2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Fault-tolerant output-feedback path planning with temporal logic constraints

Liren Yang Necmiye Ozay

Abstract— **In this paper, we consider searching for fault tolerant control strategies for linear systems to satisfy some high level requirements specified by linear temporal logic. By the term fault tolerant, we mean the obtained control strategy can respond to a fault that leads to a sudden change of the system dynamics. We first show how open-loop fault tolerant strategies (associated with each initial state) can be synthesized by leveraging Mixed Integer Linear Programming (MILP) based encodings used for linear temporal logic. These open-loop strategies, however, are not robust to the disturbances because of two reasons. First, since the disturbed system cannot be predicted precisely, the fault will be detected with a delay. Secondly, even if the faulty status is known, the true system trajectory may still deviate from the planned trajectory as the impact of the disturbance accumulates. To solve the two problems, we present a MILP formulation of the problem that incorporates finite detection delays, the open-loop strategy defined by the MILP's solution is then robustified with additional linear regulation.**

I. Introduction

With the advent of scalable numerical optimization techniques, the use of hierarchical control architectures with a high-level trajectory/path planner and a low-level regulator has become a common practice in many application domains [3]. While the low-level feedback regulator provides robustness to the uncertainties in system dynamics, if the path planning can be done online, it can provide robustness or reactiveness to the uncertainties in the environment. For instance, for simple tasks like reaching a goal while avoiding obstacles that may appear on the fly, approaches based on LQR trees [15] or contraction theory [14] have been proposed along these lines.

For more complicated tasks specified by temporal logics, there is a trade-off between reactiveness and scalability when designing controllers that provably satisfy the given task specification. On one hand, one can construct a discrete abstraction of the underlying dynamics and use reactive synthesis to design systems that can react to nondeterministic events in the environment at run-time [11]. However such approaches usually do not scale well with the continuous state-space dimension; specifically, the size of the abstraction can be exponential in the state-space dimension. On the other hand, one can design nominal trajectories offline, using mixed-integer linear programming (MILP) based encodings of temporal logic constraints, which are shown to scale better (i.e., polynomially) with the continuous state-space dimension [16]. However, since these approaches lead to open-loop controllers, they tend not to be robust/reactive due to lack of feedback. Recent work aimed at addressing this latter issue includes counter-example-guided methods [13] or searching for feedback controllers parametrized by disturbance [5] together with MILP-based trajectory planning. Though, the type of environment uncertainties against which robustness/reactiveness can be achieved by these methods is still limited. Therefore, there is a need for research that provides reactiveness for different classes of uncertainties.

In this work, we consider a special type of uncertainty, namely faults in the system, that we want the system to be robust/reactive against. We consider a linear temporal logic specification that captures the desired behavior of the system when there is no fault and (a potentially degraded) desired behavior after a fault occurs. A first natural attempt to solve this problem is to consider control strategies parametrized by fault occurrence time, that is, to design different trajectories depending on when the fault occurs and to regulate the system dynamics around these trajectories. However, when there are disturbances affecting system dynamics or measurements, a fault cannot be detected instantaneously but can only be detected after some bounded delay [9]. Therefore, the strategy should be robust against such detection delays as well, while reacting to the faults. Synthesis of fault-tolerant controllers for discrete transition systems has been addressed in [6], and some results on handling detection delays in the discrete setting are presented in [17]. While these approaches can be leveraged in an abstraction-based scheme (see, e.g., [18]), scalability with respect to the continuous state-space dimension is still a challenge.

Motivated by the above mentioned issues, we propose a hierarchical fault-tolerant controller with a MILP-based trajectory generation at the higher-level and an output-feedback regulator at the lower-level. Our MILP formulation incorporates reactiveness to faults while being robust to finite detection delays and it scales polynomially with the continuous state-space dimension. We further show that when the system dynamics are linear, the lower-level regulator design problem reduces to a quasi-convex optimization problem. Finally, we demonstrate the proposed approach numerically with a simple robot survailence task.

II. Preliminaries

Let $\mathbb{R}^n$ be the n-dimensional Euclidean space. In this paper, lowercase letters (e.g., x) are used for denoting a

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 in part by Ford Motor Co., DARPA grant N66001-14-1-4045, NSF grants CNS-1446298 and ECCS-1553873, and a NASA ECF award.

