

UC Davis

UC Davis Previously Published Works

Title

Formal interpretation of cyber-physical system performance with temporal logic

Permalink

<https://escholarship.org/uc/item/4wc4n1pw>

Journal

Cyber-Physical Systems, 4(3)

ISSN

2333-5777

Authors

Chen, Gang
Sabato, Zachary
Kong, Zhaodan

Publication Date

2018-07-03

DOI

10.1080/23335777.2018.1510857

Peer reviewed



Formal interpretation of cyber-physical system performance with temporal logic

Gang Chen, Zachary Sabato & Zhaodan Kong

To cite this article: Gang Chen, Zachary Sabato & Zhaodan Kong (2018) Formal interpretation of cyber-physical system performance with temporal logic, Cyber-Physical Systems, 4:3, 175-203, DOI: [10.1080/23335777.2018.1510857](https://doi.org/10.1080/23335777.2018.1510857)

To link to this article: <https://doi.org/10.1080/23335777.2018.1510857>



Published online: 10 Oct 2018.



Submit your article to this journal [↗](#)



Article views: 4




View Crossmark data [↗](#)

ARTICLE



Formal interpretation of cyber-physical system performance with temporal logic

Gang Chen^a, Zachary Sabato^{a,b} and Zhaodan Kong ^a

^aDepartment of Mechanical and Aerospace Engineering, University of California, Davis, California, USA; ^bHyundai Center of Excellence in Vehicle Dynamic Systems & Control at UC Davis, Davis, California, USA

ABSTRACT

The inherent and increasing complexity of many cyber-physical systems (CPSs) makes it challenging for human users or designers to comprehend and interpret their performance. This issue, without proper attention paid, may lead to unwanted and even catastrophic consequences, particularly with safety-critical CPSs. This paper presents a new methodology of enabling (i) a human to interrogate a CPS by inquiring with questions written in formal logic and (ii) the CPS to interpret its performance precisely in the context of the inquiry. This formal interpretation problem is first formulated as temporal logic inference problem, which, aided by the concept of robustness degree, can be converted into an optimisation problem with probably approximately correct solutions. A new Gaussian-process-based active learning algorithm is then proposed to address the potential computational budget issue arising from solving the optimisation problem. Both theoretical and empirical analyses are carried out to demonstrate the performance of the proposed algorithm. Finally, a detailed case study on automotive mechatronic design is provided to showcase the proposed formal interpretation methodology.

ARTICLE HISTORY

Received 7 August 2018
Accepted 8 August 2018

KEYWORDS

Cyber-physical systems; signal temporal logic; active learning; human-computer interactions; automotive mechatronics

1. Introduction

In recent years, the cyber-physical system (CPS) paradigm has found its way into many safety-critical domains, e.g. transportation, power generation, medical sectors, and military applications. Equipping CPSs with advanced sensors, autonomy, and/or distributed computation offers stakeholders a significant opportunity to reduce cost, extend operational range, and enhance system capabilities. However, the increasing complexity of many of these systems, e.g. the potentially nonlinear and sometimes unexpected interactions between their cyber and physical components, makes it hard for users or designers to comprehend and interpret the performance of these systems. Misinterpretations, without timely mitigation, may lead to a wide spectrum

of consequences, from a minor inconvenience to a major catastrophe. This motivates us to develop a tool allowing human users (or designers) to interrogate a CPS (e.g. an autonomous vehicle or an industrial robot) via inquiries.

Specifically, given an inquiry from a user (or designer), the CPS must interpret its own performance in the context of the inquiry in a rather precise (and hopefully insightful) manner.

In this paper, we are interested in inquiries expressed in a formal logic called signal temporal logic (STL) [1–3]. STL is a ‘rich’ specification language and has been widely used in specifying many high-assurance CPSs [4,5]. The parsing of STL formulas is quite easy to learn. For instance, an STL interpretation of an autonomous vehicle’s acceleration performance may be written as $F_{[0,\tau]}(v > \pi)$, which reads ‘between times 0 and τ , the speed v is eventually greater than π ’, where F is the temporal operator for ‘eventually’. Compared with natural language interpretations, which are vague and largely rely on human experts to provide (an expensive, time-consuming, and potentially error-prone task), STL interpretations are quantitative, precise, and can be algorithmically extracted (described shortly). The STL interpretation methodology presented in this paper offers human users (or designers) an effective means to actively interact with CPSs, interrogate them, and explore the interwoven threads of interacting cyber and physical components in the context of system performance. Accordingly, our methodology can potentially facilitate the human users’ (or designers’) formation of meta-knowledge of the complex CPSs they are interacting with (or designing).

We formulate the formal interpretation of CPS performance as a logic inference problem: Given a system S , e.g. a Stateflow/Simulink model of an automobile steering system, and an user inquiry, codified as a parametric STL template φ_θ with a set of unknown parameters θ , find a θ^* such that the interpretation φ_{θ^*} is satisfied by the system S . We solve this problem by first utilising the concept of *robustness degree* [2,6]. Instead of merely proving a yes or no answer about whether a system S exhibits a temporal logic property φ_{θ^*} , robustness degree quantifies how strongly the system S satisfies the property φ_{θ^*} with a real number. With the help of robustness degree, the above problem can be converted into an optimisation problem where θ^* optimises the expected robustness degree of the system S against φ_{θ^*} . Then we leverage active learning [7,8] to mitigate the need for a large number iterations during optimisation, improving efficiency and doing more with a constrained computational budget.

Related work. Our work is closely related to literature in the fields of requirement mining and active learning. Requirement mining was first proposed in the context of extracting requirements from software execution traces for the purposes of, e.g. software maintenance and legacy code understanding [9,10]. In that context, traces are of discrete, finite state; the mined requirements are generally written in regular languages or linear temporal logic (LTL). In recent years, the idea of requirement mining (also called

specification mining [11] or specification inference [12]) has been extended to the mining of requirements for CPSs [4,8,11–15], given the critical role requirements play in the formal specification, verification, validation, and controller synthesis of CPSs. In this work, the data are in the form of trajectories, which can have hybrid (i.e. continuous and discrete) states; the trajectories can be either generated online by a system or collected offline; the mined requirements are generally written in STL or metric temporal logic (MTL). One recurring theme in these latest developments is to leverage the power of optimisation and machine learning. For instance, in [11], the requirement mining problem was formulated as a parameter optimisation problem and counter-examples were used to guide the search for the optimal parameter(s); in [14,15], the requirement mining problem was formulated as a combined structure and parameter optimisation problem, with lattice search and simulated annealing used to find the satisfactory formula structure as well as the optimal parameters. We would argue that requirement mining and formal interpretation differ in not only their means but also their ends. Requirement mining stops once a requirement has been found and can be conducted offline, while formal interpretation needs to be conducted in an iterative and online fashion. That is, a user cannot wait hours in order to get an interpretation from a CPS; moreover, once an interpretation has been provided, it is quite likely that further inquiries from the user will follow. The implication is that computational efficiency is a pressing issue for formal interpretation.

One way to address this issue is by exploring the advantages offered by active learning algorithms. The main idea behind active learning is to accelerate learning by actively selecting potentially ‘informative’ samples, rather than random sampling from a pre-defined distribution [7,8]. Active learning was originally developed to reduce the number of data points needed for labelling, where a human serves as the oracle to provide the labels [16]. But recently it has been used to facilitate the verification/falsification of CPSs [8,13,17]. In this context, a system model serves as the oracle, which can generate trajectories. Existing active learning algorithms are then used to sample the search space (which can be of infinite dimension) aided by certain ‘informativeness’ metrics, in addition to knowledge gained from the already-sampled data. This practice helps to focus ongoing searches in promising ranges, thus eliminating unnecessary samples and producing increasingly better results to verify/falsify the system of interest. One benefit of using active learning is that it promises theoretical guarantees (under certain assumptions). In this paper, we will develop our own active learning algorithm while taking advantage of this benefit.

Contributions. The *main* contributions of this paper are threefold. *First*, we develop a formal interpretation methodology for CPS performance by formulating and solving a temporal logic inference problem, which is solved in the sense of obtaining solutions that are probably approximately correct (PAC), a

concept that offers rich connections to machine learning theory. We believe such a formulation paves the way for future developments in CPS interpretation (or system explanation). *Second*, instead of using existing learning algorithms, this paper develops a new active learning algorithm called Gaussian process adaptive confidence bound (GP-ACB). The performance of GP-ACB, as compared with that of other Gaussian-process-based active learning algorithms, is demonstrated both theoretically and empirically. *Third*, we implement and illustrate our interpretation methodology with a case study on automotive mechatronic design, which we believe may greatly help practitioners to understand our methodology and idea, and to gain their own insights.

An earlier version of this paper [18] appeared in the 2016 IEEE 55th Conference on Decision and Control (CDC). This paper significantly extends that paper by (i) solving a formal interpretation problem rather than a requirement mining problem, (ii) removing the assumption that the hyper-parameters of the underlying Gaussian process (GP) are known a priori, (iii) providing detailed proofs of theoretical results, and (iv) offering a detailed case study that is closer to real practices.

This rest of the paper is organised as follows. [Section 2](#) provides the necessary background on STL and GPs. [Section 3](#) defines the formal interpretation problem. [Section 4](#) discusses our GP-ACB algorithm. [Section 5](#) shows how to solve the formal interpretation problem with GP-ACB. [Section 6](#) provides two case studies to demonstrate our methodology: an academic example with the Rastrigin function, and an mechatronic design example. Finally, [Section 7](#) concludes the paper and mentions some future directions for continuing work.

2. Preliminaries

2.1. Signal temporal logic

Given two sets A and B , $\mathcal{F}(A, B)$ denotes the set of all functions from A to B . Given a time domain $\mathbb{R}^+ := [0, \infty)$, a *continuous-time, continuous-valued signal* is a function $s \in \mathcal{F}(\mathbb{R}^+, \mathbb{R}^n)$. This paper uses $s(t)$ to denote the value of signal s at time t .

STL is a temporal logic defined over signals [1]. The syntax of STL used in this paper is defined as

$$\varphi := f(s) \sim d \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid F_{[a,b]} \varphi \mid G_{[a,b]} \varphi, \quad (1)$$

where a and b are non-negative finite real numbers, $f(s) \sim d$ is a predicate where s is a signal, $f \in \mathcal{F}(\mathbb{R}^n, \mathbb{R})$ is a function, $\sim \in \{<, \geq\}$, and $d \in \mathbb{R}$ is a constant. The Boolean operators \neg , \wedge , and \vee are negation ('not'), conjunction

(‘and’), and disjunction (‘or’), respectively. The temporal operators F and G stand for ‘Finally (eventually)’ and ‘Globally (always),’ respectively.

A robustness degree [2,6] function (which serves as the semantics of STL) $r : \Psi \times \mathcal{F}(\mathbb{R}^+, \mathbb{R}^n) \rightarrow \mathbb{R}$ maps an STL formula $\varphi \in \Psi$ and a signal $s \in \mathcal{F}(\mathbb{R}^+, \mathbb{R}^n)$ to a real value, called the *robustness degree* (or the *degree of satisfaction*) of s against φ :

$$\begin{aligned} r(s, (f(s) < d), t) &= d - f(s(t)) \\ r(s, (f(s) \geq d), t) &= f(s(t)) - d \\ r(s, \varphi_1 \wedge \varphi_2, t) &= \min(r(s, \varphi_1, t), r(s, \varphi_2, t)) \\ r(s, \varphi_1 \vee \varphi_2, t) &= \max(r(s, \varphi_1, t), r(s, \varphi_2, t)) \\ r(s, G_{[a,b]}\varphi, t) &= \min_{t' \in [t+a, t+b]} r(s, \varphi, t') \\ r(s, F_{[a,b]}\varphi, t) &= \max_{t' \in [t+a, t+b]} r(s, \varphi, t'). \end{aligned}$$

A positive $r(s, \varphi, 0)$ indicates that the signal s satisfies STL formula φ at time $t = 0$; a negative $r(s, \varphi, 0)$ indicates that s violates φ at $t = 0$. If $r(s, \varphi, 0)$ is large and positive, then s would have to undergo a large deviation in order to violate φ .

