This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Temporal Logic Inference for Fault Detection of Switched Systems With Gaussian Process Dynamics

Gang Chen, Peng Wei, *Member, IEEE*, and Mei Liu

***Abstract*— In this article, we present a method for constructing the fault detector in the form of signal temporal logic (STL) formulas, which can be understood by human users and formally proven to detect faults with probabilistic satisfaction guarantees, for a class of switched nonlinear systems with partially unknown dynamics. First, the partially unknown internal dynamics are approximated by the Gaussian process with stability guarantees. Second, a novel temporal logic inference algorithm is proposed to find the fault detector, which takes advantage of the internal properties of temporal logic and searches for the optimal formula along a partially ordered direction. Moreover, the algorithm is not allowed for missing faults but allowed for false alarms during the temporal logic inference process. In addition, we simulate finitely many trajectories with Chua's circuit and infer the temporal logic formulas with the Gaussian optimization. The results show that the proposed method can find a temporal logic formula to detect the faulty trajectory with a probability guarantee.**

***Note to Practitioners*—The method proposed in this article can be used to detect faults for switched systems with partially unknown dynamics. STL is used to describe the behaviors of the system, which acts as a classifier and detector, such that all normal behaviors of the system will satisfy the description, while the faulty behaviors will violate the description. Moreover, STL formulas can be understood by human operators, which is important for the timely response to faulty events. For example, the normal behavior of a smart grid can be described as follows: "if the smart grid is safe, it should reach 9 kV within 15 min when the voltage to region A is above 12 kV," which can be expressed with STL. Due to the unknown dynamics, the Gaussian process regression is applied to estimate the model and the region that is robust to noises.**

***Index Terms*— Fault detection, Gaussian process, partially ordered direction, signal temporal logic (STL), switched system, temporal logic inference.**

Manuscript received February 27, 2021; accepted March 30, 2021. This article was recommended for publication by Associate Editor B. Zhang and Editor C. Seatzu upon evaluation of the reviewers' comments. *(Corresponding author: Mei Liu.)*

Gang Chen is with the Department of Automation, School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: megangchen@gmail.com).

Peng Wei is with the Department of Mechanical and Aerospace Engineering, University of California at Davis, Davis, CA 95616 USA (e-mail: penwei@ucdavis.edu).

Mei Liu is with the Department of Automation, School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: liumeimei@tju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2021.3074548.

Digital Object Identifier 10.1109/TASE.2021.3074548

I. Introduction

In recent years, an increasing number of researchers pay attention to the study of fault diagnosis and prognosis for cyber–physical systems (CPSs) and have reported many impressive results [1]–[5]. CPSs are often modeled as switched nonlinear systems. These systems with partially unknown dynamics have played increasing roles in the operation of critical infrastructures, such as autonomous systems [6], [7] and power grids [8], [9]. Due to complex operational environments, these systems are vulnerable to external attacks or disruptions [10]. Faults often occur in these systems, weaken the system's performance, destroy the system stability, and cause catastrophe accidents. Moreover, the *interpretability* of the fault detection process is important to understand the operation status of the systems and take fault-recovery actions quickly when a potential fault. Therefore, accurately and timely detecting the faults with a human-understandable approach for these systems are critical tasks in practical applications.

Model-based fault detection for switched systems has been widely studied [11], [12]. The basic idea of these fault detection systems is to estimate a residual signal and, based on this signal, determine a residual evaluation function to compare it with a predefined threshold. When the residual estimation function has a value higher than the threshold, an alarm is triggered [13], [14]. However, almost all of the existing fault detection approaches are about switched linear systems [15], [16]. Up until now, very few results of fault detection for switched nonlinear systems with unknown dynamics have appeared [17]. One of the main reasons is that the linear matrix inequalities (LMIs) are usually no longer effective for nonlinear systems. In [18], the observer-based actuator fault detection of uncertain nonlinear systems was considered, in which the design of an observer is the key step in fault detection. In [19], the nonlinear switched system is first transferred into a Takagi–Sugeno fuzzy switched model by using fuzzy IF-THEN rules; then, the fault detector is designed based on the persistent dwell-time switching signal and the quasi-time-dependent Lyapunov function technique. However, the linear observers may not well estimate the states of nonlinear systems. To address the unknown dynamics of the systems, Tang and Zhao [17] used the radial basis function neural networks to estimate the unknown internal dynamics; then, based on the estimated dynamics, a switched nonlinear

1545-5955 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

observer is developed. However, no guarantee can be obtained from the neural network-based observer. Therefore, how to choose the form of the observer for switched nonlinear systems with unknown dynamics is very crucial.

On the other hand, data-driven methods do not depend extensively on the model of the system and can build black-box data models without prior knowledge of the systems, such as the neural network-based method in [17]. However, these black-box data models lack the ability to capture and interpret the system knowledge, which is important for operators to understand the operation status of the system and take fault-recovery actions promptly when a fault occurs [20]. One feasible solution would be using the logic-based method to increasing the interpretability of the fault detector. In recent years, there has been increasing interest in describing fault behaviors of CPS with temporal logic and using a temporal logic for monitoring tasks [10], [21], [22]. In these works, the temporal logic formulas are used as observers that learned offline with labeled data sets and used online to detect the faults. Since temporal logic formulas describe temporal patterns between events in a form close to humans' way of reasoning, they can be intelligible and easily acceptable by humans. Using temporal logic for fault detection and monitoring tasks needs to infer a temporal logic formula, which has attracted extensive attention among researchers. For example, Chen *et al.* [10] formulated the temporal logic inference problem as a Markov decision process and solved it with a reinforcement learning algorithm. Bombara *et al.* [23] introduced a decision-tree approach to infer temporal logic formulas for classification. However, these methods do not provide guarantees for the results, i.e., a feasible temporal logic formula to detect the fault cannot be guaranteed even if there exists one. Moreover, existing learning-based temporal logic inference ignores the models of the systems and cannot provide probability guarantees for the fault detection results.

A. Contributions and Advantages

In this article, we focus on inferring temporal logic to diagnose the faults of switched systems with partially unknown dynamics. The proposed logic inference algorithm searches for the optimal formula based on a partially ordered relation, which only allows false alarm but does not allow missing fault. Thus, we call the searching method a safe temporal logic inference. Safe temporal logic inference does not search optimal formula with reinforcement learning strategy as in [10] but using a partially ordered relation to guide the search. The contributions of this article are twofold as follows.

1) *Fault Detection for Switched Systems With Partial Unknown Dynamics With Probability Guarantee:* We modify the methods in [24] and [25] and extend previous works to allow temporal logic to be applied to monitoring tasks for switched systems with unknown dynamic and uncertainties. To capture the unknown dynamics and uncertainties, we use the Gaussian process that is robust to uncertainties to approximate the unknown dynamics and estimate the region of attraction (ROA) with probability guarantees, which will be used for temporal logic inference.
2) *Safe Temporal Logic Inference for Fault Detection:* We propose a novel temporal logic inference algorithm, which guarantees that the detector can be found if it exists. Moreover, instead of searching the optimal formula via brute force or REINFORCE, the proposed method finds the optimal formula via partially ordered direction; thus, the proposed temporal logic inference algorithm obeys a safe manner during the inference procedure. Particularly, the monitoring system will not miss fault but allows false alarm at any time even when the estimated model is not accurate but allows false alarms during logic inference procedure.

This article is organized as follows. Section II introduces the preliminary knowledge and assumptions made in this article. Section III provides the main theoretical results and the solutions for the problem. Section IV demonstrates the performance of the proposed method with Chua's circuit-switched system. Section V concludes this article.

II. Preliminaries and Problem Statement

We consider a nonlinear, closed-loop continuous-time switched system $\mathcal{S}$ with dynamics

$$\begin{aligned} \dot{x}(t) &= h_m(x(t)) = \underbrace{f_m(x(t))}_{\text{known model}} + \underbrace{g_m(x(t))}_{\text{unknown model}} \\ y(t) &= x(t) \end{aligned} \tag{1}$$

where $x(t) \in \mathcal{X}_m \subset \mathcal{X} \subset \mathbb{R}^q$ is the state at time t within a connected set $\mathcal{X}_m$, and $m \in \mathcal{M}$ is a mode among the mode set $\mathcal{M}$. Denote Ξ a subset of $\mathcal{M} \times \mathcal{M}$, which contains the switch event. If an event $d = (m, m') \in \Xi$ takes place, then the system switches from mode m to m'. The system dynamics consist of a set of known models $f_m(x) \subset \mathcal{F}$ and a set of unknown models $g_m(x) \subset \mathcal{G}$. The latter accounts for unknown dynamics and model uncertainties, where the uncertainties are assumed to be Gaussian noises in this article. $y(t)$ is the system output, which assumes that the states are fully observable.

A. Postfault Dynamic Model

There are two kinds of events in the definition of switched systems: external events and internal events. We only consider the switchings triggered by external events (faults, state-dependent switches, and so on). Specifically, we consider two types of common faults: *parameter faults*, that is, faults that manifest in passive or switching elements and *sensor faults*, that is, faults that cause the measured values in $y(t)$ to deviate from the actual values of $x(t)$ [26].

1) Parameter Faults: Generally, parameter faults are common in real applications, which manifest as additive deviations $\Delta h_m(x(t))$ from the nominal $h_m(x(t))$ in (1). Thus, the state dynamics in the faulted condition can be modeled as

$$\dot{x}(t) = h_m(x(t)) + \Delta h_m(x(t)). \tag{2}$$

With algebraic manipulation, we can rewrite (2) as the sum of (1) and product of time-varying scalar component fault

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

magnitude function $\eta_m(x(t))$ and the time-invariant vector fault signature P_i, that is

$$\dot{x}(t) = f_m(x(t)) + g_m(x(t)) + \eta_m(x(t))P_i \tag{3}$$

where $i = 1, 2, \ldots, I$ is the number of possible types of faults.

2) Sensor Faults: Sensor faults manifest as affine deviations in the output readout values, which can be rewritten as the sum of the nominal sensor reading and the product of a scalar sensor fault magnitude function $\zeta_j(x(t))$ and a vector sensor fault signature Q_j That is, the output readout map in the faulted condition can be modeled as

$$y(t) = x(t) + \zeta_j(x(t))Q_j \tag{4}$$

where $j = 1, 2, \ldots, J$ and J is the number of possible types of sensors faults.

During the temporal logic inference process, we need to learn partially unknown dynamics. This needs further assumptions to restrict the type of the models. Since we want to estimate the model with uncertainties, we assume that the unknown model $g_m(\cdot)$ has low *complexity*, as measured under the norm of a reproducing kernel Hilbert space (RKHS) [27]. This assumption enables us to estimate the ROA that is relevant to the inference process with the Gaussian process for the exploration analysis [28], and the assumption can be defined formally as follows.

Assumption 1 (Well-Calibrated Model): Let $\mu_{m,n}(\cdot)$ and $\Sigma_{m,n}(\cdot)$ denote the posterior mean and covariance matrix functions of the statistic model of the dynamics (1) conditioned on n noisy measurements under mode m. With $\sigma_{m,n}(\cdot) = \text{trace}(\Sigma_{m,n}(\cdot))^{1/2}$, there exists a $\beta_{m,n} > 0$ such that, with probability at least $(1-\delta)$, it holds that, for all $n \geq 0$ and $x \in \mathcal{X}_m$, $||h_m(x) - \mu_{m,n}||_1 \leq \beta_{m,n}^{1/2}\sigma_{m,n}(x)$ for $m \in \mathcal{M}$.

This assumption ensures that we can estimate the model with bounded confidence intervals with a given probability. In the following, we assume that the system is stable in normal states, and there exists a Lyapunov function among the ROA, in which the system can be described with this Lyapunov function. We assume the following.

Assumption 2 (Lyapunov Function): A fixed and twice continuously differentiable Lyapunov function $V_m(x)$ is given for $m \in \mathcal{M}$. Moreover, there exists a constant γ such that $((\partial V_m(x))/\partial x)(f_m(x) + g_m(x)) < 0$ for all $x \in \mathcal{X}$, where $\mathcal{V}_m(\gamma) = \{x \in \mathcal{X}_m | V_m(x) < \gamma\}$.

Assumption 2 implies that the system is asymptotically stable within a region defined by $\mathcal{V}_m(\gamma)$, which indicates that the states of the system will be bounded under normal modes.

B. Signal, Trace, and Trajectory

Given a time domain $\mathbb{N} := 0, 1, \ldots,$ a discrete-time, continuous-valued *signal* is a function $s \in \mathcal{F}(\mathbb{N}, \mathbb{R}^n)$, where $\mathcal{F}(\mathbb{N}, \mathbb{R}^n)$ denotes the set of all functions from $\mathbb{N}$ to $\mathbb{R}^n$. Here, we use $s(t)$ to denote the value of signal s at time t and $s[t]$ to denote the suffix of signal s from time t, i.e., $s[t] = \{s(\tau)|T \geq \tau \geq t\}$, where T is the duration of the signal. In this article, the signal $s(t)$ denotes the observed states of the system $\mathcal{S}$ at time t, and we assume that the states of the system are fully observable. Now, we define the trace of system $\mathcal{S}$ with respect to the signal $s[0]$.

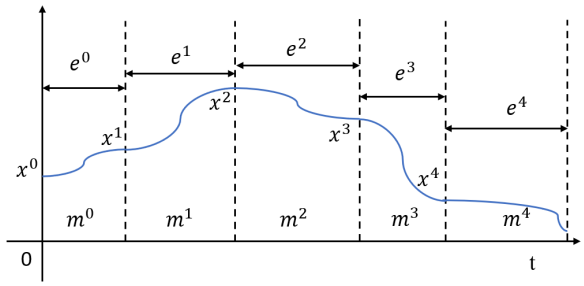

Fig. 1. Illustration of a trajectory of the system $\mathcal{S}$.

Definition 1 (Trace): A trace of the system $\mathcal{S}$ is a labeled signal denoted as $\xi = \{(m(t), s(t))|0 \leq t \leq T\}$, where $m(t)$ is the mode of the system at time t, $s(t)$ is the state of the system at time t, and T is the duration of the signal.

Here, we use $\omega(\xi, t) = s(t)$ to map a trace to a signal and $\omega(\xi)$ to denote $s[0]$. The signal s combines all system states of the trace. The trace of the system does not indicate the event switching time, while it is important for fault detection. We assume that there is a minimum dwell time for each event and define the event trajectory as follows.

