Barrier Certificate Computation using Gaussian Process Optimization*

Xinyu Ge[1,*], Sadegh Soudjani[2] and Paolo Zuliani[3]

[1]Department of Computer Science, Aalborg University, Aalborg 9220, Denmark
[2]Max Planck Institute for Software Systems, Kaiserslautern D-67663, Germany
[3]Dipartimento di Informatica, Università di Roma "La Sapienza", Rome 00185, Italy

Abstract

Autonomous agents deployed in safety-critical environments must operate with a clear awareness of safety constraints embedded in their dynamics. Barrier certificates (BCs) are a powerful formalism for verifying such safety properties in continuous dynamical systems. However, synthesizing BCs remains a computationally difficult problem. In this work, we propose a novel approach for constructing BCs using the Augmented Lagrangian framework, combined with Bayesian optimization over Gaussian Process models to efficiently guide the search for valid certificates. We demonstrate our algorithm on several test cases, illustrating the success of our proposed scheme.

Keywords

Barrier certificate, Dynamical systems, Gaussian process, Safety verification

1. Introduction

Autonomous agents operating in complex environments must not only achieve task performance but also remain continuously aware of safety constraints that govern their permissible behaviors. Dynamical systems theory, which studies the long-term behavior of evolving systems [1], provides a formal foundation for modeling such agents. This is especially relevant in hybrid systems, where continuous and discrete dynamics interact to define the agent's trajectory in the physical world, as commonly encountered in cyber-physical systems.

Ensuring safety in such systems is critical in domains such as aviation, autonomous driving, and chemical process control [2, 3]. A central safety requirement is to verify that an agent's trajectory, starting from any admissible initial condition, does not lead into unsafe regions of the state space [4]. However, this is a computationally demanding problem due to the high dimensionality and nonlinear behavior of many dynamical systems. Traditional approaches include reachability analysis [5, 6], often supported by model checking [7] and deductive reasoning [8]. For specific system classes, symbolic and decision-based methods have shown promise [9, 2, 10] , and various tools have been developed to automate such analysis (see [11] for a comprehensive overview).

As an important numerical method for safety verification of autonomous agents, *barrier certificates* (BCs) have been proposed recently [4, 12, 13]. The idea of BCs is to depict a 'barrier' between the possible system trajectories and the given unsafe region in order to prove that the system is safe. In other words, the existence of BCs serves as evidence for safety. Depending on the specific application, some studies are committed to relax the conditions instead of using the definition in [4]. Combining multiple functions to build a BC have been studied in [14] while discussing weaker BC conditions. Compositional synthesis of BCs has been studied in [15] for hybrid systems consisting of many interconnected subsystems. Extension of barrier certificates for stochastic systems and temporal specifications beyond safety is studied in [16]. Input-to-state safe control barrier functionals (ISSf-CBFs) guarantee the safety of

2025 Workshop on Awareness in Learning Agents (ALA), October 25-26, 2025, Bologna, Italy
*Corresponding author.
xinyuge@cs.aau.dk (X. Ge); sadegh@mpi-sws.org (S. Soudjani); zuliani@di.uniroma1.it (P. Zuliani)
https://ge-xinyu.github.io/ (X. Ge); https://hycodev.com/ssoudjani (S. Soudjani); https://pzuliani.github.io/ (P. Zuliani)
0000-0002-7392-9107 (X. Ge); 0000-0003-1922-6678 (S. Soudjani); 0000-0001-6033-5919 (P. Zuliani)

time-delay systems is proposed in [17]. Recent application of barrier certificates for checking properties of quantum systems is addressed in [18].

With the advances in learning methods, various data-driven approaches have emerged to address safety verification and synthesis of autonomous agents, showcasing the potential integration of machine learning with traditional formal methods. Data-driven abstraction with formal guarantees is studied in [19, 20] for satisfying temporal specifications. Data-driven reachable set computation with formal guarantees is studied in [21]. Learning algorithms for computing BCs that are in the form of neural networks are also proposed [22, 23]. Safety verification of autonomous agents with unknown dynamics using noisy data from observations, Gaussian process regression, and abstracting the system into a finite Markov model has been studied in [24].

Gaussian process (GP) regression is an efficient and accurate data-driven approach for approximating computationally complex functions. An advantage of GPs is their ability to offer an efficient analytical approximation over the entire domain of the target function. Furthermore, other approaches such as deep neural networks typically need vast amounts of training data to ensure convergence of the learning phase. This contrasts with GPs, which in many cases can learn good approximations even with relatively small quantities of training data. GPs are used in optimization through a Bayesian approach known as Gaussian process optimization (GPO) [25], which is often more sample-efficient than traditional optimization methods.

GPs are used for verifying properties of agents with unknown dynamics. The authors of the papers [26, 24, 27] use GPs to learn a model of the agent based on data. This is extended with deep kernel learning in [28], where the GP kernel is augmented with a neural network preprocessing its inherent feature map.

We concentrate on using GPO for the safety verification of a given autonomous agent through the computation of BCs. Building upon insights from [29] and leveraging the augmented Lagrangian framework as described in [30], we propose a framework that transforms the BC synthesis problem into a constrained optimization problem using the augmented Lagrangian method. Our approach computes the parameters of a BC from a fixed template through GPO for handling constraints. Given that the constraints are in the form of parameterized optimizations, we fit a GP to estimate their values from a finite number of evaluations. This gives a probabilistic interpretation of the constraints, which are transformed to chance-constraints and subsequently incorporated into a reliability-based design optimization (RBDO) [31]. We further validate the feasibility and effectiveness of our method through implementation in three case studies. The main contributions of this article are threefold:

1. We introduce an optimization approach that synthesizes a BC from a parameterized set of functions for autonomous agents evolving continuously in time. We reconfigure the problem into a robust parameter optimization task and use augmented Lagrangian.
2. Gaussian process regression and RBDO are utilized to estimate with finite number of evaluations the functions appearing as maximum over a continuous domain and satisfy the related constraints.
3. The validity of the computed BCs are formally verified using Satisfiability Modulo Theories (SMT) solvers.

The remainder of this paper is organized as follows. A concise overview of essential details regarding safety verification, BCs, and Gaussian process regression is provided in Section 2. Our proposed solution approach is presented along with the algorithm in Section 3 containing the details of the augmented Lagrangian, RBDO, and verification of the BC using SMT Solvers. Case studies are provided in Section 4 to illustrate the results with concluding remarks in Section 5.

2. Preliminaries and Problem Statement

In the following, $\mathbb{R}$ denotes the set of real numbers; $\arg\min(\cdot)$ returns an argument that minimizes the input function, $\max(\cdot)$ represents the largest value taken by the input function; $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, and $\phi(\cdot)$ is the probability density function of the standard normal distribution.

