2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Fault Detectability Analysis of Switched Affine Systems with Linear Temporal Logic Constraints

Liren Yang Necmiye Ozay

Abstract— **In this paper, we consider detectability analysis for faults in systems governed by switched affine dynamics. By a fault, we mean a sudden and permanent change in the system dynamics. Given the model of the healthy system, such a fault can be detected via model invalidation, i.e., by collecting past observations over a finite horizon and checking whether these observations can be generated by the healthy system model. Whenever the faulty system model is also available, it is possible to find T, the minimum length of the horizon, with which the fault is guaranteed to be detected eventually (with a T-step delay at most). The procedure for finding such T is known as the fault detectability analysis, and can be accomplished by solving a mixed integer linear program (MILP) for switched affine systems. The main contribution of this work is to show the possibility of reducing the value of T, by augmenting the fault detectability analysis with additional linear temporal logic (LTL) constraints on the switching signals, if any. We express the LTL constraints (restricted in a finite horizon) with a nondeterministic finite state machine called a monitor, which is then transformed into a set of mixed integer linear constraints that can be easily integrated in the MILP used for the detectability analysis. The effectiveness of the proposed approach is illustrated with a drone altitude consensus protocol with switching communication topology.**

I. Introduction

Run-time anomaly and fault detection is crucial for safe operation of cyber-physical systems so that potential faults can be quickly detected and contained before they lead to system-wide catastrophes. Early anomaly and fault detection is even more important for situations like space missions where maintenance is very costly if at all possible. For discrete (e.g., software-based) systems, several monitoring and run-time verification techniques have been proposed [1], [7], [13], [14]. Similarly, fault detection algorithms have also been studied for continuous-state dynamical systems using ideas from machine learning, filtering or optimization [8], [4]. The goal of this paper is to bring together ideas from these different communities to more effectively detect faults for a class of hybrid systems, governed by both discrete and continuous variables.

In particular, we consider systems whose dynamics are given by a switched affine model. Such models can be used to describe, for instance, physical systems with discrete actuation or closed-loop systems with continuous plants and logic-based controllers. Switching mode captures the discrete, logic-based variables. We further assume that when the system operation is normal (i.e., the system is healthy), the switching mode signal satisfies a certain LTL formula; and in case of faults, it satisfies a different (possibly trivial) formula, capturing the potential switching patterns during normal operation and anomalies, respectively. Given also a potentially uncertain fault model, our goal is to analyze if the fault can be detected and if so, how fast it can be detected.

We approach this problem from the perspective of model invalidation [6], [12], [15]. Model invalidation problem is to check whether some given input output data can be explained by a given model. This is very similar to the model conformance problem studied in computer science community [2], [16], in particular to the input/output conformance for discrete-time systems [16], [17]. Given two models, one for the healthy system and one for the faulty system, it is also possible to ask whether, in the worst-case, the faulty system can be distinguished from the healthy system in finite time within the framework of model invalidation [6]. This is called detectability analysis, and it essentially amounts to checking whether the reachable sets of the healthy and faulty models become disjoint at some time. In [5], [6], it is shown that for switched affine systems, this analysis can be reduced to a first order logic statement, which can be checked with MILP or SMT solvers.

In essence, the approaches mentioned above can all be viewed as worst case analysis. They may sometimes conclude conservatively that a fault is not detectable in finite time in the worst case, while this may not be true in practice. For a less conservative detectability analysis, these approaches hence need to be enhanced with extra side information, if any. From a control point of view, it is also important to know that a fault can always be detected within finite time because this knowledge can be easily incorporated in correct-by-construction control frameworks, for example, see [18], [19].

Our main contribution in this paper is to show that combining dynamical models (given as switched affine systems) and behavioral models (given as LTL formula), one can reduce the worst-case fault detection time a receding horizon algorithm guarantees. The fault detection with such LTL switching mode constraints are motivated by many practical scenarios. For example, consider a swarm of robots trying to achieve consensus over a communication network. In particular, they have to switch between several different subnetworks due to limited communication bandwidth. If the fault leads to one subnetwork to be broken, the fault will not be revealed unless the broken subnetwork is eventually used. This hence indicates that the worst-case detection delay is

The authors are with the Dept. of Electrical Engineering and Computer Science, Univ. of Michigan, Ann Arbor, MI 48109, USA yliren,necmiye@umich.edu. This work is supported by an Early Career Faculty grant from NASA's Space Technology Research Grants Program.

978-1-7281-1398-2/19/$31.00 ©2019 IEEE

infinitely long without extra specifications w.r.t. the switching pattern. As a result, we cannot confidently perform the detection with a finite memory without missing the fault. On the other hand, switching among all the subnetworks within certain amount of time is necessary for the robot swarm to achieve consensus, and such extra information can be captured by restricting the switching sequence satisfying an LTL formula, with which we may reduce the worst-case detection delay to finite time. In this scenario, knowledge of the fact that the robots, trying to achieve consensus, will visit each subnetwork frequently enough, constitutes a side information.

In this work, we show that detectability analysis in this setting can be conducted by creating a monitor finite state machine for the LTL formula and encoding the restriction the formula imposes on the system behavior as mixed integer linear constraints, which is then incorporated into the MILP for the offline worst-case detectability analysis. Using a receding horizon approach, we avoid automaton determinization. Therefore, our method provides a good trade-off between conservatism and complexity. These ideas are illustrated with an example where a collection of unmanned aerial vehicles are implementing a consensus protocol over a communication network with time-varying connectivity. We show how detectability can be guaranteed when network connectivity patterns change using the proposed approach.

II. Preliminaries

We first define some basic notations. Let $\mathbb{R}^n$ be the n-dimensional Euclidean space and $\mathbb{N}$ be the set of nonnegative integers. For a given set Σ, Σ^* denotes the set of all finite words over Σ, and Σ^ω denotes the set of all ω-words over Σ. Given two positive integer a, b such that $a < b$, we use $[\![a, b]\!]$ as a short notation of the set $\{c \in \mathbb{N} \mid a \leq c \leq b\}$.

A. Fault Detection for Switched Affine Systems via Model Invalidation

1) Switched Affine Systems: A discrete-time switched affine system $\mathcal{S}$ is described by the following difference equations:

$$\begin{aligned} x_{t+1} &= A_{s_t}x_t + B_{s_t}u_t + E_{s_t}w_t + K_{s_t}, \\ y_t &= C_{s_t}x_t + D_{s_t}u_t + F_{s_t}v_t, \end{aligned} \tag{1}$$

where

- $x \in X \subseteq \mathbb{R}^n$ is the **unobserved** internal state,
- $u \in U \subseteq \mathbb{R}^m$ is the **observed** input,
- $w \in W \subseteq \mathbb{R}^l$ is the **unobserved** input,
- $y \in Y \subseteq \mathbb{R}^p$ is the **observed** output,
- $v \in V \subseteq \mathbb{R}^q$ is the **unobserved** measurement noise (can be viewed as input),
- s is the **observed** switching mode from a finite set $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$.

We also assume that sets U, V, W, X, Y are polytopes in their spaces, and that A_s, B_s, C_s, D_s, E_s, F_s, K_s are matrices with proper sizes.

