IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 June 2024, accepted 29 July 2024, date of publication 1 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3436515

RESEARCH ARTICLE

BEACON: A Bayesian Evolutionary Approach for Counterexample Generation of Control Systems

JOSHUA YANCOSEK[1] **AND ALI BAHERI**[2]**, (Member, IEEE)**
[1]Department of Mechanical, Materials and Aerospace Engineering, West Virginia University, Morgantown, WV 26506, USA
[2]Department of Mechanical Engineering, Rochester Institute of Technology, Rochester, NY 14623, USA

Corresponding author: Ali Baheri (akbeme@rit.edu)

ABSTRACT The rigorous safety verification of control systems in critical applications is essential, given their increasing complexity and integration into everyday life. Simulation-based falsification approaches play a pivotal role in the safety verification of control systems, particularly within critical applications. These methods systematically explore the operational space of systems to identify configurations that result in violations of safety specifications. However, the effectiveness of traditional simulation-based falsification is frequently limited by the high dimensionality of the search space and the substantial computational resources required for exhaustive exploration. This paper presents BEACON, a novel framework that enhances the falsification process through a combination of Bayesian optimization and covariance matrix adaptation evolutionary strategy. By exploiting quantitative metrics to evaluate how closely a system adheres to safety specifications, BEACON advances the state-of-the-art in testing methodologies. It employs a model-based test point selection approach, designed to facilitate exploration across dynamically evolving search zones to efficiently uncover safety violations. Our findings demonstrate that BEACON not only locates a higher percentage of counterexamples compared to standalone BO but also achieves this with significantly fewer simulations than required by CMA-ES, highlighting its potential to optimize the verification process of control systems. This framework offers a promising direction for achieving thorough and resource-efficient safety evaluations, ensuring the reliability of control systems in critical applications. A Python implementation of the algorithm can be found at https://github.com/SAILRIT/BO-CMA.

INDEX TERMS Falsification, Bayesian optimization, covariance matrix adaptation evolutionary strategy, safety-critical systems.

I. INTRODUCTION

The growth of cyber-physical systems, such as autonomous vehicles and robotics, has significantly raised the importance of ensuring controllers operate safely and reliably [1]. Traditional approaches to safety verification, such as formal verification, simulation-based testing, and model checking, each contribute valuable insights but also face significant limitations [2]. Formal verification methods, grounded in rigorous mathematical proofs, offer a high degree of assurance by proving system properties. However, their practical application is often constrained by the complex nature of control systems and the computational intensity required to analyze large state spaces—a phenomenon known as the state explosion problem. The state explosion problem refers to the exponential growth of the number of states in a system as its complexity increases, making exhaustive exploration computationally infeasible. This challenge limits the scalability of formal verification techniques, particularly in the context of high-dimensional and nonlinear control systems [3]. Model checking automates the process of verifying whether a system's model meets specified criteria [4]. While effective for discrete systems, model checking struggles with scalability and the complexities of systems with continuous states.

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

Simulation-based falsification, on the other hand, has become an essential aspect of safety validation in control systems, particularly in safety-critical systems [5], [6], [7]. Falsification pertains to the systematic discovery of counterexamples, or conditions under which the system fails to meet safety specifications. It serves as a tool in the design-time assurance process, especially when handling complex systems where conventional verification approaches fall short due to nonlinearities and high dimensionality. Software tools such as S-TaLiRo [8], [9], Breach [10], C2E2 [11], and DryVR [12] have been instrumental in system falsification. These tools are designed to automate and facilitate the falsification process. They provide frameworks for systematically exploring the system's behavior under various conditions, seeking configurations that lead to specification violations. Various falsification methods, including search-based testing [13], [14], [15], [16], [17], [18], [19], optimization-based testing [20], [21], [22], [23], [24], [25], [26] and machine learning approaches [27], [28], [29], [30] offer different strengths.

Several works have demonstrated the potential of hybrid methodologies in the domain of safety verification [31], [32], [33]. Some approaches have integrated symbolic methods, which provide strong guarantees on correctness and completeness, with numeric methods, known for their efficiency and scalability [34]. The integration of machine learning algorithms with traditional search or optimization-based falsification methods has been explored to predict areas of the parameter space more likely to yield counterexamples [35]. This predictive capability can guide the falsification process more effectively, reducing the number of simulations required. Approaches that adaptively adjust their search have shown promise [36], [37].

Despite these advancements, a significant gap remains in the ability to efficiently identify safety violations in complex control systems. Many of the existing hybrid approaches still face challenges in balancing exploration and exploitation, dealing with high-dimensional spaces. We propose the `BEACON` framework, a **B**ayesian **E**volutionary **A**pproach for **CO**u**N**terexample Generation. It is designed to tackle these challenges by integrating Bayesian optimization (BO) with covariance matrix adaptation evolutionary strategy (CMA-ES) into a cohesive hybrid strategy.

The rationale behind the integration of BO and CMA-ES in `BEACON` is twofold. At its core, BO excels in efficiently exploring search spaces by using a probabilistic model, typically a Gaussian process (GP), to guide the search process. This model-based approach enables BO to make informed decisions about where to sample next, balancing the trade-off between exploration (searching in new areas) and exploitation (focusing on areas with known potential). However, BO's performance can be limited by the accuracy of the surrogate model and its tendency to focus too narrowly on regions of perceived interest, potentially overlooking other critical areas of the search space. CMA-ES addresses some of the limitations inherent in BO by employing an evolutionary strategy that adaptively refines the search based on the fitness of previous candidates. CMA-ES maintains a multivariate Gaussian distribution over the search space, characterized by a mean vector and a covariance matrix. The mean vector represents the current best estimate of the optimal solution, while the covariance matrix captures the correlations and scaling of the search distribution. By updating the mean and covariance based on the most promising solutions in each iteration, CMA-ES can dynamically adjust the search distribution and adapt to the structure of the search space. This ability to adapt the search distribution based on evolutionary principles allows CMA-ES to effectively explore the search space.

The integration of BO and CMA-ES in `BEACON` combines the strengths of both approaches while mitigating their individual limitations. BO provides a global search mechanism, efficiently exploring the search space and identifying promising regions based on the surrogate model. CMA-ES, on the other hand, brings a powerful local search capability, adaptively refining the search within the identified regions and effectively navigating complex fitness landscapes. By alternating between these two search strategies, `BEACON` can strike a balance between exploration and exploitation, leveraging the surrogate model of BO to guide the global search and the evolutionary adaptation of CMA-ES to refine the solutions locally. This synergistic combination allows `BEACON` to efficiently explore high-dimensional search spaces, handle multiple local optima, and converge towards globally optimal solutions.

Our Contributions: In this paper, we present the following contributions:

- We propose a novel framework that synergistically merges BO and CMA-ES to efficiently uncover counterexamples in complex, high-dimensional uncertainty spaces.
- We conduct an extensive evaluation of our framework, emphasizing its adaptability and effectiveness in refining the search strategy for optimal falsification results.
- We release the `BEACON` framework as open-source software to facilitate the adoption of our approach within the safety verification community.

Paper Organization: Section II presents the foundational background to the proposed framework and the problem statement. Section III describes the `BEACON` framework. Section IV presents our experimental setup, results, and discussion. Finally, Section VI concludes the paper with a summary of the findings and future works.

II. PRELIMINARIES AND PROBLEM SETUP

A. SIGNAL TEMPORAL LOGIC

Specifications consist of properties (predicates) ψ over a continuous time signal. These properties are expressed in the formal language of signal temporal logic (STL) [38]. STL

TABLE 1. Reference of symbols commonly used throughout the paper.

Symbol	Definition
φ	Safety specification
$\rho_\varphi(.)$	Robustness function
$\mathcal{U}_G$	Global search space
$\mathcal{U}_L$	Local search zone
$\mathbf{e}$	A set of environmental parameters
e_n	An environmental parameter
$S(\mathbf{e})$	Signal/trajectory associated with the corresponding set of environmental parameters
b	Simulation budget
P	Simulation budget in a local parameter zone
n	Number of environmental parameters

TABLE 2. Quantitative semantics.