2.1. Autonomous Agents and the Safety Problem

An autonomous agent modeled as a d-dimensional dynamical system over the continuous state space $X \subset \mathbb{R}^d$ is described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} = [x_1, \dots, x_d]^T$ is a column vector, $\dot{\boldsymbol{x}}$ denotes the derivative of $\boldsymbol{x}$ with respect to time, $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \dots, f_d(\boldsymbol{x})]^T$ is a Lipschitz-continuous vector field, and $X \subseteq \mathbb{R}^d$ is an open set defining the state space of the system. The region for the initial state of the system $\boldsymbol{x}_0$ is denoted by X_0; the unsafe region of the system is denoted by X_u.

Definition 1 (Safety verification problem). *Let $\Psi(\boldsymbol{x}_0, t)$ be the solution of the state equation (1) from initial state $\boldsymbol{x}_0$ at time $t \geq 0$. If the reachable set*

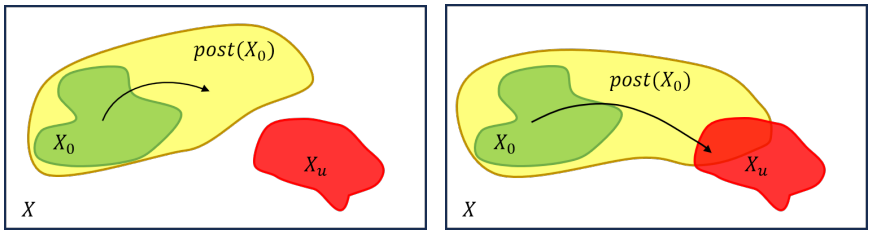
$$post(X_0) := \{\boldsymbol{x} \in X | \exists \boldsymbol{x}_0 \in X_0, t \geq 0 \,:\, \Psi(\boldsymbol{x}_0, t) = \boldsymbol{x}\} \tag{2}$$

satisfies

$$post(X_0) \cap X_u = \emptyset, \tag{3}$$

then the system (1) is considered safe. On the contrary, if there exists $\boldsymbol{x}_p \in X$ and $t_p \geq 0$ such that $\Psi(\boldsymbol{x}_p, t_p) \in post(X_0) \cap X_u$, then X_u is reachable, which indicates the system is unsafe.

In Fig. 1 we depict examples of safe and unsafe systems.



(a) A safe system (b) An unsafe system

Figure 1: The Safety Verification Problem. The black box is the state space X, which covers the set of initial states X_0 (in green), the reachable set $post(X_0)$ (in yellow) and the unsafe region X_u (in red). The arrow represents a trajectory of the system. Panel (1a): if $post(X_0)$ does not intersect X_u, then X_u is not reachable from X_0, and therefore the system is safe. Panel (1b): if there exists a trajectory that reaches X_u from X_0, then the system is unsafe.

2.2. Barrier Certificate

Due to the difficulty of computing the reachable set $post(X_0)$, the concept of *barrier certificate* (BC) [4] has emerged as an effective way to prove the safety of a system. The idea of a BC is to find a so-called barrier function that separates the reachable set and the unsafe set of the system, which means there is no trajectory of the system that reaches X_u from X_0. Thus, as a sufficient condition, the system is considered safe. A BC is a function of state that satisfies a set of inequalities on both the function itself and its time derivative along the flow of the system [4]. We recall that for a continuously differentiable function $B(\boldsymbol{x})$, the gradient of $B(\boldsymbol{x})$ is defined by the column vector

$$\nabla B(\boldsymbol{x}) = \frac{\partial B(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left[\frac{\partial B(\boldsymbol{x})}{\partial x_1}, \dots, \frac{\partial B(\boldsymbol{x})}{\partial x_d}\right]^T. \tag{4}$$

Definition 2 (Lie derivative). *Given a vector field $\boldsymbol{f}(\boldsymbol{x})$, the Lie derivative of a continuously differentiable function $B(\boldsymbol{x})$ is defined as the inner product of $\nabla B(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x})$:*

$$\dot{B}(\boldsymbol{x}) = \nabla B(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{d} \left(\frac{\partial B(\boldsymbol{x})}{\partial x_i} \cdot f_i(\boldsymbol{x})\right). \tag{5}$$

Proposition 1 (BCs for Safety Verification[4]). *Given a system described as in* (1) *with sets* X_0, X_u, *and* X *for which there exists a BC, namely a function* $B(\boldsymbol{x})$ *that is differentiable with respect to its argument and satisfies the following conditions:*

$$B(\boldsymbol{x}) \leq 0 \;\; \forall \boldsymbol{x} \in X_0, \quad B(\boldsymbol{x}) > 0 \;\; \forall \boldsymbol{x} \in X_u, \text{ and } \quad \dot{B}(\boldsymbol{x}) \leq 0 \;\; \forall \boldsymbol{x} \in X, \tag{6}$$

then the safety of the system is guaranteed. That is, there is no system trajectory starting from an initial state in X_0 *and reaching a state in* X_u.

The verification problem statement is as follows:

Input: A dynamical system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$, initial region X_0, unsafe region X_u and state space X.
Problem: Synthesize a BC $B(\boldsymbol{x})$ that guarantees the safety of the system within X.

The notion of BC has played a predominant role in the investigation of dynamical systems, particularly for safety verification. A BC represents a formal proof of safety for the system. For a system that is safe, the existence of multiple BCs is plausible: multiplying a BC with a positive constant will give another BC, and the set of BCs is a convex set meaning that any convex combination of BCs is again a BC. These are immediate from (6).

2.3. Gaussian Process Regression

Gaussian process regression (GPR) is an approach for non-parametric Bayesian inference on functions over continuous domains [32]. Theoretical and practical developments over the last decade have made GPR a suitable approach for machine learning applications, in particular when training data are limited or costly. This non-parametric approach assumes that a function $g : X \to \mathbb{R}$ is a Gaussian process (GP) meaning that its values at any $\boldsymbol{x} \in X$ is a random variable that is distributed according to a Gaussian distribution. Therefore, the full probabilistic information on the function is characterized by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$, defined as follows:

$$m(\boldsymbol{x}) = E[g(\boldsymbol{x})] \quad \text{and} \quad k(\boldsymbol{x}, \boldsymbol{x'}) = E[(g(\boldsymbol{x}) - m(\boldsymbol{x}))(g(\boldsymbol{x'}) - m(\boldsymbol{x'}))]. \tag{7}$$

Such a GP gives the prior distributions on $g(\cdot)$ and is denoted by

$$g(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})). \tag{8}$$

When a set of N observations $\boldsymbol{y} = [y_1, y_2, \dots, y_N]^T$ on the values of the function $g(\cdot)$ at data points $[\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]$ is available, the posterior distribution conditioned on the training data $\hat{g}(\boldsymbol{x}|D)$, is again a GP with mean $\hat{\mu}$ and variance $\hat{\sigma}$:

$$\hat{\mu} = \boldsymbol{\kappa}(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{y} \quad \text{and} \quad \hat{\sigma} = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{\kappa}(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{\kappa}(\boldsymbol{x})^T \tag{9}$$

with the covariance matrix $\boldsymbol{K} := [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^N$ and the vector $\boldsymbol{\kappa}(\boldsymbol{x}) := [k(\boldsymbol{x}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}, \boldsymbol{x}_N)]$.

