2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Active Learning Based Requirement Mining for Cyber-Physical Systems

Gang Chen, Zachary Sabato, Zhaodan Kong

***Abstract*— This paper uses active learning to solve the problem of mining signal temporal requirements of cyber-physical systems or simply the requirement mining problem. By utilizing robustness degree, we formulate the requirement mining problem as an optimization problem. We then propose a new active learning algorithm called Gaussian Process Adaptive Confidence Bound (GP-ACB) to help in solving the optimization problem. We show theoretically that the GP-ACB algorithm has a lower regret bound–thus a larger convergence rate–than some existing active learning algorithms, such as GP-UCB. We finally illustrate and apply our requirement mining algorithm with two case studies: the Ackley's function and a real world automotive power steering model. Our results demonstrate that there is a principled and efficient way of extracting requirements for complex cyber-physical systems.**

I. Introduction

In this paper, we propose the use of active learning for mining signal temporal logic (STL) requirements of cyber-physical systems (CPSs). CPS is a modeling paradigm in many safety-critical domains, such as automotive, medical, and aerospace industries, where the correctness of the end product is of significant importance [1]. However, in many industrial settings, the requirements intended to enforce the correctness guarantee are vague and in many cases expressed in natural languages, such as "smooth steering" and "good fuel efficiency." Further, due to the complex nature of many CPSs, writing down the appropriate requirements to reflect the desirable system properties can be challenging even for experienced designers.

Given a system S, e.g., a Stateflow/Simulink model of a steering system, and a requirement template φ with a set of unknown parameters θ, the goal of this paper is to develop an improved method which can automatically infer a requirement φ_θ written in signal temporal logic, i.e., the system S satisfies the requirement φ_θ. Such a problem can be called a *requirement mining* problem. In the past few years with the introduction of the concept of *robustness degree* [2], [3], the problem has received increased attention and has achieved significant progress. With the help of robustness degree, the requirement mining problem of a CPS can be converted to an optimization problem for the expected robustness [4]–[6]. Various techniques, such as particle swarm optimization [7], simulated annealing [5], Nelder-Mead [8], and stochastic gradient descent algorithm [6] can then be used to solve the optimization problem.

All three authors are with the Department of Mechanical and Aerospace Engineering, University of California, Davis. Z. Sabato is also affiliated with the Hyundai Center of Excellence in Vehicle Dynamic Systems & Control at UC Davis. Z. Kong is the corresponding author. (email: zdkong@ucdavis.edu)

Due to the complex nature of CPSs, the robustness degree function can be highly nonlinear. Furthermore, many CPSs are stochastic because of various uncertainties inherent to the system. These complexities added together make uniform sampling method inefficient to solve the optimization problem (there are even cases in which the exact probability distribution over the parameter space is unknown a priori). Monte-Carlo techniques have been shown to be an effective sampling method to tackle the issue [8], [9]. It is worth pointing out that these techniques may suffer from slow convergence, meaning that the inference procedure may take a long time. In many applications, a quick verdict is needed, consider for instance an online diagnosis of a faulty safety-critical system.

In this paper, we develop a new active learning algorithm called Gaussian Process Adaptive Confidence Bound (GP-ACB) and use it to partially mitigate the need for a large number of iterations during optimization. The idea behind active learning is to accelerate convergence by actively selecting potentially "informative" samples, in contrast with random sampling from a predefined distribution [10]. Our paper unifies two complementary camps of requirement mining and verification philosophies: one is model based [8], [9], [11], and the other is data driven [5], [12]. In our method, models are used as oracles, generating data which enables our method to gain knowledge of the system. This helps focus ongoing searches in promising parameter ranges, and thus eliminates unnecessary samples.

This paper is divided into the following sections. Section II discusses the relevant background on signal temporal logic and Gaussian processes. Section III formally defines the requirement mining problem. Section IV discusses our GP-ACB algorithm. Section V provides two case studies to demonstrate our algorithm. Section VI concludes the paper.

II. Preliminaries

A. 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 $s \in \mathcal{F}(\mathbb{R}^+,\mathbb{R}^n)$. We use $s(t)$ to denote the value of signal s at time t. *Signal temporal logic* (STL) [13] is a temporal logic defined over signals. STL is a predicate logic with interval-based temporal semantics. The syntax of STL is defined as

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

where a and b are non-negative finite real numbers, and $f(s) \sim d$ 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 $\neg$ and $\wedge$ are negation ("not") and conjunction ("and"), respectively. The other Boolean operators are defined as usual. The temporal operators F and G stand for "Finally (eventually)" and "Globally (always)", respectively.

STL is equipped with a *robustness degree* [2], [3] (also called "degree of satisfaction") that quantifies how well a given signal s satisfies a given formula φ. The robustness is calculated recursively as follows

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

We use $r(s,\varphi)$ to denote $r(s,\varphi,0)$. If $r(s,\varphi)$ is large and positive, then s would have to deviate substantially 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 [11]. We denote them as *scale* parameters $\pi = [\pi_1, ..., \pi_{n_\pi}]$ and *time* parameters $\tau = [\tau_1, ..., \tau_{n_\tau}]$, respectively. A full parameterization is given as $[\pi,\tau]$. The syntax and semantics of PSTL are the same as those of STL. A *valuation* θ is a mapping that assigns real values to the parameters appearing in an PSTL formula. A valuation θ of an PSTL formula φ induces an STL formula φ_θ. For example, if $\varphi = F_{[\tau_1,\tau_2)}(x<\pi_1)$ and $\theta([\pi_1,\tau_1,\tau_2]) = [0,0,3]$, then $\varphi_\theta = F_{[0,3)}(x<0)$.