$\rho(S_\mathbf{e}(t), \psi)$	$=$	$c - \psi(\mathbf{x}[t])$
$\rho(S_\mathbf{e}(t), \neg\varphi)$	$=$	$-\rho(S_\mathbf{e}(t), \varphi)$
$\rho(S_\mathbf{e}(t), \varphi_1 \wedge \varphi_2)$	$=$	$\min(\rho(S_\mathbf{e}(t), \varphi_1), \rho(S_\mathbf{e}(t), \varphi_2))$
$\rho(S_\mathbf{e}(t), \varphi_1 \vee \varphi_2)$	$=$	$\max(\rho(S_\mathbf{e}(t), \varphi_1), \rho(S_\mathbf{e}(t), \varphi_2))$
$\rho(S_\mathbf{e}(t), \mathbf{G}_{[a,b]}\varphi)$	$=$	$\min_{t' \in [t+a, t+b]} \rho(S_\mathbf{e}(t'), \varphi)$
$\rho(S_\mathbf{e}(t), \mathbf{F}_{[a,b]}\varphi)$	$=$	$\max_{t' \in [t+a, t+b]} \rho(S_\mathbf{e}(t'), \varphi)$
$\rho(S_\mathbf{e}(t), \varphi_1 \mathbf{U} \varphi_2)$	$=$	$\max_{t' \in [t+a, t+b]}(\min(\rho(S_\mathbf{e}(t'), \varphi_2), \min_{t'' \in [0, t']} \rho(S_\mathbf{e}(t''), \varphi_1)))$

formulas are defined by:

$$\varphi := \top \mid \psi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{G}_{[a,b]}\varphi \mid \mathbf{F}_{[a,b]}\varphi \mid \varphi_1 \mathcal{U}_{[a,b]}\varphi_2 \tag{1}$$

Besides ψ which stands for a predicate or requirement, each of the following symbols represents a Boolean operator. $\top$ denotes the *true* operator; $\neg$ denotes *negation*; $\wedge$ denotes *and*; $\vee$ denotes *or*. $\mathbf{G}_{[a,b]}$ denotes globally true (*always*) over a time range $[a, b]$ where $a, b \in \mathbb{R}_{[0,\infty]}$ and $a \leq b$. $\mathbf{F}_{[a,b]}$ denotes *eventually*. *Eventually* states that the specification is true at some point within the time range. $\mathbf{U}_{[a,b]}$ denotes *until* which states that φ_1 remains true until φ_2 has been met.

Quantitative STL Semantics: STL formulas reveal whether a specification has been violated or not. By incorporating quantitative semantics, defined in Table 2, one acquires a measure of robustness, how well a signal follows the specification [39]. A positive robustness value indicates the specification is satisfied; a negative robustness value indicates the specification has been violated.

B. BAYESIAN OPTIMIZATION

BO is a sample-efficient, global optimization technique for expensive-to-evaluate, black-box objective functions that lack explicit analytical forms or are non-convex and non-differentiable, widely applied across various domains including machine learning, robotics, control, and design optimization problems [40], [41], [42], [43], [44]. BO has emerged as a tool for falsification tasks, where the goal is to discover system configurations that lead to undesirable behaviors or safety specification violations [23], [45], [46]. Given the unknown relationship between the robustness function and environment parameters, BO uses surrogate modeling, typically through Gaussian processes (GP), to approximate this function based on observed data.

Consider we have a set of n observations from previously evaluated environmental parameters, represented as $\mathbf{y}_n = \left[\hat{\rho}_\varphi(\mathbf{e}_1), \ldots, \hat{\rho}_\varphi(\mathbf{e}_n)\right]$ at environmental parameters $\mathbf{e}_1, \ldots, \mathbf{e}_n$. Here, $\hat{\rho}_\varphi(\mathbf{e}) = \rho_\varphi(\mathbf{e}) + \omega$ incorporates Gaussian noise $\omega \sim \mathcal{N}(0, \sigma^2)$. The posterior distribution of $\rho_\varphi(\mathbf{e})$ is characterized by the following equations for the mean $m_n(\mathbf{e})$, covariance $k_n(\mathbf{e}, \mathbf{e}')$, and variance $\sigma_n(\mathbf{e})$:

$$m_n(\mathbf{e}) = \mathbf{k}_n(\mathbf{e})(\mathbf{K}_n + \mathbf{I}_n\sigma^2)^{-1}\mathbf{y}_n \tag{2}$$

$$k_n(\mathbf{e}, \mathbf{e}') = k(\mathbf{e}, \mathbf{e}') - \mathbf{k}_n(\mathbf{e})(\mathbf{K}_n + \mathbf{I}_n\sigma^2)^{-1}\mathbf{k}_n^T(\mathbf{e}') \tag{3}$$

$$\sigma_n^2(\mathbf{e}) = k_n(\mathbf{e}, \mathbf{e}') \tag{4}$$

The covariance between a new set of environmental parameters and the previous ones is captured in the vector $\mathbf{k}_n(\mathbf{e}) = [k(\mathbf{e}, \mathbf{e}_1), \ldots, k(\mathbf{e}, \mathbf{e}_n)]$. Here, $\sigma_n^2(\mathbf{e})$ denotes the variance, $\mathbf{I}_n$ represents the identity matrix, and $\mathbf{K}_n$ refers to the kernel matrix with entries $\left[k_n(\mathbf{e}, \mathbf{e}')\right]$.

C. COVARIANCE MATRIX ADAPTATION EVOLUTIONARY STRATEGY (CMA-ES)

The covariance matrix adaptation evolutionary strategy (CMA-ES) is a powerful optimization algorithm that belongs to the class of evolutionary algorithms. It is particularly well-suited for solving high-dimensional, non-convex optimization problems [47]. Central to CMA-ES is its strategy of exploiting the correlations among variables to steer the search towards the global optimum efficiently. This is achieved through a mechanism that dynamically adapts the search strategy based on the history of previous evaluations.

CMA-ES initiates its process by generating a population of P candidate solutions $\mathbf{e}_1, \ldots, \mathbf{e}_P$, each representing a set of environmental parameters. These candidates are sampled from a multivariate normal distribution defined as:

$$\mathbf{e}_i \sim \mathcal{N}(\boldsymbol{\mu}, \lambda^2\mathbf{C}), \quad i = 1, \ldots, P \tag{5}$$

where $\boldsymbol{\mu}$ represents the mean vector, $\mathbf{C}$ denotes the covariance matrix capturing the relationship between variables, and λ signifies the scale or step size of the search. To refine its search strategy, CMA-ES evaluates the generated candidates based on the robustness function $\rho_\varphi(\mathbf{e})$ and ranks them according to their performance, from the least to the most robust outcomes:

$$\rho_\varphi(\mathbf{e}_1) \leq \ldots \leq \rho_\varphi(\mathbf{e}_P) \tag{6}$$

Subsequently, the algorithm updates the mean $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{C}$ based on the top-performing candidates, identified as P_{best}, where $P_{best} \leq P$. The updated mean and covariance matrix are calculated as follows:

$$\boldsymbol{\mu} = \frac{1}{P_{best}} \sum_{i=1}^{P_{best}} \mathbf{e}_i \tag{7}$$

$$\sigma_{nn'}^2 = \frac{1}{P_{best}} \sum_{i=1}^{P_{best}} (\mathbf{e}_{i,n} - \mu_n)(\mathbf{e}_{i,n'} - \mu_{n'}), \tag{8}$$

where $\sigma^2_{nn'}$ represents the elements of the covariance matrix $\mathbf{C}$, and $\mathbf{e}_{i,n}$ and μ_n are the n^{th} components of the i^{th} environmental parameter vector and the mean vector, respectively.