A frequently used covariance function is the squared exponential function

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_Y^2 \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_M^2) \tag{10}$$

with prior variance σ_Y^2 and scaling matrix M. The scaling matrix M, such as $\text{diag}(1/l_1^2, \dots, 1/l_d^2)$, assigns individual scaling factors to each component of the vector $\boldsymbol{x}$. The vector of length scales $l_1, \dots, l_d$, and σ_Y^2 are called hyperparameters, which are typically tuned through methodologies such as maximum likelihood estimation.

3. Solution Approach

The construction of barrier functions is often not simple. For barrier functions that are polynomials with unknown coefficients, the verification problem is transformed into obtaining the value of said coefficients. In this section, as a first step, we transform the problem into an optimization problem on the coefficients and state. Then, we propose a computational method and illustrate the algorithm.

Let $B(\boldsymbol{a}, \boldsymbol{x})$ denote a polynomial with fixed structure, where $\boldsymbol{a}$ is the vector of parameters in B, and $\boldsymbol{x} = [x_1, \dots, x_d]$ are the polynomial variables; $\boldsymbol{\theta} = [\theta_{11}, \theta_{12}, \dots, \theta_{d1}, \theta_{d2}]$ is a $2d$ vector; $\Delta(\boldsymbol{\theta})$ is a hyperrectangle with edges that are parallel to the parameter axes defined as

$$\Delta(\boldsymbol{\theta}) = \{\boldsymbol{x} \in X \,|\, \theta_{i1} \leq x_i \leq \theta_{i2}, i = 1, \dots, d\}. \tag{11}$$

We aim at finding parameters $\boldsymbol{a}$ that maximize the volume of $\Delta(\boldsymbol{\theta})$, in order to find a BC that proves the system's safety on the largest possible space. Reworking the general constraints described by (6), we present the optimization problem on variables $\boldsymbol{x}$ with parameters $\boldsymbol{a}$ and $\boldsymbol{\theta}$:

$$\arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \tag{12}$$

$$s.t. \quad c_1(\boldsymbol{a}) \leq 0, c_2(\boldsymbol{a}) < 0, c_3(\boldsymbol{a}, \boldsymbol{\theta}) \leq 0, \tag{13}$$

where

$$c_1(\boldsymbol{a}) := \max_{\boldsymbol{x} \in X_0} B(\boldsymbol{a}, \boldsymbol{x}), \quad c_2(\boldsymbol{a}) := \max_{\boldsymbol{x} \in X_u} -B(\boldsymbol{a}, \boldsymbol{x}), \quad \text{and} \quad c_3(\boldsymbol{a}, \boldsymbol{\theta}) := \max_{\boldsymbol{x} \in \Delta(\boldsymbol{\theta})} \dot{B}(\boldsymbol{a}, \boldsymbol{x}), \tag{14}$$

and $f(\boldsymbol{\theta})$ is described as

$$f(\boldsymbol{\theta}) = -\prod_{i=1}^{d} |\theta_{i1} - \theta_{i2}|. \tag{15}$$

For simplicity, the constraints in (14) are denoted collectively as $c_i(\boldsymbol{a}, \boldsymbol{\theta})$, $i = 1, 2, 3$, even though c_1 and c_2 do not depend on θ.

The objective of the optimization in (12) is to maximize the size of the hyperrectangle $\Delta(\boldsymbol{\theta})$ while ensuring that $\boldsymbol{a}$ satisfies (13). As showed in Fig. 2, if there exist $\boldsymbol{a}$ and $\boldsymbol{\theta}$ such that (13) holds with

$$X \subseteq \Delta(\boldsymbol{\theta}), \tag{16}$$

then (6) holds with $B(\boldsymbol{a}, \boldsymbol{x})$. That is to say, $B(\boldsymbol{a}, \boldsymbol{x})$ is a valid BC for the system, which is thus safe.

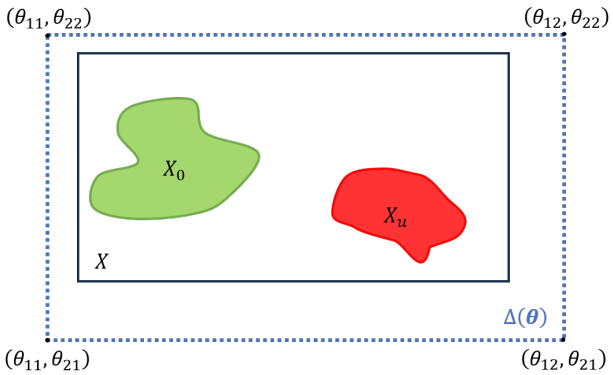


Figure 2: Diagram of an ideal $\Delta(\boldsymbol{\theta})$. For $B(\boldsymbol{a}, \boldsymbol{x})$, if the constraints in (13) are satisfied and $\Delta(\boldsymbol{\theta})$ covers X, then the systems is safe.

To find a solution for the robust optimization (12), we utilize the concept of augmented Lagrangian and employ GPR for generalizing its finite evaluation to the whole state space.

3.1. Augmented Lagrangian with Gaussian Process Regression

Augmented Lagrangian is used primarily for constrained nonlinear optimization [30]. This gives the following objective function for an unconstrained optimization

$$L(\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\lambda},\rho)=f(\boldsymbol{\theta})+\sum_{i=1}^{3}\lambda_i c_i(\boldsymbol{a},\boldsymbol{\theta})+\frac{1}{2\rho}\sum_{i=1}^{3}\max(0,c_i(\boldsymbol{a},\boldsymbol{\theta}))^2, \tag{17}$$

where the first term $f(\boldsymbol{\theta})$ is the objective function of the original constraint optimization (12). The second term encodes the functions $c_i(\boldsymbol{a},\boldsymbol{\theta})$ in the three constraints with coefficients $\lambda_i \geq 0$ called Lagrangian multipliers. The third term encodes the requirements on the functions $c_i(\boldsymbol{a},\boldsymbol{\theta})$ to be negative with the coefficients $\rho > 0$ being penalty parameters that penalize the positive values of $c_i(\boldsymbol{a},\boldsymbol{\theta})$.