B. Gaussian Processes

A Gaussian process (GP) is defined as a collection of random variables, any finite linear combination of which have a joint Gaussian distribution [14]. Any GP is completely specified by its mean function $m(\vec{x})$ and its covariance function or kernel $k(\vec{x},\vec{x'})$

$$\begin{aligned} &m(\vec{x}) = E[f(\vec{x})], \\ &k(\vec{x},\vec{x'}) = E[(f(\vec{x}) - m(\vec{x}))(f(\vec{x'}) - m(\vec{x'}))]. \end{aligned}$$

A flat (or even zero) mean function $m(\vec{x})$ 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 in not confined to zero. There is a large set of available kernels $k(\vec{x},\vec{x'})$. Two common ones are [14]

- Gaussian kernel with length-scale $l > 0, k(\vec{x}_1,\vec{x}_2) = \exp(-|\vec{x}_1 - \vec{x}_2|^2/(2l^2))$, where $|.|$ is the Euclidean length.
- Matérn kernel with length-scale $l > 0$

$$k(\vec{x}_1,\vec{x}_2) = \frac{2^{1-\nu}}{\Gamma(\nu)}\Big(\frac{\sqrt{2\nu}|\vec{x}_1-\vec{x}_2|}{l}\Big)^\nu K_\nu\Big(\frac{\sqrt{2\nu}|\vec{x}_1-\vec{x}_2|}{l}\Big)$$

where $\Gamma(\nu)$ is the Gamma function, K_ν is the modified Bessel function and ν is a positive parameter.

III. Problem Formulation

In this section, we first define the requirement mining problem. Then, we show how to formulate the requirement mining problem as a search for critical level sets.

A. Problem Statement

Notations from [9], [15] are adopted here. A system S maps an initial condition (or uncontrolled environmental conditions, e.g., road conditions) $\vec{x}_0 \in X_0 \subset \mathbb{R}^{n_x}$ to a discrete-time output signal $\vec{y} \in \mathcal{F}([0,T],Y)$ with $Y \subset \mathbb{R}^{n_y}$ and T as the finite maximal simulation time. We assume both X_0 and Y can be represented as the Cartesian product of intervals $[a_1,b_1]\times[a_2,b_2]\times\ldots[a_n,b_n]$, where $a_i,b_i \in \mathbb{R}$.

In this paper, we solve the following requirement mining problem with the help of robustness degree.

Problem 1. Given a system S with a set of sampled traces $\bar{Y} = \{\vec{y}^i \in Y \subset \mathbb{R}^{n_y}, i = 1,\cdots,n_s\}$ starting from $\bar{X}_0 = \{\vec{x}_0^i \in X_0 \subset \mathbb{R}^{n_x}, i = 1,\cdots,n_s\}$, where $\vec{y}^i = S(\vec{x}_0^i)$ and n_s is the number of traces, and a PSTL formula φ_θ with unknown parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, find a valuation set of θ to solve

$$\max_\theta(0, \epsilon - \min_{\vec{x}_0\in\bar{X}_0}(r(S(\vec{x}_0),\varphi_\theta))), \tag{1}$$

where $\epsilon > 0$ us a user-specified bound.

The max function $\max(0,\epsilon-\cdot)$ in Eqn. (1) is a modified hinge loss function. As the minimum of the robustness, $\min(r(S(\vec{x}_0),\varphi_\theta))$, is positive, the loss function rewards values that are close to the bound 0 and at the same time positive. It is utilized here to tackle the issue related to the non-uniqueness of solutions to the requirement mining problem, as pointed out in [8]. For a particular θ, the min function rewards initial states that lead to negative robustness degrees. The goal of the min function is to find an initial state which leads to a trace that does not meet the requirement φ_θ or simply to solve a falsification problem. The falsification problem can be solved by optimization as elaborated in [16]. The outcome of the falsification problem is the set of the sampled traces $\bar{Y}$. The max-min function in Eqn. (1) then finds parameters θ, and, in turn, requirements φ_θ such that for any initial state $x_0 \in X_0$, the output of the system has a positive robustness degree that is smaller than ϵ. This means that the system satisfies the requirements φ_θ, but only barely.

B. Solution as Critical Level Sets

To investigate the topology of a parameter space Θ, consider the following function $F : \Theta \to \mathbb{R}$, which is a scalar field:

$$F(\theta) = \min_{\vec{x}_0\in\bar{X}_0}(r(S(\vec{x}_0),\varphi_\theta)). \tag{2}$$

Denote the connected components of the parameter space Θ as $cc(\Theta)$. A *critical level set* of the function F in the parameter space Θ is a connected component ξ, which satisfies [17]

$$\xi \subset cc(\{\theta\in\Theta : F(\theta) = 0\}). \tag{3}$$

The set of all critical sets of F in the parameter space Θ can be denoted by $Cr(F, \Theta)$. Finally, the *topology induced partition* of the function F can be defined by [17]

$$M(F, \Theta) = cc(\Theta \setminus Cr(F, \Theta)). \tag{4}$$

An example of such a topology induced partition is shown in Fig. 1(a), where the white curves are the critical level sets.

With the definition of critical level sets, it is straightforward to conclude that solving Problem 1 is equivalent to finding the critical level sets of function F. Fig. 7 illustrates the scalar field of F with an automobile model and a PSTL formula $\varphi_\theta := G_{[0,\tau]}(a_{lat} < \pi)$, where a_{lat} is the vehicle's lateral acceleration. The critical level set, corresponding to the curve with zero robustness degree, divides the parameter space Θ into two partitions, one partition Θ^+ consists of all parameters with positive robustness degrees and the other partition Θ^- consists of all parameters with negative robustness degrees. In other words, the model satisfies the requirement φ_θ if $\theta \in \Theta^+$ and violates the requirement φ_θ if $\theta \in \Theta^-$. After the boundary of Θ^+ and Θ^- is found, the requirement mining problem is solved. We will show in next section how to find such boundaries or critical level sets with active learning.

IV. Active Learning Based Requirement Mining

In the section, we propose an active learning algorithm and show how to use it to solve the problem of finding critical level sets, as mentioned in Section III-B.

A. Gaussian Process Adaptive Confidence Bound Algorithm

Active learning algorithms were originally developed to solve classification problems when an oracle is needed to provide labels [10]. The process of obtaining labels from the oracle can be expensive in terms of both time and money. Thus, the goal of any active learning algorithm is to achieve high classification or regression accuracy by using the fewest labeled instances. The requirement mining problem is similarly constrained. Our oracle is a simulator, e.g., a Stateflow/Simulink model. Given the complexity of many CPS models, to obtain a trace from the simulator can be costly in time. Thus, we need to decrease the number of simulations needed to learn a formula.