2) Guaranteed Fault Detection via Model Invalidation: In this work, we consider fault detection of switched affine systems. By a fault, we mean a sudden and permanent change of the system dynamics in Eq. (1), due to physical component failures or extreme operating conditions. Such changes can be reflected by dynamics being governed by different system matrices, and by having larger admissible uncertainty set V and W. Since the uncertainty w and v may "hide" a fault in the worst case, the behavior of the faulty system may not be distinguishable from the healthy one immediately after the fault occurs. Our goal is to detect the fault occurrence as soon as possible. In particular, the correctness of the detection needs to be guaranteed, meaning that the fault must have already happened, once detected.

It is shown in [6] that such guaranteed fault detection can be done using a model invalidation approach. The model invalidation problem addresses the following question: at time instant t_0, given a sequence $\{u_t, s_t, y_t\}_{t=t_0-N+1}^{t_0}$ of past inputs and outputs over a finite window of length N, can we find an admissible unobserved sequence $\{x_t, w_t, v_t\}_{t=t_0-N+1}^{t_0}$ such that the output $\{y_t\}_{t=t_0-N+1}^{t_0}$ is indeed generated by the system in Eq. (1) under input $\{u_t, s_t, w_t, v_t\}_{t=t_0-N+1}^{t_0}$? If no such unobserved sequences can be found, the actual observation cannot possibly be generated by the healthy system model (i.e., the model is invalidated). We can hence claim that a fault that changes the system dynamics must have occurred within the examined time window.

For switched affine systems, the model invalidation problem can be formulated as a linear program (LP)

$$\begin{aligned} \text{find} \quad & \{x_t, w_t, v_t\}_{t=t_0-N+1}^{t_0} \\ \text{s.t.} \quad & x_{t+1} = \textstyle\sum_{k=1}^{|\Sigma|} a_k^t \Big(A_{\sigma_k}x_t + B_{\sigma_k}u_t + E_{\sigma_k}w_t + K_{\sigma_k} \Big), \\ & \forall t \in [\![t_0 - N + 1, t_0 - 1]\!], \\ & y_t = \textstyle\sum_{k=1}^{|\Sigma|} a_k^t \Big(C_{\sigma_k}x_t + D_{\sigma_k}u_t + F_{\sigma_k}v_t \Big), \\ & \forall t \in [\![t_0 - N + 1, t_0]\!], \\ & x_t \in X, w_t \in W, v_t \in V, \forall t \in [\![t_0 - N + 1, t_0]\!], \end{aligned} \tag{2}$$

where a_k^t is a binary indicator that takes value 1 if and only if (iff) $s_t = \sigma_k$. Note that $\{u_t, y_t\}_{t=t_0-N+1}^{t_0}$ and a_k^t (which is known from $\{s_t\}_{t=t_0-N+1}^{t_0}$) are all parameters rather than variables in the above feasibility problem, as $\{u_t, s_t, y_t\}_{t=t_0-N+1}^{t_0}$ are observed at each time. This means the feasibility problem in Eq. (2) does not contain integer variables and is hence an LP.

To perform model invalidation based fault detection at run-time, one needs to update the time window (i.e., horizon) to incorporate newly collected data. As pointed out in [6], there are two ways of changing the horizon at run-time: one is called the growing horizon scheme (Fig. 1, left) and the other is known as a receding horizon scheme (Fig. 1, right). With the growing horizon scheme, we start at time $t = 0$ with a horizon of length $N = 0$, and increase N by one at each time step. In this case, $N \to \infty$ as time grows. Under the receding horizon scheme, we stop growing the horizon length whenever it reaches a certain value, and we

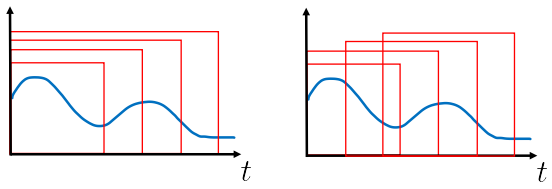

Fig. 1: Growing horizon scheme (left) versus receding horizon scheme (right). The red boxes marked the growing/shifted time window.

start to shift the time window after that. That is, at every time instant t_0, we collect the most recent history of the observed variables from time window $[\![t_0 - N + 1, t_0]\!]$ and perform the above model invalidation procedure. If the system is invalidated, we claim a fault; otherwise we shift the time window to $[\![t_0 - N + 2, t_0 + 1]\!]$ and repeat the procedure with the updated data in the shifted window. Theoretically, the growing horizon scheme may lead to an earlier detection than the receding horizon scheme because the latter one drops older observations. We say the receding horizon detector is more conservative compared to the growing horizon detector in the sense the former may miss a fault that is detectable by the latter. However, the growing horizon scheme is not practical because it requires the memory to grow to infinity. We hence always implement the receding horizon scheme in practice to keep the memory finite.

3) Detectability Analysis: Note that the fault detection technique via model invalidation may not be complete, in the sense that a fault may remain undetected indefinitely. There are two sources of this: (i) the fault dynamics can be inherently indistinguishable from the nominal dynamics, (ii) the invalidation process is conservative, e.g., due to using a fixed horizon. For the latter issue, a longer window tends to make the detector "closer" to being complete. On the other hand, if the models of both the healthy system $\mathcal{S}$ and the faulty system $\mathcal{S}^{\mathrm{f}}$ are known, it is possible to verify if the detection is complete with a given window length N. That is, if a fault occurs whether it will be detected within N time steps by the receding horizon detector. We call a healthy-faulty system pair $(\mathcal{S}, \mathcal{S}^{\mathrm{f}})$ to be "N-detectable" if this is the case.

For a given healthy-faulty system pair $(\mathcal{S}, \mathcal{S}^{\mathrm{f}})$ and a positive integer N, the detectability analysis answer the following question: is system pair $(\mathcal{S}, \mathcal{S}^{\mathrm{f}})$ N-detectable? If yes, what is the minimal N such that the pair is N-detectable? From a theoretical point of view, it is important if we can prove N-detectability of a system pair because it allows us to use a receding horizon detector without missing any faults due to its conservativeness compared to the growing horizon detector. From a practical point of view, it is also important to find the minimal N so that the receding horizon detector does not need to keep an unnecessarily long memory.

To analyze the detectability of system pair $(\mathcal{S}, \mathcal{S}^{\mathrm{f}})$, we construct the so called N-behavior set $\mathfrak{B}_N(\mathcal{S})$ (and $\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}})$, respectively), i.e., the set of all observed input-output sequences of length N that can be possibly generated by the healthy system $\mathcal{S}$ (or the faulty system $\mathcal{S}^{\mathrm{f}}$, respectively), and check if the two sets intersect. If $\mathfrak{B}_N(\mathcal{S}) \cap \mathfrak{B}_N(\mathcal{S}^{\mathrm{f}}) = \emptyset$, then the healthy and the faulty behavior must differ within N time steps. In this case, the minimal horizon length T that is necessary for the detection to be complete (i.e., $T := \min\{N \mid \mathfrak{B}_N(\mathcal{S}) \cap \mathfrak{B}_N(\mathcal{S}^{\mathrm{f}}) = \emptyset\}$) can be computed by a line search over ascending N, starting from $N = 1$.