The functions $c_i(\boldsymbol{a},\boldsymbol{\theta})$ in (14) do not admit a closed-form solution in general since they are in the form of parameterized maximization over a continuous domain. Therefore, we move towards using Gaussian process regression that can build a likelihood for the values of such functions when a finite number of evaluations of these functions is available. Denote by $\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta})$ the GP model of $c_i(\boldsymbol{a},\boldsymbol{\theta})$ and define the following objective function based on the expectation of the assigned GPs:

$$f_L(\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\lambda},\rho)=f(\boldsymbol{\theta})+\sum_{i=1}^{3}\lambda_i E[\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta})]+\frac{1}{2\rho}\sum_{i=1}^{3}E[\max(0,\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta}))^2], \tag{18}$$

where the GP model $\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta})$ has the mean $\hat{\mu}_i(\boldsymbol{a},\boldsymbol{\theta})$ and variance $\hat{\sigma}_i(\boldsymbol{a},\boldsymbol{\theta})$, and we have from [30] that

$$\begin{aligned} E[\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta})] =&\hat{\mu}_i(\boldsymbol{a},\boldsymbol{\theta}),\\ E[\max(0,\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta}))^2] =&\hat{\sigma}_i^2(\boldsymbol{a},\boldsymbol{\theta})\Big[(1+\frac{\hat{\mu}_i(\boldsymbol{a},\boldsymbol{\theta})}{\hat{\sigma}_i(\boldsymbol{a},\boldsymbol{\theta})})^2\Phi(\frac{\hat{\mu}_i(\boldsymbol{a},\boldsymbol{\theta})}{\hat{\sigma}_i(\boldsymbol{a},\boldsymbol{\theta})})+\phi(\frac{\hat{\mu}_i(\boldsymbol{a},\boldsymbol{\theta})}{\hat{\sigma}_i(\boldsymbol{a},\boldsymbol{\theta})})\Big]. \end{aligned} \tag{19}$$

3.2. Reliability-based Design Optimization

By interpreting the values of the functions in the constraint as Gaussian random variables, the constraints can no longer be hold, but need to be interpreted probabilistically and hold with a high probability. Therefore, the constraint $c_i(\boldsymbol{a},\boldsymbol{\theta}) \leq 0$ can be replaced with

$$\text{Prob}(\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta}) \leq 0) \geq 1 - p_i^{target} \tag{20}$$

for some small admissible failure probability p_i^{target}. To handle such chance constraints, so-called reliability-based design optimization (RBDO) [31] approaches are employed that are capable of considering constraints on the failure probability. Numerical methods for RBDO are studied previously, e.g., in [31]. The core idea is to estimate the failure probability (i.e., the complement of the right-hand side of (20)) using Monte Carlo estimation

$$p_i(\boldsymbol{a},\boldsymbol{\theta}) \approx \frac{1}{m}\sum_{j=1}^{m}\mathbb{I}(\hat{c}_{ij}(\boldsymbol{a},\boldsymbol{\theta}) > 0), \tag{21}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\hat{c}_{ij}$, $j = 1, 2, \ldots, m$, are the samples drawn from the GP model $\hat{c}_i(\boldsymbol{a},\boldsymbol{\theta})$. Then the difference $p_i(\boldsymbol{a},\boldsymbol{\theta}) - p_i^{target}$ has to be negative and is considered as part of the optimization.

3.3. Optimization Algorithm

The constructed optimization approach is presented in Algorithm 1. The details of the algorithm and its subroutines are given as follows with a flowchart of the algorithm presented in Figure 3.

Initialization The initialization consists of: 1) selecting values for the Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]$ and the penalty factor ρ; 2) randomly sampling $(\boldsymbol{a},\boldsymbol{\theta})$ over their domain and forming the initial dataset S; 3) computing the functions $c_i(\boldsymbol{a},\boldsymbol{\theta})$ on S and storing the values in a set C.

Objective function With the optimization problem described as (13), the augmented Lagrangian of (18) is used for the optimization.

Loop 1 (lines 8, 17-18) The outer loop is for updating $\boldsymbol{\lambda}$ and ρ to find the parameters for the augmented Lagrangian approach according to the heuristic update rule [30]. This is based on increasing the penalty factor when the constraints are violated in new data points. The iterations are performed until an $\boldsymbol{s}^*$ is found that satisfies (16) and (13).

Loop 2 (lines 6-13) The inner loop solves the maximization in (18) on $\boldsymbol{a}$ and $\boldsymbol{\theta}$, given the most recent candidates for $\boldsymbol{\lambda}$ and ρ from the outer loop. The optimization (line 10) is done using Bayesian optimization, during which GP models of $c_i(\boldsymbol{a}, \boldsymbol{\theta})$ are learned. Then s' is included in the dataset (line 14). With updated $f_L'(\boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \rho)$ (line 15), the iterations are performed until an ideal $\boldsymbol{s}^*$ is found or f_L' does not improve over f_L with current $\boldsymbol{\lambda}$ and ρ.

In practice, the implementation of Algorithm 1 employs a further parameter that caps the number of iterations, to prevent untimely or prolonged execution, ensuring acceptable performance within reasonable time constraints.

3.4. Verifying the BC Using SMT Solvers

In order to verify the BC computed by the optimization algorithm, Satisfiability Modulo Theories (SMT) solvers [33, 34] are employed for formally checking the validity of the BC. An SMT solver is a tool for deciding *correctly*, *i.e.*, without numerical errors, the satisfiability of a logical formula over a mathematical theory, *e.g.*, linear real arithmetic. In our case, each *negated* constraint can be seen as a formula that we aim to refute (which means the constraint is satisfied). Define the logical formula

$$(\exists \boldsymbol{x} \in X_0,\ -B(\boldsymbol{x}) < 0) \vee (\exists \boldsymbol{x} \in X_u,\ B(\boldsymbol{x}) \leq 0) \vee (\exists \boldsymbol{x} \in X,\ -\dot{B}(\boldsymbol{x}) < 0), \tag{22}$$

which is the negation of constraints in (6). Hence, a candidate BC is verified when the SMT solver returns *unsatisfiable* for the logical formula in (22). Barrier certificates reported in the case study section of this paper are verified by the SMT solvers Z3 and dReal [33, 34].

4. Case studies

In this section, we present four case studies that demonstrate the validity of our approach. All experiments are performed with Matlab R2025a with an Apple M3, 24 GB RAM Macbook Air. The estimation

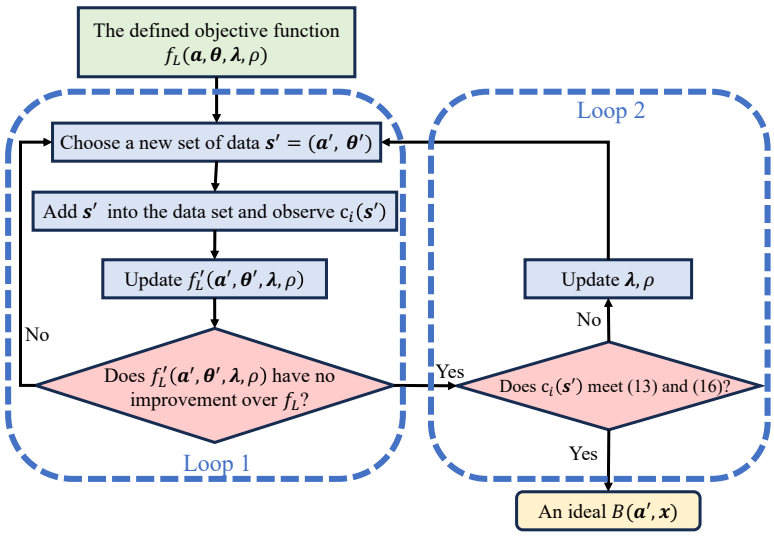


Figure 3: Flowchart of Algorithm 1.

