12 \cdot w$	V_G	$0.16 \cdot w$
F_{01}	$0.0097 \cdot w$	t_{maxG}	40	F_R	0
EGP_0	$0.0161 \cdot w$	A_G	0.8	k_{int}	0.025

Fuel Control System Model. The dynamics of the engine (plant) are given by the following set of ODEs:

$$\dot{p} = c_1\left(\dot{m}_{af} - \dot{m}_c\right), \dot{\theta} = 10(\theta_{in} - \theta), \dot{\lambda} = c_{26}\left[\frac{\dot{m}_c}{c_{25}F_c} - \lambda\right],$$

where p (bar) is the intake manifold pressure; θ (degrees) is the throttle angle; λ is the air/fuel ratio; θ_{in} (degrees) is the throttle angle input disturbance; $\hat{\theta}$ is the throttle plate angle and is defined by: $\hat{\theta} = c_6 + c_7\theta + c_8\theta^2 + c_9\theta^3$; $\dot{m}_c$ (g/s) is the air inflow rate to cylinder and is defined by:

$$\dot{m}_c = c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega^2 p);$$

ω (rad/s) is the engine speed disturbance; $\dot{m}_{af}$ is the inlet air mass flow rate, defined by:

$$\dot{m}_{af} = 2\hat{\theta}\sqrt{p/c_{10} - (p/c_{10})^2};\ \text{and}$$

F_c is the commanded fuel input defined as $F_c = (1 + u(t))\dot{m}_c/\bar{\lambda}$, where $u(t)$ is the control input and $\bar{\lambda}$ is the ideal air/fuel ratio.

Parameter values are: $c_1 = 0.41328$, $c_2 = -0.366$, $c_3 = 0.08979$, $c_4 = -0.0337$, $c_5 = 0.0001$, $c_6 = 2.821$, $c_7 = -0.05231$, $c_8 = 0.10299$, $c_9 = -0.00063$, $c_{10} = 1$, $c_{12} = 0.9$, $c_{25} = 1$, $c_{26} = 4$.

Quadruple-Tank Process Model. The dynamics of the Quadruple-Tank Process are given by the following set of ODEs [21]:

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}u_1\\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}u_2\\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}u_2\\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}u_1,
\end{aligned}$$

where $h_i, a_i, A_i,\ i \in \{1, 2, 3, 4\}$ are the water level, cross-section of the outlet hole, and cross-section of tank i, respectively. Inputs u_1, u_2 indicate the voltages applied to the pumps and the corresponding flows are k_1u_1, k_2u_2. The parameters $\gamma_1, \gamma_2 \in (0, 1)$ show the settings of the valves. The flow to tank 1 is $\gamma_1 k_1 u_1$ and the flow to tank 4 is $(1-\gamma_1)k_1u_1$ (similarly for the other two tanks). The acceleration of gravity is denoted by g. The water levels of tanks 1, 2 are measured by sensors as $k_c h_1, k_c h_2$. The parameter values are: $A_1 = A_3 = 28\,\text{cm}^2$, $A_2 = A_4 = 32\,\text{cm}^2$, $a_1 = a_3 = 0.071\,\text{cm}^2$, $a_2 = a_4 = 0.057\,\text{cm}^2$, $k_c = 0.5\,\text{V/cm}$, $g = 9.81\,\text{m/s}^2$. We have chosen the steady state values $h_1^0 = 12.4\,\text{cm}$, $h_2^0 = 12.7\,\text{cm}$, $h_3^0 = 1.8\,\text{cm}$, $h_4^0 = 1.4\,\text{cm}$, $u_1^0 = 3.00\,\text{V}$, $u_2^0 = 3.00\,\text{V}$, $k_1 = 3.33\,\text{cm}^3/\text{Vs}$, and $k_2 = 3.35\,\text{cm}^3/\text{Vs}$.