This adaptive process allows CMA-ES to iteratively refine its search distribution, progressively focusing on more promising regions of the search space. The algorithm continues this evolutionary cycle, adjusting $\boldsymbol{\mu}$ and $\mathbf{C}$ with each generation, until a predefined termination criterion is met.

D. PROBLEM STATEMENT

Given a system under test (SUT) embedded within a high-dimensional parameter space $\mathcal{U}_G \subseteq \mathbb{R}^n$, the goal of falsification is to identify sets of environmental parameters $\mathbf{e} \in \mathcal{U}_G$ that lead the SUT to violate one or more predefined safety specifications. These safety specifications are formalized as constraints over the system's output trajectories $S(\mathbf{e})$, where $S : \mathbb{R}^n \to \mathbb{R}^m$ is a function mapping environmental parameters to system responses. A safety specification φ is inherently defined in relation to the trajectories of a SUT. We interpret φ to include all finite-horizon trajectories $S(\mathbf{e})$ that adhere to the defined system safety requirements. A trajectory $S(\mathbf{e})$, resulting from a set of environmental parameters $\mathbf{e}$, is deemed compliant with the specification φ if and only if $S(\mathbf{e}) \in \varphi$. This condition is denoted as $S \models \varphi$, meaning that φ evaluates the trajectory $S(\mathbf{e})$ as satisfying the safety specification. The structure of φ is derived from multiple individual conditions, termed predicates. These predicates act as the elemental logical units that, through a combination of logical operations, construct the overall safety specification. Each predicate μ is considered a continuous function evaluated along the trajectory $S(\mathbf{e})$. Satisfaction of a predicate occurs when $\mu(S(\mathbf{e})) > 0$, indicating adherence to the safety criterion; otherwise, the predicate—and by extension, the trajectory—is considered falsified. Instead of merely assessing the Boolean satisfaction of a predicate, the notion of robust or quantitative semantics is introduced to measure the extent of satisfaction [39]. This approach introduces a more refined safety assessment by associating a real-valued function $\rho_\phi(S(\mathbf{e}))$ with each predicate, which is evaluated along the system trajectory $S(\mathbf{e})$. This function serves as a "measure" of how significantly the safety specification is satisfied. The falsification task can thus be represented as an optimization problem:

$$\underset{\mathbf{e}}{\operatorname{argmin}}\, \rho_\varphi\left(S\left(\mathbf{e}\right)\right) \tag{9}$$

where $\rho_\varphi(S(\mathbf{e}))$ is a robustness metric quantifying the degree of safety specification violation by the system's output for a given set of parameters $\mathbf{e}$.

The BEACON framework addresses this optimization problem through a hybrid strategy that combines the exploratory strengths of BO with the adaptive capabilities of CMA-ES. Specifically, the framework partitions the global search space $\mathcal{U}_G$ into localized search zones $\mathcal{U}_L$, each potentially containing parameter sets that lead to specification violations.

1) BAYESIAN OPTIMIZATION COMPONENT

BO is applied within each $\mathcal{U}_L$ to efficiently identify parameter sets that are likely to violate safety specifications. This is achieved by constructing a probabilistic model (e.g., a GP) of the robustness metric $\rho_\varphi(S(\mathbf{e}))$ and using acquisition functions to guide the selection of new parameter sets for evaluation.

2) COVARIANCE MATRIX ADAPTATION EVOLUTIONARY STRATEGY (CMA-ES) COMPONENT

CMA-ES is used to adaptively refine the search within $\mathcal{U}_L$ based on the outcomes of previous evaluations. It adjusts the sampling distribution (mean and covariance) to concentrate future simulations in regions of the parameter space more likely to uncover falsifying examples.

III. METHODOLOGY

This section details the BEACON framework, a novel approach that integrates BO and CMA-ES strategy to advance the falsification of control systems. The framework is designed to efficiently identify counterexamples in complex, high-dimensional search spaces characterized by numerous local optima. By synergistically combining the explorative capabilities of BO with the global search strategy of CMA-ES, BEACON aims to significantly reduce the number of simulations required to locate violations of safety specifications. The core strategy is depicted in Figure 1.

The methodology uses a strategic division of the global search space, $\mathcal{U}_G$, into localized search zones, $\mathcal{U}_L \subseteq \mathcal{U}_G$, each defined by the adaptive mechanisms inherent in CMA-ES. This partitioning enables focused exploration and exploitation within subsets of the search space. Within each localized search zone $\mathcal{U}_L$, the BO constructs a GP model to serve as a surrogate for the system's robustness function, $\rho_\varphi(\mathbf{e})$. To guide the selection of new test points within $\mathcal{U}_L$, the framework uses the lower confidence bound (LCB) acquisition function, balancing the exploration of unexplored regions against the exploitation of known areas of interest:

$$\mathbf{e}_n = \underset{\mathbf{e}\in\mathcal{U}_L}{\operatorname{argmin}}\, m_{n-1}(\mathbf{e}) - \xi^{\frac{1}{2}}\sigma^{BO}_{n-1}(\mathbf{e}) \tag{10}$$

where ξ dynamically adjusts the focus between exploring new areas and exploiting existing knowledge to efficiently converge on global minima.

After simulating a set of P environmental parameters within the current localized search zone $\mathcal{U}_L$, the framework applies CMA-ES's adaptive mechanisms to update the search strategy. This process begins by evaluating the robustness of the P_{best} performing parameters, which then informs the calculation of the mean vector, μ, and the variance, σ^2. These statistical parameters are crucial for shaping the boundaries of the next local search zones:

$$\mathcal{U}_L = \bigcup_{i=1}^{n} [\mu_i - \sigma_i, \mu_i + \sigma_i] \tag{11}$$

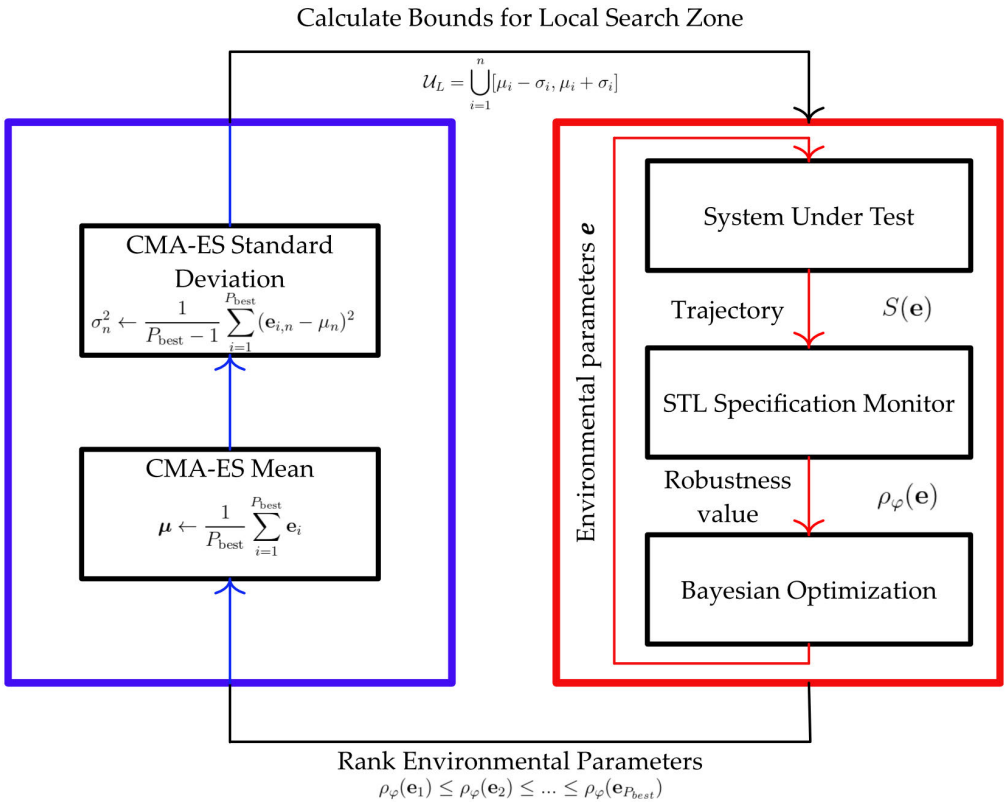


FIGURE 1. Schematic Representation of the BEACON Falsification Framework. This framework constructs a model for evaluating system specifications within a defined local search zone, $\mathcal{U}_L \subseteq \mathcal{U}_G$, as highlighted by the red box. Over P iterations, BO is used to select new environmental parameters for simulation within $\mathcal{U}_L$. Upon exhausting the iteration budget P, the framework uses the P_{best} environmental parameters to derive the mean and standard deviation, using principles from the CMA-ES, as indicated in blue. These statistical measures are then used to determine the upper and lower bounds of the subsequent local search zone, setting the stage for the next cycle of the process.