Parametric signal temporal logic (PSTL) is an extension of STL where the bound d and the endpoints of the time intervals $[a, b]$ are parameters instead of constants [19]. This paper calls variables used to parameterise d as *scale* parameters π and those parameterising a and b as *time* parameters τ . A full parameterisation is given as $\theta := [\pi, \tau]$. The syntax and semantics of PSTL are the same as those of STL. A *valuation* $\bar{\theta}$ is a mapping that assigns real (numerical) values to the parameters θ appearing in a PSTL formula. A valuation $\bar{\theta}$ of a PSTL formula φ_θ parameterised by θ induces an STL formula $\varphi_{\bar{\theta}(\theta)}$. For example, if $\varphi_\theta = F_{[\tau_1, \tau_2]}(x < \pi_1)$ with $\theta = [\pi_1, \tau_1, \tau_2]$ and $\bar{\theta}(\theta) = [0, 0, 3]$, then $\varphi_{\bar{\theta}(\theta)} = F_{[0, 3]}(x < 0)$. In the following, we will use $\varphi_{\bar{\theta}}$ to denote $\varphi_{\bar{\theta}(\theta)}$ for simplicity. We will also use φ_{θ_i} (or φ_{θ^*}) to denote the STL formula resulting from valuating the parameters θ of the PSTL formula φ_θ at $\theta_i(\theta)$ (or $\theta^*(\theta)$).

2.2. Gaussian processes

A *GP* is defined in a continuous domain as a collection of random variables, any finite linear combination of which has a joint Gaussian distribution [20]. Any GP over $\mathbf{d} \in D$ can be defined completely with its mean function $\boldsymbol{\mu}_{\mathbf{d}}(\mathbf{d})$ and its covariance function $\Sigma_{\mathbf{d}}(d, d')$ as follows:

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{d}}(\mathbf{d}) &= E[f(\mathbf{d})], \\ \Sigma_{\mathbf{d}}(\mathbf{d}, \mathbf{d}') &= E[(f(\mathbf{d}) - \boldsymbol{\mu}(\mathbf{d}))(f(\mathbf{d}') - \boldsymbol{\mu}(\mathbf{d}'))]. \end{aligned} \quad (2)$$

A flat (or even zero) mean function $\boldsymbol{\mu}_{\mathbf{d}}(\mathbf{d})$ is chosen in the majority of cases in the literature. Such a choice does not cause many issues since the mean of the

posterior process is not confined to zero. There is a large set of available kernels $k_{\mathbf{d}}(\mathbf{d}, \mathbf{d}')$ to construct $\mu_{\mathbf{d}}(\mathbf{d})$ and $\Sigma_{\mathbf{d}}(\mathbf{d}, \mathbf{d}')$ (see e.g. Equation (5)) but not all kernel functions lead to universal approximation [21].

3. System definition and problem statement

3.1. System definition

We make the following assumptions regarding the class of CPSs studied in this paper.

Assumption 1. The dynamics of the CPSs are deterministic, but subject to stochastic initial condition perturbations, i.e. their dynamics can be described by [22,23]

$$\mathbf{x}(t) = f(\mathbf{x}(t), \zeta), \quad (3)$$

Assumption 1. where $\mathbf{x}(t)$ is the state vector and ζ is a perturbation vector, a zero mean Gaussian with an unknown covariance Σ_{ζ} , i.e. $\zeta \sim \mathcal{N}(\mathbf{0}, \Sigma_{\zeta})$. Moreover, there is no measurement noise.

Remark 1. Based on this assumption, a CPS S maps an initial condition $\mathbf{x}_0 \in X_0 \subset \mathbb{R}^{n_x}$ and a perturbation ζ (due to e.g. uncontrolled and unknown environmental conditions) to a discrete-time output signal $\mathbf{y}(t|\mathbf{x}_0, \zeta) \in \mathcal{F}([0, T], Y)$ with $Y \subset \mathbb{R}^{n_y}$ and T as the finite maximal simulation time. Then $r(\mathbf{y}(t|\mathbf{x}, \zeta), \varphi_{\bar{\theta}}, 0)$ stands for the robustness degree of an output signal $\mathbf{y}(t|\mathbf{x}, \zeta)$ starting from \mathbf{x} at time $t = 0$ against an STL property $\varphi_{\bar{\theta}}$. In the following, unless specified otherwise, we will use $r(\mathbf{x}, \varphi_{\bar{\theta}})$ to denote $r(\mathbf{y}(t|\mathbf{x}, \zeta), \varphi_{\bar{\theta}}, 0)$.

Assumption 2. Both X_0 and Y can be represented as the Cartesian product of intervals $[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$, where $a_i, b_i \in \mathbb{R}$.

3.2. Problem statement

The formal CPS performance interpretation solved in this paper can be defined as follows:

Problem 1. (Formal CPS Performance Interpretation) *Given a CPS S with an initial set $X_0 \subset \mathbb{R}^{n_x}$, two user-specified bounds $\delta \in (0, 1), \iota \in \mathbb{R}^+$, and a user-*

specified PSTL φ_θ with parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, where Θ is the set of all feasible valuations, find a valuation θ^* such that

$$\mathbb{P}(0 < \min_{\mathbf{x} \in X_0} r(\mathbf{x}, \varphi_{\theta^*}) < \iota) > 1 - \delta, \quad (4)$$

where $\mathbb{P}(0 < \min_{\mathbf{x} \in X_0} r(\mathbf{x}, \varphi_{\theta^*}) < \iota)$ is the probability that $0 < \min_{\mathbf{x} \in X_0} r(\mathbf{x}, \varphi_{\theta^*}) < \iota$.

Remark 2. The condition $\min_{\mathbf{x} \in X_0} r(\mathbf{x}, \varphi_{\theta^*}) > 0$ says that the output signal $\mathbf{y}(t|\mathbf{x}, \zeta)$ starting from any initial state $\mathbf{x} \in X_0$ has a positive robustness degree against the STL formula φ_{θ^*} . According to the concept of the robustness degree, this implies that all output signals starting from X_0 satisfy φ_{θ^*} , which will be called the formal interpretation of the CPS S 's performance in this paper. The condition $\min_{\mathbf{x} \in X_0} r(\mathbf{x}, \varphi_{\theta^*}) < \iota$ dictates that the found interpretation should only be satisfied by the CPS S barely. Moreover, notice that there are infinitely many such interpretations (formulas), which together form a Pareto-front-like surface in Θ (see [11] and Section 5.2 for details).

Problem 1 suffers from the curse of dimensionality with a search space of dimension $n_x \times n_\theta$. To mitigate this complexity, this paper solves Problem 1 by solving two separate sub-problems iteratively: one with a search space of dimension n_x and the other with a search space of dimension n_θ . Before elaborating on these two sub-problems, let us first introduce an assumption regarding the property of the robustness degree function $r(\mathbf{x}, \varphi_\theta)$.

Assumption 3. For a CPS S , its marginal robustness degree functions $r(\cdot, \varphi_{\bar{\theta}}) \in \mathcal{F}(X_0, \mathbb{R})$ and $r(\mathbf{x}, \varphi) \in \mathcal{F}(\Theta, \mathbb{R})$ can both be approximated by GPs. $r(\cdot, \varphi_{\bar{\theta}}) \in \mathcal{F}(X_0, \mathbb{R})$ maps an initial state $x \in X_0$ to a robustness degree taken with respect to an STL formula $\varphi_{\bar{\theta}}$ (i.e. a PSTL formula φ_θ with a fixed valuation $\bar{\theta}$), while $r(\mathbf{x}, \varphi) \in \mathcal{F}(\Theta, \mathbb{R})$ maps a valuation of $\theta \in \Theta$ to a robustness degree taken with respect to a fixed initial state \mathbf{x} . Specifically, given a set of initial-state-robustness-degree pairs $\mathcal{L}_{\mathbf{x}, r} = \{(\mathbf{x}_i, \hat{r}(\mathbf{x}_i, \varphi_{\bar{\theta}}))\}_{i=1, \dots, N}$ for a fixed valuation $\bar{\theta}$, $r(\cdot, \varphi_{\bar{\theta}})$ can be approximated by a GP $\hat{r}_{\bar{\theta}}(\cdot)$ with its mean function and covariance function defined as follows [20]:

$$\begin{aligned} \mu_{\mathbf{x}}(\mathbf{x}) &= k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_{\mathcal{L}}) k_{\mathbf{x}}(\mathbf{x}_{\mathcal{L}}, \mathbf{x}_{\mathcal{L}})^{-1} \hat{r}_{\mathcal{L}}^{\bar{\theta}} \\ \Sigma_{\mathbf{x}}(\mathbf{x}, \mathbf{x}') &= k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}') - k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}_{\mathcal{L}}) k_{\mathbf{x}}(\mathbf{x}_{\mathcal{L}}, \mathbf{x}_{\mathcal{L}})^{-1} k_{\mathbf{x}}(\mathbf{x}_{\mathcal{L}}, \mathbf{x}'), \end{aligned} \quad (5)$$

where $\mathbf{x}_{\mathcal{L}} = [x_1, \dots, x_N]^T$, $\hat{r}_{\mathcal{L}}^{\bar{\theta}} = [\hat{r}(x_1, \varphi_{\bar{\theta}}), \dots, \hat{r}(x_N, \varphi_{\bar{\theta}})]^T$, and $k_{\mathbf{x}}(\cdot, \cdot)$ is a kernel function, which is chosen to be the automatic relevance determination (ARD) kernel in this paper

$$k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}') = \sigma_{\mathbf{x}, \text{ARD}}^2 \exp\left(-\frac{1}{2} \sum_{j=1}^{n_x} w_{\mathbf{x}, j} (\mathbf{x}_j - \mathbf{x}'_j)^2\right). \quad (6)$$

The ARD kernel is characterised by a set of hyper-parameters $\sigma_{\mathbf{x}} = \left\{ \sigma_{\mathbf{x},ARD}^2, w_{\mathbf{x},1}, w_{\mathbf{x},2}, \dots, w_{\mathbf{x},n_x} \right\}$, where $w_{\mathbf{x},j}, j = 1, \dots, n_x$ weights the importance of the j -th dimension of X_0 . The reason that the notation $\hat{r}(\mathbf{x}_i, \varphi_{\bar{\theta}})$ instead of $r(\mathbf{x}_i, \varphi_{\bar{\theta}})$ is used will be detailed in Section 5.1. Before that, $\hat{\cdot}$ can be read as indicating ‘estimated’ or ‘observed’ robustness degrees (due to, for instance, the perturbation ζ or the GP approximation) rather than the ‘real’ ones. Similarly, given a set of valuation-robustness-degree pairs $\mathfrak{L}_{\theta,r} = \{(\theta_i, \hat{r}(\mathbf{x}, \varphi_{\theta_i}))\}_{i=1,\dots,N}$ (or $\mathfrak{L}_{\theta,r} = \{\theta_{\mathfrak{L}}, \hat{r}_{\mathfrak{L}}^{\mathbf{x}}\}$ with $\theta_{\mathfrak{L}} = [\theta_1, \dots, \theta_N]^T, r_{\mathfrak{L}}^{\mathbf{x}} = [\hat{r}(\mathbf{x}, \varphi_{\theta_1}), \dots, \hat{r}(\mathbf{x}, \varphi_{\theta_N})]^T$) for a fixed initial state \mathbf{x} , $r(\mathbf{x}, \varphi)$ can be approximated by another GP $\hat{r}_{\mathbf{x}}(\theta)$ with its kernel function, mean function, covariance function, and hyper-parameters denoted as $k_{\theta}(\cdot, \cdot), \mu_{\theta}(\cdot), \Sigma_{\theta}(\cdot, \cdot)$, and $\sigma_{\theta} = \left\{ \sigma_{\theta,ARD}^2, w_{\theta,1}, w_{\theta,2}, \dots, w_{\theta,n_{\theta}} \right\}$, respectively.

Remark 3. Since GPs is a universal functional approximator [24,25], the above assumption is a quite reasonable one. In comparison to isotropic squared exponential kernels with equal weight for each dimension, our choice of ARD kernels allows different weights for different dimensions, and is thus more flexible.