Algorithm 1: Barrier Certificate Synthesis

Input: A dynamical system as in (1); sets X_0, X_u, X;

Select initial values for the Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]$ and the penalty factor ρ;
Select a dataset S by randomly choosing $(\boldsymbol{a}, \boldsymbol{\theta})$;
Compute the functions $c_i(\boldsymbol{s}, \boldsymbol{\theta})$ using (14) on S and store the values in a set C;
Compute the augmented Lagrangian $f_L(\boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \rho)$ on the dataset S;
if $c_i(\boldsymbol{a}, \boldsymbol{\theta})$ *meet* (13) *and* $\Delta(\boldsymbol{\theta})$ *meets* (16) *for some* $(\boldsymbol{a}, \boldsymbol{\theta}) \in S$ **then**
    $\boldsymbol{s}^* \leftarrow (\boldsymbol{a}, \boldsymbol{\theta})$
end
repeat
    repeat
        Select a new $\boldsymbol{s}' = (\boldsymbol{a}', \boldsymbol{\theta}')$ by maximizing (18);
        if $c_i(\boldsymbol{s}')$ *meet* (13) *and* $\Delta(\boldsymbol{\theta}')$ *meets* (16) **then**
            $\boldsymbol{s}^* \leftarrow \boldsymbol{s}'$;
        end
        $S \leftarrow S \cup \boldsymbol{s}'$, $C \leftarrow C \cup c_i(\boldsymbol{a}', \boldsymbol{\theta}')$;
        update the Lagrangian $f_L'(\boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \rho)$ with the GPs on the new dataset ;
    until $f_L \leq f_L'$ *or* $\boldsymbol{s}^* \neq \emptyset$;
    $\lambda_i \leftarrow \max(0, \lambda_i + \frac{c_i(\boldsymbol{a}', \boldsymbol{\theta}')}{\rho})$;
    Set $\rho \leftarrow \frac{1}{2}\rho$ if $c_i(\boldsymbol{a}', \boldsymbol{\theta}') > 0$
until $\boldsymbol{s}^* \neq \emptyset$;
Verify that (13) is satisfied with $\boldsymbol{s}^* = (\boldsymbol{a}^*, \boldsymbol{\theta}^*)$ using SMT solvers;

Output: $\boldsymbol{s}^* = (\boldsymbol{a}^*, \boldsymbol{\theta}^*)$ satisfying (13) and (16).

of hyperparameters for the GPR is executed using the `fitrgp` function in MATLAB with its default settings.

Case 4.1. *Consider the two-dimensional system*

$$\begin{cases} \dot{x_1} &= x_1 + 2x_2, \\ \dot{x_2} &= x_1 x_2 - \frac{1}{2}x_2^2, \end{cases}$$

with initial set $X_0 = [-1, 1] \times [-1, 1]$, *unsafe set* $X_u = [-1, 1] \times [3, 5]$, *and* $X = [-5, 5] \times [-5, 5]$. *Assume the BC is polynomial as* $B(\boldsymbol{a}, \boldsymbol{x}) = a_1 x_1^2 + a_2 x_2 + a_3$.

For this case study, two experiments are conducted as Row 1 and Row 2 in Table 1. In Row 1, the value of the a_i's are $a_i \in \{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$ for $i = 1, 2, 3$, thereby resulting in $5^3 = 125$ data points for the initial dataset S. However, the algorithm terminates at the 27th data point, returning a valid result well

Table 1
Barrier Certificate Synthesis using Algorithm 1. The a_i column shows the initial dataset for each component of $\boldsymbol{a}$. The $\boldsymbol{s}^*$ column reports $[\boldsymbol{a}^*, \boldsymbol{\theta}^*]$ returned by Algorithm 1.

Row	Case Study	a_i	$\boldsymbol{s}^* = [\boldsymbol{a}^*, \boldsymbol{\theta}^*]$
1	Case 1	$\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$	[-0.5000, 1.0000, -1.0000, -5.0000, 5.0000, -5.0000, 5.0000]
2	Case 1	$\{-1, 0, 1\}$	[-1.3993, 2.4492, -2.9646, -9.3977, 5.2681, -9.4590, 7.1630]
3	Case 2	$\{-\frac{1}{2}, 0, \frac{1}{2}\}$	[-0.2465, -0.6794, -0.8787, -1.1328, -0.8086, -4.9977, 4.8801, -4.9696, 4.7986]
4	Case 3	$\{-1, 0, 1\}$	[0.7654, -2.7858, 1.8012, -1.8210, 2.2383, -0.0945, 1.4833]
5	Case 3	$\{-1, 1\}$	[2.9760, -2.9401, 1.6344, -2.9823, 1.0333, 0.2645, 1.3431]
6	Case 4	$\{-1, 1\}$	[1.0000, 1.0000, 1.0000, -1.0000, 0.1000, 1.0000, 0.1000, 1.0000]

before the generation of the entire dataset. Hence, no iterations are performed. With a smaller initial dataset shown in Row 2, $\boldsymbol{s}^*$ is found on the 79^{th} iteration. As shown in Table 1 Row 2, the algorithm returns

$$\boldsymbol{s}^* = [\boldsymbol{a}^*, \boldsymbol{\theta}^*] = [\underbrace{-1.3993, 2.4492, -2.9646}_{\boldsymbol{a}^*}, \underbrace{-9.3977, 5.2681, -9.4590, 7.1630}_{\boldsymbol{\theta}^*}],$$

which means that the candidate BC $B(\boldsymbol{x}) = -1.3993x_1^2 + 2.4492x_2 - 2.9646$ meets (13) for $\Delta(\boldsymbol{\theta}) = [-9.3977, 5.2681] \times [-9.4590, 7.1630]$, which covers $X = [-5, 5] \times [-5, 5]$. Hence, (6) holds for the computed $B(\boldsymbol{x})$ with the given X. Thus, it is a valid BC for the system, which indicates the system is safe. In Figure 4 (top panels) we depict the results of Row 2 of Table 1.

Case 4.2. *Consider the two-dimensional system*

$$\begin{cases} \dot{x_1} &= x_2, \\ \dot{x_2} &= -x_1 + \frac{1}{3}x_1^3 - x_2, \end{cases}$$

with initial set $X_0 = \{(x_1 - 1.5)^2 + x_2^2 \leq 0.25\}$, *unsafe set* $X_u = \{(x_1 + 1)^2 + (x_2 + 1)^2 \leq 0.16\}$, *and* $X = [-\frac{7}{5}, 2] \times [-\frac{7}{5}, \frac{1}{2}]$. *Assume the BC is a polynomial as* $B(\boldsymbol{x}) = a_1x_1^2 + a_2x_1x_2 + a_3x_1 + a_4x_2 + a_5$.