REFERENCES

[1] A. Abate, I. Bessa, D. Cattaruzza, L. Chaves, L. Cordeiro, C. David, P. Kesseli, D. Kroening, and E. Polgreen, "DSSynth: An automated digital controller synthesis tool for physical plants," in *Proc. 32nd IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Oct. 2017, pp. 919–924.

[2] W. F. Ames and B. G. Pachpatte, "Inequalities for differential and integral equations," in *Mathematics in Science and Engineering*. New York, NY, USA: Academic, 1997.

[3] D. P. Bertsekas, *Dynamic Programming and Stochastic Control*. Orlando, FL, USA: Academic, 1976.

[4] E. F. Camacho and C. B. Alba, *Model Predictive Control*. London, U.K.: Springer, 2013.

[5] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 257–280, Feb. 2011.

[6] X.-H. Chang, Q. Liu, Y.-M. Wang, and J. Xiong, "Fuzzy peak-to-peak filtering for networked nonlinear systems with multipath data packet dropouts," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 3, pp. 436–446, Mar. 2019.

[7] Y. Chang, Y. Wang, F. E. Alsaadi, and G. Zong, "Adaptive fuzzy output-feedback tracking control for switched stochastic pure-feedback nonlinear systems," *Int. J. Adapt. Control*, vol. 33, no. 10, pp. 1567–1582, 2019.

[8] J. Ding and C. J. Tomlin, "Robust reach-avoid controller synthesis for switched nonlinear systems," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Dec. 2010, pp. 6481–6486.

[9] Y. Dodge, Ed., *The Concise Encyclopedia of Statistics, Chapter Binomial Test*. New York, NY, USA: Springer, 2008, pp. 47–49.

[10] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Proc. Int. Conf. Formal Modeling Anal. Timed Syst. (FORMATS)*. Berlin, Germany: Springer, 2010, pp. 92–106.

[11] P. Mohajerin Esfahani, D. Chatterjee, and J. Lygeros, "The stochastic reach-avoid problem and set characterization for diffusions," *Automatica*, vol. 70, pp. 43–56, Aug. 2016.

[12] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Math. Program.*, vol. 171, nos. 1–2, pp. 115–166, Sep. 2018.

[13] C. Fan, U. Mathur, S. Mitra, and M. Viswanathan, "Controller synthesis made real: Reach-avoid specifications and linear dynamics," in *Proc. 30th Int. Conf. Comput. Aided Verification (CAV)*. Cham, Switzerland: Springer, 2018, pp. 347–366.

[14] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. Soudjani, "Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3324–3331, Aug. 2019.

[15] S. Gao, J. Avigad, and E. M. Clarke, "Delta-decidability over the reals," in *Proc. 27th Annu. IEEE Symp. Log. Comput. Sci.*, Jun. 2012, pp. 305–314.

[16] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *Proc. 24th Int. Conf. Automated Deduction (CADE)* in Lecture Notes in Computer Science, vol. 7898. Berlin, Germany: Springer, 2013, pp. 208–214.

[17] T. Glad and L. Ljung, *Control Theory: Multivariable Nonlinear Methods*. Oxfordshire U.K.: Taylor & Francis, 2000.

[18] S. Haesaert and S. Soudjani, "Robust dynamic programming for temporal logic control of stochastic systems," 2018, *arXiv:1811.11445*. [Online]. Available: https://arxiv.org/abs/1811.11445

[19] R. Hovorka, V. Canonico, L. J. Chassin, U. Haueter, M. Massi-Benedetti, M. O. Federici, T. R. Pieber, H. C. Schaller, L. Schaupp, T. Vering, and M. E. Wilinska, "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes," *Physiol. Meas.*, vol. 25, no. 4, p. 905, 2004.

[20] X. Jin, J. V. Deshmukh, J. Kapinski, K. Ueda, and K. Butts, "Powertrain control verification benchmark," in *Proc. 17th Int. Conf. Hybrid Syst., Comput. Control HSCC*, 2014, pp. 253–262.

[21] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 456–465, May 2000.

[22] S. S. Kanderian, Jr., and G. M. Steil, "Apparatus and method for controlling insulin infusion with state variable feedback," U.S. Patent 8 777 924, Jul. 15, 2014.