Definition 2 (Trajectory): A trajectory of the event of the system $\mathcal{S}$ is denoted as $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^{Q}$, where m^i is the ith mode of the trajectory for the system, e^i is the dwell time of model m^i, Q is the total number of modes, and the following holds.

1) $\forall i \in \mathbb{N}$, s.t. $0 \leq i \leq Q$: $x^i \in \mathcal{X}$ is the initial state at mode $m^i \in \mathcal{M}$. The state of the system is governed by the dynamics defined in (1) with initial state x^i.
2) $\forall i \in \mathbb{N}$, s.t. $0 \leq i \leq Q$: $e^i \geq e_{\min}$, where $e_{\min}$ is the minimal dwell time, and the event occurring time for mode m^i can be calculated as $\sum_{l=0}^{i-1} e^l$ for $i \neq 0$ and zero for $i = 0$.
3) $\forall i \in \mathbb{N}$, s.t. $0 \leq i < Q$: $(m^i, m^{i+1}) \in \Xi$.

A trajectory is a sequence of mode, state, and duration. Here, we use $\alpha(\xi) = s_\xi$ to map a trace to a trajectory. Given an estimation of the unknown function $g_m(\cdot)$ with the Gaussian regression model $\hat{g}_m(\cdot)$, the system dynamics is assumed to admit a unique global solution $\varpi_{m^i}(x^i, \tau)$, where ϖ_{m^i} satisfies $((\partial \varpi_{m^i}(x^i, \tau))/\partial t) = f_{m^i}(x(t)) + \hat{g}_{m^i}(x(t))$, and $\varpi_{m^i}(x^i, 0) = x^i$. Namely, we use $\varpi_{m^i}(x^i, \tau)$ to denote the state of the system for a simulated signal s at time $t = \tau + \sum_{j=0}^{i-1} e^j$, $\tau \leq e^i$ in mode m^i starting from initial state x^i. $\varpi_{m^i}(x^i, \tau)$ is also called the continuous flow of the dynamic system $\mathcal{S}$. We use $\varpi(s_\xi, t)$ to denote the simulated signal based on trajectory s_ξ at time t and use $\varpi(s_\xi)$ to denote the whole signal.

Example 1: Fig. 1 gives an example of a trajectory of the system with $Q = 4$. The dwell time for each mode are e^0, e^1, e^2, e^3, and e^4. The system changes its mode at initial state x^0, x^1, x^2, x^3, and x^4, respectively. The trajectory of the system can be denoted as $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^{4}$.

C. Signal Temporal Logic

In this section, we introduce the concept of signal temporal logic (STL) [29] and its quantitative semantic.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Definition 3: STL is a temporal logic defined over signals. Its syntax is defined recursively as

$$\varphi ::= \top | \mu | \neg\varphi | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \Diamond_{\mathcal{I}}\varphi | \Box_{\mathcal{I}}\varphi \tag{5}$$

where $\top$ stands for the Boolean constant true, and μ is a predicate over a signal, which can be defined as $l(s(t)) \sim c$ with $l \in \mathcal{F}(\mathbb{R}^n, \mathbb{R})$ being a function, $\sim \in \{\leq, \geq\}$, and $c \in \mathbb{R}$ being a constant. The Boolean operators $\neg$, $\vee$, and $\wedge$ are negation ("not"), disjunction ("or"), and conjunction ("and"), respectively. The temporal operators $\Diamond$ and $\Box$ stand for "Eventually" and "Always," respectively. $\mathcal{I}$ is a time interval of the form $\mathcal{I} = [a, b]$, where a and b are nonnegative finite real numbers.

STL is equipped with a quantitative semantics called *robustness degree* $\rho : \Psi \times \mathcal{F}(\mathbb{R}^+, \mathbb{R}^n) \to \mathbb{R}$, which maps an STL formula $\varphi \in \Psi$ and a signal $s \in \mathcal{F}(\mathbb{R}^+, \mathbb{R}^n)$ to a real number. $\rho(\varphi, s)$ indicates how far a signal s is away from satisfying STL formula φ and is defined in [29]. The robustness is sound, meaning that $\rho(\varphi, s, t) > 0$ implies that signal s satisfies φ at time t, denoted as $s[t] \models \varphi$, and $\rho(\varphi, s, t) < 0$ implies that signal s violates φ at time t, denoted as $s[t] \not\models \varphi$. In the rest of this article, we denote the robustness of specification φ at time 0 with respect to signal s by $\rho(\varphi, s)$ for short.

D. Fault Detection With Signal Temporal Logic

In this article, we define the behaviors $\mathcal{B}$ of a switched system $\mathcal{S}$ as the collection of all possible traces of $\mathcal{S}$. We define the set of normal modes as $\mathcal{M}_N$ and the set of faulty modes as $\mathcal{M}_F$, respectively. In addition, we have $\mathcal{M} = \mathcal{M}_N \cup \mathcal{M}_F$. A trace $\xi = \{(m(t), s(t)) | 0 \leq t \leq T\} \in \mathcal{B}$ is normal if and only if $\forall i$, $m^i \in \mathcal{M}_N$ and $m^i \in \alpha(\xi)$. Similarly, a trace is faulty if $\exists i, m^i \in \mathcal{M}_F$ and $m^i \in \alpha(\xi)$. Here, we denote the set of all normal behaviors as $\mathcal{B}_N$ and all faulty behaviors as $\mathcal{B}_F$, respectively. We say the fault of system is detectable if $\mathcal{B}_N \cap \mathcal{B}_F = \emptyset$. However, a fault is detectable does not indicate that it can be detected by an STL formula unless it is STL-detectable. An STL formula φ defines a language, which defines a set of trajectories as follows.

Definition 4: Given an STL formula φ, and $\sigma \in \mathbb{R}$, the σ-language of φ of system $\mathcal{S}$ is defined as the following set:

$$\mathcal{L}(\varphi, \sigma) := \{\xi \in \mathcal{B} | \rho(\varphi, \omega(\xi)) \geq \sigma\}. \tag{6}$$

If $\sigma_1 \geq \sigma_2$, then $\mathcal{L}(\varphi, \sigma_1) \subseteq \mathcal{L}(\varphi, \sigma_2)$. In a Boolean sense, if a trace $\xi \in \mathcal{B}_N$, then there exists a signal s with respect to the trace ξ, and s satisfies the formula, denoted as $s[0] = \omega(\xi) \models \varphi$, i.e., $\omega(\xi) \in \mathcal{L}(\varphi, 0)$ and $s[0] \in \mathcal{L}(\varphi, 0)$. In the rest of this article, we use $\mathcal{L}(\varphi)$ to denote $\mathcal{L}(\varphi, 0)$ for short. Given an STL formula φ and a signal s with length L, $\mathcal{L}(\varphi) \subset \mathbb{R}^{n(L+1)}$, which n is the dimension of the signal. With the rectangular predicates, the bounded-time language becomes a finite union of hyperrectangles. The formula φ then can be regarded as an external observation map, which maps the behaviors of the switched system $\mathcal{S}$ to the space of the language defined by φ. Therefore, if all normal behaviors of the system satisfy formula φ, we have $\mathcal{B}_N \subset \mathcal{L}(\varphi)$. In contrast, if all faulty behaviors of the system violate φ, we have $\mathcal{B}_F \subset \mathcal{L}(\neg\varphi)$. Based on this observation, we can define the concept of STL-detectability as follows.

Definition 5 (STL-Detectable System): A system $\mathcal{S}$ is STL-detectable if and only if there exists an STL formula φ such that: 1) $\forall \xi \in \mathcal{B}_N, \omega(\xi) \models \varphi$ and 2) $\forall \xi \in \mathcal{B}_F$, $\omega(\xi) \not\models \varphi$. Intuitively, an STL-detectable system indicates that faulty and normal behaviors can be classified with an STL formula. Due to the properties of the rectangular predicates used in the STL formula, a system that is STL-detectable means that the normal behaviors are among a polytope, which is a finite union of hyperrectangles, and the faulty behaviors are outside the polytope. The STL-detectable system requires that there exists a hyperplane that can separate the faulty and normal behaviors. However, in many practical systems, due to the existence of noises and uncertainties, it is almost impossible for us to find a hyperplane that can perfectly classify the two kinds of behaviors. In these cases, we hope to find an STL formula that can classify the behaviors correctly with a given probability. We define (σ, δ)-diagnosable to address this issue as follows.

Definition 6 ((σ, δ)-Diagnosable): Given an STL formula φ, which introduces a language space $\mathcal{L}(\varphi)$, a set of normal behaviors $\mathcal{B}_N$, a set of faulty behaviors $\mathcal{B}_F$, two real numbers $\sigma \in \mathbb{R}^+$ and $\delta \in (0, 1)$, and a metric d_φ among the language space, the fault is (σ, δ)-diagnosable if, $\forall \xi \in \mathcal{B}_N$, $\forall \hat{\xi} \in \mathcal{B}_F$, there exists an STL formula such that

$$d_\varphi\big(\omega(\xi, t), \omega\big(\hat{\xi}, t\big)\big) \geq \sigma \tag{7}$$

holds with probability at least $(1 - \delta)$.

Remark 1: The idea of (σ, δ)-diagnosable borrows from the concept of (δ_d, δ_m)-diagnosable in [30], which shows a system is (δ_d, δ_m)-diagnosable if any fault can be detected δ_d time units after its occurrence. δ_m is the observation accuracy of the time intervals with respect to a specific metric. This concept is further generalized to (δ_d, ε)-diagnosable to allow continue time trajectories in [24]. In this article, the concept is extended to allow uncertainties among the system. Thus, the fault detection results allow probabilistic satisfaction.

We can build a switched estimator of the system from (1) as follows:

$$\begin{aligned} \dot{z}(t) &= f_m(z(t)) + g_m(z(t)) \\ r(t) &= d_\varphi(e(t)) \end{aligned} \tag{8}$$

where $z(t)$ is an estimation of the state vector $x(t)$ from (1), $e(t) = y(t) - z(t)$, and $r(t)$ is the residual of the difference between the measurement output $y(t)$ and the estimated output $z(t)$ under metric d_φ. Fault detection is achieved by monitoring the value of $r(t)$ at each time step and comparing it with a predefined *fault-detection threshold* γ. When $||r(t)||_1 > \gamma$, the algorithm detects a fault.

To detect the fault, we need a metric d_φ to measure the distance between two signals and get the residual values, which is defined as

$$d_\varphi(s, \hat{s}) = |\rho(\varphi, s) - \rho(\varphi, \hat{s})| \tag{9}$$

where s and $\hat{s}$ are signals with respect to traces ξ and $\hat{\xi}$, respectively. φ is an STL formula, and $|\cdot|$ is the absolution operator.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Lemma 1: The function d_φ defined by (9) is a semimetric on $\mathcal{L}(\varphi)$.

Proof: A semimetric function should have three conditions: 1) $\xi_1 = \xi_2 \Rightarrow d_\varphi(s_1, s_2) = 0$; 2) $d_\varphi(s_1, s_2) = d_\varphi(s_2, s_1)$; and 3) $d_\varphi(s_1, s_2) \leq d_\varphi(s_1, s_3) + d_\varphi(s_3, s_2)$ [31]. It is obvious that, when $d_\varphi(s, \hat{s}) = |\rho(\varphi, s) - \rho(\varphi, \hat{s})|$, the three conditions hold. Thus, the lemma has been proven. □

Lemma 1 shows that the distance between two signals can be calculated with the robustness degree of the signals with response to an STL formula. Based on Definition 6, the fault is detectable if the minimum distance between faulty signals and normal signals is larger than σ.

III. Gaussian Process With Temporal Logic

A. Fault Detection With Gaussian Process

In this article, we try to find an STL formula φ such that the associated metric d_φ used in (8) can detect the fault with bounded error and bounded probability. In order to find the admissible formula, we need to infer the structure and the associated parameters safely such that missing fault can be avoided. However, there are uncertainties among the model in (1); we cannot infer the formula safely directly from the model. Here, we use a set of simulated trajectories of the system to approximate the behaviors and find the admissible formula based on the approximation of the system. We denote $\hat{\mathcal{B}} = \{\xi^1, \xi^2, \ldots, \xi^N\}$ as the set of traces that approximate $\mathcal{B}$, where $\alpha(\xi^k) = \{(m^{i,k}, x^{i,k}, e^{i,k})\}_{i=0}^{Q}$ and $k \in \{1, 2, \ldots, N\}$. Similarly, we denote $\hat{\mathcal{B}}_N$ and $\hat{\mathcal{B}}_F$ as the sets of trajectories that approximate $\mathcal{B}_N$ and $\mathcal{B}_F$, respectively. Using the idea of *Approximate Bisimulation*, the behavior $\mathcal{B}$ can be approximated with $\hat{\mathcal{B}}$ [32]. This bisimulation relation can be defined with the bisimulation function.

Definition 7 (See [33]): For each mode $m \in \mathcal{M}$, a continuously differentiable function $\mathcal{A}_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}_{\geq}$ is defined as a bisimulation function if

$$\mathcal{A}_m(x, \hat{x}) \geq 0 \quad \forall x, \hat{x} \in \mathcal{X}_m,$$
$$\frac{\partial \mathcal{A}_m(x, \hat{x})}{\partial x} h_m(x) + \frac{\partial \mathcal{A}_m(x, \hat{x})}{\partial \hat{x}} h_m(\hat{x}) \leq 0. \quad (10)$$

Note that the bisimulation function is nonincreasing with respect to the flow. The following proposition can describe this property formally.

Proposition 1 (See [33]): For $\forall x, \hat{x} \in \mathcal{X}_m, m \in \mathcal{M}$, the bisimulation function evaluated along the flows of initial conditions x^0 and $\hat{x}^0$ is nonincreasing, i.e., for any $t_2 \geq t_1 \geq 0$, it is $\mathcal{A}_m(\varpi_m(x^0, t_1), \varpi_m(\hat{x}^0, t_1)) \geq \mathcal{A}_m(\varpi_m(x^0, t_2), \varpi_m(\hat{x}^0, t_2))$.

Based on Assumption 2, there exists a Lyapunov function for each mode. If we denote the Lyapunov function as $V_m(x) = [x^T M_m x]^{1/2}$, where M_m is a positive matrix; then, according to the above definition, we can construct a bisimulation function based on the following condition.