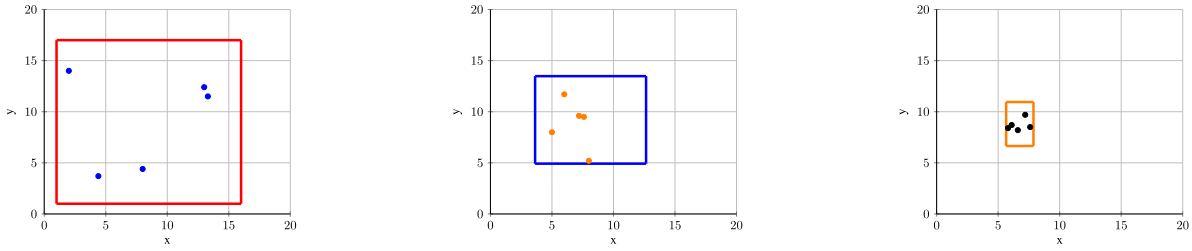
The Eq. 11 shows how the BEACON framework dynamically tailors local search zones, $\mathcal{U}_L$, through a union of intervals across each dimension of the input space. Each interval is centered around the mean, μ_i, of the best-performing parameters, expanded by their standard deviation, σ_i. This adjustment ensures that the search zones are not only concentrated around the most informative regions identified thus far but also sufficiently broad to explore areas that may harbor undiscovered counterexamples. Figure 2 provides a visual illustration of how the local search zones evolve over three consecutive iterations in a 2-dimensional global search space. Each subsequent search zone contracts around the P_{best} environmental parameters highlighted in the prior search zone. The visualization helps to understand how the BEACON framework progressively refines the search regions based on the information gained from previous iterations, enabling a more targeted exploration of the parameter space.

In the iterative exploration of local search zones by BEACON, we closely monitor the evolution of robustness values to identify any signs of stagnation. Stagnation occurs when the algorithm fails to find lower robustness values in successive local search zones, indicating a potential trap in a local optimum or having reached the vicinity of the global minimum. Mathematically, this is evaluated by comparing the minimum robustness values between consecutive search zones, as follows. Let $\hat{\mathbf{z}} = [\mathbf{e}_1^{(g)}, \ldots, \mathbf{e}_P^{(g)}]$ denote the set of environmental parameters evaluated in the current search zone (g), and $\hat{\mathbf{y}} = [\mathbf{e}_1^{(g-1)}, \ldots, \mathbf{e}_P^{(g-1)}]$ represent those from the previous zone $(g-1)$. The corresponding robustness vectors for these sets are $\mathbf{z} = [\rho_\varphi(\mathbf{e}_1^{(g)}), \ldots, \rho_\varphi(\mathbf{e}_P^{(g)})]$ and $\mathbf{y} = [\rho_\varphi(\mathbf{e}_1^{(g-1)}), \ldots, \rho_\varphi(\mathbf{e}_P^{(g-1)})]$, respectively. Stagnation is formally detected when:

$$\min(\mathbf{z}) > \min(\mathbf{y}) \tag{12}$$

implying no improvement in robustness has been achieved in the most recent search zone compared to its predecessor.

To address potential stagnation and avoid exhaustive focus on suboptimal regions, the BEACON framework incorporates a stagnation monitoring mechanism. This mechanism activates when the search fails to yield improved robustness outcomes over successive iterations. A predetermined threshold, δ, represents the maximum number of consecutive local search zones allowed without observing any improvement. The counter, γ, tracks the number of such zones, and once γ equals or exceeds δ, it triggers a shift. This shift entails relocating the search focus to an unexplored area of the global search space, $\mathcal{U}_G$, in pursuit of new counterexamples.



(a) First Local Search Zone (b) Second Local Search Zone (c) Third Local Search Zone

FIGURE 2. Illustration of the BEACON methodology applied within a 2-dimensional global search space $\mathcal{U}_G = [0, 20]^2$. Each subfigure shows the evolving boundaries of local search zones, with the highlighted points representing the P_{best} environmental parameters selected to refine the subsequent search space. This sequential adaptation showcases the framework's progression through the search space to efficiently explore regions of interest.

Algorithm 1 BEACON: **B**ayesian **E**volutionary **A**pproach for **CO**u**N**terexample Generation

1: **Input:** global search space $\mathcal{U}_G$, global simulation budget b, search zone simulation budget P, and stagnation factor δ
2: **Initialize:** simulate random samples
3: **Until** simulation count = b **do**:
4: **for** $i = 1 to n$ **do**
5: $\boldsymbol{\mu} \leftarrow \frac{1}{P_{\text{best}}} \sum_{i=1}^{P_{\text{best}}} \mathbf{e}_i$ ▷ Calculate CMA-ES mean
6: $\sigma_n^2 \leftarrow \frac{1}{P_{\text{best}}-1} \sum_{i=1}^{P_{\text{best}}} (\mathbf{e}_{i,n} - \mu_n)^2$ ▷ Calculate CMA-ES variance
7: $\mathcal{U}_L \leftarrow \bigcup_{i=1}^{n} [\mu_i - \sigma_i, \mu_i + \sigma_i]$
8: **end for**
9: Initialize local GP model of $\mathcal{U}_L$
10: **for** $p = 1 to P$ **do**
11: $\mathbf{e}_n \leftarrow \text{argmin}_{\mathbf{e} \in \mathcal{U}_L} m_{n-1}(\mathbf{e}) - \xi^{\frac{1}{2}} \sigma_{n-1}^{BO}(\mathbf{e})$ ▷ Select parameters with BO
12: Update GP model with $\rho_\varphi(\mathbf{e})$ ▷ Refine GP model with new data
13: **end for**
14: Previous search zone: $\mathbf{y} \leftarrow [\rho_\varphi(\mathbf{e}_1^{(g-1)}), \ldots, \rho_\varphi(\mathbf{e}_P^{(g-1)})]$
15: Current search zone: $\mathbf{z} \leftarrow [\rho_\varphi(\mathbf{e}_1^{(g)}), \ldots, \rho_\varphi(\mathbf{e}_P^{(g)})]$
16: **if** $\min(\mathbf{z}) \leq \min(\mathbf{y})$ **then** ▷ Stagnation monitor process
17: Update $\mathcal{U}_L$
18: **else**
19: $\gamma \leftarrow \gamma + 1$
20: **if** $\gamma \geq \delta$ **then**
21: Shift $\mathcal{U}_L$ to an unexplored region of $\mathcal{U}_G$
22: **end if**
23: **end if**

We present the BEACON framework in Algorithm 1. The algorithm initiates with a set of random samples for simulation and records the associated robustness values. Using equations from CMA-ES, the mean and variance are calculated to determine the boundaries of the subsequent local search space (line 6 – 8). Within this new local search space, BO selects environmental parameters for simulation based on the LCB acquisition function (line 11). After simulating the chosen parameters, a GP model is constructed to map the inputs to their robustness values (line 12). If stagnation is observed, the search is redirected to a new region (line 16 – 23).