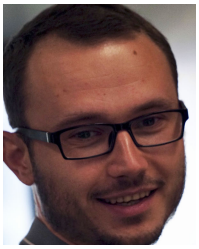
[23] S. Karaman, R. G. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 2117–2122.

[24] H. K. Khalil, *Nonlinear Systems*. London, U.K.: Pearson, 2002.

[25] E. S. Kim, S. Sadraddini, C. Belta, M. Arcak, and S. A. Seshia, "Dynamic contracts for distributed temporal logic control of traffic networks," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 3640–3645.

[26] D. A. Lawrence, "Stability analysis of nonlinear sampled-data systems," in *Proc. 36th IEEE Conf. Decis. Control*, Dec. 1997, pp. 365–366.

[27] D. A. Lawrence, "A stability property of nonlinear sampled-data systems with slowly varying inputs," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 592–596, Mar. 2000.
[28] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*. New York, NY, USA: Springer, 2006.
[29] J. Li, P. Nuzzo, A. Sangiovanni-Vincentelli, Y. Xi, and D. Li, "Stochastic contracts for cyber-physical system design under probabilistic requirements," in *Proc. 15th ACM-IEEE Int. Conf. Formal Methods Models Syst. Design*, Sep. 2017, pp. 5–14.
[30] Z.-M. Li and J. H. Park, "Dissipative fuzzy tracking control for nonlinear networked systems with quantization," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Sep. 24, 2018, doi: 10.1109/TSMC.2018.2866996.
[31] L. Ma, X. Huo, X. Zhao, and G. D. Zong, "Observer-based adaptive neural tracking control for output-constrained switched MIMO nonstrict-feedback nonlinear systems with unknown dead zone," *Nonlinear Dyn.*, vol. 99, no. 2, pp. 1019–1036, Jan. 2020.
[32] D. Nesic and A. R. Teel, "A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models," *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1103–1122, Jul. 2004.
[33] D. Nešić and A. R. Teel, "Input-to-state stability of networked control systems," *Automatica*, vol. 40, no. 12, pp. 2121–2128, Dec. 2004.
[34] K. Ogata, *Discrete-time Control Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
[35] Y. V. Pant, H. Abbas, and R. Mangharam, "Smooth operator: Control using the smooth robustness of temporal logic," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2017, pp. 1235–1240.
[36] V. Raman, A. Donze, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 81–87.
[37] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proc. 18th Int. Conf. Hybrid Syst. Comput. Control HSCC*, 2015, pp. 239–248.
[38] R. Y. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodol. Comput. Appl. Probab.*, vol. 1, no. 2, pp. 127–190, Sep. 1999.
[39] W. Rüemelin, "Numerical treatment of stochastic differential equations," *SIAM J. Numer. Anal.*, vol. 19, no. 3, pp. 604–613, Jun. 1982.
[40] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Robotics: Science and Systems XII*. Ann Arbor, MI, USA: Univ. Michigan, 2016.
[41] H. Shen, S. Huo, J. Cao, and T. Huang, "Generalized state estimation for Markovian coupled networks under round-robin protocol and redundant channels," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1292–1301, Apr. 2019.
[42] F. Shmarov, N. Paoletti, E. Bartocci, S. Lin, S. A. Smolka, and P. Zuliani, "SMT-based synthesis of safe and robust PID controllers for stochastic hybrid systems," in *Proc. 13th Int. Haifa Verification Conf. (HVC)* in Lecture Notes in Computer Science, vol. 10629. Cham, Switzerland: Springer, 2017, pp. 131–146.
[43] F. Shmarov and P. Zuliani, "ProbReach: Verified probabilistic delta-reachability for stochastic hybrid systems," in *Proc. 18th Int. Conf. Hybrid Syst. Comput. Control HSCC*, 2015, pp. 134–139.
[44] F. Shmarov and P. Zuliani, "Probabilistic hybrid systems verification via SMT and Monte Carlo techniques," in *Proc. 12th Int. Haifa Verification Conf. (HVC)* in Lecture Notes in Computer Science, vol. 10028. Cham, Switzerland: Springer, 2016, pp. 152–168.
[45] A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, and V. Saraswat, "Combinatorial sketching for finite programs," *ACM SIGARCH Comput. Archit. News*, vol. 34, no. 5, pp. 404–415, Oct. 2006.
[46] E. D. Sontag, "Input to state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*. Berlin, Germany: Springer, 2008, pp. 163–220.
[47] S. Soudjani, "Formal abstractions for automated verification and synthesis of stochastic systems," Ph.D. dissertation, Delft Center Syst. Control, Technische Univ. Delft, Delft, The Netherlands, 2014.
[48] S. E. Z. Soudjani and A. Abate, "Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes," *SIAM J. Appl. Dyn. Syst.*, vol. 12, no. 2, pp. 921–956, Jan. 2013.
[49] G. M. Steil, C. C. Palerm, N. Kurtz, G. Voskanyan, A. Roy, S. Paz, and F. R. Kandeel, "The effect of insulin feedback on closed loop glucose control," *J. Clin. Endocrinol. Metabolism*, vol. 96, no. 5, pp. 1402–1408, 2011.
[50] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2817–2830, Nov. 2012.