The two sub-problems are formulated as follows:

Problem 2. (Marginal Robustness Degree Function Regression) *Given a CPS S with an initial set $X_0 \subset \mathbb{R}^{n_x}$, two user-specified bounds $\rho \in (0, 1), \iota \in \mathbb{R}^+$, an STL formula $\varphi_{\bar{\theta}}$ with a fixed valuation $\bar{\theta}$, find a set of initial-state-robustness-degree pairs $\mathfrak{L}_{x,r} = \{x_{\mathfrak{L}}, \hat{r}_{\mathfrak{L}}^{\bar{\theta}}\}$ and a set of hyper-parameters $\tilde{\sigma}_x^*$ such that*

$$\mathbb{P}(|\hat{r}_{\bar{\theta}}(x^*) - r(x^*, \varphi_{\bar{\theta}})| < \iota | \mathfrak{L}_{x,r}, \tilde{\sigma}_x^*) > 1 - \rho \quad (7)$$

where

$$\begin{aligned} \tilde{\sigma}_x^* &= \operatorname{argmax}_{\tilde{\sigma}_x} \log \mathbb{P}(\hat{r}_{\mathfrak{L}}^{\bar{\theta}} | x_{\mathfrak{L}}, \sigma_x), \\ x^* &= \operatorname{argmin}_{x \in X_0} \hat{r}_{\bar{\theta}}(x), \end{aligned} \quad (8)$$

$r(\cdot, \varphi_{\bar{\theta}})$ is the real marginal robustness function for a fixed $\varphi_{\bar{\theta}}$, $\hat{r}_{\bar{\theta}}(\cdot)$ is its GP approximation (see Assumption 3), and $\mathbb{P}(\hat{r}_{\mathfrak{L}}^{\bar{\theta}} | x_{\mathfrak{L}}, \tilde{\sigma}_x)$ is the marginal likelihood taken with respect to hyper-parameters $\tilde{\sigma}_x$.

Remark 4. $\tilde{\sigma}_x = \sigma_x \cup \sigma_v^x$, where σ_x is a set of hyper-parameters defined in Assumption 3 and σ_v^x is another hyper-parameter. Details regarding σ_v^x as well as the marginal likelihood $\mathbb{P}(\hat{r}_{\mathfrak{L}}^{\bar{\theta}} | \mathbf{x}_{\mathfrak{L}}, \tilde{\sigma}_x)$ will be provided in Section 5.1. A by-product of generating $\mathfrak{L}_{x,r}$ is a set of sampled output signals $y_{\mathfrak{L}} = \{y_i(t | \mathbf{x}_i, \zeta)\}_{i=1,\dots,N}$. In practice, these signals can be obtained from a real system or a through simulation of a high-fidelity Stateflow/Simulink model.

Problem 3. (Parameter Estimation) *Given a CPS S with an initial state x (or equivalently an output signal $y(t|x, \zeta)$), three user-specified bounds $\rho \in (0, 1)$, $\iota \in \mathbb{R}^+$ and $\varepsilon \in \mathbb{R}^+$, and a user-specified PSTL formula φ_θ with parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, find a set of valuation-robustness-degree pairs $\mathfrak{L}_{\theta,r} = \{\theta_\mathfrak{L}, \hat{r}_\mathfrak{L}^x\}$, a valuation θ^* , and a set of hyper-parameters $\tilde{\sigma}_\theta^*$ such that*

$$\mathbb{P}(|\hat{r}_x(\varphi_{\theta^*}) - r(x, \varphi_{\theta^*})| < \iota | \mathfrak{L}_{\theta^*,r}, \tilde{\sigma}_{\theta^*}^*) > 1 - \rho \quad (9)$$

where

$$\begin{aligned} \tilde{\sigma}_\theta^* &= \operatorname{argmax}_{\tilde{\sigma}_\theta} \log \mathbb{P}(\hat{r}_\mathfrak{L}^x | \theta_\mathfrak{L}, \tilde{\sigma}_\theta) \\ \theta^* &= \operatorname{argmax}_{\theta \in \Theta} \max(0, \varepsilon - |\hat{r}_x(\varphi_\theta)|) \end{aligned} \quad (10)$$

where $r(x, \varphi)$ is the real marginal robustness function for a fixed x , $\hat{r}_x(\varphi)$ is its GP approximation (see Assumption 3) and $\mathbb{P}(\hat{r}_\mathfrak{L}^x | \theta_\mathfrak{L}, \tilde{\sigma}_\theta)$ is the marginal likelihood taken with respect to hyper-parameters $\tilde{\sigma}_\theta = \sigma_\theta \cup \sigma_\nu^\theta$ (details regarding σ_ν^θ will be provided in Section 5.1).

Remark 5. The function $\max(0, \varepsilon - \cdot)$ in Equation (10) is a modified hinge loss function. We utilise this function to favour those interpretations i.e. STL formulas, that are only barely satisfied by the CPS S . See Remark 2.

Theorem 1. Given a CPS S with an initial set $X_0 \subset \mathbb{R}^{n_x}$, three user-specified bounds $\delta \in (0, 1)$, $\iota \in \mathbb{R}^+$, $\varepsilon \in \mathbb{R}^+$, a ρ satisfying $(1 - \rho)^2/2 \geq 1 - \delta$, and a user-specified PSTL φ_θ with parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, where Θ is the set of all feasible valuations, if there exists a set of 4-tuples $\mathfrak{L} = \{(x_i, \theta_i, y_i(t|x_i, \zeta), r(x_i, \varphi_{\theta_i}))\}_{i=1, \dots, N}$, two sets of hyper-parameters σ_x^* and σ_θ^* , and a valuation θ^* such that together they solve Problems 2 and 3 simultaneously, then θ^* is a solution to Problem 1.

Remark 6. The proof of this theorem is provided in Appendix 8.1. The implication is that solving Problem 1, which has a search space of dimension $n_x \times n_\theta$, can be converted into solving two sub-problems iteratively, Problem 2 with a search space of dimension n_x and Problem 3 with a search space of dimension n_θ .

4. Gaussian process adaptive confidence bound active learning algorithm

A close inspection of Problems 2 and 3 reveals that a crucial computational bottleneck for the formal interpretation problem is the generation of the output signal $y(t|\mathbf{x}, \zeta)$ given an initial state \mathbf{x} and a CPS S . In practices, this involves either conducting tests with a real CPS or generating signals with a

high-fidelity Stateflow/Simulink CPS model; both can be experimentally/computationally expensive. In this paper, we assume that:

Assumption 4. A Stateflow/Simulink model of the CPS S under investigation is available and can serve as an oracle i.e. given an initial state \mathbf{x} , it automatically generates an output signal $y(t|\mathbf{x}, \zeta)$. The formal CPS performance interpretation problem is restricted by a computational budget, manifesting as a constraint on the number of simulations, N_{sim} .

Many sampling based optimisation methods, such as particle swarm optimisation [26], simulated annealing [14], Nelder–Mead [11], and the stochastic gradient descent algorithm [15] may not be suitable for the formal interpretation problem then, due to the large number of simulations needed. Instead this paper develops an active learning algorithm called GP-ACB, inspired by the Gaussian process upper confidence bound (GP-UCB) approach [24], but with improved performance (to be demonstrated both theoretically and empirically in this work).

GP-ACB is used to solve both Problem 2 and Problem 3. According to Assumption 3, functions $r(\cdot, \varphi_{\bar{\theta}}) \in \mathcal{F}(X_0, \mathbb{R})$ and $r(\mathbf{x}, \varphi) \in \mathcal{F}(\Theta, \mathbb{R})$ can be approximated by GPs $\hat{r}_{\bar{\theta}}(\cdot)$ and $\hat{r}_{\mathbf{x}}(\cdot)$, respectively. Then from a computational perspective, the two problems are quite similar: Problem 2 involves finding enough samples $\mathcal{L}_{\mathbf{x},r} = \{\mathbf{x}_{\zeta}, \hat{r}_{\zeta}^{\bar{\theta}}\}$ to construct the underlying $\hat{r}_{\bar{\theta}}(\cdot)$ (exploration) and at the same time locating the \mathbf{x}^* minimising the current $\hat{r}_{\bar{\theta}}(\cdot)$ (exploitation); Problem 3 involves finding enough samples of $\mathcal{L}_{\theta,r} = \{\theta_{\zeta}, \hat{r}_{\zeta}^{\mathbf{x}}\}$ to construct the underlying $\hat{r}_{\mathbf{x}}(\cdot)$ (exploration) and at the same time locating the θ^* maximising Equation (10), which is determined by $\hat{r}_{\mathbf{x}}(\cdot)$ (exploitation). Thereby a necessary mechanism needs to be in place to formally address the trade-off between exploration and exploitation. Moreover, according to Assumption 4, the number of simulations of the CPS model should be as small as possible.¹ These needs are addressed by GP-ACB.

Specifically, GP-ACB chooses the next sample based on the following strategy:

$$\mathbf{d}_t = \underset{\mathbf{d} \in D}{\operatorname{argmax}} (\mu_{t-1}(\mathbf{d}) + \mathbf{n}_{\mu}(\mathbf{d})^{\frac{1}{2}} \beta_t^{\frac{1}{2}} \Sigma_{t-1}(\mathbf{d})), \quad (11)$$

where \mathbf{d} is \mathbf{x} for Problem 2 and θ for Problem 3 (the search space D is X_0 in the former case and Θ in the latter one); subscripts t and $t - 1$ indicate iteration steps e.g. \mathbf{d}_t is the instance that will be sampled at step t ; $\mu_{t-1}(\cdot)$ and $\Sigma_{t-1}(\cdot)$ are the underlying GP's mean and covariance functions at time step $t - 1$, respectively, which are evaluated based on all the data obtained until step $t -$

1 (see Equations (2) and (5)); the underlying GP is $\hat{r}_{\bar{\theta}}(\cdot)$ for Problem 2 and $\hat{r}_{\mathbf{x}}(\cdot)$ for Problem 3; $\eta_{\mu}(\mathbf{d})$ normalises the mean $\mu_{t-1}(\mathbf{d})$:

$$\eta_{\mu}(\mathbf{d}) = \frac{\mu_{t-1}(\mathbf{d}) - \min_{d \in D} (\mu_{t-1}(\mathbf{d}))}{\max_{d \in D} (\mu_{t-1}(\mathbf{d})) - \min_{d \in D} (\mu_{t-1}(\mathbf{d}))}; \quad (12)$$

and β_t is a function of t and independent of \mathbf{d} (see Theorem 2). Note that Equation (11) is formulated as maximisation (i.e. to maximise reward) instead of minimisation (i.e. to minimise cost) to be consistent with active learning literature.

The GP-ACB algorithm balances the classical exploitation-exploration trade-off as follows: the term $\mu_{t-1}(\mathbf{d})$ tends to pick those points that are expected to achieve high rewards (exploitation), and the term $\Sigma_{t-1}(\mathbf{d})$ tends to pick those points that are uncertain (exploration). The normalisation term $\eta_{\mu}(\mathbf{d})$ ($0 \leq \eta_{\mu}(\mathbf{d}) \leq 1$) acts as an adaptive factor favouring exploration directions associated with higher rewards. Its role is summarised in the following proposition:

Proposition 1. *Set $L_t = \max(\mu_t(d)) - \min(\mu_t(d))$, $\forall d \in D$, and let β_t be defined as in Lemma 2 (see [Appendix 8.2](#)), then*

$$1 - \eta_{\mu}(d_t)^{1/2} \leq \beta_t^{1/2} \Sigma_{t-1}(d_t) / L_t \quad \forall t \geq 1.$$

Remark 7. Proposition 1 shows that when the scaling function L_t is large, $\eta_{\mu}(\mathbf{d})$ will be close to 1, meaning the GP-ACB algorithm degrades to GP-UCB [27]. Conversely, when L_t is large, $\mu_t(\mathbf{d})$ will drive the algorithm to be greedy. The proof of this proposition is provided in [Appendix 8.2](#).

Algorithm 1: GP-ACB algorithm

Input: Search space D ; kernel function with hyperparameters $\sigma_{\mathbf{d}}$; maximal simulation time T .

Output: A set of sample-observation pairs $\{(d_i, r_i)\}_{i=1}^T$.