IV. EXPERIMENTAL SETUPS AND CASE STUDIES

We evaluate the proposed method against vanilla BO and CMA-ES on several benchmark problems. For each case study, the methods are exposed to the same uncertainty space $\mathcal{U}_G$. BEACON and vanilla BO are subject to simulation budgets of 100, 200, 300, 400, and 500 where they perform 150 tests for each budget. CMA-ES is subject to 150 tests per case study and is not restricted to a simulation budget.

For BEACON, we choose several user-defined settings prior to testing that remain consistent across experiments. The local search zone simulation budget P is set to 20 simulations. For this study, BEACON uses the top quarter $P_{best} = 5$ of the environmental parameters to calculate $\mathcal{U}_L$. Finally, we set the stagnation constant $\delta = 2$ so that BEACON can effectively exploit local areas, but retain resources for wider exploration. Vanilla BO starts with an initial sampling of 20 parameters, the same as with BEACON. Finally, each generation of CMA-ES performs 20 simulations from which the top 5 are used to adapt the covariance matrix.

A. CASE STUDY 1: MOUNTAIN CAR

The mountain car environment is an autonomous car situated at the bottom of a valley on a one-dimensional track. The car's objective is to ascend the right hill by employing acceleration in both the left and right directions. The car has two observable states: its position x and its velocity $\dot{x}$. We consider four sources of uncertainty whose ranges are provided in Table 3. The uncertain parameters are the initial position x, initial velocity $\dot{x}$, the car's maximum velocity $\dot{x}_{max}$, and maximum power magnitude ρ_{max}.

The car is controlled by a policy trained with deep deterministic policy gradient (DDPG) [48]. The controller is subject to two safety specifications simultaneously represented in STL format in Table 3. The first specification states that the car's velocity should *always* remain below 0.0735 when its position is less than -1.1 or greater than 0.5. Second, the car's velocity should remain below 0.055 *until* it has reached the position 0.1.

TABLE 3. Mountain car specifications and environmental parameters.

Specification	STL specification	
φ_1	$\mathbf{G}((x \leq -1.1) \vee x \geq 0.5) \wedge (\dot{x} < 0.0735))$	
φ_2	$(\dot{x} < 0.055)\mathbf{U}(x > 0.1)$	
Parameter	**Symbol**	**Range**
Position	x	$[-0.6, -0.4]$
Velocity	$\dot{x}$	$[-0.025, 0.025]$
Maximum velocity	$\dot{x}_{max}$	$[0.040, 0.075]$
Power magnitude	ρ_{max}	$[0.0005, 0.0025]$

TABLE 4. Automatic transmission specifications and environmental parameters.

Specification	STL specification	
φ_1	$\mathbf{G}(\dot{x} \leq 80)$	
φ_2	$\mathbf{G}(\omega \leq 1400)$	
Parameter	**Symbol**	**Range**
Throttle	θ_{thr}	$[0, 100]^2$
Brake	θ_{brk}	$[0, 100]^2$

B. CASE STUDY 2: AUTOMATIC TRANSMISSION

The automatic transmission environment is a scenario that simulates the speed of a 4-gear vehicle with an automatic transmission. The simulation has two observable properties: the vehicle's speed $\dot{x}$ and engine speed ω. We consider four sources of uncertainty whose ranges are provided in Table 4. We consider two input signals, the throttle angle θ_{thr} and brake angle θ_{brk}.

We explore the uncertainty space for a combination of environmental parameters that cause the vehicle to violate one of the following two specifications presented in Table 4. First, the vehicle's speed should *always* remain below 80mph. Second, the engine speed should *always* remain below 1400rpm.

C. CASE STUDY 3: NEURAL NETWORK CONTROLLER

This environment models a magnet levitating above an electromagnet, maintaining a specific reference height. The simulation tracks the height h of the magnet, with the only input being the reference position. Thus, the model incorporates eight sources of uncertainty, detailed in Table 5.

We evaluate the nonlinear autoregressive moving average (NARMA) neural controller's capability to move the magnet to a reference position by controlling the current [49]. The neural controller consists of a neural network with nine hidden layers. The controller is subjected to two safety specifications given in STL format in Table 5. First, the controller should *always* keep the magnet below 3.9mm. Second, the magnet should *always* settle to a new reference position within two seconds. This can be reworded as the magnet should *eventually* remain within the specified range of the reference position for one second.

D. CASE STUDY 4: F16-GROUND COLLISION AVOIDANCE

This environment simulates the F-16 control system, with a specific focus on the aircraft's ground collision avoidance inner-loop controller, modeled by 16 continuous piecewise nonlinear differential equations [50]. Although the F-16 environment features a wide range of observable properties, such as roll, pitch, and yaw angles, the primary concern for the falsification problem is the aircraft's altitude.

We falsify the controller against five sources of uncertainty listed in Table 6. The uncertainty parameters consist of the altitude *alt*, initial velocity $\dot{x}$, roll angle θ, pitch angle ϕ, and yaw angle ω. The controller is subject to the singular safety specification that the aircraft should *always* avoid colliding with the ground during evasive maneuvers.

E. CASE STUDY 5: AIR FUEL CONTROL

The air fuel control system model captures the dynamics of fuel regulation, focusing on the air-fuel ratio in response to varying inputs such as throttle angle and engine speed. This model allows us to analyze the behavior of the air-fuel mixture across different operational conditions [51]. The uncertainty space consists of 11 parameters whose ranges are given in Table 7. The system is subjected to one safety parameter provided in Table 7. The specification states that the air-fuel ratio should *always* remain with 0.7% of the value of 14.7, otherwise, the system may emit undesirable quantities of noxious fumes.

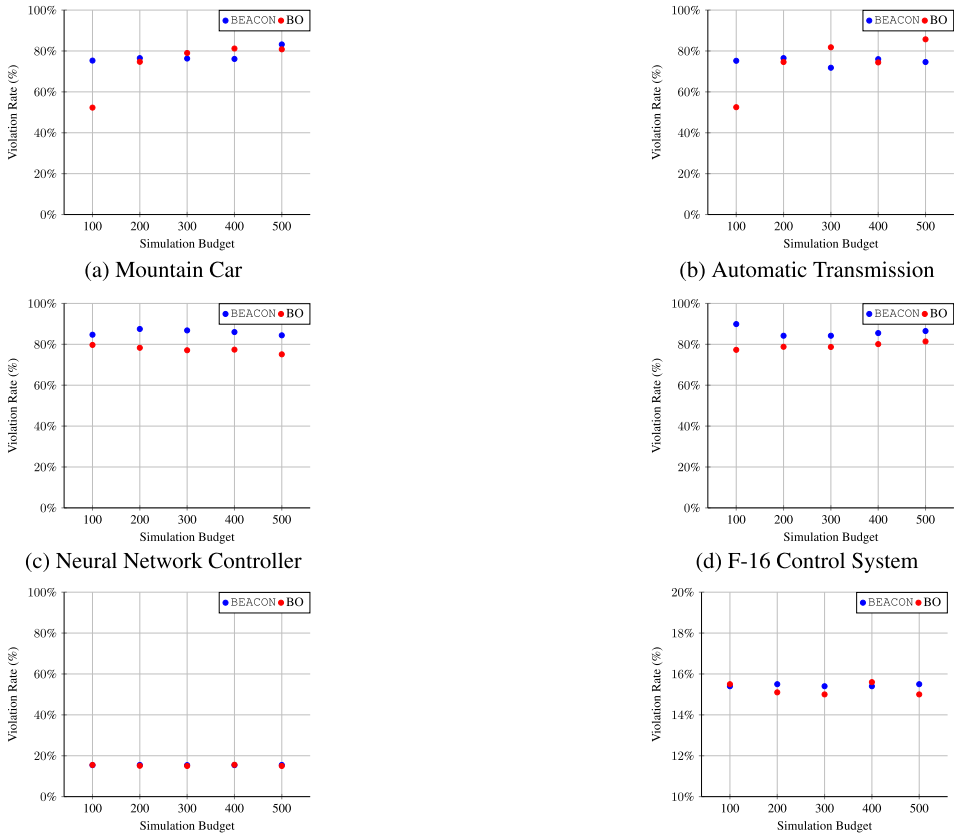
V. RESULTS AND DISCUSSION

We present the violation rates and simulations required to locate a violation for each experiment from BEACON, BO, and CMA-ES in Table 8. The highlighted numbers indicate the highest violation rate achieved within a case study between BEACON and BO. Numbers in bold represent the higher violation rate obtained in each experiment. The results for BEACON and BO are visualized in Figure 3 for each case study.