If the dynamics of system $\mathcal{S}$ satisfies Eq. (1), the N-behavior set of system $\mathcal{S}$ can be described by mixed integer linear constraints. In this case, $\mathfrak{B}_N(\mathcal{S}) \cap \mathfrak{B}_N(\mathcal{S}^{\mathrm{f}}) = \emptyset$ is equivalent to a MILP being infeasible. Formally, $\mathfrak{B}_N(\mathcal{S})$ is defined by Eq. (3). where the constraints in Eq. (3) can be expressed with exactly the same set of the formulas in Eq. (2) (after shifting the time window to $[\![1, N]\!]$), except that now the observed sequences $\{u_t, y_t\}_{t=1}^N$ and the auxiliary binary variables a_k^t are also variables rather than parameters of the constraints, and that u_t, a_k^t, y_t must satisfy

$$u_t \in U, \quad y_t \in Y, \quad \forall t \in [\![1, N]\!], \tag{4}$$

$$a_k^t \in \{0, 1\}, \quad \forall t \in [\![1, N]\!], k \in [\![1, |\Sigma|]\!], \tag{5}$$

and a_k^t must also satisfy

$$\sum_{\sigma_k \in \Sigma} a_k^t = 1, \quad \forall t \in [\![1, N]\!]. \tag{6}$$

Note that with a_k^t being variables, the constraints describing $\mathfrak{B}_N(\mathcal{S})$ now contain bilinear terms $a_k^t u_t, a_k^t x_t$, $a_k^t w_t$, $a_k^t v_t$ (see Eq. (2)). These bilinear constraints can be transformed into linear ones by introducing some continuous-valued auxiliary variables, which leads to a set of (mixed integer) linear constraints. The detailed transformation procedure can be found in [5]. To simplify the notations, we will denote the obtained overall mixed integer linear constraints by

$$\mathbf{H}_N^{\mathcal{S}}\left(\left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t, w_t, v_t, \right\}_{t=1}^N, \boldsymbol{\xi}_{\mathrm{ob}}, \boldsymbol{\xi}_{\mathrm{un}}\right) \le 0,$$
$$a_k^t \in \{0, 1\}, \quad \forall t \in [\![1, N]\!], k \in [\![1, |\Sigma|]\!], \tag{7}$$

where $\boldsymbol{\xi}_{\mathrm{ob}}$ is the continuous-valued auxiliary variable that comes from $a_k^t u_t$, and $\boldsymbol{\xi}_{\mathrm{un}}$ is the auxiliary variable that comes from $a_k^t x_t$, $a_k^t w_t$, $a_k^t v_t$, and $\mathbf{H}_N^{\mathcal{S}}$ is an affine function that depends on the system matrices of $\mathcal{S}$ and horizon length N. With this notation, $\mathfrak{B}_N(\mathcal{S}) \cap \mathfrak{B}_N(\mathcal{S}^{\mathrm{f}}) = \emptyset$ is equivalent to the following MILP being infeasible

$$\begin{aligned}
\text{find} \quad & \left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t, x_t^{\mathrm{f}}, w_t, w_t^{\mathrm{f}}, v_t, v_t^{\mathrm{f}}\right\}_{t=1}^N, \boldsymbol{\xi}_{\mathrm{ob}}, \boldsymbol{\xi}_{\mathrm{un}}, \boldsymbol{\xi}_{\mathrm{un}}^{\mathrm{f}}, \\
\text{s.t.} \quad & \mathbf{H}_N^{\mathcal{S}}\left(\left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t, w_t, v_t, \right\}_{t=1}^N, \boldsymbol{\xi}_{\mathrm{ob}}, \boldsymbol{\xi}_{\mathrm{un}}\right) \le 0, \\
& \mathbf{H}_N^{\mathcal{S}^{\mathrm{f}}}\left(\left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t^{\mathrm{f}}, w_t^{\mathrm{f}}, v_t^{\mathrm{f}}, \right\}_{t=1}^N, \boldsymbol{\xi}_{\mathrm{ob}}, \boldsymbol{\xi}_{\mathrm{un}}^{\mathrm{f}}\right) \le 0, \\
& a_k^t \in \{0, 1\}, \quad \forall t \in [\![1, N]\!], k \in [\![1, |\Sigma|]\!] \le 0.
\end{aligned} \tag{8}$$

B. Linear Temporal Logic Constraints on Switching Modes

In this work we consider fault detectability analysis of switched affine systems whose mode sequences $\mathbf{s} = s_1 s_2 s_3 \ldots$ must satisfy certain LTL formulas. In particular, our goal is to show how such side information can improve detectability analysis. In what follows, we briefly recall LTL and some related concepts from automata theory that will be useful to encode the LTL constraints on $\mathbf{s}$.

$$\mathfrak{B}_N(\mathcal{S}) = \left\{ \begin{array}{c} \{u_t, s_t, y_t\}_{t=1}^N \in \\ (U \times \Sigma \times Y)^N \end{array} \;\middle|\; \begin{array}{c} \exists \{x_t, w_t, v_t\}_{t=1}^N \in (X \times W \times V)^N : \\ \{u_t, v_t, w_t, x_t, y_t, s_t\}_{t=1}^N \text{ satisfy } \mathcal{S}\text{'s dynamics} \end{array} \right\}, \tag{3}$$

1) Linear Temporal Logic: We first give the syntax and the semantics of LTL.

a) Syntax: Let Σ be a finite set of modes, the syntax of LTL formulas over Σ is given by

$$\varphi ::= \sigma \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \; \mathcal{U} \; \varphi_2 \tag{9}$$

where $\sigma \in \Sigma$. With the grammar given in Eq. (9), we define $\varphi_1 \wedge \varphi_2 := \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \to \varphi_2 := \neg\varphi_1 \vee \varphi_2$, $\Diamond\varphi := True \; \mathcal{U} \; \varphi$, $\Box\varphi := \neg\Diamond\neg\varphi$,

b) Semantics: Let $\mathbf{s}$ be an ω-word over Σ (i.e., $\mathbf{s} \in \Sigma^\omega$), whose t^{th} element is denoted by $\mathbf{s}(t)$. We interpret an LTL formula over such sequences as follows:

- $\mathbf{s}, t \models \sigma$ iff $\mathbf{s}(t) = \sigma$,
- $\mathbf{s}, t \models \neg\varphi$ iff $\mathbf{s}, t \not\models \varphi$,
- $\mathbf{s}, t \models \varphi_1 \vee \varphi_2$ iff $\mathbf{s}, t \models \varphi_1$ or $\mathbf{s}, t \models \varphi_2$,
- $\mathbf{s}, t \models \bigcirc \varphi$ iff $\mathbf{s}, t+1 \models \varphi$,
- $\mathbf{s}, t \models \varphi_1 \; \mathcal{U} \; \varphi_2$ iff $\exists s \geq t : \mathbf{s}, s \models \varphi_2$ and $\forall r < s : \mathbf{s}, r \models \varphi_1$.

We write $\mathbf{s} \models \varphi$ if $\mathbf{s}, 1 \models \varphi$.

2) Monitor: We introduce several concepts related to LTL monitoring that will be used to encode the LTL constraints in the fault detectability analysis.