Lemma 2: Given the system described by (1), $\mathcal{A}_m(x, \hat{x}) = V_m(x - \hat{x}) = [(x - \hat{x})^T M_m (x - \hat{x})]^{1/2}$ is a bisimulation function if, for any $x, \hat{x} \in \mathcal{X}_m$

$$M_m \geq 0, (x - \hat{x})^T \big(M_m^T + M_m\big)(h_m(x) - h_m(\hat{x})) \leq 0. \quad (11)$$

Proof: If $M_m \geq 0$, then, $\forall x, \hat{x} \in \mathcal{X}_m$, we have $\mathcal{A}_m(x, \hat{x}) = [(x - \hat{x})^T M_m (x - \hat{x})]^{1/2} \geq 0$. If

$$(x - \hat{x})^T \big(M_m^T + M_m\big)(h_m(x) - h_m(\hat{x})) \leq 0 \quad (12)$$

it follows that, $\forall x, \hat{x} \in \mathcal{X}_m$:

$$\frac{\partial \mathcal{A}_m(x, \hat{x})}{\partial x} h_m(x) + \frac{\partial \mathcal{A}_m(x, \hat{x})}{\partial \hat{x}} h_m(\hat{x}) = \frac{(x - \hat{x})^T \big(M_m^T + M_m\big)(h_m(x) - h_m(\hat{x}))}{2\big[(x - \hat{x})^T M_m (x - \hat{x})\big]^{1/2}} \leq 0. \quad (13)$$

Therefore, $\mathcal{A}_m(x, \hat{x}) = [(x - \hat{x})^T M_m (x - \hat{x})]^{1/2}$ is a bisimulation function of the system described by (1). □

Here, we define $B_m(x, \gamma) \triangleq \{\hat{x} | V_m(x - \hat{x}) < \gamma\}$ as the (spatial) robust neighborhood of x, and γ denotes the (spatial) robustness radius. Based on Proposition 1, the trajectories starting with an initial state among the neighborhood of x will stay inside the neighborhood. The property means that the trajectories starting from a neighborhood will share similar properties. However, since $g_m(x)$ is unknown, we cannot check whether the condition in (11) holds. Moreover, in many cases, (11) does not hold for all $x \in \mathcal{X}_m$ but a subset of $\mathcal{X}_m$. Therefore, finding the maximum neighborhood radius is an issue and should be addressed when using this neighborhood concept. The following lemma and theorem provide the conditions of the bisimulation function for the system in (1).

Lemma 3 ([34, Lemma 5]): Let $\mathcal{X}_\tau \subset \mathcal{X}_m$ be a discretization of $\mathcal{X}_m$ with $|x - [x]_\tau| \leq \tau/2$ for all $x \in \mathcal{X}_m$, where $[x]_\tau$ denotes the closest point in $\mathcal{X}_\tau$ to $x \in \mathcal{X}_m$. Choosing $\beta_{m,n}$ according to [34, Lemma 1], the following holds with probability at least $(1 - \delta)$ for all $x \in \mathcal{X}_m$ and all $n > 1$:

$$|\dot{V}_m(x) - \mu_{\dot{V}_m, n-1}([x]_\tau)| \leq \beta_{m,n}^{1/2} \sigma_{\dot{V}_m, n-1}([x]_\tau) + L\tau \quad (14)$$

where L is a Lipschitz constant number in [34], and

$$\mu_{\dot{V}_m, n}(x) = \frac{\partial V_m(x)}{\partial x}\big(\mu_{m,n}(x) + f_m(x)\big)$$
$$\sigma_{\dot{V}_m, n}(x) = \left|\frac{\partial V_m(x)}{\partial x}\right| \sigma_{m,n}(x). \quad (15)$$

$\mu_{m,n}(x)$ and $\sigma_{m,n}(x)$ are the GP predictions of the unknown model $g_m(x)$, which can be obtained based on the measurement of $g_m(x)$.

Lemma 3 provides a probability bound on $\dot{V}$ in the continuous domain $\mathcal{X}_m$ by using the GP confidence intervals (15) on the discrete set $\mathcal{X}_\tau$. With this result, we have the following theorem.

Theorem 1: Consider a nominal trajectory $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^{Q}$ of a switched system $\mathcal{S}$ in (1) and a discretization of $\mathcal{X}_{m^i}$; $\mathcal{A}_{m^i}$ is a bisimulation function, and the switched system is asymptotically stable for all $y \in B_{m^i}(x^i, \gamma^i) \cap \mathcal{X}_\tau$ with probability at least $(1 - \delta)$ if

$$\mu_{\dot{V}_{m^i}, n-1}(y) \leq -\beta_{m^i,n}^{1/2} \sigma_{\dot{V}_{m^i}, n-1}(y) - L\tau \quad (16)$$
$$AB - A\mu_{m^i, n-1} \leq -||A||_1 \Big(\beta_{m^i,n}^{1/2} \sigma_{m^i, n-1}(y) + L\tau\Big) \quad (17)$$

where $A = (x^i - y)^T (M_{m^i}^T + M_{m^i})$, $B = h_{m^i}(x^i) - f_{m^i}(y)$, and $||\cdot||_1$ is the one-norm of a matrix.

Proof: See the Appendix. □

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Theorem 1 provides a way to calculate the radius for a switched system with unknown dynamics around a trajectory. Algorithm 1 shows the detail of robustness radius calculation. In Algorithm 1, we are given the discretization of the state space $\mathcal{X}_\tau$, GP prior $k(x, x')$, initial robustness radius γ_0 (usually very small), sample limit N for model estimation, and a trajectory $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^{Q}$. Line 3 finds the maximum robustness radius based on the current estimation of the model. Line 5 samples a new state that maximizes the variance, which samples the most uncertain state, and line 6 updates the estimation with newly sampled data. Since inequality (17) is a conservative version of the condition for bisimulation, the proposed algorithm will lead to a conservative estimation for the robustness radius.

Algorithm 1 Neighborhood Exploration

Input: Domain $\mathcal{X}_m$ and discretization with τ, $\mathcal{X}_\tau$, GP prior $k(x, x')$, initial robustness radius γ_0, sample limit N, initial neighborhood $\mathcal{N}_0$ and a trajectory $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^{Q}$ of system $\mathcal{S}$.

Output: A sequence of robustness radius $\{\gamma_n^i\}_{i=0}^{Q}$.

1: **for** $i = 1, \ldots, Q$ **do**
2: **for** $n = 1, \ldots, N$ **do**
3: $\gamma_n^i \leftarrow \text{argmax}_{\gamma>0}\, \gamma$, subject to (16) and (17) for all $x \in \mathcal{X}_\tau \cap B_m(x^i, \gamma)$;
4: $\mathcal{N}_n \leftarrow \mathcal{N}_0 \cup B_m(x^i, \gamma)$
5: $x_n \leftarrow \text{argmax}_{x \in \mathcal{N}_n} \sigma_{m^i, n-1}(x)$
6: Update GP with measurement of $g_{m^i}(x_n)$.

Proposition 1 shows the bisimulation function is nonincreasing through time. If the conditions in Theorem 1 hold, the robust neighborhood defined by the bisimulation function $\mathcal{A}_m$ is invariant with respect to the flow of the dynamic system. This property can be described as a tube defined as follows.

Definition 8 (Tube): A tube corresponding to the system $\mathcal{S}$ in (1) and its behavior $\mathcal{B}$ is a set of trajectories that start from a set of bounded initial states around a nominal (simulated) trajectory $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^{Q}$, denoted as $\mathcal{T}(\nu, \gamma, s_\xi)$ and defined as follows:

$$\mathcal{T}(\nu, \gamma, s_\xi) = \Big\{\hat{s}_\xi = \Big\{\Big\{(m^i, \hat{x}^i, \hat{e}^i)\Big\}_{i=0}^{Q} \Big| \hat{x}^0 \in B_{m^0}(x^0, \gamma)\Big\}, |e^i - \hat{e}^i| < \nu\Big\} \quad (18)$$

where ν is a parameter such that there exists a sequence $\Gamma = \{\gamma^i\}_{i=0}^{T}$, and for $\forall i > 0$

$$\bigcup_{\tau \in (e^{i-1} + [-\nu, \nu])} B_{m^{i-1}}\left(\varpi\left(x^{i-1}, \tau\right), \gamma^{i-1}\right) \subset B_{m^i}\left(x^i, \gamma^i\right) \quad (19)$$

holds with probability at least $(1 - \delta)$, where $B_{m^i}(x^i, \gamma^i) = \{\hat{x} | V_{m^i}(\hat{x} - x^i) \le \gamma^i\}$.

A tube is a robust forward set for system $\mathcal{S}$ that consists of a set of trajectories with bounded initial state variations and bounded switching time variations. The parameter ν allows the variance of the duration for each mode among the tube. As shown in Fig. 2, the blue regions are the tubes around trajectory s_ξ. This definition is the modification of time robustness tube in [24], in which the parameter ν should be carefully chosen, such that the tube is a robust forward set and decreases

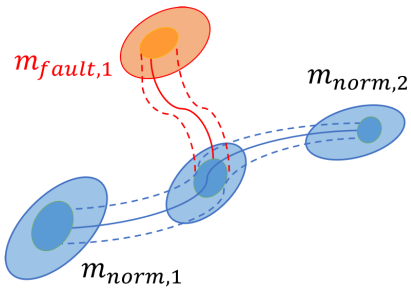

Fig. 2. If the ending neighborhoods are always covered by the initial neighborhoods in the next mode, then any normal trajectory will stay inside the tubes in the normal mode for an infinitely long horizon, while faulty trajectories will diverge from the normal trajectories within the finite-time horizon.

along with the flow of the system. Here, we incorporate the condition for ν in the definition of a tube. The condition in (19) guarantees that, if the ending robust neighborhoods are always covered by the initial robust neighborhoods in the next mode, then any trajectory will stay inside the tube. This property is important for fault detection since the faulty trajectories will be outside the tube (see Fig. 2). Formally, we can define the fault generator as follows.

1) Parameter Faults' Detection: When the ith component fault occurs, the dynamics of the faulted systems can be modeled, as in (3). Thus, in this case, the dynamics of $e(t)$ are governed by

$$\begin{aligned} \dot{e}(t) &= \Delta\hat{g}_m(e(t)) - \Upsilon_m(x(t))P_i \\ r(t) &= d_\varphi\left(\int_0^t (\Delta\hat{g}_m(e(t)) - \Upsilon_m(x(t))P_i)dt\right) \end{aligned} \quad (20)$$

where $\Delta\hat{g}_m(e(t))$ is the estimation error for the Gaussian process, which will vanishes as $t \to \infty$, since, with more and more data, the error for the Gaussian process can reach a small enough value. However, the (scalar) integral term $\int_0^t \Upsilon_m(x(t))P_i)dt$ will become nonzero depending on the dynamics of $\Upsilon_m(x(t))P_i$. Thus, when the magnitude of this term is bigger than a threshold, a fault will be detected.

2) Sensor Faults' Detection: When the jth sensor fault occurs, the dynamics of the faulted system can be modeled, as in (4). Thus, the dynamics of $e(t)$ are governed by

$$\begin{aligned} \dot{e}(t) &= \Delta\hat{g}_m(e(t)) \\ r(t) &= d_\varphi\left(e(t) - \Psi_j(x(t))Q_j\right). \end{aligned} \quad (21)$$

The first term $e(t)$ will vanish as $t \to \infty$. However, the term $\Psi_j(x(t))Q_j$ will become nonzero depending on the dynamics of $\Psi_j(x(t))Q_j$. Thus, when the magnitude of this term is bigger than a threshold, the algorithm will detect a fault.

Remark 2: In the above fault detection analysis, we claim that $\Delta\hat{g}_m(e(t))$ and $e(t)$ will vanish as $t \to \infty$ with enough estimation accuracy from the Gaussian process regression. However, the trajectories have finite lengths; thus, it is impossible for us to have nonerror estimation. Fortunately, Lemma 3 and Theorem 1 indicate that, when the estimation is bounded within the tube, the system trajectories will stay inside the tubes in the modes for an infinitely long horizon. Namely, within some bounds, $\Delta\hat{g}_m(e(t))$ and $e(t)$ will not affect the

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

fault detection results within a horizon. Moreover, the neighborhood radius can be seen as a threshold to check whether the state of the system is normal or abnormal. In addition, since STL is sensitive to noise, using $r(t)$ directly will make the decision be sensitive to noise. To address this issue, we use the following logic relationship for fault detection:

$$\begin{aligned} J(r) > J_{\text{th}} &\Rightarrow \text{ with faults } \Rightarrow \text{ alarm} \\ J(r) \leq J_{\text{th}} &\Rightarrow \text{ no faults} \end{aligned} \tag{22}$$

where the residual evaluation function is selected as $J(r) = \sum_t^{t+W} r(t)/W$ such that W is a finite-time window. It is obvious that $J(r)$ is the average residual within a time interval W.

Consider a predicate μ of STL, which defines a threshold of the signals; we have bounds for the signal as follows.

Lemma 4: Consider a trajectory $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^Q$ of a switched system $\mathcal{S}$, for any state $\varpi(x,t) \in B_{m^i}(\varpi_{m^i}(x^i,t), \gamma^i)$ and a predicate μ; we have

$$\begin{aligned} \rho\big(\mu, \varpi_{m^i}\big(x^i,t\big)\big) - \hat{\gamma} \leq \rho(\mu, \varpi_{m^i}(x,t)) \\ \leq \rho\big(\mu, \varpi_{m^i}\big(x^i,t\big)\big) + \hat{\gamma} \end{aligned} \tag{23}$$

with probability at least $(1-\delta)$, where $\hat{\gamma} = \gamma^i ||M_{m^i}||^{1/2}$ ($||\cdot||$ denoted the largest singular value of a matrix).

Proof: Since $\varpi(x,t) \in B_{m^i}(\varpi_{m^i}(x^i,t), \gamma^i) = \{\hat{x} | V_{m^i}(\hat{x} - \varpi_{m^i}(x^i,t)) \leq \gamma^i)\}$, we have that $|\varpi(x,t) - \varpi_{m^i}(x^i,t)| \leq \hat{\gamma}$ holds with probability at least $(1-\delta)$ if φ is a predicate. Therefore, $\rho(\mu, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \leq \rho(\mu, \varpi_{m^i}(x,t)) \leq \rho(\mu, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$. The lemma has been proven. □

Lemma 4 shows the up and low bounds of the robustness for any trajectory in the neighborhood of a given trajectory for a predicate. Based on this result, we can extend the result to any STL formula as follows.