- 1: Set GP priors $\mu_0(\mathbf{d}) = 0$ and $\Sigma_0(\mathbf{d}) = 0$;
 - 2: **for** $t = 1$ to T
 - 3: Update $\mu_{t-1}(\mathbf{d})$ and $\Sigma_{t-1}(\mathbf{d})$ with Equation (2);
 - 4: Calculate $\eta_{\mu}(\mathbf{d})$ with Equation (12);
 - 5: Calculate \mathbf{d}_t with Equation (11);
 - 6: Obtain r_t that corresponds to \mathbf{d}_t .
 - 7: **end for**
-

Algorithm 1 describes the GP-ACB in pseudocode. At Line 5, if D is X_0 , i.e. for Problem 2, r_t is obtained by first using the oracle (the Stateflow/Simulink model of the CPS of interest) to generate an output signal $y_t(t|\mathbf{x}, \zeta)$ starting from initial state x_t and then calculating the corresponding robustness degree with respect to the fixed STL formula $\varphi_{\bar{\theta}}$; if D is Θ i.e. for Problem 3, r_t is obtained by directly calculating the robustness degree of the given output signal $y(t|\mathbf{x}, \zeta)$ with respect to the current STL formula φ_{θ_t} .

Theorem 2. Let $\delta \in (0, 1)$, $\beta_t = 2 \log(|D|t^2\pi^2/6\delta)$, $p = \min_{t=(1,\dots,T)}(\eta_\mu(d_t))$, $q = \max_{t=(1,\dots,T)}(\eta_\mu(d_t))$, and $|D| \leq 1$. Running GP-ACB results in a regret bound as follows

$$Pr\left\{R_T \leq \sqrt{qC_1T\beta_T\gamma_T}, \forall T \geq 1\right\} \geq 1 - \delta^p, \quad (13)$$

where $C_1 = 8/\log(1 + \sigma^{-2})$.

Remark 8. Based on the bound on $\eta_\mu(\mathbf{d})$ in Proposition 1 and the definition of the information gain γ_T (see Appendix 8.2), it is obvious that $\Sigma_{t-1}(\mathbf{d}_t)$ is close to 0 when the iteration step t is large, implying that p and q in Theorem 2 are close to one in this case. This means that GP-ACB algorithm is greedy at first (when the iteration step t is small) and then gradually degrades to GP-UCB algorithm (when t is large). The regret bound of the GP-UCB algorithm is [27]

$$Pr\left\{R_T \leq \sqrt{C_1T\beta_T\gamma_T}, \forall T \geq 1\right\} \geq 1 - \delta.$$

With the same parameter setting, the regret bound of the GP-ACB algorithm is Equation (13). Since $0 < p, q \leq 1$, we can conclude that the GP-ACB algorithm can get the same regret bound more efficiently than the GP-UCB algorithm. Since the regret bound can be easily translated into the convergence rate, we can also conclude that, on average, the GP-ACB algorithm has a higher convergence rate than GP-UCB. The setting $|D| \leq 1$ implies that the search space X_0 or Θ needs to be normalised before the application of GP-ACB. The proof of this theorem is provided in Appendix 8.2.

5. Solutions

As already pointed out in Section 3.2, the solution to Problem 1 is not one single satisfactory valuation θ^* but a set of satisfactory valuations. In this section, we will first show how to find single satisfactory valuations and then briefly describe how to get sets of satisfactory valuations.

5.1. Single satisfactory valuation

Algorithm 2: Formal CPS performance interpretation with a single valuation as the output

Input: A CPS S with an initial set X_0 ; a user-specified PSTL φ_θ with parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$; kernel functions k_x and k_θ with unknown hyper-parameters $\tilde{\sigma}_x$ and $\tilde{\sigma}_\theta$, respectively; bounds ρ , ι and ε ; computational budget N_{sim} .

output: A satisfactory valuation θ^* .

- 1: Initialise θ to a random value $\bar{\theta}$ within Θ ;
- 2: Set $\tilde{\sigma}_x$ and $\tilde{\sigma}_\theta$ to some prior values;
- 3: Randomly select a set of $c - 1$ ($c \ll N_{sim}$) initial states within X_0 ,
 $\mathbf{x}_\mathcal{L} \leftarrow \{\mathbf{x}_i\}_{i=1, \dots, c-1}$;
- 4: Use S as an oracle to simulate a set of output signals $y_\mathcal{L} \leftarrow \{y_i(t|\mathbf{x}_i)\}_{i=1, \dots, c-1}$ with $\mathbf{x}_\mathcal{L}$ as initial states;
- 5: $\hat{r}_\mathcal{L}^\theta \leftarrow \{-\hat{r}(\mathbf{x}_i, \varphi_{\bar{\theta}})\}_{i=1, \dots, c-1}$;
- 6: **repeat**
- 7: **repeat**
- 8: Construct GP approximation $\hat{r}_{\bar{\theta}}(\cdot)$ i.e. $\mu_{c-1}^{\bar{\theta}}(\cdot)$ and $\Sigma_{c-1}^{\bar{\theta}}(\cdot)$, with k_x , $\tilde{\sigma}_x$, $\mathbf{x}_\mathcal{L}$, and $\hat{r}_\mathcal{L}^\theta$;
- 9: $\tilde{\sigma}_x^* \leftarrow \operatorname{argmax}_{\tilde{\sigma}_x} \log \mathbb{P}(\hat{r}_\mathcal{L}^\theta | \mathbf{x}_\mathcal{L}, \tilde{\sigma}_x)$;
- 10: $\mathbf{x}_c \leftarrow \operatorname{argmax}_{\mathbf{x} \in X_0} (\mu_{c-1}^{\bar{\theta}}(\mathbf{x}) + \eta_\mu^{\bar{\theta}}(\mathbf{x})^{\frac{1}{2}} \beta_c^{\frac{1}{2}} \Sigma_{c-1}^{\bar{\theta}}(\mathbf{x}))$;
- 11: Use S to simulate $y_c(t|\mathbf{x}_c)$ with \mathbf{x}_c as initial state;
- 12: $\mathbf{x}_\mathcal{L} \leftarrow \mathbf{x}_\mathcal{L} \cup \mathbf{x}_c$; $y_\mathcal{L} \leftarrow y_\mathcal{L} \cup y_c$; $\hat{r}_\mathcal{L}^\theta \leftarrow \hat{r}_\mathcal{L}^\theta \cup \{-\hat{r}(y_c(t|\mathbf{x}_c), \varphi_{\bar{\theta}})\}$;
- 13: $c \leftarrow c + 1$;
- 14: **until** $2\beta_c^{1/2} \Sigma_{c-1}^{\bar{\theta}}(\mathbf{x}_c) < \iota$
- 15: Initialise $\theta_\mathcal{L}$; $t = |\theta_\mathcal{L}|$;
- 16: $\hat{r}_\mathcal{L}^x \leftarrow \{\max(0, \varepsilon - |\hat{r}(\mathbf{x}_c, \varphi_{\theta_i})|)\}_{i=1, \dots, t}$;
- 17: **repeat**
- 18: Construct GP approximation $\hat{r}_x(\cdot)$ i.e. $\mu_t^x(\cdot)$ and $\Sigma_t^x(\cdot)$, with k_θ , $\tilde{\sigma}_\theta$, $\theta_\mathcal{L}$, and $\hat{r}_\mathcal{L}^x$;
- 19: $\tilde{\sigma}_\theta^* \leftarrow \operatorname{argmax}_{\tilde{\sigma}_\theta} \log \mathbb{P}(\hat{r}_\mathcal{L}^x | \theta_\mathcal{L}, \tilde{\sigma}_\theta)$;
- 20: $\theta_{t+1} \leftarrow \operatorname{argmax}_{\theta \in \Theta} (\mu_t^x(\theta) + \eta_\mu^x(\theta)^{\frac{1}{2}} \beta_{t+1}^{\frac{1}{2}} \Sigma_t^x(\theta))$;
- 21: $\theta_\mathcal{L} \leftarrow \theta_\mathcal{L} \cup \theta_{t+1}$; $\hat{r}_\mathcal{L}^x \leftarrow \hat{r}_\mathcal{L}^x \cup \{\max(0, \varepsilon - |\hat{r}(y_c(t|\mathbf{x}_c), \varphi_{\theta_{t+1}})|)\}$;
- 22: **until** $2\beta_{t+1}^{1/2} \Sigma_t^x(\theta_{t+1}) < \iota$
- 23: $\bar{\theta} \leftarrow \theta_{t+1}$;
- 24: **until** $c \geq N_{sim}$

The pseudocode of the algorithm to solve Problem 1 is provided in Algorithm 5.1. The procedures are rather self-explanatory: lines 7–14 use GP-ACB to solve Problem 2 and lines 17–22 use GP-ACB to solve Problem 3.

Moreover, notice that constructing $\hat{r}_\xi^{\bar{\theta}}$ (lines 5 and 12) and \hat{r}_ξ^* (lines 16 and 21) conform with Equations (8) and (10).

For the remainder of this subsection, we will elaborate on the hyper-parameters $\tilde{\sigma}_x$ (line 9 in Algorithm 2 and Equation (8) in Problem 2) and $\tilde{\sigma}_\theta$ (line 19 in Algorithm 2 and Equation (10) in Problem 3). Interest in leveraging GPs for the purpose of formal verification, mining, and inference of CPSs has surged as of late [8,13,17,18,28]. Among the current works, many assume that the true hyper-parameters are known a priori and also fixed. This is obviously not the case in the majority of practical cases. In this paper, we instead estimate relevant hyper-parameters from data. To demonstrate, we shall use the marginal robustness degree function $r(\cdot, \varphi_{\bar{\theta}})$ and the set of its hyper-parameters $\tilde{\sigma}_x$ as an example (the other marginal robustness degree function $r(\mathbf{x}, \varphi)$ and the set of its hyper-parameters $\tilde{\sigma}_\theta$ can be understood accordingly).

Let us return to the original notation of the robustness degree. The robustness degree function $r(y(t|\cdot, \xi), \varphi_{\bar{\theta}}, 0) \in F(X_0, \mathbb{R})$ corresponding to observed output signals $y(t|\cdot, \xi)$ can be approximated as the summation of a latent function $r(y(t|\cdot, 0), \varphi_{\bar{\theta}}, 0) \in F(X_0, \mathbb{R})$, which is exactly the marginal robustness degree function $r(\cdot, \varphi_{\bar{\theta}})$ mentioned in Assumption 3, and an additive perturbation term $v(\cdot)$, namely

$$r(y(t|\cdot, \xi), \varphi_{\bar{\theta}}, 0) = r(y(t|\cdot, 0), \varphi_{\bar{\theta}}, 0) + v(\cdot). \tag{14}$$

Then from the interpretation algorithm’s perspective, only $r(y(t|\mathbf{x}, \xi), \varphi_{\bar{\theta}}, 0)$ can be accessed, while $r(y(t|\mathbf{x}, 0), \varphi_{\bar{\theta}}, 0)$ is hidden. Recall that we use $r(\mathbf{x}, \varphi_{\bar{\theta}})$ to denote $r(y(t|\mathbf{x}, 0), \varphi_{\bar{\theta}}, 0)$ and $\hat{r}(\mathbf{x}, \varphi_{\bar{\theta}})$ to denote $r(y(t|\mathbf{x}, \xi), \varphi_{\bar{\theta}}, 0)$.² According to Assumption 3, given a set of initial-state-robustness-degree pairs $\mathcal{L}_{x,r} = \{\mathbf{x}_\xi, r_\xi^{\bar{\theta}}\}$, $r(\mathbf{x}, \varphi_{\bar{\theta}})$ can be approximated by a GP with its prior as

$$\mathbb{P}(r(\mathbf{x}_\xi, \varphi_{\bar{\theta}}) | \mathbf{x}_\xi, \sigma_x) \sim \mathcal{N}(0, \Sigma_x) \tag{15}$$

where $r(\mathbf{x}_\xi, \varphi_{\bar{\theta}}) = [r(\mathbf{x}_1, \varphi_{\bar{\theta}}), \dots, r(\mathbf{x}_{|\mathcal{L}_{x,r}|}, \varphi_{\bar{\theta}})]^T$ and $\sigma_x = \{\sigma_{\mathbf{x},ARD}^2, w_{\mathbf{x},1}, w_{\mathbf{x},2}, \dots, w_{\mathbf{x},n_x}\}$ (see Assumption 3). Assume that v in Equation (14) is approximately Gaussian and uniform. Then the likelihood for a pair $(r(\mathbf{x}_i, \varphi_{\bar{\theta}}), \hat{r}(\mathbf{x}_i, \varphi_{\bar{\theta}}))$ is given by

$$\mathbb{P}(\hat{r}(\mathbf{x}_i, \varphi_{\bar{\theta}}) | r(\mathbf{x}_i, \varphi_{\bar{\theta}})) \sim \mathcal{N}(r(\mathbf{x}_i, \varphi_{\bar{\theta}}), \sigma_v^2) \tag{16}$$

where σ_v is the covariance for the perturbation v . The likelihood function for $\mathcal{L}_{x,r} = \{\mathbf{x}_\xi, r_\xi^{\bar{\theta}}\}$ is then given by

$$\mathbb{P}(r_\xi^{\bar{\theta}} | r(\mathbf{x}_\xi, \varphi_{\bar{\theta}}), \sigma_v) = \prod_{i=1}^{|\mathcal{L}_{x,r}|} \mathbb{P}(\hat{r}(\mathbf{x}_i, \varphi_{\bar{\theta}}) | r(\mathbf{x}_i, \varphi_{\bar{\theta}})). \tag{17}$$