One active learning algorithm is modified from Gaussian Process Upper Confidence Bound (GP-UCB) [18]. At each step t, it solves the following problem

$$\vec{x}_t = \underset{\vec{x} \in D}{\operatorname{argmax}}\, m_{t-1}(\vec{x}) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(\vec{x}), \tag{5}$$

where D is the search space, β_t is a function of t and independent of $\vec{x}$ (an example of β_t will be given later), $m_{t-1}(.)$ and $\sigma_{t-1}(.)$ are the mean and covariance functions of the Gaussian process, respectively, and $\vec{x}_t$ is the instance that will be inquired at step t, meaning the label of $\vec{x}_t$ will be obtained from the oracle.

The second term of Eqn. (5) only depends on the covariance function $\sigma(\vec{x})$, which can potentially make the exploration process random and inefficient. To address this problem, we propose an algorithm called Gaussian Process Adaptive Confidence Bound (GP-ACB) by adding a normalization term $\eta_m(x)$ to Eqn. (5) as follows:

$$\vec{x}_t = \underset{\vec{x} \in D}{\operatorname{argmax}}\, m_{t-1}(\vec{x}) + \eta_m(\vec{x})^{\frac{1}{2}} \beta_t^{\frac{1}{2}} \sigma_{t-1}(\vec{x}), \tag{6}$$

where $\eta_m(\vec{x})$ normalizes the mean $m_{t-1}(\vec{x})$ and can be written explicitly as

$$\eta_m(\vec{x}) = \frac{m_{t-1}(\vec{x}) - \min(m_{t-1}(\vec{x}))}{\max(m_{t-1}(\vec{x})) - \min(m_{t-1}(\vec{x}))}.$$

It is obvious that $0 \leq \eta_m(\vec{x}) \leq 1$. $\eta_m(\vec{x})$ acts as an adaptive factor to uncertainty (covariance) and favors exploration directions associated with increasing rewards. In this paper, we assume that the observation at time t, $y_t \in \mathbb{R}$, is $y_t = f(\vec{x}) + \varepsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ and σ^2 known. Pseudocode for the GP-ACB algorithm is provided in Algorithm 1.

Algorithm 1: GP-ACB Algorithm

Input:
Search space D; GP priors $m(\vec{x})_0 = 0$ and σ_0;
Kernel function k; Maximal simulation time T.

1: **for** $i = 1$ to T **do**
2: Bayesian update $m_{t-1}(\vec{x})$ and $\sigma_{t-1}(\vec{x})$;
3: Calculate the normalization factor $\eta_m(\vec{x})$;
4: Choose $\vec{x}_t = \operatorname{argmax}_{\vec{x} \in D}\, m_{t-1}(\vec{x}) + \eta_m(\vec{x})^{\frac{1}{2}} \beta_t^{\frac{1}{2}} \sigma_{t-1}(\vec{x})$;
5: Calculate $\vec{y}_t = f(\vec{x}_t) + \varepsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$.

B. Regret Bound of GP-ACB

The goal of any learning algorithm can be stated as follows: given an unknown reward function $f \in \mathcal{F}(D, \mathbb{R})$, maximize the sum of rewards $\sum_{t=1}^{T} f(\vec{x}_t)$, which is equivalent to finding a $\vec{x}^*$ such that $\vec{x}^* = \operatorname{argmax}_{\vec{x} \in D} f(\vec{x})$. A concept called regret bound can be used to quantify the convergence rate of a learning algorithm [10], [19]. First, the instantaneous regret at time t is defined as $r_t = f(\vec{x}^*) - f(\vec{x}_t)$. Then, the cumulative regret R_T after T rounds is $R_T = \sum_{t=1}^{T} r_t$. A desired property of the learning algorithm is then to guarantee $\lim_{T \to \infty} R_T/T = 0$, implying the convergence to the global maximum $\vec{x}^*$. Finally, the bounds on the average regret R_T/T are directly related to the convergence rate of the learning algorithm. The lower the bound is, the faster the algorithm converges. This section investigates the regret bound of the GP-ACB algorithm.

Define γ_T as the maximum information gain after T rounds as follows [19]:

$$\gamma_T = \max_{T' \leq T} \frac{1}{2} \sum_{t=1}^{T'} \log(1 + \sigma^{-2} \sigma_{t-1}^2(\vec{x}_t))$$

If the search space $D \in \mathbb{R}^d$ is compact and convex, where d is the dimension of the search space, we can get the following theorem (the proof is in the Appendix).

Theorem 1. *Let* $\delta \in (0,1)$, $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$, $m = \min_{t=(1,\cdots,T)}(\eta_m(\vec{x}_t))$ *and* $n = \max_{t=(1,\cdots,T)}(\eta_m(\vec{x}_t))$. *Running GP-ACB results in a regret bound as follows*

$$Pr\{R_T \leq \sqrt{nC_1T\beta_T\gamma_T}, \forall T \geq 1\} \geq 1 - \delta^m, \quad (7)$$

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

Remark 1. The regret bound of GP-UCB is [19]

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

With the same parameter setting, the regret bound of GP-ACB is shown as Eqn. (7). Since $0 < m, n \leq 1$, we can conclude that the GP-ACB algorithm can get the same regret bound more efficiently than the GP-UCB algorithm. The maximum $\max_{t\leq T} f(\vec{x}_t)$ in the first T iterations is no further from $f(\vec{x}^*)$, where $\vec{x}^*$ is the global optimum, than the average regret R_T/T. Thus, compared with GP-UCB, on average, the GP-ACB algorithm has a higher convergence rate.

C. Active Requirement Mining

Many optimization methods, such as particle swarm optimization [7], simulated annealing [5], Nelder-Mead [8], and stochastic gradient descent algorithm [6] are not suitable to solve Problem 1 due to the large number of simulations or objective function evaluations needed. To reduce the number of simulations, many researchers have shown the effectiveness of using response surfaces for the optimization of computationally expensive problems [20]. For algorithms based on finite samples, it should be noticed that there is an inherent trade-off between exploration and exploitation. In this paper, we use active learning to solve the exploration and exploitation trade-off in a principled manner: the first term in Eqn. (6) tends to pick points in the decision space that are expected to achieve high rewards (exploitation); and the second term in Eqn. (6) tends to pick points that favor uncertainty (exploration).