Theorem 2: Consider a trajectory $s_\xi = \{(m^i, x^i, e^i)\}_{i=0}^Q$ of a switched system $\mathcal{S}$, an STL formula φ and the set $\sigma = \rho(\varphi, \varpi(s_\xi))$; then, for any $\hat{s}_\xi \in \mathcal{T}(\nu, \gamma^0, s_\xi)$, there exist $\delta, \kappa \in \mathbb{R}$ such that

$$\hat{s}[0] \in \mathcal{L}(\varphi, \sigma - \gamma_{\max})/\mathcal{L}(\varphi, \sigma + \gamma_{\max}) \tag{24}$$

with probability at least $(1-\delta^\kappa)$, where $\hat{s} = \varpi(\hat{s}_\xi)$ and $\gamma_{\max} = \max_{i=0}^Q \gamma^i ||M||_m^{1/2}$. κ is a factor related to the length of the formula.

Proof: See the Appendix. □

Theorem 2 is the key result in this article, which is a modification of [24, Th. 1]. In this revised version, we do not consider the time-varying part of the neighborhood radius since the up and low bounds only depend on the initial radius. Moreover, we consider the probability satisfaction of the formula, which is important to deal with the uncertainty of the system.

The following theorem can build a relationship between neighborhood radius and the distances between faulty signals and normal signals.

Theorem 3: Consider a set of labeled behaviors $\hat{\mathcal{B}} = \hat{\mathcal{B}}_N \cup \hat{\mathcal{B}}_F$ of a switched system $\mathcal{S}$, two real numbers $\sigma \geq 0$ and $\delta \in (0,1)$, and an STL formula φ; if, $\forall \xi \in \hat{\mathcal{B}}_N$ and $\forall \hat{\xi} \in \hat{\mathcal{B}}_F$, two independent traces $\omega(\xi) \in \mathcal{L}(\varphi, \sigma)$ and $\omega(\hat{\xi}) \in \mathcal{L}(\neg\varphi, \sigma)$ hold with probability at least $(1-\delta)$, then $d_\varphi(s, \hat{s}) > \sigma$ holds with probability at least $(1-\delta)$ for all $\xi, \hat{\xi}$, where $\omega(\xi, t) = s(t)$ and $\omega(\hat{\xi}, t) = \hat{s}(t)$.

Proof: See the Appendix. □

B. Temporal Logic Inference via Partially Ordered Direction

In this section, we solve the problem with the results in Section III-A. The problem tries to find an STL formula φ, such that the fault is (σ, δ)-diagnosable. Here, we try to maintain a safe exploration process and define the following time robustness signature to address the safe exploration problem.

Definition 9 (Robustness Signature): Given a set of labeled behaviors $\hat{\mathcal{B}} = \hat{\mathcal{B}}_N \cup \hat{\mathcal{B}}_F$ of a switched system $\mathcal{S}$, two real numbers $\sigma \geq 0$ and $\delta \in (0,1)$ and an STL formula φ, for any trace $\hat{\xi} \in \mathcal{B}$, and the associated trajectory $\alpha(\hat{\xi}) = \hat{s}_\xi = \{(m^i, \hat{x}^i, \hat{e}^i)\}_{i=0}^Q$ and its neighborhood trajectory $s_\xi = \{(m^i, \bar{x}^i, e^i)\}_{i=0}^Q \in \mathcal{T}(\nu, \sigma, \hat{s}_\xi)$, the robustness signature is denoted as $\lambda(\hat{s}_\xi, \varphi, \sigma, \delta, t)$ and defined as follows:

$$\lambda\big(\hat{s}_\xi, \varphi, \sigma, \delta, t\big) = \begin{cases} \text{SS}, & \text{if } \forall \bar{\xi} \in \hat{\mathcal{B}}_F, d_\varphi\big(\omega\big(\bar{\xi}, t\big), \varpi\big(\bar{s}_\xi, t\big)\big) > \sigma, \text{holds with} \\ & \text{probability at least } 1-\delta, \text{ and } \rho\big(\varphi, \omega\big(\hat{\xi}, t\big)\big) > 0, \\ \text{SV}, & \text{if } \forall \bar{\xi} \in \hat{\mathcal{B}}_F, d_\varphi\big(\omega\big(\bar{\xi}, t\big), \varpi\big(\bar{s}_\xi, t\big)\big) > \sigma, \text{holds with} \\ & \text{probability at least } 1-\delta, \text{ and } \rho\big(\neg\varphi, \omega\big(\hat{\xi}, t\big)\big) > 0, \\ \text{US}, & \text{if } \exists \bar{\xi} \in \hat{\mathcal{B}}_F, d_\varphi\big(\omega\big(\bar{\xi}, t\big), \varpi\big(\bar{s}_\xi, t\big)\big) \leq \sigma, \text{holds with} \\ & \text{probability at most } 1-\delta, \text{ and } \rho\big(\varphi, \omega\big(\hat{\xi}, t\big)\big) > 0, \\ \text{UV}, & \text{if } \exists \bar{\xi} \in \hat{\mathcal{B}}_F, d_\varphi\big(\omega\big(\bar{\xi}\big), \varpi\big(\bar{s}_\xi, t\big)\big) \leq \sigma, \text{holds with} \\ & \text{probability at most } 1-\delta, \text{ and } \rho\big(\neg\varphi, \omega\big(\hat{\xi}, t\big)\big) < 0, \\ \text{RS}, & \text{if } \hat{\xi} \in \mathcal{B}_N, d_\varphi\big(\omega\big(\hat{\xi}, t\big), \varpi\big(\hat{s}_\xi, t\big)\big) \leq \sigma, \text{holds with} \\ & \text{probability at least } 1-\delta, \text{ and } \rho\big(\varphi, \omega\big(\hat{\xi}, t\big)\big) > 0, \\ \text{RV}, & \text{if } \hat{\xi} \in \mathcal{B}_F, d_\varphi\big(\omega\big(\hat{\xi}, t\big), \varpi\big(\hat{s}_\xi, t\big)\big) > \sigma, \text{holds with} \\ & \text{probability at least } 1-\delta, \text{ and } \rho\big(\neg\varphi, \omega\big(\hat{\xi}, t\big)\big) > 0. \end{cases}$$

Remark 3: SS, SV, US, UV, RS, and RV are abbreviations for "Safe Satisfaction," "Safe Violation," "Unsafe Satisfaction," "Unsafe Violation," "Robust Satisfaction," and "Robust Violation," respectively. When the trace is normal, the distance to its neighborhood is bounded by σ with a probability of at least $(1-\delta)$. Therefore, the above conditions for SS, SV, US, and UV include, $\forall \hat{\xi} \in \hat{\mathcal{B}}_N, d_\varphi(\omega(\hat{\xi}, t), \varpi(s_\xi, t)) < \sigma$ holds with probability at least $(1-\delta)$. The SS signature requires that the distance between any faulty trajectory and its simulated trajectory is larger than σ, and the distance between any normal trajectory and its simulated trajectory is no larger than σ, and the trace satisfies the formula. The SV signature requires that the distance between any faulty trajectory and its simulated trajectory is larger than σ; there exists a normal trajectory such that its distance to the simulated trajectory is larger than σ and the trace violates the formula. Moreover, the probability guarantee can be satisfied when using the Gaussian process to approximate the dynamic function. Namely, if all the fault behaviors can be detected correctly with distance $d_\varphi(\omega(\bar{\xi}, t), \varpi(\bar{s}_\xi, t))$ as the metrics, the formula φ is a safe detector. Otherwise, φ is not a safe detector. The requirements for US, UV, RS, and RV can be understood accordingly.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

In the robustness signature requirements, the distance metric d_φ is used to detect the fault, while the robustness degree metrics $\rho(\varphi, \omega(\hat{\xi}, t))$ are used to measure how much the trace satisfies or violates the formula. SS, SV, US, and UV focus on the safety of the formula, while RS and RV focus on how well the formula describes the behaviors. Therefore, if a perfect formula has been found, the robustness signature for faulty behaviors should be SV and RV, and the robustness signature for normal behaviors should be SS and RS. During the safe temporal logic inference process, we do not allow US and UV, and the goal is to find a formula that sets the behaviors' robustness signature to be RS or RV accordingly. The definition of robustness signature is based on two basic concepts: 1) any normal trajectory initiated from a neighborhood will stay inside the tube around the simulated trajectory before another event occurs, while, in contrast, a faulty trajectory will be outside the tube and 2) safe temporal logic inference allows false alarm but does not allow missing alarm.

In the supervised learning setting, the STL formula φ_θ is chosen from a set of templates, denoted as Φ, where θ denotes the parameter vector that defines the formula. The search starts from a set of primitive STL formulas in the form of $\Diamond_{[a_1,b_1]}\Box_{[a_2,b_2]}\mu$ or $\Diamond_{[a,b]}\mu$, and we extend the formula by adding Boolean connectives between newly added formulas until a satisfactory formula is found. Note that each of the primitives starts with a $\Diamond$ operator, which will allow the fault to happen at any time. Let the formula obtained at the ith step be φ_i; since we require the temporal logic inference procedure to obey a safe manner, there is a relationship between φ_{i-1} and φ_i, called partial order, denoted as $\varphi_{i-1} \preceq \varphi_i$, defined as follows.

Definition 10 (Partial Order): Given two labeled sets of behaviors $\hat{\mathcal{B}}_N$ and $\hat{\mathcal{B}}_F$ of a switched system $\mathcal{S}$ in (1); for any two STL formulas φ_1 and φ_2, we say $\varphi_1 \preceq \varphi_2$ iff the following conditions hold.

1) $\forall \hat{\xi} \in \hat{\mathcal{B}}_F$, if $\lambda(\hat{s}_\xi, \varphi_1, \sigma, \delta, t) = \text{RV} \wedge \text{SV}$, then $\exists \kappa$, $\lambda(\hat{s}_\xi, \varphi_2, \sigma, \delta^\kappa, t) = \text{RV} \wedge \text{SV}$.
2) $\forall \hat{\xi} \in \hat{\mathcal{B}}_N$, if $\lambda(\hat{s}_\xi, \varphi_1, \sigma, \delta, t) = \text{RS} \wedge \text{SS}$, then $\exists \kappa$, $\lambda(\hat{s}_\xi, \varphi_2, \sigma, \delta^\kappa, t) = \text{RS} \wedge \text{SS}$.

Based on the definition of partial order and Theorem 2, if φ_{i-1} can detect all faulty behaviors safely, then φ_i can also detect all faulty behaviors safely. Moreover, the number of false alarms caused by φ_i is no larger than the false alarm caused by φ_{i-1}. As shown in Definition 6, every STL formula defines a language, which also defines a region among the signal space. Fig. 3 illustrates the safe temporal logic inference procedure, where the rectangle regions are defined by STL formulas. The central red region is expended by the abnormal behaviors, and the temporal logic inference tries to find an STL formula, which defines a region to approximate the abnormal behaviors. Before reaching the satisfactory formal, the formula considers many normal trajectories as faulty trajectories, i.e., the region covered by φ_{i-1} is no smaller than the region covered by φ_i. The following theorem shows that we can find a satisfactory formula by searching along with the partial order if the fault is detectable.

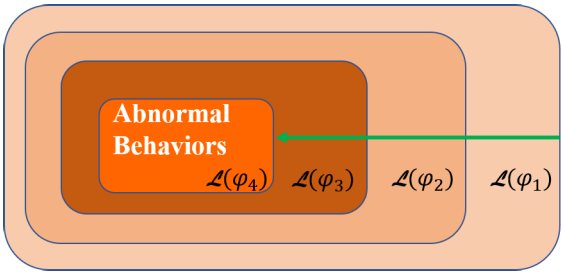

Fig. 3. Illustration of the temporal logic inference process. At each step, the rectangle regions are defined by the STL formulas, and the central red region is the true abnormal behavior. The temporal logic inference procedure tries to approximate the abnormal behavior with an STL formula by searching along a partially ordered direction.

Algorithm 2 Temporal Logic Inference Algorithm

Require: two set of behaviors $(\hat{\mathcal{B}}_F, \hat{\mathcal{B}}_N)$ and their label set $\mathcal{C}$, length limit Z for formula φ, a history list $\mathcal{H} = \emptyset$.
Ensure: The STL formula φ for fault detection.
1: Initialize $\varphi \leftarrow \varphi_1$ as a primitive formula with random parameters and calculate cost function $J(\varphi)$, set $i = 0$.
2: **repeat**
3: $i \leftarrow i + 1$,
4: $\rho_i \leftarrow calculateRobust(\varphi_i, \hat{\mathcal{B}}_F, \hat{\mathcal{B}}_N)$,
5: $\mathcal{D}^+, \mathcal{D}^-, \mathcal{U}^+, \mathcal{U}^+ \leftarrow assignLable(\rho_i, C)$,
6: **for** $\varphi_a \in \Phi$ **do**
7: $\varphi_{\text{and},\theta} \leftarrow \varphi_{\theta^*} \wedge \varphi_a$,
8: $J(\varphi_{\theta^*}), \varphi_{\theta^*} \leftarrow optAnd(\mathcal{D}^+, \mathcal{D}^-, \mathcal{U}^-, \varphi_{\text{and},\theta})$,
9: $\mathcal{H} = \mathcal{H}.add(J(\varphi_{\theta^*}), \varphi_t)$,
10: $\varphi_{\text{or},\theta} \leftarrow \varphi_i \vee \varphi_a$,
11: $J(\varphi_{\theta^*}), \varphi_{\theta^*} \leftarrow optOr(\mathcal{D}^-, \mathcal{U}^+, \mathcal{U}^-, \varphi_{\text{or},\theta})$,
12: $\mathcal{H} = \mathcal{H}.add(J(\varphi_{\theta^*}), \varphi_{\theta^*})$,
13: $\varphi \leftarrow \text{argmax}_{J(\varphi_t) \in \mathcal{H}} J(\varphi_{\theta^*})$, $\mathcal{H} \leftarrow \emptyset$.
14: **until** $i \geq Z$ or $J(\varphi_\theta)$ is non-increasing.