Finally,

$$\mathbb{P}(r_{\Sigma}^{\bar{\theta}}|\mathbf{x}_{\Sigma}, \tilde{\sigma}_{\mathbf{x}}) = \int^{\mathbb{P}} (r_{\Sigma}^{\bar{\theta}}|r(\mathbf{x}_{\Sigma}, \varphi_{\bar{\theta}}), \sigma_v) \mathbb{P}(r(\mathbf{x}_{\Sigma}, \varphi_{\bar{\theta}})|\mathbf{x}_{\Sigma}, \sigma_{\mathbf{x}}) dr(\mathbf{x}_{\Sigma}, \varphi_{\bar{\theta}})$$

where $\tilde{\sigma}_{\mathbf{x}} = [\sigma_{\mathbf{x}}, \sigma_v]$. Then the hyper-parameters $\tilde{\sigma}_{\mathbf{x}}$ can be estimated by optimising the log marginal likelihood $\mathbb{P}(r_{\Sigma}^{\bar{\theta}}|\mathbf{x}_{\Sigma}, \tilde{\sigma}_{\mathbf{x}})$, namely

$$\tilde{\sigma}_{\mathbf{x}}^* = \underset{\sigma_{\mathbf{x}}}{\operatorname{argmax}} \log \mathbb{P}(r_{\Sigma}^{\bar{\theta}}|\mathbf{x}_{\Sigma}, \tilde{\sigma}_{\mathbf{x}}). \quad (18)$$

This is exactly Equation (8). In this paper, the above optimisation problem is solved with the Gaussian processes for machine learning (GPML) toolbox with the Polack–Ribiere conjugate gradients implementation [29].

5.2. Sets of satisfactory valuations: critical level sets

The way to obtain a set of satisfactory valuations is quite straightforward given Algorithm 2. Such sets can be mathematically characterised by critical level sets. A critical level set of a function F e.g. a marginal robustness degree function $r(\mathbf{x}, \varphi)$, with a parameter space D is a connected component χ satisfying $\chi \subset \operatorname{cc}(\{\mathbf{d} \in D : F(\mathbf{d}) = cl\})$, where cl is the corresponding critical level e.g. a robustness degree value [30]. In the context of formal CPS performance interpretation, a human user is in the loop, who wants to understand the intricacies and tendencies of the CPS. We are therefore not only interested in one valuation θ^* that explains the CPS's performance (Problem 1), but also how the robustness degree changes with respect to different parameters and valuations (an illustrative example will be provided in Section 6.2).

The construction of critical level sets can be achieved by (i) setting cl (a robustness degree value in this case) to a specific value; (ii) running the method elaborated in the last sub-section M times to find M parameters $\{\theta_i\}_{i=1, \dots, M}$, each of which solves Problem 1, specifically each θ_i satisfies

$$\mathbb{P}(0 < \min_{\mathbf{x} \in X_0} r(\mathbf{x}, \varphi_{\theta_i}) < \iota) > 1 - \delta; \quad (19)$$

(iii) using ε -SVR (epsilon-insensitive support vector regression) to approximate the critical level set corresponding to cl , and (iv) choosing a different cl and continuing (i)-(iv) until halted by the user.

6. Case studies

In this section, we will first illustrate the performance of GP-ACB (the algorithm provided in Section 4) on a known benchmark, the global optimisation of the Rastrigin function. Then we will illustrate our formal CPS performance interpretation methodology (using algorithms provided in Section 5) with a case study in automotive mechatronic design.

6.1. Global optimisation of Rastrigin function

The performance of four active learning algorithms based on GPs are compared: (i) GP-UCB with fixed hyper-parameters [27], (ii) GP-UCB with optimised hyper-parameters [31], (iii) GP-ACB with fixed hyper-parameters, and (iv) GP-ACB with optimised hyper-parameters; the last strategy is the one being advocated in this paper. We choose GP-UCB since it is a state-of-the-art active learning algorithm. The usage of ‘optimised’ means that the real hyper-parameters are unknown a priori, and actually estimated from data. The global optimisation of the Rastrigin function (shown in Figure 1(a)) is used as the benchmark here. This surface is described with the following formula:

$$f(x, y) = 20 + x^2 - 10 \cos(2\pi x) + y^2 - 10 \cos(2\pi y) + e$$

where e is a Gaussian noise with zero mean and variance σ^2 of 1. The search space $D = [-5, 5]^2$ is randomly discretised into 1000 points. Each algorithm is run for $T = 300$ iterations. Since the global minimum of the Rastrigin function (x^*, y^*) is known (though unknown to the learning algorithms), for the i -th trial, if (x_t^i, y_t^i) is the solution obtained by running the algorithm for t iterations, then the mean regret for the algorithm at time t is $\bar{R}_t = \sum_{i=0}^{N_t} [f(x_t^i, y_t^i) - f(x^*, y^*)] / N_t$, where N_t is the number of trials. In this case study, N_t is set to 100. Each trial is initialised randomly.

Figure 1(b) shows the mean regrets \bar{R}_t incurred by the four Gaussian-process-based algorithms. GP-ACB with optimised hyper-parameters

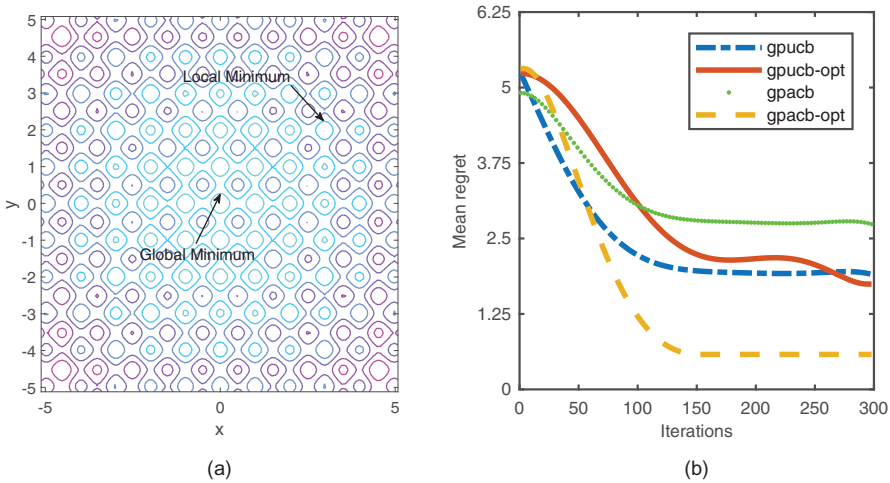


Figure 1. (a) The contour map of the Rastrigin function. (b) Comparison of the performances of the four active learning algorithms. The notation ‘-opt’ in the legend indicates that the hyper-parameters are allowed to be re-optimised after each iteration. The mean regret over 100 trails $\bar{R}(t) := \bar{R}_t = \sum_{i=0}^{100} [f(x_t^i, y_t^i) - f(x^*, y^*)] / 100$ is chosen as the performance metric.

outperforms the others. For instance, GP-UCB arrives at its minimum regret in an average of 300 iterations; while GP-ACB arrives at its minimum regret in an average of 150 iterations. Moreover, the mean regret \bar{R}_t of the GP-ACB is much lower than that of the GP-UCB.

6.2. Formal interpretation for automotive mechatronic design

Next, the utility of the proposed interpretation approach elaborated in [Section 5](#) is demonstrated in the context of automotive mechatronic design; an engineer explores the influence of certain cyber and physical system parameters on satisfaction of performance targets, in addition to the completeness and valuation of these objectives.

6.2.1. Model description

In this case study, our modelling assumptions yield a computational description that (i) defies traditional analytic guarantees on closed-loop stability, and (ii) can adequately represent some established vehicle design trade-offs. We depict a typical modern mid-size passenger car with 4-corner semi-active suspension shock absorbers (whose effective damping coefficients can be modulated on-the-fly). Though a complete derivation of the simulation model is outside the scope of this paper, here we would like to highlight a few interpretation and design challenges related to the following output signals: the lateral component of the vehicle inertial trajectory (Y), lateral velocity (v), yaw rate (ω_y), sideslip angle (θ_{side}), sprung mass vertical acceleration (a_{heave}), and the normal force on the road wheels (F_{zi} at each corner).

An important and safety-critical controller found in most all modern vehicles is one that provides torque assistance (and haptic feedback) to the driver for the steering task. This controller is usually at least partially tuned using heuristics, embedding a manufacturer-specific feeling. This nuanced configuration may take the form of non-linear multi-dimensional look-up tables; a basic example is shown in [Figure 2\(a\)](#). Note that the steering ‘boost’ generally decreases with forward velocity: the scrubbing effect associated with the tire’s steered (diametral) axis is reduced with increased rolling rate, owing to its viscoelastic properties and the friction conditions at the road interface. A simulation model designed to steer at high and low forward velocities cannot fairly represent these dynamics with a linear function. Moreover, analytic guarantees on the closed-loop steering performance can be hard to obtain without significant simplifying assumptions regarding this subsystem.

The vehicle model is designed to be valid under large centre-of-gravity accelerations, which cause large changes in chassis attitude (by comparison with nominal driving conditions). These can be a result of emphatic user inputs, or simply changes in roadway elevation taken at speed. Such

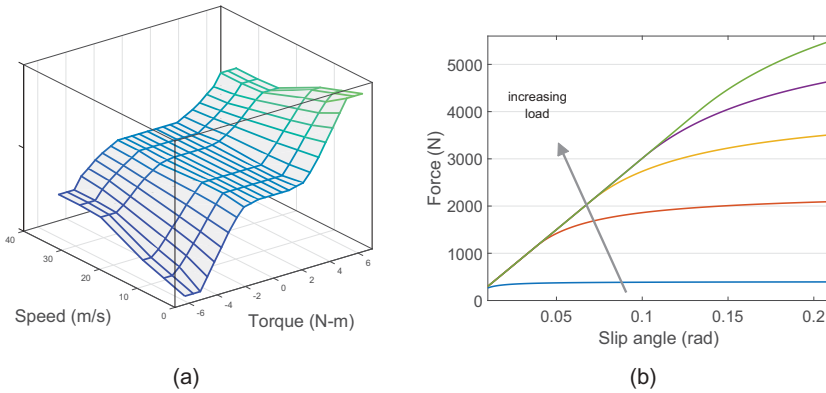


Figure 2. (a) Boost curve for a generic electric power-assist steering system (EPAS). (b) Non-linear relationship for lateral tire force generation as a function of normal loading. Note the saturating effect of added weight transfer.

occurrences result in weight transfer—changes in the vertical (normal) force under each wheel. An important non-linear relationship relates changes in the normal and lateral tire forces, the latter of which motivates changes in the vehicle’s physical trajectory (an important component of vehicle handling assessment). [Figure 2\(b\)](#) shows the lateral force generation with increasing slip angle (slip angles result from steering, or the vehicle dynamics), for high and low normal loads. Not only is the amplitude impacted, but the rate of change of lateral force is non-constant with increasing vertical force—an essential trend in the analysis and design of both suspension parts and feedback compensators. For this case study, we consider two parameters (one each cyber and physical) as a design demonstration. The physical parameter is the fore-aft distribution of the (fixed) total vehicle roll stiffness, in the form of a dimensionless ratio of anti-roll bar rates (N-m/rad). This has a significant effect on handling properties, particularly closer to the limits of road adhesion [32,33].