We solve Problem 1 by locating the critical level sets as elaborated in Section III-B in three steps:

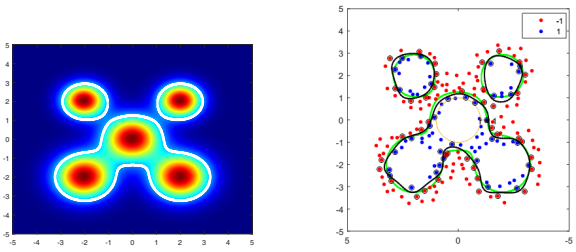
1) Generate N samples by using the GP-ACB algorithm with the reward function

$$\max_{\theta\in\Theta} -|F(\theta)|, \quad (8)$$

where $F(.)$ is defined in Eqn. (2). The output of the first step is a set of N data points $Si := \{(\theta_1, F(\theta_1)), \cdots, (\theta_N, F(\theta_N)) \in \mathbb{R}^{n_\theta} \times \mathbb{R}\}$.
2) Generate the training set by keeping the data points having robustness degrees whose absolute values are smaller than or equal to ϵ, i.e., $Si' = \{(\theta, F(\theta)) \in Si | -\varepsilon \leq F(\theta) \leq \varepsilon\}$.
3) Approximate the critical level sets by using the ϵ-SVR algorithm [21] and data set Si'.

In the following, we call the above procedure *active requirement mining*.

An example of using the active requirement mining to locate critical level sets is demonstrated in Fig.1. The robustness function F of the example is a Gaussian mixture function as shown in Fig. 1(a). In Fig. 1(b), samples generated by the GP-ACB algorithm are shown as red and blue dots. The critical level sets obtained by the ϵ-SVR algorithm are shown as black curves. It can be observed that the GP-ACB algorithm is able to sample points around critical level sets and it is possible to approximate the level sets by using the ϵ-SVR algorithm with only a few training data points.



(a) (b)

Fig. 1. An active requirement mining example. (a) A Gaussian mixture function with critical level sets shown in white. (b) Samples (red and blue dots) generated by the GP-ACB algorithm and the critical level sets (black curves) obtained by the ϵ-SVR algorithm.

V. Case Studies

A. Global Optimization of Ackley's Function

To verify the performance of the proposed GP-ACB algorithm, we compare the GP-ACB algorithm with four types of Gaussian-Process-based strategies: (i) GP-UCB active learning, (ii) Batch-greedy UCB active learning [22], (iii) pure exploration, i.e., choosing points of maximum variance at each step, and (iv) pure exploitation or greedy, i.e., choosing points of maximum mean at each step. We use Ackley's function as follows:

$$f(x,y) = -20e^{-0.2\sqrt{0.5(x^2+y^2)}} - e^{0.5(\cos(2\pi x)+\cos(2\pi y))} + e + 20$$

where e is the observation noise with zero mean and variance σ^2 at 0.025. The search space $D = [-5,5]^2$ is randomly discretized into 1000 points. We run each algorithm for $T = 58$ iterations with sampling time $\delta = 0.1$. Since the global minimum of the Ackley's function (x^*, y^*) is known (unknown to the learning algorithms though), for the i-th trial, if (x_t^i, y_t^i) is the solution obtained by running the algorithm for t iterations, then 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, we set $N_t = 1000$. Each trial is initialized randomly.

Fig. 2(a) and Fig. 2(b) show the comparison of the mean regret $\bar{R}_t$ incurred by the different Gaussian Process based algorithms with Gaussian kernel and Matérn kernel, respectively. With both kernels, GP-ACB outperforms the others. For instance, GP-UCB arrives at its minimum regret in an average of 58 iterations; while GP-ACB arrives at its minimum regret in an average of 45 iterations. Further, for this particular case, Matérn kernel outperforms the Gaussian kernel. This is not surprising, given that the Ackley's function

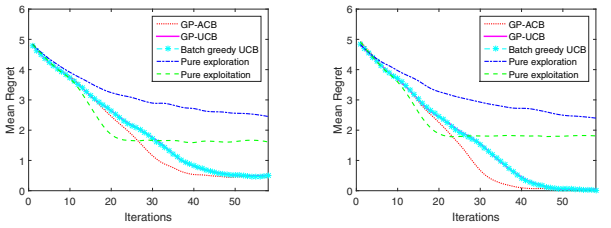


Fig. 2. Performances of GP-ACB and other strategies. The mean regret over 1000 trails $\bar{R}(t) := \bar{R}_t = \sum_{i=0}^{1000}[f(x_t^i, y_t^i) - f(x^*, y^*)]/1000$ is chosen as the performance metric. (a) shows the results with Gaussian kernel. (b) shows the results with Matérn kernel.

is quite "non-smooth" and Matérn kernel is designed to capture non-smoothness.

B. Active Requirement Mining for Automotive Power Steering

In this sub-section, the utility of our active requirement mining algorithm elaborated in Section IV-C is demonstrated in the context of a road vehicle power steering system.

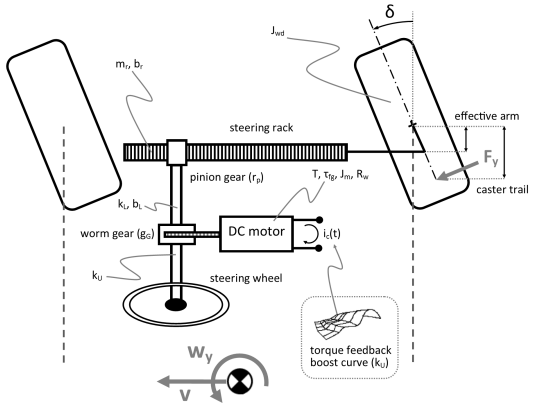


Fig. 3. A vehicle steering system with dynamics modeled for the steering rack, pinion gear, worm gear assembly, steering geometry (for small angular displacement), electronic power assist steering motor and feedback control algorithm. Of interest is the interaction between targets and selected design parameters. This work addresses upper column stiffness (k_U), EPAS motor winding centroidal mass moment of inertia (J_m) and the worm gear overall ratio (g_G).