Theorem 4: Given two labeled sets of behaviors $\hat{\mathcal{B}}_N$ and $\hat{\mathcal{B}}_F$ of a switched system $\mathcal{S}$ in (1), if there exists an STL formula φ defined by syntax in (5), such that, $\forall \hat{\xi} \in \hat{\mathcal{B}}_F$, $\rho(\neg\varphi, \omega(\hat{\xi})) > 0$ and, $\forall \hat{\xi} \in \hat{\mathcal{B}}_N$, $\rho(\varphi, \omega(\hat{\xi})) > 0$, then there exists a sequence of STL formulas $\varphi_1, \varphi_2, \ldots, \varphi_n$ with proper parameters such that $\varphi_1 \preceq \varphi_2, \ldots, \preceq \varphi_n \preceq \varphi$ for $n \geq 1$ and $|\varphi_i| - |\varphi_{i-1}| = 1$, where $|\varphi_i|$ denotes the number of predicates in φ_i.

Proof: See the Appendix. □

Based on Theorem 4, we can infer the formula step by step. Algorithm 2 shows the process to infer the safe STL formula. The inputs of the Algorithm 2 are the class label set $\mathcal{C}$ and two sets $(\hat{\mathcal{B}}_F, \hat{\mathcal{B}}_N)$. Line 1 initializes φ_1 with a random formula from two primitive formulas with random parameters. Then, the algorithm calculates the robustness of all traces in $(\hat{\mathcal{B}}_F, \hat{\mathcal{B}}_N)$ in line 4. Based on the robustness, line 5 checks whether the trajectories are detected correctly with the current formula φ. $\mathcal{D}^+$ denotes that the traces in $\hat{\mathcal{B}}_N$ are classified correctly, and $\mathcal{U}^+$ denotes that the traces in $\hat{\mathcal{B}}_F$ are classified correctly. $\mathcal{D}^-$ and $\mathcal{U}^-$ are defined vice versa. For example, if a trajectory in $\hat{\mathcal{B}}_N$ has a positive robustness degree, the trajectory is assigned to the set $\mathcal{D}^+$. If the trajectory

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

has negative robustness, it will be assigned to the set $\mathcal{D}^-$. Line 6 checks all template formulas. Line 7 extends the current formula to get a new formula $\varphi_{\text{and},\theta}$ with parameter vector θ by conjunction operator. Line 8 optimizes the parameter vector and calculates cost function to achieve a temporary optimal $(J(\theta^*), \varphi_\theta)$. Line 9 saves the results in history $\mathcal{H}$. Lines 10–12 extend the formula with a disjunction operator and find the optimal parameter vector. Line 13 chooses the best formula in history. This procedure will be continued until a length limit has been reached. The cost function is defined as follows:

$$J(\varphi_\theta) = \sum_{\hat{\xi}^N \in \hat{\mathcal{B}}_N} J_{\text{SRV}}\left(\varphi_\theta, \hat{s}_\xi^N\right) + \sum_{\hat{\xi}^F \in \hat{\mathcal{B}}_F} J_{\text{SRS}}\left(\varphi_\theta, \hat{s}_\xi^F\right) \quad (25)$$

where $J_{\text{SRV}}(\varphi_\theta, \hat{s}_\xi^N)$ is 1 if $\lambda(\hat{s}_\xi^N, \varphi, \sigma, \delta, 0) = \text{RS} \wedge \text{SS}$, else $J_{\text{SRV}}(\varphi_\theta, \hat{s}_\xi^N$ is 0, and $J_{\text{SRS}}(\varphi_\theta, \hat{s}_\xi^F)$ is 1 if $\lambda(\hat{s}_\xi^F, \varphi, \sigma, \delta, 0) = \text{RV} \wedge \text{SV}$, else $J_{\text{SRS}}(\varphi_\theta, \hat{s}_\xi^F)$ is 0. $J(\varphi_\theta)$ calculates the number of traces that have robustness signature $\text{RS} \wedge \text{SS}$ and $\text{RV} \wedge \text{SV}$, which is equivalent to decrease the number of trajectories in $\mathcal{U}^-$ and $\mathcal{D}^-$. The optimization problems are defined as follows.

Parameter Optimization: The goal of each optimization problem is to find an optimal parameter vector θ^* with safe explore manner such that $\varphi \preceq \varphi_{\text{and},\theta^*}$, $\varphi \preceq \varphi_{\text{or},\theta^*}$, and the value for $J(\varphi_\theta)$ is maximum. Therefore, the two optimization problems in Lines 8 and 11 can be defined as

$$\theta^* = \text{argmax}\, J(\varphi_\theta) \quad (26)$$

subject to

$$\forall \hat{\xi} \in \mathcal{D}^+, \quad \lambda\left(\hat{s}_\xi, \varphi_{\text{and}}, \sigma, \delta, 0\right)\big) = \text{RS} \wedge \text{SS} \quad (27a)$$
$$\forall \hat{\xi} \in \mathcal{U}^+, \quad \lambda\left(\hat{s}_\xi, \varphi_{\text{or}}, \sigma, \delta, 0\right)\big) = \text{RV} \wedge \text{SV}. \quad (27b)$$

According to the semantics of STL, we can ignore the constraint in (27a) during the optimization process in line 11 and ignore the constraint in (27b) during the optimization process in line 8 since they always hold. We solve the optimization problem defined in (26) with an active learning algorithm called the *Gaussian process adaptive confidence bound* (GP-ACB) defined in [35]. Algorithm 2 can lead to the following theoretical results.

Theorem 5: Denote φ_i to be the formula found from the ith iteration in line 11 of Algorithm 2; then, the following statements hold.

1) $\varphi_1 \preceq \varphi_2, \ldots, \varphi_i, \ldots, \preceq \varphi_M$.
2) If there exists an STL formula φ^* with proper parameters that can detect the fault correctly in $\hat{\mathcal{B}}$, then, with a large enough M, we have $\varphi^* \preceq \varphi_M$.

Proof: See the Appendix. □

The overall procedure for the proposed method is shown in Fig. 4. The method first uses Algorithm 1 to find the neighborhood radius and estimate the system model. Second, with the estimated model and the normal behaviors and abnormal behaviors, the estimated behaviors are obtained. Third, with the neighborhood radius and the behaviors, Algorithm 2 is applied to infer the optimal formula. Finally, the fault detection is performed with the learned formula by comparing the residual signals with a predefined threshold, which is related to the neighborhood radius.

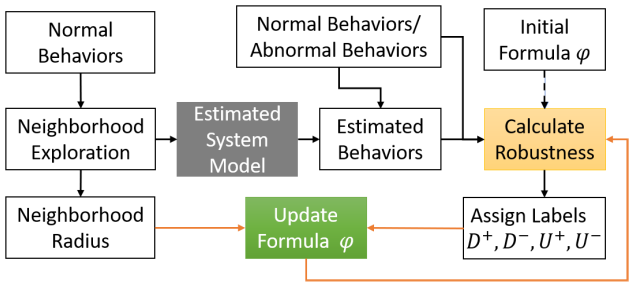

Fig. 4. Overall procedure of temporal logic inference for fault detection with partially unknown dynamics.

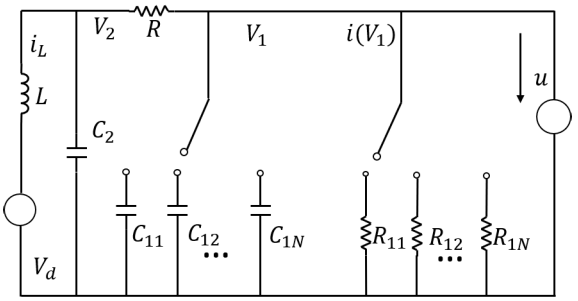

Fig. 5. Switched Chua's circuit.

Remark 4: Safe temporal logic inference performs an inference process based on the estimation of unknown dynamics with the Gaussian process. With more data collected, we will have a better approximation of the unknown dynamics. Moreover, the safe learning process decreases the number of false alarms and missing fault gradually (missing fault does not exist in our setting). These properties make the proposed method suitable for online fault detection, in which the algorithm updates the approximation of unknown dynamics online and improves the approximation with newly collected data. The safe temporal logic inference algorithm updates the formula if the new trajectory's robustness signature is not RS or SV.

Remark 5: The obtained formula can be seen as an interpretable classifier. Therefore, it has two roles: one is a classifier, and the other is a decision explanator. It classifies the conditions of the system and gives explanations for the decision process with the semantics of STL formulas. Since we use labeled data to train a model, this problem is a supervised learning problem previously addressed in [25] and [36]. However, in these works, the learning algorithm cannot provide guarantees for the results. Moreover, these works assumed that they can find the optimal classifier that will detect the fault surely. This hypothesis is too strong in practical. The problem and solution presented in this article are more general and will consider uncertainties of the system along with the noise among the trajectories. More specifically, we do not provide a solution that detects the fault surely but with a probabilistic satisfaction guarantee. Compared with other residual signal-based fault detection, whose thresholds are determined empirically, the thresholds used in this article are estimated based on the bisimulation function.

IV. Fault Detection for Chua's Circuit System

In this section, we apply the proposed method to at switched Chua's circuit system [17], [37], which is shown in Fig. 5 and

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

can be described as follows:

$$\begin{cases} \dot{V}_1 = -\dfrac{1}{C_{1m}}V_1 + \dfrac{1}{C_{1m}R}V_2 - \dfrac{1}{C_{1m}}g_m(V_1) - \dfrac{1}{C_{1m}}u \\ \dot{V}_2 = \dfrac{1}{C_2R}V_1 - \dfrac{1}{C_2R}V_2 - \dfrac{1}{C_2}i_L \\ \dot{i}_L = \dfrac{1}{L}V_2 - \dfrac{1}{L}V_d \end{cases} \tag{28}$$

where V_1 and V_2 are the two states and represent the voltage across the capacitors C_{1m} and C_2 with $m = 1, 2$ being the switched signal. i_L stands for the current in the inductor L and is also the system state. u is the current from the generator, and u acts as a noise source in this article. V_d is the voltage loss of $R_0 i_L$. $g_k(V)$ is the kth current in resistor R_{1k}, which is a nonlinear function defined as

$$g_m(V) = \begin{cases} -1.43V_1 + V_1^2 + V_2^2, & m = 1 \\ -0.78 * V_1 + V_1^2, & m = 2. \end{cases} \tag{29}$$

Here, we follow [17], and let $x_1 = V_1$, $x_2 = V_2$, $x_3 = i_L$, $\varrho_{1m} = 1/C_{1m}R$, $\varrho_{2m} = 1/C_{1m}$, $\varrho_3 = 1/C_2R$, $\varrho_4 = 1/C_2$, and $\varrho_5 = 1/L$; then, the system in (28) can be transformed into the following form:

$$\dot{x} = A_m x + B_m[g_m(x_1) + u] \tag{30}$$

$$A_m = \begin{bmatrix} -\varrho_{1m} & \varrho_{1m} & 0 \\ \varrho_3 & -\varrho_3 & -\varrho_4 \\ 0 & \varrho_5 & -\varrho_5 R \end{bmatrix}, \quad B_m = \begin{bmatrix} -\varrho_{2i} \\ 0 \\ 0 \end{bmatrix} \tag{31}$$

where $x = [x_1, x_2, x_3]^T$ is the system state vector, and we assume that the states are fully observable.

In this experiment, we choose parameters of the switched Chua's circuit system as $C_{11} = 0.764$, $C_{12} = 3.215$, $R = 1.637$, $C_2 = 10$, $L = 1.1$, and $R_0 = 0.012$. Then, the system matrices and vectors are given by

$$A_1 = \begin{bmatrix} -0.799 & 0.799 & 0 \\ 0.061 & -0.061 & -0.1 \\ 0 & 0.0909 & -0.011 \end{bmatrix}, \quad B_1 = \begin{bmatrix} -1.309 \\ 0 \\ 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -0.311 & 0.311 & 0 \\ 0.061 & -0.061 & -0.1 \\ 0 & 0.0909 & -0.011 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -2.146 \\ 0 \\ 0 \end{bmatrix}.$$

A. Temporal Logic Inference

When a fault is injected, the considered system is in the form of

$$\begin{aligned} \dot{x} &= A_m x + B_m[g_m(x) + u] + D_m \eta_m(x) \\ y &= x \end{aligned} \tag{32}$$

where $D_1 = D_2 = [1, 1, 1]^T$ and the fault functions are selected as

$$\eta_1(x) = \begin{cases} 0.3\cos(t), & t_1 \le t \le t_2 \\ 0, & \text{others} \end{cases}$$

$$\eta_2(x) = \begin{cases} 1, & t_1 \le t \le t_2 \\ 0, & \text{others} \end{cases} \tag{33}$$

where $[t_1, t_2]$ is the fault injection dwell time. Note that, here, we define the model of $g_m(\cdot)$, and the noise u is assumed to be Gaussian during the simulation process such that we can simulate the system and obtain trajectories for demonstration purpose. However, the algorithms proposed in this article are not given the model of $g_m(\cdot)$, and it should approximate the model with the Gaussian process regression approach.

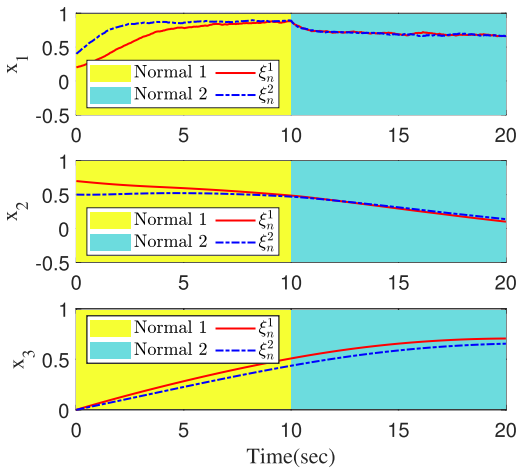

Fig. 6. Two traces ξ_n^1, ξ_n^2 in the normal cases. The system switches mode at time = 10 s.