Due to the conservativeness of the convex condition, the method based on the condition (6) succeeded only in one case (degree = 4) borrowed from [35]. Instead, here we employ the exponential conditions from (3) in [35], where $\dot{B}(\boldsymbol{a}, \boldsymbol{x}) - \varphi B(\boldsymbol{a}, \boldsymbol{x}) \leq 0$, for all $\boldsymbol{x} \in X$, with $\varphi = -1$. The algorithm's output is

$$\boldsymbol{s}^* = [\boldsymbol{a}^*, \boldsymbol{\theta}^*] = [\underbrace{-0.2465, -0.6794, -0.8787, -1.1328, -0.8086}_{\boldsymbol{a}^*}, \underbrace{-4.9977, 4.8801, -4.9696, 4.7986}_{\boldsymbol{\theta}^*}],$$

which is listed in Table 1 Row 3. Therefore, the BC $B(\boldsymbol{x}) = -0.2465x_1^2 - 0.6794x_1x_2 - 0.8787x_1 - 1.1328x_2 - 0.8086$ satisfies the condition (13) for $\Delta(\boldsymbol{\theta}) = [-4.9977, 4.8801] \times [-4.9696, 4.7986]$ that covers $X = [-\frac{7}{5}, 2] \times [-\frac{7}{5}, \frac{1}{2}]$. Hence, (6) holds for the candidate $B(\boldsymbol{x})$ with the given X. Thus, it is a BC for the system, and the system is safe.

Case 4.3. *The complex structure of the human brain, with billions of neurons and trillions of synaptic connections, presents a modeling challenge. The Wilson-Cowan (WC) model [36] is a neurophysiological model that captures the brain's function through simplified differential equations grounded in neurophysiological principles. The WC model has shown correlation with experimental evidence, affirming its utility in studying neurological phenomena, including epilepsy. As mentioned in [37], the presence of seizure activity in a specific brain region is considered an unsafe state of the model. As such, it is important to study whether for a given combination of initial state and parameters the WC model is safe,* i.e.*, it shows no seizures. With the parameters set in [37], the system is described as*

$$\dot{x_1} = \frac{1}{0.0264}\left[-x_1 + \frac{1}{1 + e^{-(17x_1 + 10x_2 + 4.3 - d_1)}}\right],$$
$$\dot{x_2} = \frac{1}{0.012}\left[-x_2 + \frac{1}{1 + e^{-(25x_1 + 10 - d_2)}}\right],$$

where x_1 *and* x_2 *are the populations of excitatory and inhibitory neurons, respectively;* d_1, d_2 *represent disturbances to the input of* x_1, x_2, *respectively, and are both in* $[-0.2, 0.2]$. *In our experiments we set* $X_0 = [0, \frac{1}{5}] \times [\frac{4}{5}, 1]$, $X_u = [\frac{1}{3}, \frac{1}{2}] \times [\frac{1}{3}, \frac{1}{2}]$, *and* $X = [0, \frac{1}{2}] \times [\frac{1}{3}, 1]$. *Assume the BC is a polynomial structured as* $B(\boldsymbol{x}) = a_1x_1 + a_2x_2 + a_3$.

As shown in Table 1 Row 4, the algorithm returns

$$\boldsymbol{s}^* = [\boldsymbol{a}^*, \boldsymbol{\theta}^*] = [\underbrace{0.7654, -2.7858, 1.8012}_{\boldsymbol{a}^*}, \underbrace{-1.8210, 2.2383, -0.0945, 1.4833}_{\boldsymbol{\theta}^*}]$$

affirming that $B(\boldsymbol{x}) = 0.7654x_1 - 2.7858x_2 + 1.8012$ fulfills condition (13) for $\Delta(\boldsymbol{\theta}) = [-1.8210, 2.2383] \times [-0.0945, 1.4833]$, which covers $X = [0, \frac{1}{2}] \times [\frac{1}{3}, 1]$. Consequently, (6) is satisfied for the computed $B(\boldsymbol{x})$ under the given X, which makes it a valid BC for the system. In Figure 4 (mid panels) we display the BC found and the output of several intermediate steps performed by our synthesis algorithm.

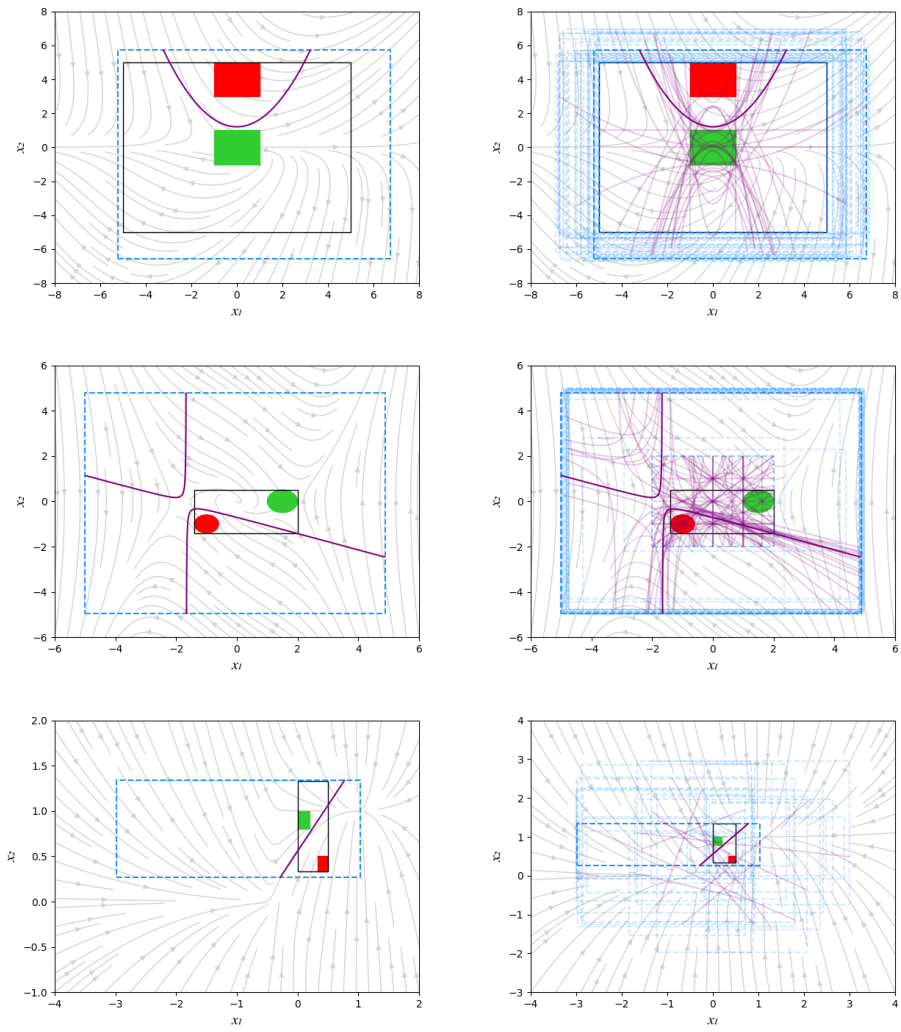


Figure 4: Graphical representation of barrier certificate synthesis. Grey arrows: vector fields of the system; green boxes and ovals: initial region X_0; red boxes and ovals: unsafe set X_u; black line: state space X; blue dashed rectangles: $\Delta(\boldsymbol{\theta})$; purple lines: graph defined by $B(\boldsymbol{x}) = 0$. Left-hand side column: final results of BC synthesis corresponding to Rows 2 (top), 3 (middle), and 5 (bottom) in Table 1, using the color legend above. Right-hand side column: intermediate steps of the BC synthesis leading to the final output displayed in the corresponding graph in the left-hand side column.