The software governing the semi-active suspension houses the second (tuning) parameter we investigate in this study. A modal chassis control algorithm addresses the basic vibratory patterns of the body [34]: here we develop expressions for the desired (i) rolling torque, (ii) pitching torque and (iii) heaving (vertical) force that are proportional to the roll/pitch/heave velocities. This creates the effect of ‘inertial damping,’ a concept germane to work focusing on the improvement of vehicle ride qualities. The following items are used to make this calculation: (i) the vehicle roll, pitch and heave momenta, $p = [p_R, p_P, p_H]^T$, (ii) the body mass and centroidal mass moments of inertia in the roll and pitch principle directions (m, J_R, J_P), (iii) platform dimensions including those locating the centre-of-gravity longitudinally relative to the axles and track widths (a, b, w_1, w_2), and (iv) the inertial roll, pitch and heave

damping coefficients (C_r, C_p, C_h). The heave coefficient C_h is allowed to vary for this study. The required force and torques is mapped to the desired forces at each corner (to be generated by the damper, if possible), $F_{L1}, F_{R1}, F_{L2}, F_{R2}$, through the platform geometry:

$$-\mathbf{p} \circ \begin{bmatrix} C_r \\ J_R \\ C_p \\ J_p \\ C_h \\ m \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{w_1}{2} & \frac{w_1}{2} & -\frac{w_1}{2} & \frac{w_1}{2} \\ -a & -a & b & b \end{bmatrix} \begin{bmatrix} F_{L1} \\ F_{R1} \\ F_{L2} \\ F_{R2} \end{bmatrix},$$

where \circ is element-wise multiplication.

These forces, as designed above, can demand power from corner actuators. In our example application, semi-active dampers are selected, only capable of dissipating power, albeit at a variable rate [35]. We must therefore restrict the corner commands to represent those which can be created by semi-active force generators. This is achieved with the process described in Algorithm 3. This switching condition serves as a final example of system properties that are realistic, difficult to analytically verify, though necessary for representing the relevant dynamics and objectives.

Algorithm 3: Calculation of suspension semi-active damper forces using passivity

Input: Allocated forces $\mathbf{F} = [F_{L1}, F_{R1}, F_{L2}, F_{R2}]^T$; suspension relative velocities $\mathbf{v}_{rel} = [v_{L1}, v_{R1}, v_{L2}, v_{R2}]^T$.

Output: Desired passive suspension damper forces \mathbf{F}_p .

- 1: Restrict computation to appreciable relative velocities $|\mathbf{v}_{rel}| < \epsilon_{4 \times 1}$;
- 2: **if** $\mathbf{F} \circ \mathbf{v}_{rel} > 0$
- 3: $\mathbf{F}_p = \mathbf{F}$;
- 4: **else**
- 5: $\mathbf{F}_p = \mathbf{F}_{min}$, the damper OFF state.
- 6: **end if**

6.2.2. Test conditions and PSTL templates

We select two test modes suitable for evaluating the ride and handling properties of a road vehicle. For testing the handling performance, a step steer manoeuvre is chosen (ISO 7401 provides guidance and specifications). To inspect one aspect of ride, we traverse a bump on the right track only—thus exciting heave, roll and pitch motion.

Bump Test: During the bump test, the vehicle with a forward speed of 30 kph runs over a rounded obstacle as shown in Figure 3(a) with the wheels on the right-hand side. The PSTL templates for the bump test (that can be used by a designer to formally query the performance of the vehicle) are as follows:

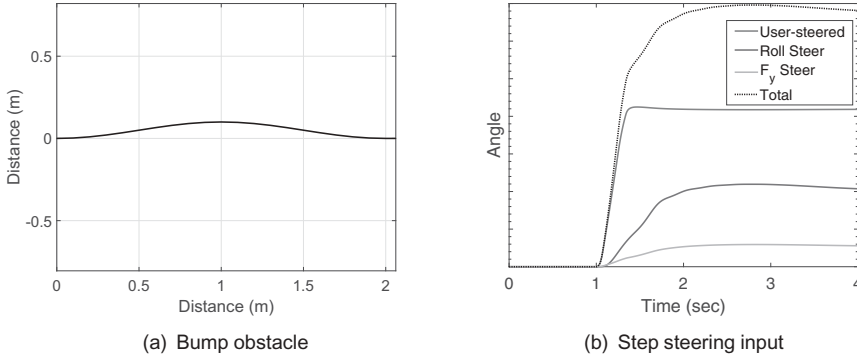


Figure 3. Test input for (a) the bump test, in road elevation profile for the right track and (b) the step steering test, in steering angular displacement. Note how the total wheel angle setting is a combination of the handwheel input and passive contributions from the suspension (both kinematic and compliant).

- The vehicle should have a straight trajectory as much as possible after traversing the bump, $\varphi_1 := G(|Y| < \pi_1)$;
- Normal force variation should be minimised, for both safety and handling reasons, $\varphi_2 := G(|F_z - F_0| < \pi_2)$;
- Small heave acceleration obtains a good ride experience, $\varphi_3 := G(a_{heave} < \pi_3)$.

Thus, the formal interpretation of a particular automobile design in the context of a rounded obstacle test can be the conjunction of the above three PSTL templates:

$$\phi_{bump} := \varphi_1 \wedge \varphi_2 \wedge \varphi_3. \quad (20)$$

Step Steering Test: During the step steering test, an open-loop steering input is used; the final steering angle is first calibrated to the desired output, then the driver has no effect on the results (a mechanised driver can be used). The input is shown in Figure 3(b), the standard forward speed is 100 kph (about 62 mph). Starting with yaw rate and lateral velocity close to zero, the steering wheel is turned quickly to that value which yields a selected nominal lateral acceleration in the steady state. The PSTL templates for the step steering test are as follows:

- Design changes should minimise body sideslip angle, $\varphi_4 := G(\theta_{side} < \pi_4)$;
- Excessive heave acceleration should be avoided, $\varphi_5 := G(a_{heave} < \pi_5)$;
- Maximise steady-state yaw rate and bounded maximum yaw rate, $\varphi_6 := F_{[3,T]}(\omega_{ys} > \pi_6) \wedge G(\omega_{ym} < \pi_7)$.

Thus, the formal interpretation in the context of the step steer test can be the conjunction of the above three templates:

$$\phi_{step} := \varphi_4 \wedge \varphi_5 \wedge \varphi_6. \quad (21)$$

This example set of PSTL templates for the bump and step tests clearly do not represent all conceivable objective metrics for automotive ride and handling performance. They can, however, serve as a basis to illustrate how a designer might interact with our formal CPS interpretation methodology, and gain insight into the design of complex CPSs.

6.2.3. Algorithmic performance

Here we use the bump test as an example to demonstrate the performance of our proposed algorithm in solving the formal interpretation problem i.e. Problem 1, when only a single satisfactory valuation is required (the corresponding algorithm was provided in Section 5.1). The iteration limit N_{sim} is set to 400. Figure 4 shows the performance of four active learning algorithms (the same as those mentioned in Section 6.1) over 10 randomly-initialised runs when the user-specified PSTL template is $\varphi_1 := G(|Y| < \pi_1)$. The error percentages are calculated by dividing estimation errors by the approximated true valuation, which is obtained by sampling the parameter space with 10,000 points and choosing the one that leads to a robustness closest to zero. The results show that the active learning algorithms with optimised hyper-parameters outperform those with fixed hyper-parameters. The performances of GP-ACB and GP-UCB with optimised hyper-parameters are indistinguishable for this performance metric.

Next we use the step input test as an example to demonstrate the performance of our proposed algorithm in solving the formal interpretation problem when a set of satisfactory valuations is required (the corresponding algorithm was provided in Section 5.2). In this case, we set the critical level c_l to zero and

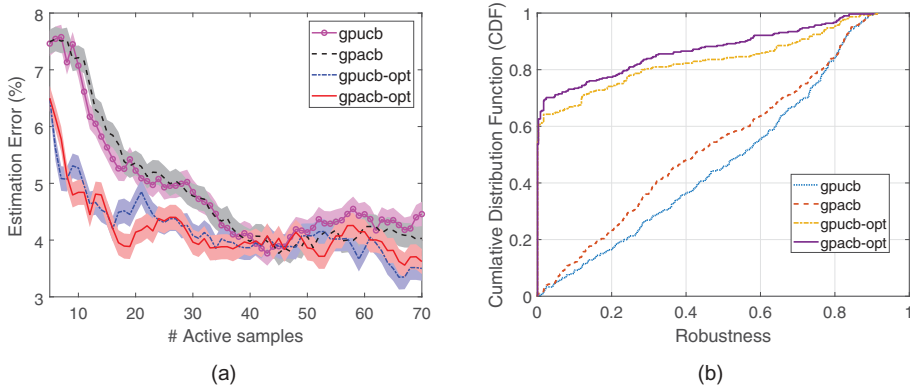


Figure 4. (a) Comparison of the performance of the four active learning algorithms over 10 randomly-initialised runs when the user-specified PSTL template is $\varphi_1 := G(|Y| < \pi_1)$. The x-axis corresponds to the number of samples selected by the active learning algorithms (c in Algorithm 2). Standard deviation intervals correspond to σ bounds. (b) The cumulative distribution functions (CDFs) of the absolute values of all the sampled output signals' robustness degrees for the four active learning algorithms. The user-specified PSTL template is chosen to be ϕ_{step} .

plot the cumulative distribution function (CDF) of the absolute values of the robustness degrees of the sampled output signals, i.e. the \hat{r}_σ^θ in Algorithm 2. Ideally a learning algorithm with high efficiency tends to sample signals with robustness degrees closer to the set critical level (0 in this case). Therefore, the CDF can serve as a good indicator of the sampling efficiency of learning algorithms. To compare the performance of the four active learning algorithms, we let all of them start with the same 10 randomly-selected initial states and the computational budget N_{sim} is set to 500. Figure 4(b) shows the CDFs of the absolute values of all the sampled signals' robustness degrees for the four algorithms. It clearly shows that GP-ACB outperforms GP-UCB in term of sampling efficiency and GP-ACB with optimised hyper-parameters has the best performance.

6.2.4. Formal-interpretation-aided design

In this sub-section, we show two effective ways that the formal interpretation methodology proposed in this paper can be utilised to help designer to analyse and design complex CPS.

The *first* way is to investigate the pair-wise relationships between interpretation parameters and their joint effects on the system performance, as described by STL formulas. In this procedure, the designer only allows one parameter to change while keeping all the others fixed. The designer can then investigate the effect of the changes in interpretation parameterisation on the robustness degree with respect to a particular performance that he or she is interested in studying which is codified by a PSTL template. Figure 5(a) shows

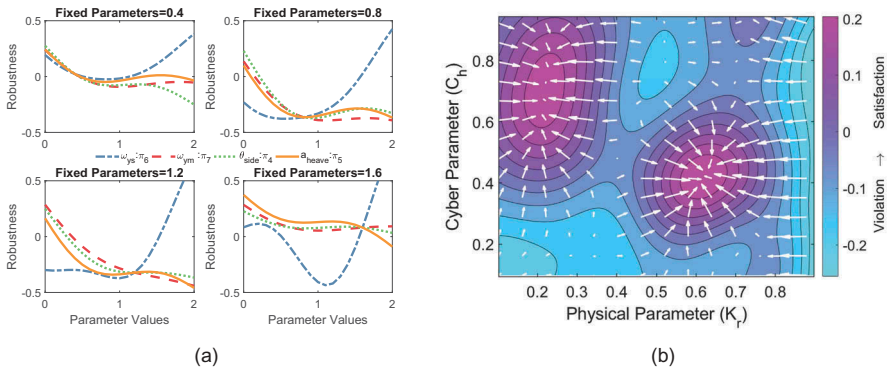


Figure 5. (a) Pair-wise relationships between the four parameters ($\pi_4, \pi_5, \pi_6, \pi_7$) and the robustness degree with respect to ϕ_{step} . The sub-figures show the effect of a single parameters' change in interval $[0, 2]$ to the robustness degree, during which the other three parameters are fixed to 0.4 (top left), 0.8 (top right), 1.2 (bottom left), and 1.6 (bottom right). (b) Critical level sets of the vehicle systems robustness degree for the bump test (the corresponding STL formula is $\phi_{bump} := \varphi_1 \wedge \varphi_2 \wedge \varphi_3$ with valuation $[1, 1, 1]$) with respect to two design parameters: a physical one, the roll bar stiffness ratio K_r , and a cyber one, the inertial heave damping coefficient C_h .

the pair-wise relationship between the robustness degree with respect to ϕ_{step} and four parameters for the step steering test: sideslip angle θ_{side} , heave acceleration a_{heave} , steady yaw rate ω_{ys} , and maximum yaw rate ω_{ym} . To get these results, the algorithm described in Section 5.2 is used to find the sets of valuations of the varying parameter corresponding to a range of robustness degree critical levels. In the sub-figures, one of the four parameters changes in interval $[0, 2]$, and the other three parameters are fixed. For example, the bottom-left figure indicates that all four parameters impact the robustness degree when the others are held fixed at 1.2. The bottom right belies the insensitivity of the robustness degree to the changes of maximum yaw rate ω_{ym} and sideslip angle θ_{side} when the fixed value is high (effectively relaxed requirements). Moreover, Figure 5(a) also shows the rates of change of the robustness degrees with the slope of these curves, a quite useful information for conditioning the PSTL template structure. With the knowledge obtained from analysing the pair-wise relationships, designers can adjust their design and balance the influence of different cyber-physical parameters knowing their impact on the overall satisfaction of the system requirements.