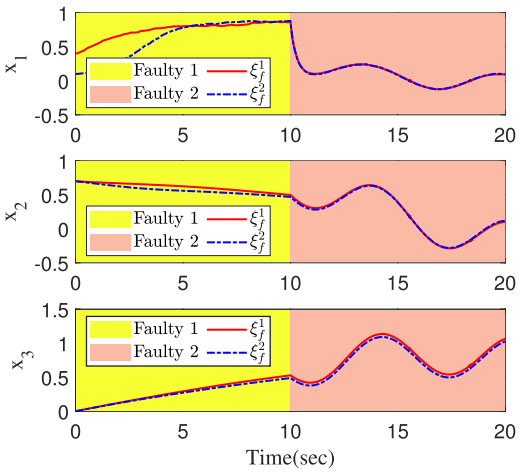

Fig. 7. Two traces ξ_f^1, ξ_f^2 in the faulty cases, where the fault mode is introduced at time = 10 s.

In this case study, we assume that a possible event occurs at t_e and the minimal dwell time $e_{\min} = 5$ s, which means that, if there is an event that occurs at t_e, no other events can occur between $(t_e - e_{\min})$ and $(t_e + e_{\min})$. Since the two kinds of primitives, $\Diamond_{[a_1,b_1]}\Box_{[a_2,b_2]}\mu$ and $\Diamond_{[a,b]}\mu$, start with "Eventually" operator, the formula can capture the fault with suitable parameters and even the fault happens at random time. To make it simple, in this setting, we have $t_e = 10$ s for all generated trajectories in the training sets.

We simulate 20 traces (ten normal traces in $\hat{\mathcal{B}}_N$ and ten faulty traces in $\hat{\mathcal{B}}_F$) for 20 s. The traces are various due to uncertainties and the variation of initial states at each switching instant. Fig. 6 and 7 show examples of normal and faulty traces, respectively. For the normal traces, two initial states are simulated, and the system switches from mode 1 to mode 2 after 10 s. For the faulty traces, one is injected fault

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

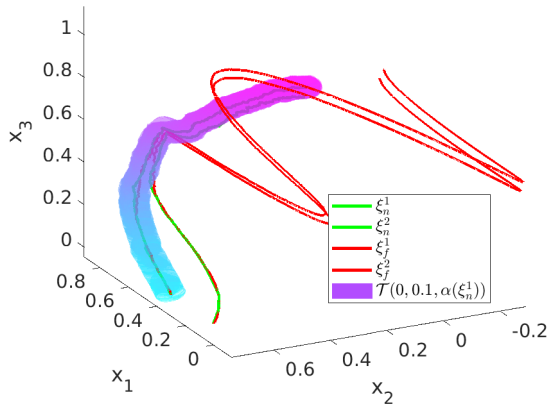

Fig. 8. Tube around a normal trace, two normal traces (green, one is hidden in the tube), and two faulty traces (red), respectively. The normal trace is always within the tube, while the faulty traces are outside the tube after mode switch.

function 1, and the other is injected fault function 2. Both of them start from normal mode 1 and switch to fault mode after 10 s. Here, we set the fault event occurring time the same as the fault injection time even though it may not lead to a failure immediately. Fig. 8 shows the tube around one normal trace with a neighborhood radius of 0.1, which illustrates that the normal traces are constrained in a bounded region, and the faulty traces are outside that bounded region.

Since the uncertainty of Chua's circuit is assumed to be Gaussian, the uncertainty caused by noises can be covered by the neighborhoods around the simulated traces. In order to estimate the neighborhood radius, we first estimate the unknown dynamic with the Gaussian process regression. During the model estimate process, we sample a set of points of the system for each mode and use the GP-ACB algorithm to approximate the model for $g_m(\cdot)$. Then, we estimate the neighborhood radius. This process is completed with Algorithm 1. Fig. 9 shows the estimation process. In order to have a better visualization of the process, we only plot two dimensions of the state (V_1 and V_2) and set $M_m = \text{diag}(1, 1, 0)$. The top row of Fig. 9 shows the estimation of $g_m(\cdot)$, which shows that the estimated model can approximate the true model better with more samples. The bottom row of Fig. 9 shows the states within the neighborhood radius of a simulated trace. Starting from an initial set, our algorithm can approximate the true neighborhood of the simulated traces with samples of the system. The more data we have, the smaller the estimation error we can achieve. The results of Algorithm 1 show that the maximum neighborhood radius for mode 1 is 0.3 and the maximum neighborhood radius for mode 2 among all normal traces is 0.5. The probability bounds for all learning processes are set to 0.05, which means that the approximation error is bounded to have a small value with a probability of at least 0.95. The bottom two figures in Fig. 9 show the estimated and true neighborhood of normal traces at the initial state (mode 1), which again proves that the proposed algorithm can approximate the neighborhood with limited samples of the system. Note that, if we discretize the system with smaller values of τ and collect more sample data, the estimation will improve and, in the limit, converge to the true neighborhood. Overall, Algorithm 1 provides a powerful tool to learn the neighborhood radius.

Based on the estimation of the neighborhood radius, we can obtain the robustness tubes $\cup_{s_\xi^\xi \in \hat{\mathcal{B}}} \mathcal{T}(0, 0.1, s_\xi^\xi)$ for each trace. In Fig. 8, we plot the robustness tube around one simulated normal trace. Fig. 8 also shows that the distance between two normal traces does not increase, indicating that the true radius is larger than the estimated one, which is in line with our analysis that the proposed algorithm provides a conservative estimation of the robustness radius. With the neighborhood radius, we can infer the temporal logic formula with Algorithm 2. In this example, we set $Z = 4$, $\lambda = 0.1$, $\sigma = 0.1$, and $\delta = 0.05$. A sequence of satisfactory formulas is obtained as follows:

$$\varphi_1 = \left(\Diamond_{[0.3,10.4]}\Box_{[0.2,6.7]}(x_1 \leq 0.21)\right) \tag{34}$$

$$\varphi_2 = \left(\Diamond_{[0.3,10.4]}\Box_{[0.2,6.7]}(x_1 \leq 0.21)\right) \wedge \left(\Diamond_{[15.3,17.2]}(x_1 \geq -0.09)\right) \tag{35}$$

$$\varphi_3 = \left(\Diamond_{[0.3,10.4]}\Box_{[0.2,6.7]}(x_1 \leq 0.21)\right) \wedge \left(\Diamond_{[15.3,17.2]}(x_1 \geq -0.09)\right) \vee \left(\Diamond_{[1.3,6.8]}\Box_{[10.2,15.4]}(x_3 \geq 0.42)\right) \tag{36}$$

$$\varphi_4 = \left(\Diamond_{[0.3,10.4]}\Box_{[0.2,6.7]}(x_1 \leq 0.21)\right) \wedge \left(\Diamond_{[15.3,17.2]}(x_1 \geq -0.09)\right) \vee \left(\Diamond_{[1.3,6.8]}\Box_{[10.2,15.4]}(x_3 \geq 0.42)\right) \vee \left(\Diamond_{[2.6,7.7]}\Box_{[3.2,8.2]}(x_2 \geq -0.12)\right) \tag{37}$$

where φ_4 can be read by plain English as follows: "eventually, within 0.3–10.4 s, always within 0.2–6.7 s, V_1 should be no larger than 0.21, and eventually, within 15.3–17.2 s, V_1 should be no smaller than -0.09, or eventually, within 1.3–6.8 s, always within 10.2–15.4 s, i_L should be no smaller than 0.42, or eventually, within 2.7–7.7 s, always within 3.2–8.2 s, V_1 should be no smaller than -0.12."

The robustness signatures for each trajectory with respect to φ_i ($i = 1, 2, 3, 4$) are shown in Table I, where SRV is short for SV $\wedge$ RV and SRS is short for SS $\wedge$ RS. The signatures for the traces are SS or SV, which indicates that the proposed temporal logic inference algorithm searches for the formula safely. Moreover, with an increasing length of the formula, the signature for each trajectory does not change once it has been set to SRS or SRV. The value for $J(\varphi)$ is increased, which also indicates that $\varphi_1 \preceq \varphi_2 \preceq \varphi_3 \preceq \varphi_4$ and the proposed temporal logic inference algorithm searches for the formula safely.

B. Fault Detection With Temporal Logic

1) Fault Detection With Different Dwell Times: In Section IV-A, we assume that a possible event occurs at the tenth second, as shown in Fig. 7. Since we add an eventual operator for the learned formula, the trajectories can always be shifted to the trajectories that have an event at 0 s. In this section, we test and investigate the learned formula with sensor faults and different dwell times.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

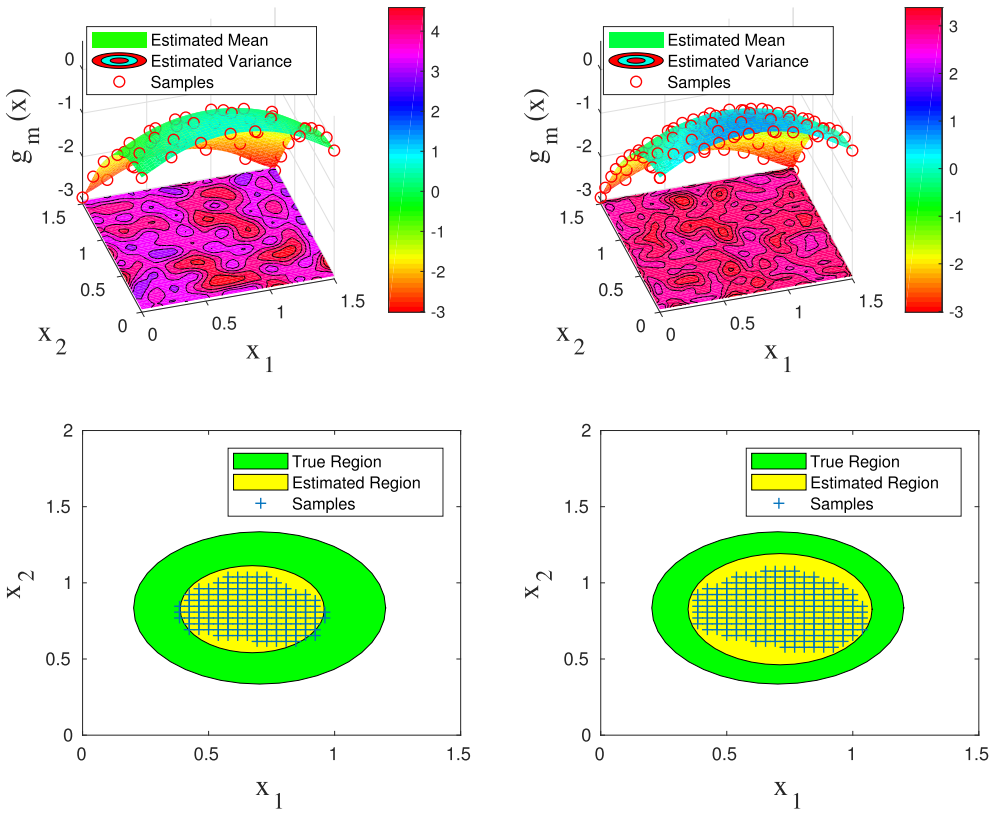

Fig. 9. 2-D example of Algorithm 1. Top: initially, the estimate of the dynamics is uncertain (the contour maps in the top figures are the variances for the estimation, and the true variances have been multiplied by 40 for better visualization). Left: we use 50 samples to estimate the model $g_m(\cdot)$. Right: we use 100 samples to estimate the model. Bottom: true and estimated neighborhood sets are plotted after 50 (left) and 100 (right) samples (blues crosses) in two dimensions. Algorithm 1 provides a conservative estimate (yellow) since it considers states with $\dot{V}(x) \leq -L\tau$, rather than $\dot{V}(x) \leq 0$ and $AB - A\mu_{m^i,n-1} \leq -||A||_1 L\tau$ in (16) and (17). The level set could be increased by discretizing the space with a smaller value of τ.

TABLE I

ROBUSTNESS SIGNATURES $\lambda(\hat{s}_\xi, \varphi_i, 0.05, 0.05)$ FOR $i = 1, 2, 3, 4$ OF THE SIMULATED TRACES

Signature	$\hat{\xi}_f^1$	$\hat{\xi}_f^2$	$\hat{\xi}_f^3$	$\hat{\xi}_f^4$	$\hat{\xi}_f^5$	$\hat{\xi}_f^6$	$\hat{\xi}_f^7$	$\hat{\xi}_f^8$	$\hat{\xi}_f^9$	$\hat{\xi}_f^{10}$	$\hat{\xi}_n^1$	$\hat{\xi}_n^2$	$\hat{\xi}_n^3$	$\hat{\xi}_n^4$	$\hat{\xi}_n^5$	$\hat{\xi}_n^6$	$\hat{\xi}_n^7$	$\hat{\xi}_n^8$	$\hat{\xi}_n^9$	$\hat{\xi}_n^{10}$	$J(\varphi)$
φ_1	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRS	SV	SV	SV	SV	SV	SV	SV	SV	SV	11
φ_2	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRS	SV	SV	SV	SV	SRS	SV	SV	SV	SRS	13
φ_3	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRS	SRS	SV	SRS	SRS	SRS	SV	SV	SRS	SRS	17
φ_4	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRV	SRS	SRS	SRS	SRS	SRS	SRS	SRS	SRS	SRS	SRS	20

In this scenario, we inject sensor fault with different dwell times. The sensor fault is in the form of

$$y = x + \zeta_m(x) \tag{38}$$

where $\zeta_m(x)$ is the fault function at mode m and defined as

$$\zeta_1(x) = \begin{cases} 0.3\sin(t), & t_1 \leq t \leq t_2 \\ 0, & \text{others} \end{cases}$$
$$\zeta_2(x) = \begin{cases} 0.5, & t_1 \leq t \leq t_2 \\ 0, & \text{others} \end{cases} \tag{39}$$

where $[t_1, t_2]$ is the fault injection dwell time.

In order to test the learned formula, we simulate 20 traces (ten normal traces in $\hat{\mathcal{B}}_N$ and ten faulty traces in $\hat{\mathcal{B}}_F$) for 100 s with different dwell times and fault injection times. Six representative trajectories of the simulated system with the minimal dwell time $e_{\min} = 5, 10$, and 15 s are shown in Fig. 10, whose fault injection times are set at 20th, 50th, and 70th second. In this setting, we set the window length to 5 s. The residual signals over time for the six representative trajectories are shown in Fig. 11. Since the residues are average robustness among a window length, the signals are smooth. The length of the residues is from 0 to 80 since the robustness evaluation length is 20 s. Fig. 11 shows that the faults can be detected with a residues threshold set to 0.1, and the fault detection results for the 20 simulated traces are shown in Table II. The results show that the learned formula can detect the fault with high accuracy.