Case 4.4. *The Moore-Greitzer jet engine model is described as*

$$\begin{cases} \dot{x}_1 &= -1.5x_1^2 - x_2 - 0.5x_1^3, \\ \dot{x}_2 &= -x_1, \end{cases}$$

with initial set $X_0 = [0.1, 0.5] \times [0.1, 0.5]$, *unsafe set* $X_u = [0.7, 1] \times [0.7, 1]$, *and* $X = [0.1, 1] \times [0.1, 1]$. *Assume the BC is a polynomial as* $B(\boldsymbol{x}) = a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4$.

Our algorithm's output is

$$\boldsymbol{s}^* = [\boldsymbol{a}^*, \boldsymbol{\theta}^*] = [\underbrace{1, 1, 1, -1}_{\boldsymbol{a}^*}, \underbrace{0.1, 1, 0.1, 1}_{\boldsymbol{\theta}^*}],$$

Therefore, the BC $B(\boldsymbol{x}) = 2x_1^2 + 2x_1 x_2 + 2x_2^2 - 2$ satisfies the condition (13) for $\Delta(\boldsymbol{\theta}) = [0.1, 1] \times [0.1, 1]$, which covers X. Hence, (6) holds for the candidate $B(\boldsymbol{x})$ with the given X. Thus, it is a BC for the system, and the system is safe.

Table 2
Hyperparameters used in the Case Studies.

	Type	λ_i	ρ	Mesh(a)	p_i^{target}	N	seed	Exploration Ratio
Case 1	1	1/2	1	1,3	e^{-4}	20	42	0.1
	2	1	1	1,3	e^{-4}	20	42	0.1
	3	1/2	1	1,3	e^{-4}	20	42	0.5
	4	1/2	1	1,3	e^{-4}	20	1	0.1
	5	1/2	1	1,5	e^{-4}	50	42	0.1
	6	1/2	1	1,3	e^{-6}	20	42	0.1
	7	1/2	1	1,3	e^{-2}	20	42	0.1
Case 2		1/2	1	1,3	e^{-4}	20	1	0.1
Case 3		1/2	1	1,3	e^{-4}	20	42	0.1
Case 4		1/2	1	1,3	e^{-4}	20	42	0.1

Table 3
CPU time statistics for 100 runs, compared to PRoTECT with the cvxopt solver.

CPU Time (s)	Our Work				PRoTECT(cvxopt)	
	min	max	avg	std	avg	std
Case 1 (type1)	1.58	6.58	2.05	0.93		
Case 1 (type2)	1.49	6.58	2.04	0.94		
Case 1 (type3)	1.57	6.55	2.11	0.93		
Case 1 (type4)	20.47	34.61	21.47	1.43	0.0010	0.0000
Case 1 (type5)	0.16	1.67	0.27	0.29		
Case 1 (type6)	1.50	6.56	2.02	0.95		
Case 1 (type7)	1.52	6.47	2.05	0.92		
Case 2	7.40	15.41	7.93	0.97	N/A	
Case 3	0.40	2384.06	147.41	308.52	N/A	
Case 4	0.05	1.31	0.17	0.22	0.0008	0.0000

Compared with Table 1 Row 4, the initial dataset is derived from the mesh grid of $\boldsymbol{a}$ with a step size of 2 in Row 5. Consequently, the initial dataset comprises 8 data points instead of 27. Notably, despite the reduction in the size of the dataset, the algorithm consistently returns valid results, of course at the expense of a higher number of iterations and thus higher CPU time. In Figure 4 (bottom panels) we display the BC found for Row 5 of Table 1.

The barrier certificates reported in Table 1 are verified by the SMT solvers Z3 [34] and dReal [33]. Specifically, according to the applicability, for polynomial-based Cases 4.1,4.2 and 4.4, the Z3 solver [34] was employed. For Case 4.3, which has transcendental elements, the dReal solver [33] was utilized for verification. Note that the *unsatisfiable* answer by dReal is formally correct.

Using a small-sized initialization and generating the dataset internally, the case studies demonstrate both the effectiveness of our approach. An insightful observation from the provided table highlights that, even for the same system, distinct initializations yield different results. Compared with computation time in the order of hours in [37] in the case study 4.3, our method successfully computes a BC over a bounded state space, requiring only a small dataset and computation times in the order of minutes on a standard laptop.

The hyperparameters are reported in Table 2. The statistics of the CPU times are reported in Table 3 and is compared against PRoTECT [38]. Note that PRoTECT is developed only for polynomial systems with sets in the form of intervals. Therefore, it cannot handle Case 2 and 3. In contrast, our approach, while being slower on polynomial cases, can handle more general nonlinear systems and sets.

5. Conclusion

In this paper, we proposed an optimization method for computing barrier certificates that give guarantees on safety of dynamical systems. The optimization utilizes the augmented Lagrangian framework and Gaussian process regression to efficiently represent black-box functions appearing in the constraints. The dataset needed for training the Gaussian process are sequentially generated based on Bayesian optimization and the augmented Lagrangian. The performance of the proposed optimization on the case studies shows that the method returns results with small-size samples, which are generated automatically, thus reducing the computational time from hours to minutes on the tested case studies. The computed barrier certificates are verified a posteriori using solvers from Satisfiability Modulo Theory. Our approach currently assumes the system dynamics are known and replaces the parameterized optimizations in the barrier certificate conditions with black-box functions that can be evaluated a finite number of times. In the future, we plan to relax this assumption by treating the system as a black-box model and couple the current optimization approach directly to the data gathered from the system.

Acknowledgments

P. Zuliani is supported by the SERICS project (PE00000014) under the Italian MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU. X. Ge is supported by the Independent Research Fund Denmark 10.46540/3120-00041B. S. Soudjani is supported by the grants EIC 101070802 and ERC 101089047.

Declaration on Generative AI

During the preparation of this work, the authors did not use Generative AI for preparation of our work.