Definition 1: Let φ be an LTL formula. A finite word $\mathbf{p} \in \Sigma^*$ is called a *bad prefix* of φ if[1] for all $\mathbf{s} \in \Sigma^\omega$, $\mathbf{ps} \not\models \varphi$, where $\mathbf{ps}$ is the ω-word obtained by concatenating $\mathbf{s}$ to $\mathbf{p}$. Otherwise we call $\mathbf{p}$ a *(valid) prefix* of φ.

Definition 2: Given and LTL formula φ, a *monitor* $\mathcal{M}^\varphi$ is a tuple $(\Sigma, Q, Q_{\text{init}}, \delta)$, where Σ is a finite set of letters, Q is a finite set of states, $Q_{\text{init}} \subseteq Q$ is a set of initial states, and partial function $\delta : Q \times \Sigma \to 2^Q$ is the nondeterministic[2] transition map. Moreover, $\mathcal{M}^\varphi$ satisfies the following condition: a finite word $\mathbf{p} = p_1 p_2 \dots p_N$ is a valid prefix of φ if and only if there exists $\mathbf{q} = q_1 q_2 \dots q_{N+1} \in Q^{N+1}$ such that $q_1 \in Q_{\text{init}}$ and $q_{i+1} \in \delta\big(q_i, p_i\big)$ for $i \in [\![1, N]\!]$.

The monitor finite state machine $\mathcal{M}^\varphi$ can be viewed as a model that generates sequences exactly from $\{\mathbf{p} \in \Sigma^* \mid \mathbf{p} \text{ is a valid prefix of } \varphi\}$. It is well known that a monitor $\mathcal{M}^\varphi$ can be constructed for every LTL formula φ [3].

C. Fault Detection versus Run-time Verification

We would like to point out the connection between the model invalidation based fault detection and run-time verification.

In run-time verification, we are given an LTL formula and desire to verify if a sequence $\mathbf{s} = s_1 s_2 \dots s_M \in \Sigma^*$ is a bad prefix of φ. To this end, we construct monitor $\mathcal{M}^\varphi = (\Sigma, Q, Q_{\text{init}}, \delta)$ and check if $\mathbf{s}$ leads to a valid run on $\mathcal{M}^\varphi$. The monitor $\mathcal{M}^\varphi$ can be viewed as a model with internal state set $q \in Q$, and the sequence $\mathbf{s}$ can be viewed as an N-behavior that may be generated by the model $\mathcal{M}^\varphi$, under some admissible nondeterministic transitions. The run-time verification procedure reduces to computing a set Q_N of the reachable states of $\mathcal{M}^\varphi$ after reading $s_1 s_2 \dots s_N$ and checking if $Q_N = \emptyset$ for some $N \leq M$. If yes, the anomaly (i.e., violation of φ) is claimed.

In the model invalidation based fault detection, we check if a sequence of observation $\{u_t, s_t, y_t\}_{t=1}^N$ is generated by a model described by Eq. (1), with internal state x that is analogue to q of a monitor, and with bounded uncertainty w, v that are analogue to the nondeterministic transition of the monitor. Very similar to the idea of run-time verification, the model invalidation reduces to checking the emptiness of a set X_N, which consists of the healthy system's reachable states that are consistent with observation $\{u_t, s_t, y_t\}_{t=1}^N$ under some admissible uncertainty sequence $\{w_t, v_t\}_{t=1}^N$. In fact, set X_N can be viewed as the projection (onto the internal state space) of a high dimensional polytope that is described exactly by the linear constraints in Eq. (2). However, unlike the case in the run-time verification where the internal state set Q_N is finite no matter what N is, X_N consists of infinite states and its representation complexity (i.e., the number of linear constraints required to describe X_N) may blow up as $N \to \infty$. This can be viewed as another interpretation of the issue that a growing horizon detector requires infinite memory. Hence we have to use the receding horizon scheme to compute an over approximation of X_N, whose description complexity is bounded. Detectability analysis tells us how tight this over approximation should be so that no fault is missed by the detector.

III. Problem Description

In this work we consider fault detectability analysis for switched affine systems whose mode sequences satisfy certain LTL constraints. To define the problem, we first define how the LTL-based side-information can be incorporated in the behavior description of the system. Let $\mathcal{S}$ be the healthy system and $\mathcal{S}^{\text{f}}$ be the faulty system, and φ and φ^{f} be the corresponding LTL formulas. We assume the side-information to be of following form:

(i) if the fault never occurs, $\mathbf{s}, 1 \models \varphi$;
(ii) if the fault occurs at time t_{o}, then $s_1 s_2 \dots s_{t_{\text{o}}-1}$ is a valid prefix of φ, and $\mathbf{s}, t_{\text{o}} \models \varphi^{\text{f}}$.

As mentioned in Section II, we collect the input-output pairs of length N in the receding horizon fault detection process. The above extra LTL constraints further restrict the sets of N-behaviors of the healthy and faulty systems, which now take the form in Eq. (10) and (11)

[1]Note that φ has no bad prefixes if it specifies a liveness property, hence a finite word being a bad prefix of φ is equivalent to the word being a bad prefix of the safety closure of the language accepted by φ.

[2]Often times in the literature, the term "monitor" are used to refer to the deterministic finite transition system that are determinzed from $\mathcal{M}$ via standard power set construction. Here we follow [3] and use the term "monitor" to refer to the nondeterministic finite transition system.

$$\mathfrak{B}_N(\mathcal{S},\varphi)=\left\{\{u_t,s_t,y_t\}_{t=1}^N \;\middle|\; \begin{array}{c}\exists\{x_t,w_t,v_t\}_{t=1}^N\in(X\times W\times V)^N:\\ \{u_t,v_t,w_t,x_t,y_t,s_t\}_{t=1}^N \text{ satisfy } \mathcal{S}\text{'s dynamics}\\ \boxed{\exists \mathbf{p}\in\Sigma^*,\mathbf{w}\in\Sigma^\omega:\mathbf{p}s_1s_2\dots s_N\mathbf{w}\models\varphi}\end{array}\right\}, \tag{10}$$

$$\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})=\left\{\{u_t,s_t,y_t\}_{t=1}^N \;\middle|\; \begin{array}{c}\exists\{x_t,w_t,v_t\}_{t=1}^N\in(X^{\mathrm{f}}\times W^{\mathrm{f}}\times V^{\mathrm{f}})^N:\\ \{u_t,v_t,w_t,x_t,y_t,s_t\}_{t=1}^N \text{ satisfy } \mathcal{S}^{\mathrm{f}}\text{'s dynamics}\\ \boxed{\exists \mathbf{w}\in\Sigma^\omega:s_1s_2\dots s_N\mathbf{w}\models\varphi^{\mathrm{f}}}\end{array}\right\}. \tag{11}$$

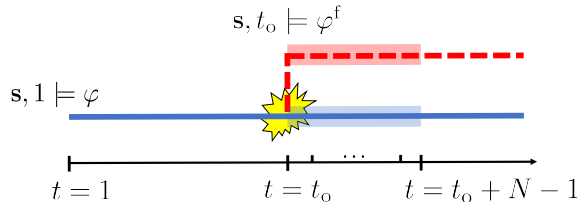

Fig. 2: Illustration of the timeline in the healthy and faulty case.