1) Model Description: The system is a passenger automobile equipped with electric power assist steering (EPAS). The steering dynamics are emphasized in the system diagram of Fig. 3 and simulation model pictured in Fig. 4. The electric motor in the studied configuration is attached to the steering column and provides torque assistance using a basic "boost curve" control concept. The simple algorithm implemented maps measurements from steering wheel torque and wheel speed sensors to motor commands through a multi-dimensional lookup table. The model is available at `http://chpsslab.com/publications.html`.

Vehicle model properties include two degree-of-freedom chassis dynamics (lateral and yaw motions). Plant features which make traditional stability analysis difficult include

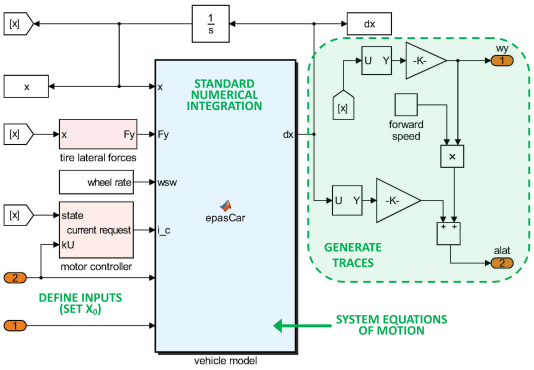


Fig. 4. MATLAB/Simulink^{TM} model of automobile with electric power-assist steering (EPAS). The blocks on the right side select/generate trajectories for requirement mining.

non-linearities owing to friction, tire force behavior and mechanism kinematics. Models for force generation of pneumatic tires range widely in their complexity, with the most complex empirically-derived examples requiring dozens of coefficients for a complete parameterization. Lateral tire forces on the front and rear axles (F_{yF}, F_{yR}) are in general generated using these highly nonlinear functions. However, for low lateral force values, a simple proportional relationship between slip angle and force matches measured data suitably. The open-loop step-steer test at $0.4g$ of lateral acceleration produces forces within the linear range.

The equations of motion for the system are as follows:

$$
\begin{aligned}
\dot{p}_V &= F_{yF} + F_{yR} - \tfrac{mu_0}{J_y} p_Y \\
\dot{p}_Y &= aF_{yF} - bF_{yR} - \tfrac{lb_L g_G}{r_p J_m} p_m - \tfrac{lk_L}{r_p} q_L + \\
&\quad \tfrac{l^2}{J_y}(b_r - \tfrac{b_L}{r_p{}^2}) p_Y + \tfrac{l^2}{J_{wd}}(\tfrac{b_L}{r_p{}^2} - b_r) p_{wd} \\
\dot{p}_{wd} &= \tfrac{lb_L g_G}{r_p J_m} p_m + \tfrac{lk_L}{r_p} q_L - \tfrac{l^2}{J_y}(b_r - \tfrac{b_L}{r_p{}^2}) p_Y - \\
&\quad \tfrac{l^2}{J_{wd}}(\tfrac{b_L}{r_p{}^2} - b_r) p_{wd} - c_a F_{yF} \\
\dot{q}_L &= \tfrac{g_G}{J_m} p_m - \tfrac{l}{r_p}(\tfrac{1}{J_{wd}} p_{wd} - \tfrac{1}{J_y} p_Y) \\
\dot{q}_U &= \omega_{sw} - \tfrac{g_G}{Jm} p_m \\
\dot{p}_m &= -\tau_{fg} + T i_c - \tfrac{b_L {g_G}^2}{J_m} p_m + \tfrac{b_L g_G}{r_p J_{wd}} p_{wd} - \\
&\quad \tfrac{b_L g_G}{r_p J_y} p_Y + g_G(k_U q_U - k_L q_L) \\
\dot{\delta} &= \tfrac{1}{J_{wd}} p_{wd} - \tfrac{1}{Jy} p_Y \\
\dot{\psi} &= \tfrac{1}{J_y} p_Y \\
\dot{X} &= u_0 \cos\psi - \tfrac{p_V}{m} \sin\psi \\
\dot{Y} &= u_0 \sin\psi + \tfrac{p_V}{m} \cos\psi
\end{aligned}
$$

where p_V is the lateral vehicle chassis momentum, p_Y is the vehicle chassis yaw angular momentum, p_{wd} is the wheel angular momentum in the diametric/steered direction, q_L is the angle of lower steering column, q_U is the angle of upper steering column, p_m is the angular momentum of EPAS motor windings, δ is the tire-steered angle, ψ is the inertial heading angle, X is the vehicle displacement in the inertial lateral direction, and Y is the vehicle displacement is inertial longitudinal direction. The model contains a single exogenous input, ω_{sw}, the angular velocity of the steering wheel. The inertial states (X, Y, ψ) can be used to evaluate fitness in the presence of requirements relating to overall

vehicle behavior, such as double lane change or obstacle avoidance maneuvers.

2) Test Conditions: The International Standards Organization (ISO) maintains a set of test methods used for studying, evaluating and reporting road vehicle dynamic performance. In this work we use ISO 7401, focusing on analyzing transient lateral response through the step-steer test. The maneuver uses an open-loop steering input: meaning once the input is calibrated to create the desired output, the driver (or driver model) has a small impact on the results. This has the advantage of isolating the vehicle and control system's response to parametric or environmental changes.

For the step steer test, 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 very quickly (hence step steer) to a value which yields a selected lateral acceleration in the steady state. Measurements made during this maneuver generate trajectories for lateral acceleration.

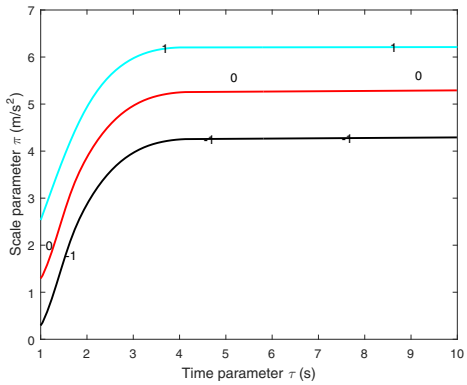


Fig. 5. Requirement mining result with $\varphi_\theta := G_{[0,\tau]}(a_{lat} < \pi)$. Sets with robustness degrees of 1, 0, and -1 are shown in blue, red and black, respectively.