The *second* way to help designers to gain insights and form meta-knowledge of CPSs is to utilise plots of critical level sets of the robustness degree against various cyber-physical parameters. Here again the algorithm described in Section 5.2 can be used. Figure 5(b) shows the effect of the two selected cyber-physical parameters, inertial heave damping coefficient C_h and roll bar stiffness distribution K_r , to the robustness degree for the bump test (ϕ_{bump}). With this plot, it is easy for the designers to know how the changes in the design parameters can affect the vehicle's performance in the context of the interpretation of interest. For example, Figure 5(b) shows two critical positions where the satisfaction reaches a local maximum, namely two peaks can be seen around point (0.21, 0.7) and point (0.63, 0.4). One can place these two viable configurations on the contours of the step steering test (which is not shown here), and perhaps find that only one of these may be suitable when the testing regimen is evaluated in total.

In a general case, it is useful to further interrogate these two points by viewing the time-series data associated with them; this often has the effect of clarifying a way that the total interpretation is incomplete. Thus the process of design with this method is iterative in nature. For instance, Figure 6(a) shows a comparison of the body yaw torque for the two maxima found in Figure 5(b), obtained by running a simulation and extracting additional interesting output signals (over and above those used to construct the interpretation). One difference apparent from this plot is that the torque imparted to the cabin appears to have either a lower average value, or energetic transfer occurring across a higher bandwidth. Depending on design requirements and occupant

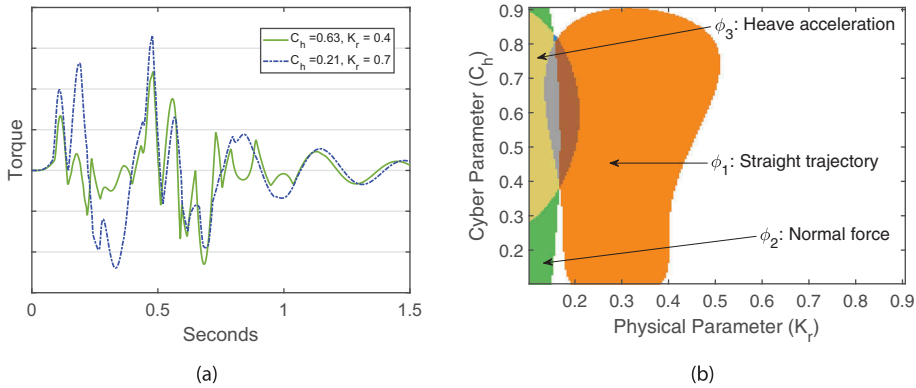


Figure 6. (a) Total torque about the body yaw axis, a factor in deciding ride quality, for both maxima found in Figure 5(b). Note the higher frequency content found in the case with higher heave damping. (b) Parameters' effective regions for three different STL formulas i.e. interpretations, ϕ_1 , ϕ_2 , and ϕ_3 . The regions with the robustness degree bigger than 0.1 are depicted.

location, one of these signals may be preferred over the other, suggesting that a more complete interpretation is needed.

In order to investigate the coupling effect between different STL formulas, or interpretations, an extra experiment is conducted here for the bump test. In this experiment, the system's performance is analysed with respect to the three individual component interpretations ϕ_1 , ϕ_2 , and ϕ_3 in $\phi_{bump} := \phi_1 \wedge \phi_2 \wedge \phi_3$ (with a valuation $[1, 1, 1]$). The way to analyse the system's performance with respect to an individual interpretation is exactly the same as the procedure for generating the map in Figure 5(b). After all three maps are generated, we select the regions having robustness degree bigger than 0.1 and put them into a single plot, as shown in Figure 6(b). Figure 5(b) and Figure 6(b) together clearly show that there are coupling effects between the three component interpretations. For instance, the peak around (0.15, 0.7) in Figure 5(b) is the place where all three interpretations reach relative high robustness degrees in Figure 6(b). However, this is not the case for the peak around (0.7, 0.4) in Figure 5(b). The difference between the peaks indicates that a good performance with respect to an overall interpretation does not necessary transfer to a good performance with respect to each individual component interpretation, at least not in an absolute sense.

7. Conclusions and future work

This paper introduced a new methodology of allowing human users or designers to interrogate the performance of complex CPSs via inquiries written in formal logic. A new active learning algorithm, called GP-ACB, was proposed to ease the possible computational cost related to the simulation or testing of

the CPSs. The paper showed both theoretically and empirically that GP-ACB has a better performance than many existing Gaussian-processes-based algorithms, such as GP-UCB. A case study on automotive mechatronic design was provided to demonstrate the power of the proposed methodology, and hopefully to provide practitioners with insights into how to design CPSs in a more formal manner. We are currently exploring possibilities of (i) integrating human-in-the-loop interactive learning into our methodology and (ii) allowing flexible rather than fixed PSTL templates.

Notes

1. It is worth pointing out that solving Problem 3 does not require generating new simulated output signals. Thus, it is not strictly subjected to the computational budget N_{sim} . None-the-less, in the context of CPS performance interpretation, there is a human user who is waiting for the answer to his or her inquiry. So it is still desirable to minimise the number of valuations $|\theta_{\mathcal{L}}|$ to improve the interpretation performance.
2. This explains the adoption of the sign $\hat{\cdot}$ in Section 3.2 and onward, implying ‘estimated’ or ‘observed’.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Office of Naval Research (ONR) [grant number N000141612027] and the Hyundai Motor Company [Gift].

ORCID

Zhaodan Kong  <http://orcid.org/0000-0002-2493-1366>

References

- [1] Maler O, Nickovic D. Monitoring temporal properties of continuous signals. In: Yassine L, Sergio Y, editors. Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. Berlin Heidelberg: Springer-Verlag. 2004. p. pages 71–76.
- [2] Donzé A, Maler O. Robust satisfaction of temporal logic over real-valued signals. In: Krishnendu C, Thomas A. H, editors. Formal Modeling and Analysis of Timed Systems. Berlin Heidelberg: Springer-Verlag. 2010. p. pages 92–106.
- [3] Bortolussi L, Nenzi L. Specifying and monitoring properties of stochastic spatio-temporal systems in signal temporal logic. In *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*, pages 66–73. ICST

- (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.
- [4] Hoxha B, Dokhanchi A, Fainekos G. Mining parametric temporal logic properties in model-based design for cyber-physical systems. *Int J Software Tools Technol Transfer*. 2017;20(1):79–93.
 - [5] Basile D, Di Giandomenico F, Gnesi S. Statistical model checking of an energy-saving cyber-physical system in the railway domain. In *Proceedings of the Symposium on Applied Computing*, pages 1356–1363. ACM, 2017.
 - [6] Fainekos GE, Pappas GJ. Robustness of temporal logic specifications for continuous-time signals. *Theor Comput Sci*. 2009;410(42):4262–4291.
 - [7] Silvetti S, Policriti A, Bortolussi L. An active learning approach to the falsification of black box cyber-physical systems. *arXiv preprint arXiv:1705.01879*, 2017.
 - [8] Quindlen JF, Topcu U, Chowdhary G, et al. Closed-loop statistical verification of stochastic nonlinear systems subject to parametric uncertainties. *arXiv preprint arXiv:1709.06645*, 2017.
 - [9] Lo D, Khoo S-C, Liu C. Mining past-time temporal rules from execution traces. In *Proceedings of the 2008 international workshop on dynamic analysis: held in conjunction with the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2008)*, pages 50–56. ACM, 2008.
 - [10] Shoham S, Yahav E, Fink SJ, et al. Static specification mining using automata-based abstractions. *IEEE Trans Software Eng*. 2008;34(5):651–666.
 - [11] Jin X, Donzé A, Deshmukh JV, et al. Mining requirements from closed-loop control models. *IEEE Trans Computer-Aided Des Integr Circuits Syst*. 2015;34(11):1704–1717.
 - [12] Vazquez-Chanlatte M, Jha S, Tiwari A, et al. Specification inference from demonstrations. *arXiv preprint arXiv:1710.03875*, 2017.
 - [13] Bartocci E, Bortolussi L, Sanguinetti G. Data-driven statistical learning of temporal logic properties. In: Axel L, Marius B, editors. *Formal Modeling and Analysis of Timed Systems*. Switzerland: Springer; 2014. p. pages 23–37.
 - [14] Kong Z, Jones A, Medina Ayala A, et al. Temporal logic inference for classification and prediction from data. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 273–282. ACM, 2014.
 - [15] Kong Z, Jones A, Belta C. Temporal logics for learning and detection of anomalous behavior. *IEEE Trans Automat Contr*. 2017;62(3):1210–1222.
 - [16] Settles B. Active learning literature survey. University of Wisconsin Madison. 2010;52(11):55–56.
 - [17] Fan C, Qi B, Mitra S, et al. Dryvr: data-driven verification and compositional reasoning for automotive systems. *arXiv preprint arXiv:1702.06902*, 2017.
 - [18] Chen G, Sabato Z, Kong Z. Active learning based requirement mining for cyber-physical systems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 4586–4593. IEEE, 2016.
 - [19] Asarin E, Donzé A, Maler O, et al. Parametric identification of temporal properties. In: Shaz Q, Serdar T, editors. *Runtime Verification*. Berlin Heidelberg: Springer-Verlag; 2012. pages 147–160.
 - [20] Rasmussen CE. *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press; 2006.
 - [21] Micchelli CA, Xu Y, Zhang H. Universal kernels. *J Machine Learn Res*. 2006;7:(Dec):2651–2667.

- [22] Abbas H, Fainekos G, Sankaranarayanan S, et al. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Trans Embedded Comput Syst (TECS)*. 2013;12(2s):95.
- [23] Sankaranarayanan S, Fainekos G. Falsification of temporal properties of hybrid systems using the cross-entropy method. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 125–134. ACM, 2012.
- [24] Srinivas N, Krause A, Kakade SM, et al. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *Inf Theory IEEE Trans*. 2012;58(5):3250–3265.
- [25] Tran D, Ranganath R, Blei DM. The variational gaussian process. *stat*. 2016;1050(23):1–14.
- [26] Haghghi I, Jones A, Kong Z, et al. Spatel: a novel spatial-temporal logic and its applications to networked systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 189–198. ACM, 2015.
- [27] Srinivas N, Krause A, Kakade S. Gaussian process optimization in the bandit setting: no regret and experimental design. *arXiv preprint arXiv*, pages 0912–3995, 2009.
- [28] Akazaki T, Kumazawa Y, Hasuo I. Causality-aided falsification. *arXiv preprint arXiv:1709.02555*, 2017.
- [29] Rasmussen CE, Nickisch H. Gaussian processes for machine learning (gpml) toolbox. *J Machine Learn Res*. 2010;11:(Nov):3011–3015.
- [30] Baronov D, Baillieul J. Decision making for rapid information acquisition in the reconnaissance of random fields. *Proc IEEE*. 2012;100(3):776–801.
- [31] K A, Desautels T, B JW. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *J Machine Learn Res*. 2014;100(3):776–801.
- [32] Varnhagen S, Anubi OM, Sabato Z, et al. Active suspension for the control of planar vehicle dynamics. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 4085–4091. IEEE, 2014.
- [33] Williams DE, Haddad WM. Nonlinear control of roll moment distribution to influence vehicle yaw characteristics. *IEEE Transactions Control Systems Technology*. 1995;3(1):110–116.
- [34] Wright P. The application of active suspension to high performance road vehicles. *J Mech E Paper C239*, 123, 1984.
- [35] Crosby M, Karnopp DC. The active damper—a new concept for shock and vibration control. *Shock and Vibration Bulletin*. 1973;43(4):119–133.
- [36] Dani V, Hayes TP, Kakade S. The price of bandit information for online optimisation. In *NIPS*. 2007;8.