2) Fault Detection With Different Noise Levels: This article uses the Gaussian process regression to estimate the unknown dynamic of the systems, which is expected to be robust to noise. To demonstrate the noise resistance property of the proposed method, this experiment generates traces with different noise levels by changing the mean and variance of u in (32). First, we simulate 20 traces (ten normal traces in $\hat{\mathcal{B}}_N$ and ten faulty traces in $\hat{\mathcal{B}}_F$ from component fault

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

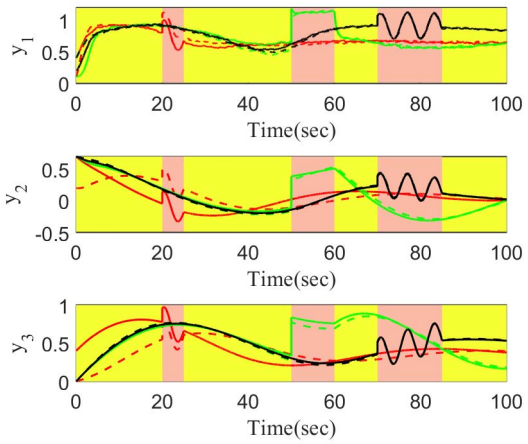

Fig. 10. Some representative trajectories of the simulated system with the minimal dwell time $e_{\min} = 5, 10, 15$ s and fault injection times are set at 20th, 50th, and 70th second, respectively. The yellow regions show the normal traces, and pink regions show the faulty traces.

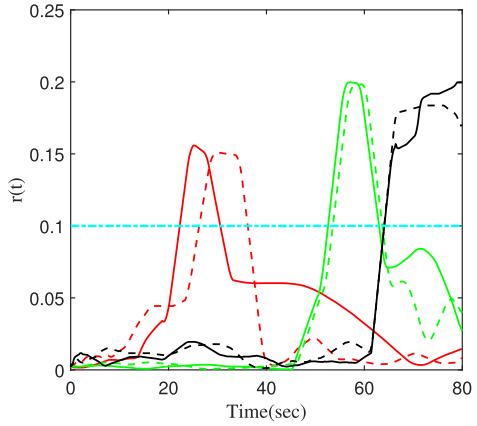

Fig. 11. Residual signals with respect to the learned formula for the trajectories in Fig. 10.

TABLE II
SIMULATION TESTING RESULTS

Test Method	Missing Fault	False Alarm
Normal Traces	0	1
Faulty Traces	0	0

injection) for 20 s with noise variance at 0.1, 0.2, and 0.3 (zero mean for all); then, we infer the temporal logic formula with the proposed method. During the temporal logic inference procedure, the robustness tube radius are set to the sum of the maximum robustness radius found in Algorithm 1 and the maximum estimated variance for the Gaussian process regression after N samples, i.e., $\sigma = \max \gamma_N^i + \max \sigma_{m^i,N-1}$. The probability factor δ is set to 0.05 for each case. Second, we test the learned formula with 100 traces (20 normal traces in $\hat{\mathcal{B}}_N$, 40 faulty traces in $\hat{\mathcal{B}}_F$ from component fault injection with different dwell times and different fault injection times, and 40 faulty traces in $\hat{\mathcal{B}}_F$ from sensor fault injection with different dwell times and different fault injection times) for 100 s. Then, we calculate the residual signals for each trace with a window length of 5 and 10 s for each case for fault detection at different noise levels. The fault detection results are shown in Table III. The results show that the temporal logic can detect the fault with high accuracy (detect errors are less than 5%) both among the training set and the testing set. Moreover, with the increase in the noise, the detection error will increase, but it will be improved by taking a larger calculation window. The reason for this is that a larger window can decrease the effect of noise, which acts as a filter.

TABLE III
SIMULATION TESTING RESULTS WITH DIFFERENT NOISE LEVELS

Noise/Length(s)	0.1/5	0.2/5	0.3/5	0.1/10	0.2/10	0.3/10
Training Error (#)	0	2	1	0	1	1
Testing Error (#)	3	5	6	2	4	5

TABLE IV
COMPARISON RESULTS FOR FAULT DETECTION

Method	Proposed Method	Method in [36]	Method in [10]
Training Error (#)	0	2	3
Testing Error (#)	1	2	4

3) Comparison Study: In order to demonstrate the performance of the partially ordered direction-based method in temporal logic inference, we compare the proposed method with other state-of-the-art methods. In this case study, we only investigate the efficiency of these algorithms in the task of temporal logic inference; thus, we assume that the robustness radius and the model of the switched system have been obtained. With the trajectories and robustness radius, the algorithms in [10] and [36] were applied to find the optimal formulas for fault detection. In this experiment, ten normal and ten faulty trajectories are simulated for training, and another 20 normal and 20 faulty trajectories are simulated for testing the performance. The noise levels for all the simulations are set to 0.1. Moreover, we constrain the training time to 600 s for all the algorithms, i.e., check the performance under a given time, and comparison results are shown in Table IV. The results show that, when given the robustness radius and trajectories, the proposed partially ordered direction-based method is more efficient to find the optimal formula.

C. Discussion

Fault detection with temporal logic brings interpretability to human users. The inferred logic formula helps maintainers to diagnose the causes of the fault and, thus, take timely actions to reduce losses. Nevertheless, fault prognosis is another important aspect in monitoring tasks. However, the quantitative semantics, i.e., robustness, of temporal logic used in this article is not continuous or differentiable; thus, it is not suitable for fault prognosis. In our future work, we will try to use differentiable quantitative semantics such that we can map the robustness to the remaining useful life of the failing

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

component for prognosis task since the robustness defines how far the system is deviated from normal.

The proposed method uses the Gaussian process regression to estimate the dynamic function of the unknown dynamics. However, many systems are non-Gaussian, which may cause a large regression error and lead to a bigger robustness radius or fault detection threshold. The large robustness radius will affect the sensitivity of the fault detector and, thus, decreases the performance of the proposed approach. When we apply the proposed method to practical systems, the noise level is another impact factor. Based on the simulation testing results, the performance degrades when the noise level is high, and it will cause a bigger robustness radius. Moreover, large noise will increase the robustness radius and, thus, decrease the performance. In addition, the proposed method needs the mode information of the trajectories for inferring the logic formulas, which is hard to obtain for many practical systems and limits the generality of the proposed method.

V. Conclusion

This article presents a method to infer temporal logic formulas for fault detection tasks of a kind of switched system with partially unknown dynamics. The proposed temporal logic inference algorithm is guided by a partially ordered relation, and the obtained temporal logic formula acted as a residual evaluation function in a transparent way. Demonstration experiments with simulated trajectories show that the proposed method can be used for fault detection for switched systems with partially unknown dynamics. Moreover, the temporal logic inference algorithm that searches along the partially ordered direction has a better performance than the state-of-the-art methods in our settings. In addition, the limitations of the proposed method have been discussed. Future research work will address these limitations and focus on: 1) fault prognosis issue with temporal logic; 2) temporal logic fault diagnosis with unsupervised learning approaches; and 3) fault detection with temporal logic for non-Gaussian systems.

Appendix

Proof of Theorem 1

Proof: Based on [34, Th. 1], when condition (16) holds with probability at least $(1-\delta)$, the system is asymptotically stable for all $y \in B_{m^i}(x^i, \gamma^i) \cap \mathcal{X}_\tau$ with the probability at least $(1-\delta)$. Based on [34, Lemma 1] and Lemma 3, we have

$$\begin{aligned}
&\left|\mu_{m^i,n-1}\left([y]_\tau\right) - g_{m^i}(y)\right| \le \beta^{1/2}_{m^i,n}\sigma_{m^i,n-1}\left([y]_\tau\right) + L\tau \\
&\quad\Rightarrow \left|A\mu_{m^i,n-1}\left([y]_\tau\right) - Ag_{m^i}(y)\right| \\
&\quad\le ||A||_1\beta^{1/2}_{m^i,n}\sigma_{m^i,n-1}\left([y]_\tau\right) + ||A||_1 L\tau \\
&\quad\Rightarrow \left|AB - Ag_{m^i}(y) - \left(AB - A\mu_{m^i,n-1}\left([y]_\tau\right)\right)\right| \\
&\quad\le ||A||_1\beta^{1/2}_{m^i,n}\sigma_{m^i,n-1}\left([y]_\tau\right) + ||A||_1 L\tau.
\end{aligned}$$

If the condition in (17) holds, we have $AB - Ag_{m^i}(y) \le 0$. Namely, we have that condition (11) holds. Therefore, $\mathcal{A}_{m^i}$ is a bisimulation function. The theorem has been proven. □

Proof of Theorem 2

Proof: Based on the result in Lemma 4, $\rho(\mu, \varpi_{m^i}(x^i,t)) \in \mathcal{L}(\varphi, \sigma - \gamma_{\max})/\mathcal{L}(\varphi, \sigma + \gamma_{\max})$ holds with probability at least $(1-\delta)$ for $i = 0, \ldots, Q$; thus, Theorem 2 holds for any predicate of STL, and then, we can prove this theorem by induction.

1) Assume that Theorem 2 holds for formula φ and prove the theorem holds for $\neg\varphi$. If Theorem 2 holds for φ, we have that $-\rho(\neg\varphi, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le -\rho(\neg\varphi, \varpi_{m^i}(x,t)) \le -\rho(\neg\varphi, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$; thus, $\rho(\neg\varphi, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\neg\varphi, \varpi_{m^i}(x,t)) \le \rho(\neg\varphi, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$.
2) Assume that Theorem 2 holds for formulas φ_1, φ_2, and the satisfaction of φ_1 and φ_2 is independent; then, we prove that the theorem holds for $\varphi_1 \wedge \varphi_2$. Based on the semantic of STL, $\rho(\varphi_1 \wedge \varphi_2, \varpi_{m^i}(x,t)) = \min(\rho(\varphi_1, \varpi_{m^i}(x,t)), \rho(\varphi_2, \varpi_{m^i}(x,t)))$. When Theorem 2 holds for formulas φ_1, φ_2, we have that $\rho(\varphi_{1,2}, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\varphi_{1,2}, \varpi_{m^i}(x,t)) \le \rho(\varphi_{1,2}, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$. Thus, there exists a κ such that $(1-\delta)^2 \ge (1-\delta^\kappa)$, and for φ_1, φ_2, we have that $\rho(\varphi_1 \wedge \varphi_2, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\varphi_1 \wedge \varphi_2, \varpi_{m^i}(x,t)) \le \rho(\varphi_1 \wedge \varphi_2, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta^\kappa)$.
3) Assume that Theorem 2 holds for formulas φ_1 and φ_2, and the satisfaction of φ_1 and φ_2 is independent; we prove that the theorem holds for $\varphi_1 \vee \varphi_2$. Based on the semantic of STL, $\rho(\varphi_1 \vee \varphi_2, \varpi_{m^i}(x,t)) = \max(\rho(\varphi_1, \varpi_{m^i}(x,t)), \rho(\varphi_2, \varpi_{m^i}(x,t)))$. When Theorem 2 holds for formulas φ_1 and φ_2, we have that $\rho(\varphi_{1,2}, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\varphi_{1,2}, \varpi_{m^i}(x,t)) \le \rho(\varphi_{1,2}, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$. Thus, there exists a κ such that $2(1-\delta) - (1-\delta)^2 \ge (1-\delta^\kappa)$, and for φ_1, φ_2, we have that $\rho(\varphi_1 \vee \varphi_2, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\varphi_1 \vee \varphi_2, \varpi_{m^i}(x,t)) \le \rho(\varphi_1 \vee \varphi_2, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta^\kappa)$.
4) Assume that Theorem 2 holds for formula φ and prove the theorem holds for $\Diamond_{\mathcal{I}}\varphi$. Based on the semantic of STL, $\rho(\Diamond_{\mathcal{I}}\varphi, \varpi_{m^i}(x,t)) = \max_{\tau\in\mathcal{I}}(\rho(\varphi, \varpi(x,\tau))$. When Theorem 2 holds for formula φ, we have that $\rho(\varphi, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\varphi, \varpi_{m^i}(x,t)) \le \rho(\varphi, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$. Thus, there exists a κ such that the combination probability is at least $1-\delta^\kappa)$, and we have that $\rho(\Diamond_{\mathcal{I}}\varphi, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\Diamond_{\mathcal{I}}\varphi, \varpi_{m^i}(x,t)) \le \rho(\Diamond_{\mathcal{I}}\varphi, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta^\kappa)$.
5) Assume that Theorem 2 holds for formula φ and prove the theorem holds for $\Box_{\mathcal{I}}\varphi$. Based on the semantic of STL, $\rho(\Box_{\mathcal{I}}\varphi, \varpi_{m^i}(x,t)) = \min_{\tau\in\mathcal{I}}(\rho(\varphi, \varpi(x,\tau))$. When Theorem 2 holds for formal φ, we have that $\rho(\varphi, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \le \rho(\varphi, \varpi_{m^i}(x,t)) \le \rho(\varphi, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta)$. Thus, there exists a κ such that the combination

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

probability is at least $(1-\delta^{\kappa})$, and we have that $\rho(\Box_{\mathcal{I}}\varphi, \varpi_{m^i}(x^i,t)) - \hat{\gamma} \leq \rho(\Box_{\mathcal{I}}\varphi, \varpi_{m^i}(x,t)) \leq \rho(\Box_{\mathcal{I}}\varphi, \varpi_{m^i}(x^i,t)) + \hat{\gamma}$ holds with probability at least $(1-\delta^{\kappa})$.