3) Requirement Mining: We choose a PSTL formula as follows:

$$\varphi_\theta = \varphi_{(\tau,\pi)} := G_{[0,\tau]}(a_{lat} < \pi), \tag{9}$$

where a_{lat} is the vehicle lateral acceleration. The initial state space X_0 is chosen to be two dimensional, consisting of the gear ratio g_G and motor inertia J_m (it is worth pointing out that, for this particular example, a more appropriate interpretation of an initial state $\vec{x}_0 \in X_0$ is that it serves as an input to the model as shown in Fig. 5). The requirement mining result is shown in Fig. 5. By implementing the algorithm elaborated in Section IV-C, the critical level set, i.e., the set consisting of all parameters $\theta := (\tau, \pi)$ with $F(\theta) = 0$ (Eqn. (2)), is identified and shown in red in Fig. 6. The robustness degree of the model with respect to any φ_θ with θ on the identified critical level set is zero.

To compare the performance of GP-ACB and GP-UCB in the context of requirement mining, we run the active requirement mining algorithm described in Section IV-C in two different ways, one with ϵ-SVR together with GP-ACB and the other one with ϵ-SVR together with GP-ACB. The comparison result in shown in Fig. 6. It shows the

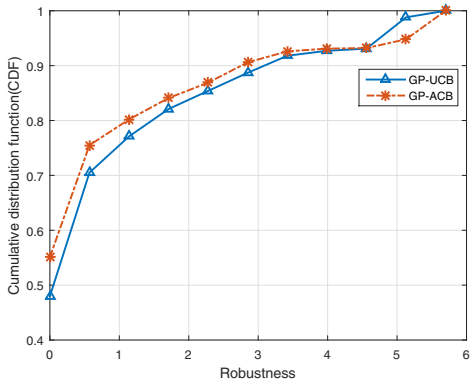


Fig. 6. The distributions of the absolute value of all the sampled points' robustness degree with the GP-ACB algorithm and the GP-UCB algorithm.

distributions of the absolute value of all the sampled points' robustness degree. For a method that can locate the critical level set quickly, it is expected that the sampled points should be concentrated around zero. Fig. 6 demonstrates that both the GP-ACB and GP-UCB algorithms can find the critical level set, since with both algorithms more than 80% of the sampled points' robustness are smaller than 20% of the maximum robustness. However, GP-ACB is superior to GP-UCB in the sense that it has more points concentrated around zero.

4) Sensitivity Analysis for Design Parameters: The knowledge of design parameters' effect on the performance of a system is crucial for the design process, especially when there exists trade-offs among multiple objectives. In the following, we demonstrate high-level parametric sensitivity analysis for a subset of steering system parameters and requirements written in the form of STL formulas. STL offers a way of specifying transient and steady-state responses. For instance,

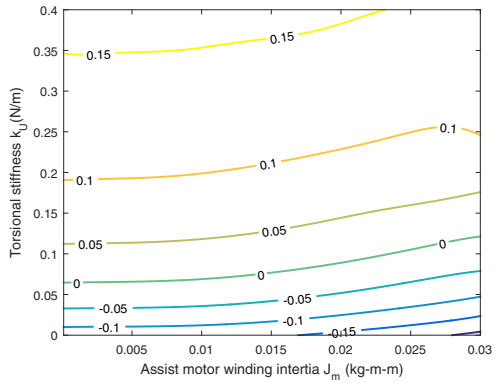
$$F_{[0,10]}(G_{[0,4.5]}(a_{lat} < 6) \wedge G_{[4.5,10]}(a_{lat} > 4.5)) \tag{10}$$

describes a requirement that is related to the settling time similar to those listed in ISO 7401. Then the robustness degree with respect to such a formula, especially how the robustness changes with multiple design parameters, offers an intuitive way of analyzing the optimal trade-off among parameters.

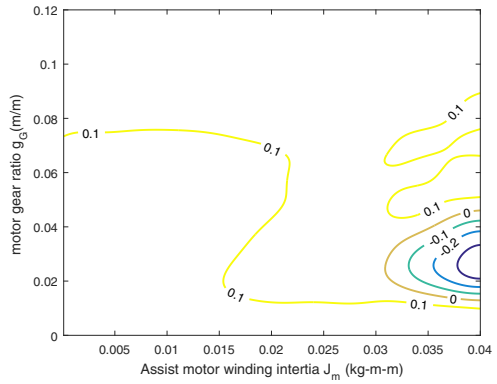
We focus on analyzing the sensitivity of the robustness degree with respect to design parameters. We first specify a PSTL formula

$$F_{[0,T]}(G_{[0,\tau]}(a_{lat} < \pi_1) \wedge G_{[\tau,T]}(a_{lat} > \pi_2)),$$

where T is the finite maximal simulation time and τ, π_1 and π_2 are parameters needed to be mined (estimated). We then mine the parameters using the active requirement mining algorithm described in Section IV-C. The end results is the STL formula (10). Finally, the same algorithm can also be used to generate the level sets of the power steering system's robustness degree with respect to different sets of design parameters as shown in Fig. 7.



(a)



(b)

Fig. 7. Level sets of the power steering system's robustness degree with respect to the STL formula $\varphi := F_{[0,10]}(G_{[0.4.5]}(a_{lat} < \pi_1) \wedge G_{[4.5,10]}(a_{lat} > \pi_2))$ and different sets of design parameters: (a) assist motor winding inertia J_m and torsional stiffness k_U; (b) J_m and motor gear ratio g_G.