8. Appendix

8.1. Proof of Theorem 1

Proof. Based on the settings of Problems 2 and 3, we have $\mathbb{P}(|\hat{r}_\theta(\mathbf{x}^*) - r(\mathbf{x}^*, \varphi_\theta)| \langle \iota \rangle) 1 - \rho$ and $\mathbb{P}(|\hat{r}_\theta(\varphi_{\theta^*}) - r(\mathbf{x}, \varphi_{\theta^*})| \langle \iota \rangle) 1 - \rho$. As the sampling processes for \mathbf{x}_ζ and θ_ζ are independent, we have $\mathbb{P}(|\hat{r}_\theta(\mathbf{x}^*) - r(\mathbf{x}^*, \varphi_\theta)| \langle \iota \rangle) = \mathbb{P}(|\hat{r}_\theta(\mathbf{x}^*) - r(\mathbf{x}^*, \varphi_\theta)| \langle \iota \rangle) \mathbb{P}(|\hat{r}_\theta(\varphi_{\theta^*}) - r(\mathbf{x}, \varphi_{\theta^*})| \langle \iota \rangle) (1 - \rho)^2$. Since $\hat{r}_{\theta^*}(\mathbf{x}^*) = 0$, then $\mathbb{P}(|r(\mathbf{x}^*, \varphi_{\theta^*})| \langle \iota \rangle) (1 - \rho)^2$. As $r(\mathbf{x}^*, \varphi_{\theta^*})$ is zero mean, then $\mathbb{P}(0 < r(\mathbf{x}^*, \varphi_{\theta^*}) \langle \iota \rangle) (1 - \rho)^2 / 2 \geq 1 - \delta$. The theorem has been proved.

8.2. Proof of Theorem 2

The proof of Theorem 2 follows the proofs of regret bound in [27,36]. Here only the cases when the search space is finite i.e. $|D| < \infty$, are considered. In this paper, the infinity norm for $|\cdot|$ will be used.

Define γ_T as the maximum information gain after T rounds as follows [27]:

$$\gamma_T = \max_{T' \leq T} \frac{1}{2} \sum_{t=1}^{T'} \log(1 + \sigma^{-2} \Sigma_{t-1}^2(\mathbf{d}_t)).$$

According to [27], if the search space $D \in \mathbb{R}^d$ is compact and convex, where d is the dimension of the search space, with the assumption that the kernel function satisfies $k(\mathbf{d}, \mathbf{d}') \leq 1$,

- $\gamma_T = \mathcal{O}((\log T)^{d+1})$ for Gaussian kernel
- $\gamma_T = \mathcal{O}(T^{d(d+1)/(2\nu+d(d+1))} (\log T))$ for Matérn kernels with $\nu > 1$.

Finally, for a unknown function g (i.e. ‘real’ robustness in this paper), the following bound for the GP-ACB algorithm can be obtained.

Lemma 1 Pick $\delta \in (0, 1)$ and set $\beta_t = 2 \log(|D| \pi_t / \delta)$, where $\sum_{t \geq 1} \pi_t^{-1} = 1$, $\pi_t > 0$. Then, with $|D| \leq 1$,

$$|g(\mathbf{d}) - \mu_{t-1}(\mathbf{d})| \leq \eta_\mu(\mathbf{d})^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}), \forall \mathbf{d} \in D, \forall t \geq 1$$

holds with probability $\geq 1 - \delta^{\eta_\mu(\mathbf{d})}$

Proof. For $\mathbf{d} \in D$ and $t \geq 1$. It is known that conditioned on $\mathbf{y}_{t-1} = (y_1, \dots, y_{t-1})$, $\{\mathbf{d}_1, \dots, \mathbf{d}_{t-1}\}$ is deterministic. Furthermore, $g(\mathbf{d}) \sim \mathcal{N}(\mu_{t-1}(\mathbf{d}), \Sigma_{t-1}^2(\mathbf{d}))$. Now if $r \sim \mathcal{N}(0, 1)$, then

$$\begin{aligned} \mathbb{P}(r > c) &= e^{-c^2/2} (2\pi)^{-1/2} \int e^{-(r-c)^2/2 - c(r-c)} dr \\ &\leq e^{-c^2/2} \mathbb{P}(r > 0) = (1/2) e^{-c^2/2}. \end{aligned}$$

for $c > 0$, as $e^{-(r-c)} \leq 1$ for $r \geq c$. Set $r = (g(\mathbf{d}) - \mu_{t-1}(\mathbf{d})) / \Sigma_{t-1}(\mathbf{d})$ and $c = \eta_\mu(\mathbf{d})^{1/2} \beta_t^{1/2}$. Then

$$\mathbb{P}(|g(\mathbf{d}) - \mu_{t-1}(\mathbf{d})| / \Sigma_{t-1}(\mathbf{d}) > \eta_\mu(\mathbf{d})^{1/2} \beta_t^{1/2}) \leq e^{-\eta_\mu(\mathbf{d}) \beta_t / 2}$$

After applying the adaptive bound,

$$|g(\mathbf{d}) - \mu_{t-1}(\mathbf{d})| \leq \eta_\mu(\mathbf{d})^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}) \quad \forall \mathbf{d} \in D$$

holds with probability $\geq 1 - e^{-\eta_\mu(\mathbf{d}) \beta_t / 2}$. Since $|D|$, the infinity norm, is less than or equal to 1, $e^{-\eta_\mu(\mathbf{d}) \beta_t / 2} \leq \delta^{\eta_\mu(\mathbf{d})} / \pi_t$ with $\pi_t = \pi^2 t^2 / 6$. Thus, the statement holds.

Remark 9. Before conducting Algorithm 1, the search space D is first normalised by its infinity norm $|D|$ to guarantee that the scaled search space D' satisfies $|D'| \leq 1$. The search space D' is then passed to Algorithm 1 as one of its inputs.

Lemma 2. Fix $t \geq 1$, if $|g(\mathbf{d}) - \mu_{t-1}(\mathbf{d})| \leq \eta_\mu(\mathbf{d})^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d})$, $\forall \mathbf{d} \in D$, then the regret r_t is bounded by $2\beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t)$.

Proof. According to the definition of d^* , $\mu_{t-1}(\mathbf{d}_t) + \eta_\mu(\mathbf{d}_t)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) \geq \mu_{t-1}(\mathbf{d}^*) + \eta_\mu(\mathbf{d}^*)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}^*) \geq g(\mathbf{d}^*)$. According to the definition of $\eta_\mu(\mathbf{d})$, then $\eta_\mu(\mathbf{d}) \leq 1$. Therefore, the instantaneous regret

$$\begin{aligned}
r_t &= g(\mathbf{d}^*) - g(\mathbf{d}_t) \\
&\leq \eta_\mu(\mathbf{d}_t)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) + \mu_{t-1}(\mathbf{d}_t) - g(\mathbf{d}_t) \\
&\leq 2\eta_\mu(\mathbf{d}_t)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) \leq 2\beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t)
\end{aligned}$$

Proposition 1 can be proved as follows:

Proof. Set $\mathbf{d}^m = \operatorname{argmax}(\mu_{t-1}(\mathbf{d}))$, $\forall \mathbf{d} \in D$, according to Equation (8), $\mathbf{d}_t = \operatorname{argmax} \eta_\mu(\mathbf{d}_t)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) + \mu_{t-1}(\mathbf{d}_t)$, thus $\eta_\mu(\mathbf{d}_t)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) + \mu_{t-1}(\mathbf{d}_t) \geq \eta_\mu(\mathbf{d}^m)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}^m) + \mu_{t-1}(\mathbf{d}^m)$, then

$$\begin{aligned}
&\mu_{t-1}(\mathbf{d}^m) - \mu_{t-1}(\mathbf{d}_{t-1}) \\
&\leq \eta_\mu(\mathbf{d}_t)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) - \eta_\mu(\mathbf{d}^m)^{1/2} \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}^m) \\
&\text{As } \mu_{t-1}(\mathbf{d}^m) - \mu_{t-1}(\mathbf{d}_{t-1}) = \mu_{t-1}(\mathbf{d}^m) - \min(\mu_{t-1}(\mathbf{d})) \\
&\quad + \min(\mu_{t-1}(\mathbf{d})) - \mu_{t-1}(\mathbf{d}_{t-1}) \\
&= L_t(1 - \eta_\mu(\mathbf{d}_t)^{1/2}) \\
&\Rightarrow 1 - \eta_\mu(\mathbf{d}_t)^{1/2} \\
&\leq \frac{\beta_t^{1/2}}{L_t} (\eta_\mu(\mathbf{d}_t)^{1/2} \Sigma_{t-1}(\mathbf{d}_t) - \eta_\mu(\mathbf{d}^m)^{1/2} \Sigma_{t-1}(\mathbf{d}^m)) \\
&\leq \frac{\beta_t^{1/2}}{L_t} (\eta_\mu(\mathbf{d}_t)^{1/2} \Sigma_{t-1}(\mathbf{d}_t) \leq \beta_t^{1/2} \Sigma_{t-1}(\mathbf{d}_t) / L_t
\end{aligned}$$

Proposition 1 has been proved.

Then Theorem 2 can be proved as follows:

Proof. According to Lemmas 1 and 2, the regret bound $\{r_t^2 \leq 4\eta_\mu(\mathbf{d}_t)\beta_t\Sigma_{t-1}^2(\mathbf{d}_t), \forall t \geq 1\}$ holds with probability $\geq 1 - \delta^{\eta_\mu(\mathbf{d}_t)} \geq 1 - \delta^p$. As β_t is non-decreasing, then

$$\begin{aligned}
4\eta_\mu(\mathbf{d}_t)\beta_t\Sigma_{t-1}^2(\mathbf{d}_t) &\leq 4q\beta_T\sigma^2(\sigma^{-2}\Sigma_{t-1}^2(\mathbf{d}_t)) \\
&\leq 4q\beta_T\sigma^2S \log(1 + \sigma^{-2}\Sigma_{t-1}^2(\mathbf{d}_t))
\end{aligned}$$

where $S = \sigma^{-2} / \log(1 + \sigma^{-2})$, since $\sigma^{-2}\Sigma_{t-1}^2(\mathbf{d}_t) \leq \sigma^{-2}k(\mathbf{d}_t, \mathbf{d}_t) \leq \sigma^{-2}$,

$C_1 = 8 / \log(1 + \sigma^{-2}) \geq 8\sigma^2$ and $h^2 \leq S \log(1 + h^2)$ for $h \in [0, \sigma^{-2}]$. As $C_1 = 8\sigma^2S$, for $T \geq 1$, then

$$\begin{aligned}
\sum_{t=1}^T r_t^2 &\leq \sum_{t=1}^T 4\eta_\mu(\mathbf{d}_t)\beta_t\Sigma_{t-1}^2(\mathbf{d}_t) \\
&\leq q\beta_T C_1 \frac{1}{2} \sum_{t=1}^T \log(1 + \sigma^{-2}\Sigma_{t-1}^2(\mathbf{d}_t)) \leq qC_1\beta_T\gamma_T.
\end{aligned}$$

According to Cauchy-Schwarz inequality, $R_T^2 \leq T \sum_{t=1}^T r_t^2$. Theorem 2 has been proven.