The key difference between the definitions of $\mathfrak{B}_N(\mathcal{S},\varphi)$ and $\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})$ is regarding the constraints on the N-sequence of modes, which are highlighted with the boxes. Fig. 2 shows an illustration that may help understanding this difference. The blue line represents the switching sequence when the system is always healthy, whereas the dashed red line represents the switching sequence assuming the fault occurs at time t_{o}. The shaded region highlights the switching mode sequence within time window $[\![t_{\mathrm{o}},t_{\mathrm{o}}+N-1]\!]$, and our goal is to check if the behavior generated by the healthy system under the blue shaded mode sequence differs from the behavior generated by the red faulty system under the red shaded mode sequence. Since we require $\mathbf{s},t_{\mathrm{o}}\models\varphi^{\mathrm{f}}$, this suggests that the most recent N-segment of the mode sequence (the red shaded area) must be a valid prefix of φ^{f}. This hence leads to the boxed condition in Eq. (11). On the other hand, we require $\mathbf{s},1\models\varphi$, the N-segment represented by the blue shaded region can be completed into a valid prefix of φ by adding $\mathbf{p}\in\Sigma^*$ in the front, which leads to the condition marked by the box in Eq. (10).

We now formally state the detectability analysis problem.

Problem 1: Assume the following are given:

(i) a healthy system $\mathcal{S}$ and a faulty system $\mathcal{S}^{\mathrm{f}}$, both of which have switched affine dynamics in form of Eq. (1),

(ii) LTL formulas φ and φ^{f} that govern the switching mode sequences of the healthy and the faulty system,

(iii) a positive integer N,

determine whether $\mathfrak{B}_N(\mathcal{S},\varphi)\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})=\emptyset$.

As discussed in Section II-A.3, the minimal horizon length $T:=\min\{N\mid\mathfrak{B}_N(\mathcal{S},\varphi)\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})=\emptyset\}$ can be found through a line search over N, starting from $N=1$. The usefulness of studying Problem 1 is that the extra LTL constraints may lead to a smaller T compared to the detectability analysis without such constraints. This is because these LTL constraints further restrict the behavior set so that the healthy and faulty behaviors differ earlier. We state this result with the following proposition.

Proposition 1: Let $T_1:=\min\{N\mid\mathfrak{B}_N(\mathcal{S},\varphi)\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})=\emptyset\}$ and $T_2:=\min\{N\mid\mathfrak{B}_N(\mathcal{S})\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}})=\emptyset\}$, we have $T_1\le T_2$.

Proof: By definition (see Eq. (3), (10), (11)), we have $\mathfrak{B}_N(\mathcal{S},\varphi)\subseteq\mathfrak{B}_N(\mathcal{S})$ and $\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})\subseteq\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}})$. This means $\mathfrak{B}_N(\mathcal{S})\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}})=\emptyset\Rightarrow\mathfrak{B}_N(\mathcal{S},\varphi)\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})=\emptyset$. Hence $\{N\mid\mathfrak{B}_N(\mathcal{S})\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}})=\emptyset\}\subseteq\{N\mid\mathfrak{B}_N(\mathcal{S},\varphi)\cap\mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}})=\emptyset\}$, which implies $T_1\le T_2$. ■

IV. Solution Approach

In this section, we present a solution to Problem 1. The main challenge is to express the condition in the boxes in Eq. (10), (11) in a way that can be easily integrated in the MILP in Eq. (8). Note that MILP encoding of bounded LTL [10] is not applicable to impose the boxed constraints in Eq. (10). Our solution is to first transfer the LTL formula into a monitor that captures the boxed conditions in Eq. (10), (11) induced from the given LTL formula. We then convert the monitor into its boolean representation that can be easily expressed as mixed integer linear constraints.

A. Monitor and System Behavior Constraints

We first connect the constraints marked by the boxes in Eq. (10), (11) with a monitor.

Let φ, φ^{f} be the LTL formulas from Eq. (10), (11), and let $\mathcal{M}^{\varphi}$, $\mathcal{M}^{\varphi^{\mathrm{f}}}$ be the associated monitors. The condition marked by the box in Eq. (11) says that $s_1s_2\dots s_N$ is not a bad prefix of φ^{f}, i.e., $s_1s_2\dots s_N$ can be generated by $\mathcal{M}^{\varphi^{\mathrm{f}}}$. On the other hand, the boxed condition in Eq. (10) says that $s_1s_2\dots s_N$ can be "completed" by adding a finite prefix $\mathbf{p}\in\Sigma^*$ in the front so that $\mathbf{p}s_1s_2\dots s_N$ can be generated by $\mathcal{M}^{\varphi}$. This suggests that $s_1s_2\dots s_N$ can be generated by another monitor $\mathcal{M}^{\varphi\prime}$, which is exactly the same as $\mathcal{M}^{\varphi}$ except for the initial conditions. We formally state this fact with the following proposition.

Proposition 2: Given an LTL formula φ and $\mathcal{M}^{\varphi}=(\Sigma,Q,Q_{\mathrm{init}},\delta)$, the monitor that recognizes the valid prefixes of φ, assume that all states in Q are reachable from Q_{init}[3], the following are equivalent:

(i) there exist $\mathbf{p}\in\Sigma^*$, $\mathbf{w}\in\Sigma^\omega$ such that $\mathbf{p}s_1s_2\dots s_N\mathbf{w}\models\varphi$;

(ii) there exists $q_1q_2\dots q_{N+1}\in Q^{N+1}$ such that $q_1\in Q$, $q_{t+1}=\delta(q_t,s_t)$ for all $t\in[\![1,N+1]\!]$.

and the following are equivalent:

(iii) there exist $\mathbf{w}\in\Sigma^\omega$ such that $s_1s_2\dots s_N\mathbf{w}\models\varphi$;

[3]This assumption can be made without loss of generality because states in Q that are not reachable from Q_{init} can be removed without changing the sequences generated by $\mathcal{M}$.

(vi) there exists $q_1 q_2 \dots q_{N+1} \in Q^{N+1}$ such that $q_1 \in Q_{\text{init}}$, $q_{t+1} = \delta(q_t, s_t)$ for all $t \in [\![1, N+1]\!]$.

B. MILP Encoding of Monitor

We present a technique to encode a monitor with mixed integer linear constraints. The idea is to use Proposition 2 to convert the two boxed constraints w.r.t φ and φ^{f} from Eq. (10), (11) into two monitors, and then write the monitors in their boolean representations and convert the boolean representations into two sets of MILP constraints. Since the encoding is for the nondeterministic monitor directly, it does not require determinizing the monitor with the power set construction and hence avoids an unnecessarily large MILP.