Fig. 7(a) shows the level sets for torsional stiffness k_U and assist motor winding inertia J_m. The figure shows that k_U has a larger impact on the robustness than J_m. Further, the sensitivity of the robustness to J_m decreases as J_m increases. Additionally, for the lateral acceleration requirement with an open-loop driver input, these trends are consistent with intuition: added compliance of the column results in less tire-steered angle. Increased steering motor winding inertia creates delay in the controlled time response and contributes to unsatisfactory performance. Fig. 7(b) looks at the winding inertia and motor gearing ratio. The figure gives some insight into the limitations of the simple control system design made for this example. The gear ratio g_G is a critical design parameter, as it allows an affordable motor to generate significant torque on the column by operating at high speed. The results show that more aggressive gearing combined with changes in the winding inertia can lead to violations of the requirement. Changes can be made to the look-up table for boost to compensate for different parameters, but a better solution would be a truly robust compensator: one which rejects the effects of limited parametric drift. Through this simple example it can be seen how the proposed technique might be useful in model-based design and when addressing manufacturing tolerances during a run of production.

VI. Conclusion

In this paper, we introduced an active learning method, called Gaussian Process Adaptive Confidence Bound (GP-ACB), for mining requirements of bounded-time temporal properties of cyber-physical systems. The theoretical analysis of the proposed algorithm showed that it has a lower regret bound and thus a higher convergence rate compared to other Gaussian-Process-based active learning algorithms, such as GP-UCB. By using two case studies, one of which was an automatic power steering model, we showed that our requirement mining algorithm outperformed other existing algorithms, e.g., those based on GP-UCB. Our results have significant implications for not only the requirement mining, but also the validation and verification of cyber-physical systems. We are currently exploring the possibility of utilizing active learning to solve the structural inference problem, i.e., to mine a requirement without any given template.

Appendix

Our proofs on the regret bound follow those in [19].

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,*

$$|f(\vec{x}) - m_{t-1}(\vec{x})| \leq \eta_m(\vec{x})^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}), \forall \vec{x} \in D, \forall t \geq 1$$

holds with probability $\geq 1 - \delta^{\eta_m(\vec{x})}$.

Proof. For $\vec{x} \in D$ and $t \geq 1$. It is known that conditioned on $\vec{y}_{t-1} = (y_1, \cdots, y_{t-1})$, $\{\vec{x}_1, \cdots, \vec{x}_{t-1}\}$ is deterministic. Further, $f(\vec{x}) \sim \mathcal{N}(m_{t-1}(\vec{x}), \sigma^2_{t-1}(\vec{x}))$. Now if $r \sim N(0,1)$, then

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

for $c > 0$, as $e^{-c(r-c)} \leq 1$ for $r \geq c$. We have $Pr\{|f(\vec{x}) - m_{t-1}(\vec{x})| > \eta_m(\vec{x})^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x})\} \leq e^{-\eta_m(\vec{x})\beta_t/2}$. Set $r = (f(\vec{x}) - m_{t-1}(\vec{x}))/\sigma_{t-1}(\vec{x})$ and $c = \eta_m(\vec{x})^{1/2}\beta_t^{1/2}$. After applying the adaptive bound, we have

$$|f(\vec{x}) - m_{t-1}(\vec{x})| \leq \eta_m(\vec{x})^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}) \quad \forall \vec{x} \in D$$

holds with probability $\geq 1 - |D|e^{-\eta_m(\vec{x})\beta_t/2}$. Choosing $|D|e^{-\eta_m(\vec{x})\beta_t/2} = \delta/\pi_t$, e.g., with $\pi_t = \pi^2 t^2/6$, and using the adaptive bound for $t \in \mathbb{N}$, the statement holds.

Lemma 2. *Fix $t \geq 1$, if $|f(\vec{x}) - m_{t-1}(\vec{x})| \leq \eta_m(\vec{x})^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x})$, $\forall \vec{x} \in D$, then the regret r_t is bounded by $2\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t)$.*

Proof. According to the definition of $\vec{x}^*$, $m_{t-1}(\vec{x}_t) + \eta_m(\vec{x}_t)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t)$
$\geq m_{t-1}(\vec{x}^*) + \eta_m(\vec{x}^*)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}^*) \geq f(\vec{x}^*)$. Therefore, the instantaneous regret

$$\begin{aligned} r_t &= f(\vec{x}^*) - f(\vec{x}_t) \\ &\leq \eta_m(\vec{x}_t)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t) + m_{t-1}(\vec{x}_t) - f(\vec{x}_t) \\ &\leq 2\eta_m(\vec{x}_t)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t) \leq 2\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t) \end{aligned}$$

Lemma 3. *Pick $\delta \in (0,1)$ and set $\beta_t = 2\log(\pi_t/\delta)$, where $\sum_{t\geq 1}\pi_t^{-1} = 1, \pi_t > 0$. Then,*

$$|f(\vec{x}) - m_{t-1}(\vec{x})| \leq \eta_m(\vec{x})^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}) \quad \forall t \geq 1$$

holds with probability $\leq 1-\delta^{\eta_m(\vec{x}_t)}$.

Proof. For $\vec{x} \in D$ and $t \geq 1$. Conditioned on $\mathbf{y}_{t-1} = \{y_1, \cdots, y_{t-1}\}, \{\vec{x}_1, \cdots, \vec{x}_{t-1}\}$ is deterministic. Further, $f(\vec{x}_t) \sim \mathcal{N}(m_{t-1}(\vec{x}), \sigma_{t-1}^2(\vec{x}))$. According to Lemma 1, $Pr\{|f(\vec{x}_t) - m_{t-1}(\vec{x}_t)| > \eta_m(\vec{x}_t)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t)\} \leq e^{-\eta_m(\vec{x}_t)\beta_t/2}$. Since $e^{-\beta_t/2} = \delta/\pi_t$, and with the adaptive bound for $t \in \mathbb{N}$, the statement holds.