Thus, Theorem 2 holds for any STL formula φ. □

Proof of Theorem 3

Proof: Since, for all $\xi \in \hat{\mathcal{B}}_N$, $\omega(\xi) \in \mathcal{L}(\varphi,\sigma)$, and for all $\hat{\xi} \in \hat{\mathcal{B}}_F$, $\hat{\xi} \in \mathcal{L}(\neg\varphi,\sigma)$ hold with probability at least $(1-\delta)$, we have $\rho(\varphi,\omega(\xi,t)) > \sigma$, and $\rho(\neg\varphi,\omega(\hat{\xi}),t)) > \sigma$ holds with probability at least $(1-\delta)$ for all $\xi,\hat{\xi}$. As $d_{\varphi}(s,\hat{s}) = |\rho(\varphi,\omega(\hat{\xi}),t) - \rho(\varphi,\omega(\xi),t)| = |\rho(\neg\varphi,\omega(\hat{\xi}),t) + \rho(\varphi,\omega(\xi),t)| > \sigma$, and the two events $\rho(\varphi,\omega(\xi),t) > \sigma$ and $\rho(\neg\varphi,\omega(\hat{\xi}),t) > \sigma$ are independent, we have that $d_{\varphi}(s,\hat{s}) > \sigma$ holds with probability at least $2(1-\delta) - (1-\delta)^2 > (1-\delta)$. This completes the proof. □

Proof of Theorem 4

Proof: When φ has only one predicate, the theorem is obviously true. When φ has more than one predicate, φ can be written as: 1) $\varphi = \varphi_n = \varphi_a \wedge \varphi_{n-1}$ or 2) $\varphi = \varphi_n = \varphi_a \vee \varphi_{n-1}$, where φ_a is the newly added formula and has only one predicate. Assume that $\mathcal{U}_i^+$ is the set for faulty behaviors that have been defected correctly with φ_i, and $\mathcal{U}_i^-$ is the set for behaviors that have been detected incorrectly with φ_i. $\mathcal{D}_i^+$ and $\mathcal{D}_i^-$ are for the normal behaviors, respectively. The proof for the two cases is shown as follows.

1) If φ_{n-1} detects the behaviors correctly, there exists a φ_a such that $\varphi_{n-1} \preceq \varphi_n$; else, in order to achieve $\varphi_{n-1} \preceq \varphi_n$, $\varphi_a \wedge \varphi_{n-1}$ should decrease the number of behaviors in $\mathcal{U}_{n-1}^-$ and does not decrease the number of behaviors in $\mathcal{D}_{n-1}^+$. If there exists φ_a such that $\exists\hat{\xi} \in \mathcal{U}_{n-1}^-$ and $\rho(\neg\varphi_a \vee \neg\varphi_{n-1}, \omega(\hat{\xi})) > 0$, then $\varphi_{n-1} \preceq \varphi_n$. Since the behaviors are in $\mathcal{U}_{n-1}^-$, we can always find a φ_a such that $\rho(\neg\varphi_a,\omega(\hat{\xi})) > 0 \Rightarrow \rho(\neg\varphi_a \vee \neg\varphi_{n-1},\omega(\hat{\xi})) > 0$. Therefore, $\varphi_{n-1} \preceq \varphi_n$.
2) If φ_{n-1} detects the behaviors correctly, there exists a φ_a such that $\varphi_{n-1} \preceq \varphi_n$; else, $\varphi_a \vee \varphi_{n-1}$ decreases the number of behaviors $\mathcal{D}_{n-1}^-$ and does not decrease the number of behaviors in $\mathcal{U}_{n-1}^+$. If there exists φ_a such that $\exists\hat{\xi} \in \mathcal{D}_{n-1}^-$ and $\rho(\varphi_a \vee \varphi_{n-1}, \omega(\hat{\xi})) > 0$, then $\varphi_{n-1} \preceq \varphi_n$. Since the behaviors are in $\mathcal{D}_{n-1}^-$, we can always find a φ_a such that $\rho(\varphi_a,\omega(\hat{\xi})) > 0 \Rightarrow \rho(\varphi_a \vee \varphi_{n-1},\omega(\hat{\xi})) > 0$. Therefore, $\varphi_{n-1} \preceq \varphi_n$.

Therefore, there exists a sequence of STL formulas such that $\varphi_1 \preceq \varphi_2 \preceq \cdots \preceq \varphi_n \preceq \varphi$. Moreover, the number of predicates satisfies the following property:

$$|\varphi_n| - |\varphi_i| = n - i \tag{40}$$

where $1 \leq i \leq n$. The theorem has been proven. □

Proof of Theorem 5

Proof: Proof for Statement 1): Based on the definition of partial order and the constraints in (27a) and (27b), statement 1) holds.

Proof for Statement 2): When we solve the optimization problem in line 8 of Algorithm 2, we have, $\forall\hat{s}_{\xi} \in \mathcal{D}^+, \lambda(\hat{\xi},\varphi_{\text{or}},\sigma,\delta)) = SS \wedge RS$. Since the trajectories in $\mathcal{U}^-$ and $\mathcal{D}^-$ are detectable, φ_t will increase the value of $J(\varphi_t)$. Based on Theorem 4, if we ignore $\mathcal{D}^-$, Algorithm 2 will eventually decrease $\mathcal{U}^-$ to zero. Similarly, when we solve the optimization problem in line 11 of Algorithm 2, we have, $\forall\hat{\xi} \in \mathcal{U}^+, \lambda(\hat{s}_{\xi},\varphi_{\text{and}},\sigma,\delta)) = SV \wedge RV$. Since the trajectories in $\mathcal{U}^-$ and $\mathcal{D}^-$ are detectable, φ_t will increase the value of $J(\varphi)$. If we ignore $\mathcal{U}^-$, Algorithm 2 will eventually decrease $\mathcal{D}^-$ to zero. Therefore, the theorem has been proven. □

REFERENCES

[1] A. Girard, G. Gossler, and S. Mouelhi, "Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models," *IEEE Trans. Autom. Control*, vol. 61, no. 6, pp. 1537–1549, Jun. 2016.

[2] L. Szabo and M. Ruba, "Segmental stator switched reluctance machine for safety-critical applications," *IEEE Trans. Ind. Appl.*, vol. 48, no. 6, pp. 2223–2229, Nov. 2012.

[3] H. Yang, Y. Zhang, and B. Jiang, "Tolerance of intermittent faults in spacecraft attitude control: Switched system approach," *IET Control Theory Appl.*, vol. 6, no. 13, pp. 2049–2056, Sep. 2012.

[4] C. Xiao, M. Yu, B. Zhang, H. Wang, and C. Jiang, "Discrete component prognosis for hybrid systems under intermittent faults," *IEEE Trans. Autom. Sci. Eng.*, early access, Sep. 15, 2020, doi: 10.1109/TASE.2020.3017755.

[5] M. Yu, C. Xiao, and B. Zhang, "Event-triggered discrete component prognosis of hybrid systems using degradation model selection," *IEEE Trans. Ind. Electron.*, early access, Oct. 21, 2020, doi: 10.1109/TIE.2020.3031515.

[6] D. Codetta-Raiteri and L. Portinale, "Dynamic Bayesian networks for fault detection, identification, and recovery in autonomous spacecraft," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 13–24, Jan. 2015.

[7] A. L. Christensen, R. O'Grady, M. Birattari, and M. Dorigo, "Fault detection in autonomous robots based on fault injection and learning," *Auto. Robots*, vol. 24, no. 1, pp. 49–67, Jan. 2008.

[8] C. Seatzu, D. Corona, A. Giua, and A. Bemporad, "Optimal control of continuous-time switched affine systems," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 726–741, May 2006.

[9] A. Farraj, E. Hammad, A. A. Daoud, and D. Kundur, "A game-theoretic analysis of cyber switching attacks and mitigation in smart grid systems," *IEEE Trans. Smart Grid*, vol. 7, no. 4, pp. 1846–1855, Jul. 2016.

[10] G. Chen, M. Liu, and Z. Kong, "Temporal-logic-based semantic fault diagnosis with time-series data from industrial Internet of Things," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4393–4403, May 2021.

[11] X. Su, P. Shi, L. Wu, and Y.-D. Song, "Fault detection filtering for nonlinear switched stochastic systems," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1310–1315, May 2016.

[12] D. Wang, W. Wang, and P. Shi, "Robust fault detection for switched linear systems with state delays," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 39, no. 3, pp. 800–805, Jun. 2009.

[13] X. Liu and S. Yuan, "Reduced-order fault detection filter design for switched nonlinear systems with time delay," *Nonlinear Dyn.*, vol. 67, no. 1, pp. 601–617, Jan. 2012.

[14] W. Xiang, J. Xiao, and M. N. Iqbal, "Robust fault detection for a class of uncertain switched nonlinear systems via the state updating approach," *Nonlinear Anal., Hybrid Syst.*, vol. 12, pp. 132–146, May 2014.

[15] G.-X. Zhong and G.-H. Yang, "Robust control and fault detection for continuous-time switched systems subject to a dwell time constraint," *Int. J. Robust Nonlinear Control*, vol. 25, no. 18, pp. 3799–3817, Dec. 2015.

[16] D. Wang, P. Shi, and W. Wang, "Robust fault detection for continuous-time switched delay systems: An linear matrix inequality approach," *IET Control Theory Appl.*, vol. 4, no. 1, pp. 100–108, Jan. 2010.

[17] L. Tang and J. Zhao, "Switched threshold-based fault detection for switched nonlinear systems with its application to Chua's circuit system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 2, pp. 733–741, Feb. 2019.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[18] F. Zhu and F. Cen, "Full-order observer-based actuator fault detection and reduced-order observer-based fault reconstruction for a class of uncertain nonlinear systems," *J. Process Control*, vol. 20, no. 10, pp. 1141–1149, Dec. 2010.
[19] D. Du, S. Xu, and V. Cocquempot, "Fault detection for nonlinear discrete-time switched systems with persistent dwell time," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2466–2474, Aug. 2018.
[20] K. Liu, H. Lin, Z. Fei, and J. Liang, "Spatially–temporally online fault detection using timed multivariate statistical logic," *Eng. Appl. Artif. Intell.*, vol. 65, pp. 51–59, Oct. 2017.
[21] Z. Xu and A. A. Julius, "Census signal temporal logic inference for multiagent group behavior analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 264–277, Jan. 2018.
[22] S. Bufo, E. Bartocci, G. Sanguinetti, M. Borelli, U. Lucangelo, and L. Bortolussi, "Temporal logic based monitoring of assisted ventilation in intensive care patients," in *Proc. Int. Symp. Leveraging Appl. Formal Methods, Verification Validation*. Berlin, Germany: Springer, 2014, pp. 391–403.
[23] G. Bombara, C.-I. Vasile, F. Penedo, H. Yasuoka, and C. Belta, "A decision tree approach to data classification using signal temporal logic," in *Proc. 19th Int. Conf. Hybrid Syst., Comput. Control*, Apr. 2016, pp. 1–10.
[24] Z. Xu and A. A. Julius, "Robust temporal logic inference for provably correct fault detection and privacy preservation of switched systems," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3010–3021, Sep. 2019.
[25] Z. Kong, A. Jones, and C. Belta, "Temporal logics for learning and detection of anomalous behavior," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1210–1222, Mar. 2017.
[26] J. Poon, P. Jain, I. C. Konstantakopoulos, C. Spanos, S. K. Panda, and S. R. Sanders, "Model-based fault detection and identification for switching power converters," *IEEE Trans. Power Electron.*, vol. 32, no. 2, pp. 1419–1430, Feb. 2017.
[27] I. Steinwart and A. Christmann, *Support Vector Machines*. Berlin, Germany: Springer, 2008.
[28] J. Wang, A. Hertzmann, and D. J. Fleet, "Gaussian process dynamical models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1441–1448.
[29] A. Donzé, "On signal temporal logic," in *Proc. Int. Conf. Runtime Verification*. Berlin, Germany: Springer, 2013, pp. 382–383.
[30] Y. Deng, A. D'Innocenzo, M. D. D. Benedetto, S. D. Gennaro, and A. A. Julius, "Verification of hybrid automata diagnosability with measurement uncertainty," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 982–993, Apr. 2016.
[31] D. Burago, I. D. Burago, Y. Burago, S. A. Ivanov, and S. Ivanov, *A Course in Metric Geometry*, vol. 33. Providence, RI, USA: American Mathematical Society, 2001.
[32] A. Girard and G. J. Pappas, "Approximate bisimulation relations for constrained linear systems," *Automatica*, vol. 43, no. 8, pp. 1307–1317, Aug. 2007.
[33] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 782–798, May 2007.
[34] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 4661–4666.
[35] G. Chen, Z. Sabato, and Z. Kong, "Active learning based requirement mining for cyber-physical systems," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 4586–4593.
[36] G. Chen, M. Liu, and J. Chen, "Frequency-temporal-logic-based bearing fault diagnosis and fault interpretation using Bayesian optimization with Bayesian neural networks," *Mech. Syst. Signal Process.*, vol. 145, Nov. 2020, Art. no. 106951.
[37] T. Zhang and G. Feng, "Output tracking of piecewise-linear systems via error feedback regulator with application to synchronization of nonlinear Chua's circuit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 8, pp. 1852–1863, Aug. 2007.

Gang Chen received the bachelor's and master's degrees in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012 and 2015, respectively, and the Ph.D. degree in mechanical and aerospace engineering from the University of California at Davis, Davis, CA, USA, in 2020.

He is currently a Visiting Scholar with the Department of Automation, School of Electrical and Information Engineering, Tianjin University, Tianjin, China, and also a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include machine learning, formal methods, control, signal processing, and fault diagnosis.

Peng Wei (Member, IEEE) received the B.S. degree in thermal and power engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015, and the M.S. degree in mechanical engineering from Arizona State University, Tempe, AZ, USA, in 2016. He is currently pursuing the Ph.D. degree in mechanical and aerospace engineering with the University of California at Davis, Davis, CA, USA.

His research interests include unmanned aerial vehicle (UAV) control, optimal control, and reinforcement learning.

Mei Liu received the B.Sc. degree in mathematics from the China University of Mining and Technology, Xuzhou, China, in 2012, and the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2017.

From August 2017 to February 2019, she was a Research Associate/Assistant with The University of Hong Kong, Hong Kong, and The Hong Kong Polytechnic University, Hong Kong. She is currently an Associate Professor with the Department of Automation, Tianjin University, Tianjin, China. Her current research interests include negative imaginary systems, positive real systems, state-space symmetric systems, and fault diagnosis.

Authorized licensed use limited to: Nanyang Technological University. Downloaded on May 09,2021 at 02:08:24 UTC from IEEE Xplore. Restrictions apply.