Let $\mathcal{M}^{\varphi} = (\Sigma, Q, Q_{\text{init}}, \delta)$ be a monitor of LTL formula φ. At each time instant t, we associate each state $q_i \in Q$ with a binary variable b_i^t, which takes value 1 if the state of $\mathcal{M}^{\varphi}$ is equal to q_i at time t and takes value 0 otherwise. Similarly, we associate each letter $\sigma_k \in \Sigma$ with a binary variable a_k^t that takes value 1 iff the monitor reads letter σ_k at time t. To guarantee that the monitor's state (or the read letter) exists and is unique at any time, we impose the following constraint:

$$\forall t \in [\![1, N+1]\!] \ : \ \sum\nolimits_{i=1}^{|Q|} b_i^t = 1, \tag{12}$$

$$\forall t \in [\![1, N]\!] \ : \ \sum\nolimits_{k=1}^{|\Sigma|} a_k^t = 1. \tag{13}$$

Moreover, the state indicator b_i^{t+1} must update according to the transition relation δ of the monitor. To this end, we require the following constraints to hold:

$$\forall t \in [\![1, N]\!], i \in [\![1, |Q|]\!] \ : \ b_i^{t+1} \leq \sum_{j,k: q_i \in \delta(q_j, \sigma_k)} p_{ijk}^t, \tag{14}$$

where p_{ijk}^t is a binary variable satisfying:

$$\begin{aligned} &\forall t \in [\![1, N]\!], i, j \in [\![1, |Q|]\!], k \in [\![1, |\Sigma|]\!] \\ &\qquad \text{such that } q_i \in \delta(q_j, \sigma_k) : \\ &1 + p_{ijk}^t \geq b_j^t + a_k^t, \quad p_{ijk}^t \leq b_j^t, \quad p_{ijk}^t \leq a_k^t. \end{aligned} \tag{15}$$

It might be useful to point out that Eq. (15) forces $p_{ijk}^t = b_j^t \wedge a_k^t$. In fact, variable p_{ijk}^t can be viewed as an indicator that takes value 1 iff there is a chance that the monitor's state is taken to q_i (at time $t+1$) from q_j, by reading letter σ_k at time t. Then Eq. (14) guarantees that b_i^{t+1} is set to 1 only if there is such a chance for the state to be equal to q_i at time $t+1$. Note that b_i^{t+1} can still be zero if some $p_{ijk}^t = 1$, but Eq. (13) and (14) together guarantee that there must be one $i' \in \{i \mid \exists j, k : q_i \in \delta(q_j, \sigma_k),\ p_{ijk}^t = 1\}$ such that $b_{i'}^{t+1} = 1$. This hence captures the nondeterministic transition relation of the nondeterministic monitor $\mathcal{M}^{\varphi}$.

Note that if a transition of $\mathcal{M}^{\varphi}$ is labeled as "True", i.e., for all $\sigma_k \in \Sigma, q_i \in \delta(q_j, \sigma_k)$, then the constraint in Eq. (15) can be simply replaced by $p_{ijk}^t = b_i^t$.

Finally, we constrain that the initial states are from Q_{init}:

$$\sum_{i: q_i \in Q_{\text{init}}} b_i^1 = 1. \tag{16}$$

The correctness of the construction so far is summarized with the following proposition, which can be easily verified using Proposition 2.

Proposition 3: Let φ be an LTL formula over mode set Σ and $\mathcal{M}^{\varphi} = (\Sigma, Q, Q_{\text{init}}, \delta)$ be its monitor. For a finite word $s_1 s_2 \dots s_N \in \Sigma^*$, assume that binary variable a_k^t is such that $a_k^t = 1$ iff $s_t = \sigma_k$, then the following are equivalent:

(i) there exists $q_1 q_2 \dots q_{N+1} \in Q^{N+1}$ such that $q_1 \in Q$, $q_{t+1} = \delta(q_t, s_t)$ for all $t \in [\![1, N+1]\!]$;

(ii) there exist binary variables b_i^t, p_{ijk}^t such that together with a_k^t, Eq. (12)-(15) hold,

and the following are equivalent:

(iii) there exists $q_1 q_2 \dots q_{N+1} \in Q^{N+1}$ such that $q_1 \in Q_{\text{init}}$, $q_{t+1} = \delta(q_t, s_t)$ for all $t \in [\![1, N+1]\!]$.

(vi) there exist binary variables b_i^t, p_{ijk}^t such that together with a_k^t, Eq. (12)-(16) hold.

Remark 1: Note that if φ is in the form of conjunction of several shorter formulas φ_i, i.e., $\varphi = \bigwedge_i \varphi_i$, the overall encoding can be done by stacking the mixed integer linear constraints derived from each $\mathcal{M}^{\varphi_i}$. This may not reduce the size of MILP formulation, but is useful when the size of the monitor for the overall φ is too large and generating the monitor becomes the bottleneck.

C. Detectability Analysis Augmented with LTL Constraints

Let φ and φ^{f} be the LTL formulas that constrain the mode sequences of the healthy and faulty system respectively. Denote the constraints in Eq. (13)-(15) that is derived from φ by

$$\mathbf{G}_N^{\varphi}\left(\{a_k^t\}_{k=1,t=1}^{|\Sigma|,N}, \boldsymbol{\eta}\right) \leq 0, \tag{17}$$

where $\boldsymbol{\eta}$ is a vector obtained by stacking auxiliary binary variable b_i^t and p_{ijk}^t together, and $\mathbf{G}_N^{\varphi}$ is an affine function that depends on φ and N. Similarly, we denote the constraints in Eq. (13)-(16) that are derived from φ^{f} by

$$\mathbf{G}_N^{\varphi^{\text{f}}}\left(\{a_k^t\}_{k=1,t=1}^{|\Sigma|,N}, \boldsymbol{\eta}^{\text{f}}\right) \leq 0. \tag{18}$$

The MILP used for detectability analysis with LTL constraints can be then formulated. That is, $\mathfrak{B}_N(\mathcal{S}, \varphi) \cap \mathfrak{B}_N(\mathcal{S}^{\text{f}}, \varphi^{\text{f}}) = \emptyset$ is equivalent to the following MILP being infeasible:

$$\begin{aligned} \text{find} \quad & \left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t, x_t^{\text{f}}, w_t, w_t^{\text{f}}, v_t, v_t^{\text{f}}\right\}_{t=1}^{N}, \\ & \boldsymbol{\xi}_{\text{ob}}, \boldsymbol{\xi}_{\text{un}}, \boldsymbol{\xi}_{\text{un}}^{\text{f}}, \boldsymbol{\eta}, \boldsymbol{\eta}^{\text{f}} \\ \text{s.t.} \quad & \mathbf{H}_N^{\mathcal{S}}\left(\left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t, w_t, v_t, \right\}_{t=1}^{N}, \boldsymbol{\xi}_{\text{ob}}, \boldsymbol{\xi}_{\text{un}}\right) \leq 0, \\ & \mathbf{H}_N^{\mathcal{S}^{\text{f}}}\left(\left\{u_t, \{a_k^t\}_{k=1}^{|\Sigma|}, y_t, x_t^{\text{f}}, w_t^{\text{f}}, v_t^{\text{f}}, \right\}_{t=1}^{N}, \boldsymbol{\xi}_{\text{ob}}, \boldsymbol{\xi}_{\text{un}}^{\text{f}}\right) \leq 0, \\ & \mathbf{G}_N^{\varphi}\left(\{a_k^t\}_{k=1,t=1}^{|\Sigma|,N}, \boldsymbol{\eta}\right) \leq 0, \\ & \mathbf{G}_N^{\varphi^{\text{f}}}\left(\{a_k^t\}_{k=1,t=1}^{|\Sigma|,N}, \boldsymbol{\eta}^{\text{f}}\right) \leq 0, \\ & a_k^t \in \{0,1\}, \forall t \in [\![1, N]\!], k \in [\![1, |\Sigma|]\!], \\ & \boldsymbol{\eta} \in \{0,1\}^{|\boldsymbol{\eta}|}, \boldsymbol{\eta}^{\text{f}} \in \{0,1\}^{|\boldsymbol{\eta}^{\text{f}}|}, \end{aligned} \tag{19}$$

where $|\boldsymbol{\eta}|$ and $|\boldsymbol{\eta}^{\text{f}}|$ are the length of vectors $\boldsymbol{\eta}$ and $\boldsymbol{\eta}^{\text{f}}$ respectively.