Lemma 4. *Set $L_t = max(m_t(\vec{x})) - min(m_t(\vec{x}))$, $\forall \vec{x} \in D$, and let β_t be defined as in Lemma 3, then*

$$1 - \eta_m(\vec{x}_t)^{1/2} \leq \beta_t^{1/2}\sigma_{t-1}(\vec{x}_t)/L_t \quad \forall t \geq 1$$

Proof. Set $m_{t-1}(\vec{x}^m) = max(m_{t-1}(\vec{x}))$, $\forall \vec{x} \in D$, according to GP-ACB, $\eta_m(\vec{x}_t)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t) + m_{t-1}(\vec{x}_t) \geq \eta_m(\vec{x}^m)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}^m) + m_{t-1}(\vec{x}^m)$, then

$$\begin{aligned}
&m_{t-1}(\vec{x}^m) - m_{t-1}(\vec{x}_{t-1}) \\
&\leq \eta_m(\vec{x}_t)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}_t) - \eta_m(\vec{x}^m)^{1/2}\beta_t^{1/2}\sigma_{t-1}(\vec{x}^m) \\
&\Rightarrow 1 - \eta_m(\vec{x}_t)^{1/2} \\
&\leq \tfrac{\beta_t^{1/2}}{L_t}(\eta_m(\vec{x}_t)^{1/2}\sigma_{t-1}(\vec{x}_t) - \eta_m(\vec{x}^m)^{1/2}\sigma_{t-1}(\vec{x}^m)) \\
&\leq \tfrac{\beta_t^{1/2}}{L_t}(\eta_m(\vec{x}_t)^{1/2}\sigma_{t-1}(\vec{x}_t) \leq \beta_t^{1/2}\sigma_{t-1}(\vec{x}_t)/L_t
\end{aligned}$$

The proof of Theorem 1 is as follows.

Proof. Define the information gain I as follows:

$$I(\mathbf{y}_T, \mathbf{f}_T) = \frac{1}{2}\sum_{t=1}^{T}\log(1+\sigma^{-2}\sigma_{t-1}^2(\vec{x}_t)),$$

where $\boldsymbol{f}_T = (f(\vec{x}_1), \ldots, f(\vec{x}_T))' \in \mathbb{R}^T$. According to Lemma 1 and Lemma 3, the regret bound $\{r_t^2 \leq 4\eta_m(\vec{x}_t)\beta_t\sigma_{t-1}^2(\vec{x}_t),\ \forall t \geq 1\}$ holds with probability $\geq 1-\delta^{\eta_m(\vec{x}_t)} \geq 1-\delta^m$. As β_t is non-decreasing, we have

$$\begin{aligned}
4\eta_m(\vec{x})\beta_t\sigma_{t-1}^2(\vec{x}_t) &\leq 4n\beta_T\sigma^2(\sigma^{-2}\sigma_{t-1}^2(\vec{x}_t) \\
&\leq 4n\beta_T\sigma^2 S\log(1+\sigma^{-2}\sigma_{t-1}^2(\vec{x}_t))
\end{aligned} \tag{11}$$

where $S = \sigma^{-2}/\log(1+\sigma^{-2})$, since $\sigma^{-2}\sigma_{t-1}^2(\vec{x}_t) \leq \sigma^{-2}k(\vec{x}_t, \vec{x}_t) \leq \sigma^{-2}$,
$C_1 = 8/\log(1+\sigma^{-2}) \geq 8\sigma^2$ and $h^2 \leq S\log(1+h^2) for \quad h \in [0, \sigma^{-2}]$. As $C_1 = 8\sigma^2 S$, for $T \geq 1$ we have

$$\begin{aligned}
&\textstyle\sum_{t=1}^T r_t^2 \leq \sum_{t=1}^T 4\eta_m(\vec{x})\beta_t\sigma_{t-1}^2(\vec{x}_t) \\
&\leq n\textstyle\sum_{t=1}^T \frac{1}{2}\beta_T C_1\log(1+\sigma^{-2}\sigma_{t-1}^2(\vec{x}_t)) \leq nC_1\beta_T\gamma_T.
\end{aligned}$$

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

REFERENCES

[1] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th Design Automation Conference*. ACM, 2010, pp. 731–736.

[2] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," *Formal Modeling and Analysis of Timed Systems*, pp. 92–106, 2010.

[3] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, 2009.

[4] E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti, "System design of stochastic models using robustness of temporal properties," *Theoretical Computer Science*, vol. 587, pp. 3–25, 2015.

[5] Z. Kong, A. Jones, A. Medina Ayala, E. Aydin Gol, and C. Belta, "Temporal logic inference for classification and prediction from data," in *Proceedings of the 17th international conference on Hybrid systems: computation and control*. ACM, 2014, pp. 273–282.

[6] Z. Kong, A. Jones, and C. Belta, "Temporal logics for learning and detection of anomalous behavior," *IEEE Transactions on Automatic Control*, accepted.

[7] I. Haghighi, A. Jones, Z. Kong, E. Bartocci, R. Gros, and C. Belta, "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*. ACM, 2015, pp. 189–198.

[8] X. Jin, A. Donzé, J. V. Deshmukh, and S. A. Seshia, "Mining requirements from closed-loop control models," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*. ACM, 2013, pp. 43–52.

[9] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivančić, and A. Gupta, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 2s, p. 95, 2013.

[10] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 11, pp. 55–56, 2010.

[11] E. Asarin, A. Donzé, O. Maler, and D. Nickovic, "Parametric identification of temporal properties," in *Runtime Verification*. Springer, 2012, pp. 147–160.

[12] A. Jones, Z. Kong, and C. Belta, "Anomaly detection in cyber-physical systems: A formal methods approach," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 848–853.

[13] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pp. 71–76, 2004.

[14] Rasmussen and C. Edward, *Gaussian processes for machine learning*. The MIT Press, 2006.

[15] S. Sankaranarayanan and G. Fainekos, "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*. ACM, 2012, pp. 125–134.

[16] G. Chen, Z. Sabato, and Z. Kong, "Active requirement mining of bounded-time temporal properties of cyber-physical systems," *arXiv preprint arXiv:1603.00814*, 2016.

[17] D. Baronov and J. Baillieul, "Decision making for rapid information acquisition in the reconnaissance of random fields," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 776–801, 2012.

[18] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *Information Theory, IEEE Transactions on*, vol. 58, no. 5, pp. 3250–3265, 2012.

[19] N. Srinivas, A. Krause, and S. Kakade, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv*, pp. 0912–3995, 2009.

[20] A. T. Krityakierne T and S. C. A, "Sop: parallel surrogate global optimization with pareto center selection for computationally expensive single objective problems," *Journal of Global Optimization*, pp. 1–21, 2015.

[21] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[22] K. A. Desautels T and B. J. W, "Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization," *The Journal of Machine Learning Research*, pp. 3873–3923, 2014.