In our initial case study, the mountain car scenario, BEACON consistently outperforms BO. BEACON achieves an average violation rate of 77.5% compared to 73.7% for BO. In particular, BEACON achieves a higher violation rate in three of the five experiments which have simulation budgets of 100, 200, and 500. CMA-ES, while exhibiting a violation rate of 86.9%, is accompanied by a significant resource demand of 4788 simulations, in contrast to BEACON's highest performance at 500 simulations with a violation rate of 83.2%. BEACON's results are within 3.7% of those achieved by CMA-ES, all while utilizing only a tenth of the resource budget. Similar trends are recorded in the automatic transmission environment. Here, BEACON maintains an average violation rate of 74.8%, surpassing BO's 73.8%. In this case study, BEACON secures higher violation rates in three of the experiments given 100, 200, and 400 simulations. CMA-ES, with an average violation rate of 83.7%, comes with a cost of 4744 simulations. In contrast, BEACON reaches its highest rate with 200 simulations at 76.5%. BEACON, requiring only 4% of the simulations to achieve within 7.2% of CMA-ES's violation rate.

TABLE 5. Neural network specifications and environmental parameters.

Specification	STL specification	
φ_1	$\mathbf{G}(h < 3.9)$	
φ_2	$\mathbf{G}_{[0,50]}(\neg(\lvert h - Ref\rvert > 0.005 + 0.04\lvert Ref\rvert) \Rightarrow \mathbf{F}_{[0,2]}\mathbf{G}_{[0,1]}(\lvert h - Ref\rvert \leq 0.005 + 0.04\lvert Ref\rvert))$	
Parameter	**Symbol**	**Range**
Reference	Ref	$[0, 3]^8$



(a) Mountain Car (b) Automatic Transmission

(c) Neural Network Controller (d) F-16 Control System

(e) Air Fuel Control (f) Air Fuel Control (Zoomed)

FIGURE 3. The illustration of violation rate vs. simulation budget for the mountain car, automatic transmission, neural network, F16, and air fuel control case studies. In these plots, BEACON's results are presented in blue, and BO's results are presented in red. BEACON performs better than BO at lower simulation budgets in the cases of mountain car and automatic transmission. In the air fuel control case study, BEACON and BO performed similarly across each budget. In the neural network and F-16 environments, BEACON achieves higher violation rates for each budget than BO.

TABLE 6. F16 specifications and environmental parameters.

Specification	STL specification	
φ_1	$\mathbf{G}_{[0,15]}(\text{altitude} > 0)$	
Parameter	**Symbol**	**Range**
Altitude	alt	[900, 4000]
Velocity	$\dot{x}$	[340, 740]
Roll angle	θ	[0.6283, 0.8900]
Pitch angle	ϕ	$[-1.6964, -1.5707]$
Yaw angle	ω	[0.7853, 1.17809]

TABLE 7. Air fuel control specification and environmental parameters.

Specification	STL specification	
φ_1	$\mathbf{G}_{[10,30]}(\lvert AF - 14.7\rvert < \text{tol} \cdot 14.7)$	
Parameter	**Symbol**	**Range**
Pedal angle	θ	$[0, 61]^{10}$
Engine speed	ω	[900, 1100]

BEACON demonstrates its second-highest performance in the neural network environment, consistently outperforming BO. BEACON achieves an average violation rate of 85.9% compared to BO's 77.5%. Although CMA-ES reaches its peak performance with a violation rate of 91.9%, this comes at the expense of an average of 1157 simulations. BEACON, on the other hand, achieves a rate of 87.5% with 200 simulations, delivering results within 4.4% of CMA-ES with only a fifth of the resources.

In the F-16 case study, BEACON shines, achieving an average violation rate of 86.1%. This rate is 7% higher

TABLE 8. Results for each case study found with BEACON, BO, and CMA-ES.

Case Study	Simulation Budget	BEACON Violation Rate	BEACON Sims/ Violation	BO Violation Rate	BO Sims/ Violation	CMA-ES Violation Rate	CMA-ES Sims
$\varphi_{1,2}^{MC}$	100	**75.3%**	1.33	52.3%	1.90	86.9%	4788
	200	**76.5%**	1.31	74.7%	1.33		
	300	76.3%	1.31	**79.0%**	1.27		
	400	76.1%	1.31	**81.2%**	1.23		
	500	**83.2%**	1.21	80.8%	1.24		
$\varphi_{1,2}^{AT}$	100	**75.2%**	1.33	52.5%	1.90	83.7%	4744
	200	**76.5%**	1.32	74.6%	1.34		
	300	71.8%	1.39	**81.8%**	1.22		
	400	**76.0%**	1.32	74.4%	1.34		
	500	74.6%	1.34	**85.7%**	1.16		
$\varphi_{1,2}^{NN}$	100	**84.7%**	1.18	79.7%	1.26	91.9%	1157
	200	**87.5%**	1.14	78.3%	1.28		
	300	**86.8%**	1.15	77.1%	1.30		
	400	**86.0%**	1.16	77.4%	1.29		
	500	**84.4%**	1.19	75.1%	1.33		
φ_{1}^{F16}	100	**89.9%**	1.11	77.3%	1.29	53.9%	6442
	200	**84.2%**	1.19	78.8%	1.26		
	300	**84.2%**	1.19	78.7%	1.27		
	400	**85.5%**	1.17	80.1%	1.25		
	500	**86.5%**	1.16	81.4%	1.23		
$\varphi_{1,2}^{AFC}$	100	15.4%	6.48	**15.5%**	6.47	16.9%	14363
	200	**15.5%**	6.47	15.1	6.61		
	300	**15.4%**	6.49	15.0%	6.66		
	400	15.4%	6.49	**15.6%**	6.43		
	500	**15.5%**	6.45	15.0%	6.66		

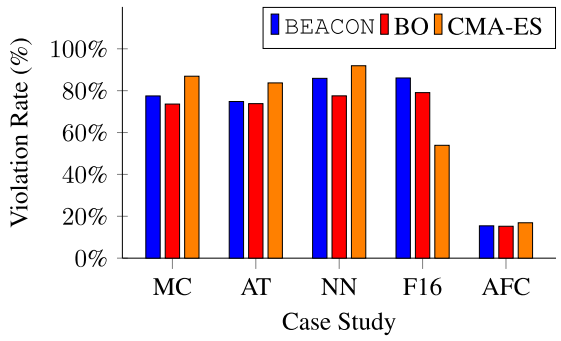


FIGURE 4. Comparative analysis of violation rates across case studies.

than that of BO's result and 36% higher than CMA-ES's 53.9%. Notably, BEACON outperforms BO and CMA-ES in each experiment. BEACON's lowest rate in this scenario, 84.2%, is achieved with 200/300 simulations, while BO's highest rate is 81.4% at 500 simulations, falling 2.8% short of BEACON's performance with a 1.5 times larger budget.