D. Run-time Fault Detection

With the detectability analysis technique presented above, we can determine the minimal N so that $\mathfrak{B}_N(\mathcal{S},\varphi) \cap \mathfrak{B}_N(\mathcal{S}^{\mathrm{f}},\varphi^{\mathrm{f}}) = \emptyset$. As a result of such analysis, we only need to check whether the latest collected $\{u_t, s_t, y_t\}_{t=t_0-N+1}^{t_0} \in \mathfrak{B}_N(\mathcal{S},\varphi)$ at the current time t_0 and claim anomaly iff this does not hold. To this end, it is sufficient to run the monitor $\mathcal{M}^{\varphi}$ (as described in Section II-C) and the model invalidation LP (Eq. (2)) with horizon N in parallel. If no fault occurs, the monitor keeps running with current state set $Q_{t_0} \neq \emptyset$ and the model invalidation keeps being feasible, and no anomaly is claimed in this case. If a fault occurs at time t_{o}, either the switching sequence turns into a bad prefix of φ before time instant $t_{\mathrm{o}} + N - 1$ and the monitor detects violation of φ immediately, or the switching sequence is still a valid prefix of φ up until time instant $t_{\mathrm{o}} + N - 1$, which validates the boxed condition in Eq. (10) and hence the model invalidation LP must turn infeasible at time $t_0 = t_{\mathrm{o}} + N - 1$ by the N-behavior isolation of the faulty and healthy systems. In other words, any fault is detected with at most N-delay without any false alarm.

V. Case Study: UAV Altitude Consensus

We use an unmanned aerial vehicle (UAV) altitude consensus protocol to demonstrate the proposed detectability analysis technique. We say that a set of UAVs reaches altitude consensus if their altitude eventually converge to the same value. There are several consensus protocols based on local communication. In particular, we assume the UAVs implement the nearest neighbor rules from [9]. Under this protocol and assuming single integrator dynamics for vertical motion, the altitude dynamics of the UAVs can be modeled as follows. We let $x = [x^1, x^2, \ldots, x^8]^T \in \mathbb{R}^8$ be the state where x^i is the altitude of the i^{th} UAV. We assume that a leader UAV, indexed by 1, reaches a set point while the other UAVs adjust their own altitude according the nearest neighbor protocol [9], induced from the UAVs' communication topologies shown in Fig. 4 (Left). Let $A_{\sigma_k}, K_{\sigma_k}$ be the system matrices representing the UAVs dynamics while implementing this protocol. The i^{th} rows of matrices $A_{\sigma_1}, K_{\sigma_1}$ take the following form:

1) $i = 1$, $(A_{\sigma_1})_{11} = 0.9$ and $(A_{\sigma_1})_{1j} = 0$ for $j \in [\![2, 8]\!]$, $(K_{\sigma_1})_1 = 0.3$ this leads to a dynamics that guarantees the leader altitude to converge a set point at $x^1 = 3$ (i.e., $x^1_{t+1} = 0.9x^1_t + 0.3$);
2) $i \neq 1$, the i^{th} UAV update x^i by averaging its own height and those of its neighbors, i.e., $(K_{\sigma_1})_i = 0$ and

$$(A_{\sigma_1})_{ij} = \begin{cases} \frac{1}{1+d(i)} & \text{if } i \text{ connects } j \text{ in topology 1} \\ 0 & \text{otherwise} \end{cases}, \tag{20}$$

where $d(i)$ is the number of edges incident to node i.

Similarly, we define A_{σ_2}, K_{σ_2} for topology 2 with the same leader UAV set point (i.e., $x^1 = 3$); and define A_{σ_3}, K_{σ_3} (and A_{σ_4}, K_{σ_4}, respectively) for topology 1 (and topology 2, respectively) with the same leader UAV dynamics but a different set point at $x^1 = 6$.

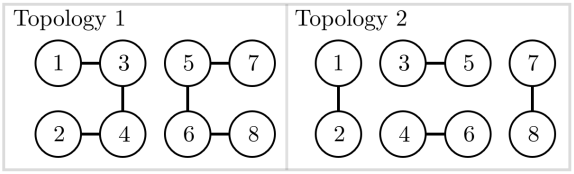

Fig. 3: Communication topologies of the UAVs, where circles represent UAVs with their indices.

We assume the changes in the communication topology and fault detection scheme (high-level decisions) run at a slower timescale than the consensus dynamics (low-level control) – i.e., 15 times slower. Then the dynamics relevant for fault detection can be written as the following switched affine system (denoted by $\mathcal{S}_{\mathrm{UAV}}$ in the rest of the paper):

$$x_{t+1} = \overline{A}_{\sigma_i} x_t + \overline{K}_{\sigma_i} + \overline{E}_{\sigma_i} w_t \tag{21}$$

where $s_t \in \Sigma := \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, and

$$\begin{aligned} &\overline{A}_{\sigma_i} = A^{15}_{\sigma_i}, \quad \overline{K}_{\sigma_i} = \sum\nolimits_{t=1}^{15} A^{t-1}_{\sigma_i} K_{\sigma_i} \\ &\overline{E}_{\sigma_i} = [A^{14}_{\sigma_i}, A^{13}_{\sigma_i}, \ldots, A_{\sigma_i}, I]. \end{aligned} \tag{22}$$

Note that in this setting there is no continuous control input u_t, and for simplicity we assume that the output $y_t = x_t$ with no measurement noise v_t. Finally, we assume that $x_t \in X := \{x \in \mathbb{R}^8 \mid 0 \leq x^i \leq 7,\ \forall i \in [\![1, 8]\!]\}$ and disturbance $w_t \in W := \{w \in \mathbb{R}^{120} \mid -0.1 \leq w^i \leq 0.1,\ \forall i \in [\![1, 120]\!]\}$.

The faulty system model $\mathcal{S}^{\mathrm{f}}_{\mathrm{UAV}}$ we analyze results from a failure in the communication links between nodes 3-5 and 4-6 in topology 2, changing the system matrices A_{σ_2} and A_{σ_4} (both induced by topology 2) and the corresponding $\overline{A}_{\sigma_2}$, $\overline{K}_{\sigma_2}$, $\overline{A}_{\sigma_4}$, $\overline{K}_{\sigma_4}$ in Eq. (22). Note that the healthy-faulty system pair $(\mathcal{S}_{\mathrm{UAV}}, \mathcal{S}^{\mathrm{f}}_{\mathrm{UAV}})$ is not N-detectable for any finite N because the fault will never be detected unless the system switch to mode σ_2 or σ_4. However, we know that switching to mode σ_2 or σ_4 infinitely often is required to achieve consensus because communication topology 1 is not a connected graph. We can hence incorporate this information in the detectability analysis to compute worst-case detection delay.