978-1-5386-1395-5/18/$31.00 ©2018 IEEE

vector in $\mathbb{R}^n$, bold font lowercase letters are used for finite sequences of vectors (e.g., $\mathbf{x} = x_1x_2x_3\dots x_N$), and blackboard bold font lowercase letters are used for infinite sequences of vectors (e.g., $\mathbb{x} = x_1x_2x_3\dots$). By convention, let $\mathbf{x}(i)$ (or $\mathbb{x}(i)$, respectively) be the i^{th} element in the sequence $\mathbf{x}$ (or $\mathbb{x}$, respectively), and let $\mathbb{x}_i = \mathbb{x}(i)\mathbb{x}(i+1)\mathbb{x}(i+2)\dots$ be the sub-sequence starting from the i^{th} position. We also define the Minkowski sum of two sets $X, Y \subseteq \mathbb{R}^n$ to be $X \oplus Y := \{x + y : x \in X, y \in Y\}$, and their Minkowski difference to be $X \ominus Y := \{x : \{x\} \oplus Y \subseteq X\}$.

A. Linear Temporal Logic

We use LTL to specify the desired closed-loop system behavior. In what follows we briefly introduce the syntax and the semantics of LTL, and refer the reader to [1] for more details.

1) Syntax: Let AP be a set of atomic propositions, the syntax of LTL formulas over AP is given by

$$\varphi ::= \pi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \ \mathcal{U} \ \varphi_2 \tag{1}$$

where $\pi \in AP$. With the grammar given in Eq. (1), we define the other propositional and temporal logic operators as follows: $\varphi_1 \wedge \varphi_2 := \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2 := \neg\varphi_1 \vee \varphi_2$, $\Diamond\varphi := True \ \mathcal{U} \ \varphi$, $\Box\varphi := \neg\Diamond\neg\varphi$, $\varphi_1 \ \mathcal{R} \ \varphi_2 := \neg(\neg\varphi_1 \ \mathcal{U} \ \neg\varphi_2)$. With these extra logical operators, an LTL formula φ can be written into a formula φ' in positive normal form, that is, all the negations in φ' only appears in front of the atomic propositions [1].

2) Semantics: Let $\mathbb{x} = x_1x_2x_3\dots$ be a infinite sequence of points in $\mathbb{R}^n$ and let AP be a set of atomic propositions. We define a labeling map $L : \mathbb{R}^n \rightarrow 2^{AP}$ and interpret an LTL formula over the labeling sequence $L(x_1)L(x_2)L(x_3)\dots$ as follows:

- $\mathbb{x} \models \pi$ iff $\pi \in L(x_1)$,
- $\mathbb{x} \models \neg\varphi$ iff $\mathbb{x} \not\models \varphi$,
- $\mathbb{x} \models \varphi_1 \vee \varphi_2$ iff $\mathbb{x} \models \varphi_1$ or $\mathbb{x} \models \varphi_2$,
- $\mathbb{x} \models \bigcirc\varphi$ iff $\mathbb{x}_2 \models \varphi$,
- $\mathbb{x} \models \varphi_1 \ \mathcal{U} \ \varphi_2$ iff $\exists s \geq 1 : \mathbb{x}_s \models \varphi_2$ and $\forall t < s : \mathbb{x}_t \models \varphi_1$.

Given an infinite word $\mathbb{x}$ and an LTL formula φ, we say φ holds for $\mathbb{x}$ (or $\mathbb{x}$ satisfies φ) iff $\mathbb{x} \models \varphi$.

B. Mixed Integer Encoding of LTL

Given an LTL formula φ, it is well-known from the literature [16] that $\mathbb{x} \models \varphi$ can be encoded with mixed integer linear constraints in the following sense. Instead of imposing constraints on the infinite sequence $\mathbb{x}$, we search for a finite sequence $\mathbf{x} = x_1x_2\dots x_k, x_{k+1\dots}x_N$ that satisfies the following linear inequality constraint:

$$H_{\varphi,k,N}(\mathbf{x}, \mathbf{b}) \leq 0, \tag{2}$$

where $\mathbf{b}$ is an N-sequence of auxiliary binary (hence integer) vectors[1], and $H_{\varphi,k,N}$ is an affine function in $(\mathbf{x}, \mathbf{b})$. In particular, $H_{\varphi,k,N}$ is constructed in such a way that the infinite sequence $\mathbb{x} := (x_1x_2\dots x_k)(x_{k+1}\dots x_N)^\omega$, obtained by unfolding $\mathbf{x} = x_1x_2\dots x_N$ at point k, is guaranteed to satisfy φ. In this case, we say that finite sequence $\mathbf{x}$ satisfy φ with a slight abuse of terminology.