In the final case study, air fuel control, all three methods obtain similar rates. On average, BEACON achieves a rate of 15.44%, compared to 15.24% with BO, and 16.9% with CMA-ES. BEACON outperforms BO in three out of five experiments when given 200, 300, and 500 simulation budgets. CMA-ES requires 14363 simulations on average to achieve its 16.9% violation rate. BEACON performs within 1.4% of CMA-ES with 3.5% of the simulations. Across the five experiments for each case study, BEACON consistently outperforms BO by an average margin ranging from 0.2% to 8.3% as depicted in Figure 4. Table 9 provides further insight for our comparison by presenting the highest violation rate achieved with each method along with the required simulation budgets (in parentheses). In Table 9, green highlighting indicates the highest violation rate achieved by either BEACON or BO in a given case study, yellow highlights denote instances where CMA-ES achieved the highest violation rate, and red highlights signify the largest simulation budget required by the three methods. In three of the case studies, CMA-ES achieves the highest violation rate. However, in all cases, this approach requires far more resources to achieve its results compared to BEACON which

TABLE 9. Comparison of violation rates and simulation counts across different methodologies for each case study. The table showcases the highest violation rates attained by BEACON, BO, and CMA-ES. Parentheses indicate the number of simulations conducted to reach the noted violation rate. Green highlights denote instances where BEACON or BO demonstrates superior performance within the comparison, whereas yellow highlights emphasize case studies where CMA-ES outperforms. Orange highlights signify the scenarios demanding the highest simulation effort to achieve the reported outcomes.

Case Study	BEACON	BO	CMA-ES
$\varphi_{1,2}^{MC}$	83.2% (500)	81.2% (400)	86.9% (4788)
$\varphi_{1,2}^{AT}$	76.5% (200)	85.7% (500)	83.7% (4744)
$\varphi_{1,2}^{NN}$	87.5% (200)	79.7% (100)	91.9% (1157)
φ_{1}^{F16}	89.9%(100)	81.4% (500)	53.9% (6442)
$\varphi_{1,2}^{AFC}$	15.5%(500)	15.6%(400)	16.9% (14363)

was discussed in each case study. BEACON achieves the highest rate out of the three methods in the F-16 environment, and higher rates than BO in mountain car and neural network. From the data, we can observe that BEACON tends to achieve its highest rates with 200 or fewer simulations compared to BO whose highest rates occur mostly from $400 - 500$ simulation budgets.

Several key conclusions can be drawn from these results. BEACON excels in situations where locating violations can prove challenging, such as in the neural network and F-16 environments. Additionally, BEACON operates more efficiently at lower simulation budgets, primarily due to its ability to perform multiple uncertainty space searches before exploring unknown regions. In contrast, BO tends to shift from exploration to exploitation as it acquires information. Overall, BEACON performs on par with or better than BO, depending on the circumstance, and is capable of achieving similar results to CMA-ES with significantly fewer resources.

Limitations: The BEACON framework, while promising, has certain limitations that should be acknowledged and addressed in future research. One limitation lies in the assumption of continuous and smooth robustness functions. In real-world systems, the robustness function may exhibit discontinuities or non-smooth behavior, which can impact the effectiveness of the GP model and the overall search process. Discontinuities can arise from abrupt changes in system dynamics or from the discrete nature of certain environmental parameters. Another limitation of BEACON is its scalability to high-dimensional search spaces. As the number of dimensions increases, the volume of the search space grows exponentially, making it challenging to efficiently explore and identify counterexamples. The curse of dimensionality can hinder the performance of BEACON, particularly in systems with a large number of environmental parameters. This limitation calls for the development of advanced sampling techniques, dimensionality reduction methods, and efficient surrogate models that can handle high-dimensional spaces.

VI. CONCLUSION

In this work, we have proposed BEACON, a novel hybrid falsification framework that integrates Bayesian optimization and covariance matrix adaptation evolutionary strategy, aiming to enhance the efficiency of safety violation detection in control systems. BEACON segments the global parameter space into localized search zones, enabling the generation of accurate surrogate models to guide the selection of environmental parameters more effectively. Through comprehensive evaluation across diverse case studies, BEACON has demonstrated its capability to not only match but in certain instances surpass the efficacy of its constituent methodologies in identifying counterexamples.

Future Work: There are several exciting avenues for future research and development that can further enhance the capabilities and performance of the BEACON. One crucial direction is to investigate the integration of dynamic parameter ranges from the CMA-ES component into the BO process. Currently, the BO assumes fixed uncertainty spaces for each parameter, which may not fully capture the evolving nature of the search space as it is adapted by CMA-ES. By incorporating techniques to update the surrogate model and acquisition function based on the dynamic parameter ranges, BEACON could more accurately model the search space and make informed decisions during the falsification process. This integration has the potential to significantly improve the efficiency of the framework in discovering counterexamples. Another promising avenue is to explore the application of BEACON to a wider range of complex scenarios and safety specifications. Conducting extensive experiments with diverse and challenging falsification tasks will provide valuable insights into the scalability, robustness, and generalizability of the framework. By considering a broader spectrum of specifications and system complexities, we can assess the performance of BEACON in real-world settings and identify areas for further improvement.

REFERENCES

[1] S. Mitra, *Verifying Cyber-Physical Systems: A Path to Safe Autonomy*. Cambridge, MA, USA: MIT Press, 2021.
[2] A. P. L. Ferreira, "Model checking," in *Proc. Workshop-School Theor. Comput. Sci.*, Aug. 2011, pp. 9–14.
[3] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Progress on the state explosion problem in model checking," in *Informatics: 10 Years Back, 10 Years Ahead*. Springer, 2001, pp. 176–194.
[4] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Falsification of LTL safety properties in hybrid systems," *Int. J. Softw. Tools Technol. Transf.*, vol. 15, no. 4, pp. 305–320, Aug. 2013.
[5] J. Kapinski, J. Deshmukh, X. Jin, H. Ito, and K. Butts, "Simulation-guided approaches for verification of automotive powertrain control systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 4086–4095.
[6] A. Baheri, "Exploring the role of simulator fidelity in the safety validation of learning-enabled autonomous systems," *AI Mag.*, vol. 44, no. 4, pp. 453–459, Dec. 2023.
[7] A. Baheri, "Safety validation of learning-based autonomous systems: A multi-fidelity approach," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 13, p. 15432.
[8] G. E. Fainekos, S. Sankaranarayanan, K. Ueda, and H. Yazarel, "Verification of automotive control applications using S-TaLiRo," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 3567–3572.
[9] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-TaLiRo: A tool for temporal logic falsification for hybrid systems," in *Proc. Int. Conf. Tools Algorithms Construct. Anal. Syst.* Cham, Switzerland: Springer, 2011, pp. 254–257.
[10] A. Donzé, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in *Proc. 22nd Int. Conf. Comput. Aided Ver.*, 2010, pp. 167–170.
[11] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok, "C2E2: A verification tool for stateflow models," in *Proc. Int. Conf. Tools Algorithms Construct. Anal. Syst.*, London, U.K. Cham, Switzerland: Springer, Apr. 2015, pp. 68–82.
[12] B. Qi, C. Fan, M. Jiang, and S. Mitra, "DryVR 2.0: A tool for verification and controller synthesis of black-box cyber-physical systems," in *Proc. 21st Int. Conf. Hybrid Systems: Comput. Control (Part CPS Week)*, Apr. 2018, pp. 269–270.
[13] Z. Zhang, G. Ernst, S. Sedwards, P. Arcaini, and I. Hasuo, "Two-layered falsification of hybrid systems guided by Monte Carlo tree search," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2894–2905, Nov. 2018.
[14] Z. Ramezani, K. Claessen, N. Smallbone, M. Fabian, and K. Åkesson, "Testing cyber–physical systems using a line-search falsification method," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 8, pp. 2393–2406, Aug. 2022.

[15] Z. Zhang, D. Lyu, P. Arcaini, L. Ma, I. Hasuo, and J. Zhao, "Effective hybrid system falsification using Monte Carlo tree search guided by QB-robustness," in *Proc. 33rd Int. Conf. (CAV)*. Berlin, Germany: Springer-Verlag, Jul. 2021, pp. 595–618.