We assume that the mode sequence satisfies the LTL formula $\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3$ under the healthy configuration, where Formula φ_1 and φ_2 together restrict the dwell time for set point changes by the leader UAVs to be within $[\![7, 20]\!]$, while formula φ_3 assures that each of the two communication topologies are used within every three time steps. They together guarantee enough time and communication for convergence to a consensus. We also assume that the mode sequence does not need to satisfy any LTL formula after the fault occurs, however in the example we choose the mode sequence after the fault to be consistent with φ, therefore a pure discrete monitor will not be able to detect this fault without taking continuous dynamics into account. Gurobi [11] is used to solve the obtained MILP. The obtained minimal length of horizon $T = 30$, which is finite and this result agrees with Proposition 1.

$$\varphi_1 = \bigwedge_{(k,l)\in\{(1,2),(3,4)\}} \Box\left(\left(\bigwedge_{t=0}^{19} \bigcirc^t(\sigma_k \vee \sigma_l)\right) \rightarrow \bigcirc^{20}\neg(\sigma_k \vee \sigma_l)\right), \tag{23}$$

$$\varphi_2 = \bigwedge_{(k,l)\in\{(1,2),(3,4)\}} \Box\left(\left(\neg(\sigma_k \vee \sigma_l) \wedge \bigcirc(\sigma_k \vee \sigma_l)\right) \rightarrow \bigwedge_{t=2}^{7} \bigcirc^t(\sigma_k \vee \sigma_l)\right), \tag{24}$$

$$\varphi_3 = \bigwedge_{(k,l)\in\{(1,3),(2,4)\}} \Box\Big((\sigma_k \vee \sigma_l) \vee \bigcirc(\sigma_k \vee \sigma_l) \vee \bigcirc^2(\sigma_k \vee \sigma_l)\Big). \tag{25}$$

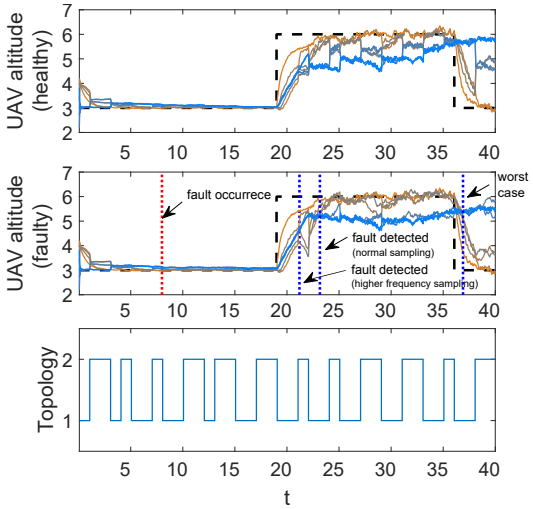

Fig. 4: Fault detection of the UAV consensus system at run time.

Fig. 4 (Right) shows the fault detection results. The upper plot shows the altitude of the eight UAVs (solid lines) and the set point profile (dashed black line) when no fault occurs. The middle plot shows the same set of trajectories of the faulty system. The lower plot shows the alternating of the two communication topologies. One can check that the given LTL formula φ is satisfied (even after fault). In this illustration, the fault occurs at time $t = 8$, at which time the consensus is already achieved (with set point at 3). Hence the fault does not lead to behavior isolation immediately. However, the fault is detected later at time $t = 23$ after the set point of the leader changes. The detection delay is 15, which is shorter than the delay bound $T = 30$. This experiment hence agrees with the theory. We also run the model invalidation at a higher frequency, using the timescale of the dynamics, for faster detection though this comes at the expense of solving LPs of larger size (15 times larger) and more often.

VI. Conclusion

In this paper, we proposed a technique to analyze fault detectability in switched affine systems whose mode sequences must satisfy certain LTL formulas, given as side information. Our approach was to transform the LTL constraints into a monitor finite state machine, which is then converted into mixed integer linear constraints that can be easily integrated into a MILP used for the detectability analysis. It was shown that the minimal length of the detector memory can be reduced with such side information, and this was illustrated with an example on UAV altitude consensus.

Acknowledgments: The authors would like to thank Dr. Dogan Ulus for helpful discussions on LTL monitoring.

References

[1] E. Asarin, A. Donzé, O. Maler, and D. Nickovic. Parametric identification of temporal properties. In *International Conference on Runtime Verification*, pages 147–160. Springer, 2011.

[2] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. P. (Eds.). *Model-based testing of reactive systems - Advanced lectures, LNCS 3472*. Springer-Verlag, Berlin, 2005.

[3] M. d'Amorim and G. Roşu. Efficient monitoring of ω-languages. In *International Conference on Computer Aided Verification*, pages 364–378. Springer, 2005.

[4] P. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of process control*, 7(6):403–424, 1997.

[5] F. Harirchi and N. Ozay. Model invalidation for switched affine systems with applications to fault and anomaly detection. In *Proceedings of the Analysis and Design of Hybrid Systems*, pages 260–266, 2015.

[6] F. Harirchi and N. Ozay. Guaranteed model-based fault detection in cyber-physical systems: a model invalidation approach. *Automatica*, 93:476 – 488, 2018.

[7] K. Havelund and G. Roşu. An overview of the runtime verification tool java pathexplorer. *Formal methods in system design*, 24(2):189–215, 2004.

[8] R. Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[9] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.

[10] S. Karaman, R. G. Sanfelice, and E. Frazzoli. Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In *2008 47th IEEE Conference on Decision and Control*, pages 2117–2122. IEEE, 2008.

[11] G. Optimization. Inc.,gurobi optimizer reference manual, 2015. *URL: http://www. gurobi. com*, 2014.

[12] N. Ozay, M. Sznaier, and C. Lagoa. Convex certificates for model (in)validation of switched affine systems with unknown switches. *IEEE Transactions on Automatic Control*, 59(11):2921–2932, 2014.

[13] T. Reinbacher, K. Y. Rozier, and J. Schumann. Temporal-logic based runtime observer pairs for system health management of real-time systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 357–372. Springer, 2014.

[14] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis. Failure diagnosis using discrete-event models. *IEEE transactions on control systems technology*, 4(2):105–124, 1996.

[15] R. S. Smith and J. C. Doyle. Model validation: A connection between robust control and identification. *IEEE Transactions on automatic control*, 37(7):942–952, 1992.

[16] G. J. Tretmans. *A formal approach to conformance testing*. Ph.D. Thesis, University of Twente, Netherlands, 1992.

[17] J. Tretmans. Test generation with inputs, outputs, and quiescence. In *Proceedings of the International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, pages 127–146, 1996.

[18] L. Yang and N. Ozay. Provably-correct fault tolerant control with delayed information. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 542–549. IEEE, 2017.

[19] L. Yang and N. Ozay. Fault-tolerant output-feedback path planning with temporal logic constraints. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4032–4039. IEEE, 2018.