FEDOR SHMAROV received the B.Sc. degree in information science and computer technology from Tambov State Technical University, Russia, in 2011, and the M.Sc. degree in advanced computing science and the Ph.D. degree in computer science from Newcastle University, U.K., in 2013 and 2018, respectively. He is currently a Postdoctoral Researcher with the School of Computing, Newcastle University. His research interests include formal methods and model checking for verifying hybrid systems with stochastic and nondeterministic behavior.

SADEGH SOUDJANI received the B.Sc. degree in pure mathematics and electrical engineering and the M.Sc. degree in control engineering from the University of Tehran, Tehran, Iran, in 2007 and 2009, respectively, and the Ph.D. degree in systems and control from the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, in November 2014. He was a Postdoctoral Researcher with the Department of Computer Science, University of Oxford, U.K., and the Max Planck Institute for Software Systems, Germany. He is currently a Lecturer (Assistant Professor) with the School of Computing, Newcastle University, U.K. His research interests include formal synthesis, abstraction, and verification of complex dynamical systems with application in cyber-physical systems, particularly involving power and energy networks.

NICOLA PAOLETTI received the Ph.D. degree in information sciences and complex systems from the University of Camerino, Italy, in 2014. He is currently a Lecturer with the Department of Computer Science, Royal Holloway, University of London, U.K. His research interests include the verification, control, and synthesis of stochastic and hybrid systems, with application to biological and biomedical systems.

EZIO BARTOCCI is currently an Associate Professor with the Faculty of Informatics, TU Wien, Austria. His research interests include the development of formal methods, computational tools, and techniques, which support the modeling and the automated analysis of complex computational systems, including software systems, cyber-physical systems, and biological systems.

SHAN LIN received the Ph.D. degree in computer science from the University of Virginia. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Stony Brook University. His research interests include networked systems, cyber physical systems, and the Internet of Things. His current research interests include wireless network protocols, medical systems and devices, smart buildings, and smart transportation systems. He received the NSF Career Award, in 2016.

SCOTT A. SMOLKA is currently a SUNY Distinguished Professor of computer science with Stony Brook University. His research interests include model checking, runtime verification, and the modeling and analysis of complex systems, including cyber-physical systems, cardiac tissue, and other biological systems. He is the Lead PI for the multi-institutional NSF CPS Frontiers Project CyberCardia. He is also a Fellow of the European Association on Theoretical Computer Science (EATCS).

PAOLO ZULIANI received the D.Phil. degree in computer science from the University of Oxford, U.K. He is currently a Senior Lecturer with the School of Computing, Newcastle University, U.K. His research interests include formal and automated methods for reasoning about computing systems, with an emphasis on probabilistic and quantum systems.