[16] M. Hekmatnejad, B. Hoxha, and G. Fainekos, "Search-based test-CASe generation by monitoring responsibility safety rules," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.

[17] Z. Zhang, P. Arcaini, and I. Hasuo, "Hybrid system falsification under (In)equality constraints via search space transformation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3674–3685, Nov. 2020.

[18] J. Deshmukh, X. Jin, J. Kapinski, and O. Maler, "Stochastic local search for falsification of hybrid systems," in *Proc. Int. Symp. Autom. Technol. Verification Anal.*, 2015, pp. 500–517.

[19] G. Ernst, S. Sedwards, Z. Zhang, and I. Hasuo, "Falsification of hybrid systems using adaptive probabilistic search," *ACM Trans. Modeling Comput. Simulation*, vol. 31, no. 3, pp. 1–22, Jul. 2021.

[20] Z. Ramezani, *On Optimization-Based Falsification of Cyber-Physical Systems*. Gothenburg, Sweden: Chalmers Tekniska Hogskola (Sweden), 2022.

[21] Z. Ramezani, J. L. Eddeland, K. Claessen, M. Fabian, and K. Åkesson, "Multiple objective functions for falsification of cyber-physical systems," *IFAC-PapersOnLine*, vol. 53, no. 4, pp. 417–422, 2020.

[22] L. Mathesen, G. Pedrielli, and G. Fainekos, "Efficient optimization-based falsification of cyber-physical systems with multiple conjunctive requirements," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 732–737.

[23] J. Deshmukh, M. Horvat, X. Jin, R. Majumdar, and V. S. Prabhu, "Testing cyber-physical systems through Bayesian optimization," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 5s, pp. 1–18, Oct. 2017.

[24] H. Abbas, M. O'Kelly, A. Rodionova, and R. Mangharam, "Safe at any speed: A simulation-based test harness for autonomous vehicles," in *Proc. Int. Workshop Design, Modeling, Eval. Cyber Phys. Syst.*, Seoul, South Korea. Cham, Switzerland: Springer, Oct. 2017, pp. 94–106.

[25] A. Aerts, B. T. Minh, M. R. Mousavi, and M. A. Reniers, "Temporal logic falsification of cyber-physical systems: An input-signal-space optimization approach," in *Proc. IEEE Int. Conf. Softw. Test., Verification Validation Workshops (ICSTW)*, Apr. 2018, pp. 214–223.

[26] Y. S. R. Annapureddy and G. E. Fainekos, "Ant colonies for temporal logic falsification of hybrid systems," in *Proc. IECON 36th Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2010, pp. 91–96.

[27] K. Kato, F. Ishikawa, and S. Honiden, "Falsification of cyber-physical systems with reinforcement learning," in *Proc. IEEE Workshop Monitor. Test. Cyber-Phys. Syst. (MT-CPS)*, Apr. 2018, pp. 456–465.

[28] Z. Zhang, I. Hasuo, and P. Arcaini, "Multi-armed bandits for Boolean connectives in hybrid system falsification," in *Proc. Int. Conf. Comput. Aided Verification*, New York, NY, USA. Cham, Switzerland: Springer, Jul. 2019, pp. 401–420.

[29] X. Qin, N. Aréchiga, A. Best, and J. Deshmukh, "Automatic testing with reusable adversarial agents," 2019, *arXiv:1910.13645*.

[30] J. J. Beard and A. Baheri, "Black-box safety validation of autonomous systems: A multi-fidelity reinforcement learning approach," 2022, *arXiv:2203.03451*.

[31] M. Al-Nuaimi, S. Wibowo, H. Qu, J. Aitken, and S. Veres, "Hybrid verification technique for decision-making of self-driving vehicles," *J. Sensor Actuator Netw.*, vol. 10, no. 3, p. 42, Jun. 2021.

[32] C. B. Burlò, A. Francalanza, and A. Scalas, "Towards a hybrid verification methodology for communication protocols (short paper)," in *Proc. Int. Conf. Formal Techn. Distrib. Objects, Compon., Syst.* Cham, Switzerland: Springer, 2020, pp. 227–235.

[33] S. Hazelhurst, G. Kamhi, O. Weissberg, and L. Fix, "A hybrid verification approach: Getting deep into the design," in *Proc. Design Autom. Conf.*, Jun. 2002, pp. 111–116.

[34] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "SpaceEx: Scalable verification of hybrid systems," in *Proc. Int. Conf. Comput. Aided Verification*, Jul. 2011, pp. 379–395.

[35] T. Dreossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components," *J. Automated Reasoning*, vol. 63, no. 4, pp. 1031–1053, Dec. 2019.

[36] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Sep. 2018, pp. 1–7.

[37] E. Bartocci, R. Bloem, B. Maderbacher, N. Manjunath, and D. Ničković, "Adaptive testing for specification coverage in CPS models," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 229–234, 2021.

[38] O. Maler and D. Ničković, "Monitoring properties of analog and mixed-signal circuits," *Int. J. Softw. Tools Technol. Transf.*, vol. 15, no. 3, pp. 247–268, Jun. 2013.

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

[40] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019.

[41] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics," *Mach. Learn.*, vol. 112, no. 10, pp. 3713–3747, Oct. 2023.

[42] A. Baheri, S. Bin-Karim, A. Bafandeh, and C. Vermillion, "Real-time control using Bayesian optimization: A case study in airborne wind energy systems," *Control Eng. Pract.*, vol. 69, pp. 131–140, Dec. 2017.

[43] A. Baheri, P. Ramaprabhu, and C. Vermillion, "Iterative in-situ 3D layout optimization of a reconfigurable ocean current turbine array using Bayesian optimization," in *Proc. ASME Dynamic Syst. Control Conf.*, 2017, Art. no. V003T40A002.

[44] A. Baheri and C. Vermillion, "Waypoint optimization using Bayesian optimization: A case study in airborne wind energy systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 5102–5017.

[45] S. Ghosh, F. Berkenkamp, G. Ranade, S. Qadeer, and A. Kapoor, "Verifying controllers against adversarial examples with Bayesian optimization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7306–7313.

[46] Z. Shahrooei, M. J. Kochenderfer, and A. Baheri, "Falsification of learning-based controllers through multi-fidelity Bayesian optimization," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2023, pp. 1–6.

[47] N. Hansen, "The CMA evolution strategy: A tutorial," 2016, *arXiv:1604.00772*.

[48] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[49] MathWorks. (2020). *Design NARMA-L2 Neural Controller in Simulink*. Accessed: May 22, 2023. [Online]. Available: https://au.mathworks.com/help/deeplearning/ug/design-narma-l2-neural-controller-in-simulink.html

[50] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in F-16 ground collision avoidance and other automated maneuvers," in *Proc. EPiC Ser. Comput.*, 2018, pp. 208–217.

[51] 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*, 2014, pp. 253–262.

JOSHUA YANCOSEK received the B.S. degree in applied physics from the West Virginia Wesleyan College, Buckhannon, in 2021. He is currently pursuing the M.S. degree in mechanical engineering with West Virginia University, Morgantown. Since 2021, he has been a Graduate Research Assistant with West Virginia University. His research interests include control systems, robotics, artificial intelligence, and the advancement of safety validation methods.

ALI BAHERI (Member, IEEE) received the Ph.D. degree from the University of North Carolina at Charlotte, in 2018. He is currently an Assistant Professor with the Department of Mechanical Engineering, Rochester Institute of Technology. His laboratory focuses on research at the intersection of autonomy, controls, and machine learning. Before joining RIT, he was a Visiting Scholar with Stanford University. Prior to that, he was an Assistant Professor (in the research track) with West Virginia University and a Postdoctoral Research Fellow with the University of Michigan, Ann Arbor.