With such mixed integer encoding technique, the path planning problem of linear systems can be formulated as a MILP. Let the system model be $x_{t+1} = Ax_t + Bu_t + F$ with state $x \in X$ and control $u \in U$, where $X \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^m$ are polytopes, also let x_{init} be the initial state, the MILP formulation is given by

$$\begin{aligned}
\text{find} \quad & \mathbf{x}, \mathbf{u}, \mathbf{b} \\
\text{s.t.} \quad & \exists 1 \leq k \leq N-1 : H_{\varphi,k,N}(\mathbf{x}, \mathbf{b}) \leq 0 \text{ and} \\
& \quad A\mathbf{x}(N) + B\mathbf{u}(N) + F = \mathbf{x}(k+1), \\
& \mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + F, \ t = 1, \dots, N-1, \\
& \mathbf{u}(t) \in U, \ \mathbf{b}(t) \in \{0,1\}, \ t = 1, \dots, N, \\
& \mathbf{x}(t) \in X, \ t = 1, \dots, N, \\
& \mathbf{x}(1) = x_{\text{init}}.
\end{aligned} \tag{3}$$

Suppose that the above optimization problem is feasible and let $(\mathbf{x}, \mathbf{u}, \mathbf{b})$ be one of its solutions, an infinite control sequence $\mathbb{u}$ can be extracted from the finite sequence $\mathbf{u}$, by unfolding $\mathbf{u}$ at some point $1 \leq k \leq N-1$. This control sequence leads to an infinite state sequence $\mathbb{x}$ that satisfies the linear dynamics and LTL specification φ.

While formulating the MILP in Eq. (3), it is a common practice to modify the state space labeling with an extra Δ-margin so that the specification is satisfied robustly [4], [12], [10]. Such modification provides robustness against uncertainties like disturbances or errors due to the sampling of a continuous-time system. For example, if an unsafe region is required to be avoided (i.e., $\varphi = \Box\neg\pi_{\text{unsafe}}$) under uncertainties, one can expand the unsafe set $X_{\text{unsafe}} := \{x \in \mathbb{R}^n : \pi_{\text{unsafe}} \in L(x)\}$ by Δ, i.e., define $\overline{X}_{\text{unsafe}} := X_{\text{unsafe}} \oplus \{x : \|x\| \leq \Delta\}$. Then avoiding $\overline{X}_{\text{unsafe}}$ means that X_{unsafe} is avoided with Δ-margin. Similarly, if some target region is required to be reached (i.e, $\varphi = \Diamond\pi_{\text{target}}$) robustly, one can shrink the target set $X_{\text{target}} := \{x \in \mathbb{R}^n : \pi_{\text{target}} \in L(x)\}$ into $\underline{X}_{\text{target}} := X_{\text{target}} \ominus \Delta$. Reaching the shrunk set $\underline{X}_{\text{target}}$ guarantees that set X_{target} is reached with Δ-margin. To do such expansion and shrinking systematically for arbitrary LTL formulas, one needs to rewrite the specification φ in positive normal form [4]. If atomic proposition π has no negation in the front, we shrink the set $X_\pi := \{x \in \mathbb{R}^n : \pi \in L(x)\}$ by Δ; and if π has a negation in the front, we expand the set X_π by Δ.

III. Problem Description

In this section, we define the fault tolerant path planning problem. The problem has two ingredients: (i) a system whose dynamics can degrade suddenly due to a fault, and (ii) an LTL formula that specifies the system's "graceful degradation".

[1]In practice, some variables in $\mathbf{b}$ need not be restricted as binary in the formulation. Instead, they can be specified as real and will be binary automatically as a result of the encoding.

A. System Model

The system model considered in this paper is defined by

$$\Sigma: \; x_{t+1} = A^{\sigma_t} x_t + B^{\sigma_t} u_t + F^{\sigma_t} + w_t, \tag{4}$$

$$\sigma_{t+1} \begin{cases} \in \{\mathrm{h}, \mathrm{f}\} & \text{if } \sigma_t = \mathrm{h} \\ = \mathrm{f} & \text{if } \sigma_t = \mathrm{f} \end{cases}. \tag{5}$$

where $x_t \in X \subseteq \mathbb{R}^n$ is the state, $u_t \in U \subseteq \mathbb{R}^m$ is the control input, $w_t \in W \subseteq \mathbb{R}^n$ is the disturbance, and $\sigma_t \in \{\mathrm{h}, \mathrm{f}\}$ is the fault status of the system. If $\sigma_t = \mathrm{h}$, the system is healthy and evolves with the dynamics defined by $(A^{\mathrm{h}}, B^{\mathrm{h}}, F^{\mathrm{h}})$; if $\sigma_t = \mathrm{f}$, this indicates that the fault has occurred and the system evolves with the dynamics defined by $(A^{\mathrm{f}}, B^{\mathrm{f}}, F^{\mathrm{f}})$. By Eq. (5), the fault is permanent because σ_t never recovers to h after it becomes f. In addition, we make the following assumption on the faults.