References

[1] M. Brin, G. Stuck, Introduction to Dynamical Systems, 1st ed., Cambridge University Press, Cambridge, 2015.
[2] P. Tabuada, Verification and control of hybrid systems: a symbolic approach, Springer Science & Business Media, 2009.
[3] C. Baier, J.-P. Katoen, Principles of model checking, MIT press, 2008.
[4] S. Prajna, A. Jadbabaie, Safety verification of hybrid systems using barrier certificates, in: Hybrid Systems: Computation and Control, Springer, Berlin, Heidelberg, 2004, pp. 477–492.
[5] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, O. Maler, SpaceEx: Scalable verification of hybrid systems, in: Computer Aided Verification, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2011, pp. 379–395.
[6] R. Majumdar, M. Salamati, S. Soudjani, Neural abstraction-based controller synthesis and deployment, ACM Transactions on Embedded Computing Systems 22 (2023) 1–25.
[7] T. A. Henzinger, P. H. Ho, Algorithmic analysis of nonlinear hybrid systems, in: Computer Aided Verification, Springer Berlin Heidelberg, 1995, pp. 225–238.
[8] A. Platzer, Differential-algebraic Dynamic Logic for Differential-algebraic Programs, Journal of Logic and Computation 20 (2010) 309–352.
[9] S. Chen, X. Ge, Reachability analysis of linear systems, Acta Informatica 61 (2024) 231–260.
[10] G. Lafferriere, G. J. Pappas, S. Yovine, Symbolic reachability computation for families of linear vector fields, Journal of Symbolic Computation 32 (2001) 231–253.
[11] L. Geretti, J. A. D. Sandretto, M. Althoff, L. Benet, P. Collins, P. Duggirala, M. Forets, E. Kim, S. Mitsch, C. Schilling, M. Wetzlinger, Arch-comp22 category report: Continuous and hybrid systems with nonlinear dynamics, in: Proceedings of ARCH22, volume 90 of *EPiC Series in Computing*, EasyChair, 2022, pp. 86–112.

[12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada, Control barrier functions: Theory and applications, in: European control conference, IEEE, 2019, pp. 3420–3431.
[13] C. Li, Z. Zhang, N. Ahmed, Q. Liu, F. Liu, M. Buss, Safe feedback motion planning in unknown environments: An instantaneous local control barrier function approach, Journal of Intelligent & Robotic Systems 109 (2023) 40.
[14] L. Dai, T. Gan, B. Xia, N. Zhan, Barrier certificates revisited, Journal of Symbolic Computation 80 (2017) 62–86. SI: Program Verification.
[15] C. Sloth, G. J. Pappas, R. Wisniewski, Compositional safety analysis using barrier certificates, in: Proceedings of the 15th International Conference on HSCC, ACM, New York, USA, 2012, p. 15–24.
[16] P. Jagtap, S. Soudjani, M. Zamani, Formal synthesis of stochastic systems via control barrier certificates, IEEE Transactions on Automatic Control 66 (2020) 3097–3110.
[17] W. Liu, Y. Bai, L. Jiao, N. Zhan, Safety guarantee for time-delay systems with disturbances, Science China Information Sciences 66 (2023) 132102.
[18] M. Lewis, P. Zuliani, S. Soudjani, Verification of quantum systems using barrier certificates, in: International Conference on Quantitative Evaluation of Systems, Springer, 2023, pp. 346–362.
[19] M. Kazemi, R. Majumdar, M. Salamati, S. Soudjani, B. Wooding, Data-driven abstraction-based control synthesis, Nonlinear Analysis: Hybrid Systems 52 (2024) 101467.
[20] L. Romao, A. R. Hota, A. Abate, Distributionally robust optimal and safe control of stochastic systems via kernel conditional mean embedding, in: CDC'23, 2023, pp. 2016–2021.
[21] S. Bogomolov, J. Fitzgerald, S. Soudjani, P. Stankaitis, Data-driven reachability analysis of digital twin FMI models, in: International Symposium on Leveraging Applications of Formal Methods, Springer, 2022, pp. 139–158.
[22] A. Peruffo, D. Ahmed, A. Abate, Automated and formal synthesis of neural barrier certificates for dynamical models, in: International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2021, pp. 370–388.
[23] H. Zhao, N. Qi, L. Dehbi, X. Zeng, Z. Yang, Formal synthesis of neural barrier certificates for continuous systems via counterexample guided learning, ACM Trans. Embed. Comput. Syst. 22 (2023).
[24] J. Jackson, L. Laurenti, E. Frew, M. Lahijanian, Safety verification of unknown dynamical systems via Gaussian process regression, in: IEEE Conference on Decision and Control, 2020, pp. 860–866.
[25] M. A. Osborne, R. Garnett, S. J. Roberts, Gaussian processes for global optimization (2009).
[26] P. Jagtap, G. J. Pappas, M. Zamani, Control barrier functions for unknown nonlinear systems using Gaussian processes, in: 59th IEEE Conference on Decision and Control, IEEE, 2020, pp. 3699–3704.
[27] J. Jiang, Y. Zhao, S. Coogan, Safe learning for uncertainty-aware planning via interval MDP abstraction, IEEE Control Systems Letters 6 (2022) 2641–2646.
[28] R. Reed, L. Laurenti, M. Lahijanian, Promises of deep kernel learning for control synthesis, IEEE Control Systems Letters 7 (2023) 3986–3991.
[29] J. Stecher, L. Kiltz, K. Graichen, Semi-infinite programming using Gaussian process regression for robust design optimization, in: 2022 European Control Conference (ECC), 2022, pp. 52–59.
[30] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. Lee, P. Ranjan, G. Wells, S. M. Wild, Modeling an augmented Lagrangian for blackbox constrained optimization, Technometrics 58 (2016) 1–11.
[31] C. A. Aoues Younes, Benchmark study of numerical methods for reliability-based design optimization, Structural and Multidisciplinary Optimization 41 (2010) 277–294.
[32] C. E. Rasmussen, C. K. I. Williams, Gaussian Process for Machine Learning, MIT Press, 2006.
[33] S. Gao, S. Kong, E. M. Clarke, Dreal: An SMT solver for nonlinear theories over the reals, in: ADE'13, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 208–214.
[34] L. De Moura, N. Bjørner, Z3: An efficient SMT solver, in: TACAS'08, Springer, 2008, pp. 337–340.
[35] H. Kong, F. He, X. Song, W. N. N. Hung, M. Gu, Exponential-condition-based barrier certificate generation for safety verification of hybrid systems, in: CAV, Springer, 2013, pp. 242–257.
[36] H. R. Wilson, J. D. Cowan, Excitatory and inhibitory interactions in localized populations of model neurons, Biophysical Journal 12 (1972).
[37] J. F. Ingham, Y. Wang, P. Zuliani, S. Soudjani, Barrier certificates for a computational model of

epileptic seizures, in: IEEE Int. Conf. on Systems, Man, and Cybernetics, 2023, pp. 4728–4733.
[38] B. Wooding, V. Horbanov, A. Lavaei, PRoTECT: Parallel construction of barrier certificates for safety verification of polynomial systems, in: Proceedings of the ACM/IEEE ICCPS, ACM, 2025.