Assumption 1: We assume that the fault is T-detectable ([9]), that is, if the fault occurs at time step t_{o}, it will be detected at t_{d} where $t_{\mathrm{o}} + 1 \leq t_{\mathrm{d}} \leq t_{\mathrm{o}} + T$.

For Assumption 1, it should be noticed that T is only an upper bound on the detection delay, and the actual online detection can be earlier than $t_{\mathrm{o}}+T$. However, this fact cannot be incorporated in the offline path planning phase because the actual detection depends on the realization of w_t.

B. LTL Specification

The desired behavior of the system with faults is specified with an LTL formula. Let AP be the set of the interested atomic propositions, and we also define an extra atomic proposition π^{f} that indicates the fault has already occurred. In particular, to relate the atomic proposition π^{f} with the fault status σ_t of the system, we require the labeling map $L : X \times \{\mathrm{h}, \mathrm{f}\} \to 2^{\mathrm{AP} \cup \{\pi^{\mathrm{f}}\}}$ to satisfy

$$\pi^{\mathrm{f}} \in L(x_t, \sigma_t) \quad \text{iff} \quad \sigma_t = \mathrm{f}. \tag{6}$$

Now we define the so called "graceful degradation" by the following LTL formula:

$$\Phi = \big((\Box\neg\pi^{\mathrm{f}}) \wedge \varphi^{\mathrm{h}}\big) \vee \big(\neg\pi^{\mathrm{f}} \,\mathcal{U}\, (\Box\pi^{\mathrm{f}} \wedge \varphi^{\mathrm{f}})\big), \tag{7}$$

where φ^{h} and φ^{f} are arbitrary LTL formulas. Eq. (7) says: either the system is always healthy and φ^{h} should hold, or the system turns faulty and φ^{f} should be satisfied immediately after the fault occurrence. In particular, typically φ^{f} is chosen to be less stringent than φ^{h}, in which case the LTL formula Φ captures a graceful degradation in the system performance.

C. Problem Statement

Problem 1: Given a system Σ defined by Eq. (4), (5), an LTL formula Φ defined by Eq. (7), and a initial state x_{init}, synthesize a control sequence $\mathfrak{u}$, under which the trajectory generated by system Σ starting from x_{init} satisfies specification Φ.

IV. Solution Approach

The main difficulties of solving Problem 1 are due to the uncertainties in the system. First, the fault occurrence time is not known at the offline path planning phase. Even when the fault is detected during the executions, the exact time of fault occurrence is still unknown. Instead, we only know that the fault occurs at most T step before the detection by the T-detectability assumption. Finally, the true trajectory may deviate from the planned trajectory as the impact of disturbance accumulates.

To tackle the above challenges, we propose an optimization-based solution approach that contains two ingredients: open-loop path planning and regulation. The control authority is split in two parts correspondingly, one part is reserved for path planning and the other part for regulation. First, we assume no disturbance acts on the system but the fault is still detected with a delay of at most length T. We then present a MILP formulation, whose solution defines a strategy $\overline{\mathbf{u}}$, which leads to a nominal trajectory $\overline{\mathbf{x}}$ that satisfy the overall specification Φ robustly against the delay. In particular, the state space labeling in formulating the MILP is modified with an extra Δ-margin, so that the true, disturbed trajectory $\mathbf{x}$ can still satisfy the specification. Secondly, in order to guarantee that the true trajectory $\mathbf{x}$ indeed stays Δ-close to the nominal trajectories $\overline{\mathbf{x}}$, we add extra linear regulation (i.e., $\mathbf{u} = \overline{\mathbf{u}} + K(\mathbf{x} - \overline{\mathbf{x}})$) with the remaining control authority. We show that the regulator gain K can be designed by solving a quasi-convex optimization problem so that the size of the required margin Δ is minimized.

A. MILP Formulation of Fault Tolerant Path Planning

In this section, we formulate the fault tolerant path planning problem as a MILP. The system is assumed to be undisturbed (an assumption to be relaxed in the next subsection) and have a fault that is T-detectable. In addition, the labeling of the state space is robustified with a Δ-margin. We will also assume that there is a way (to be presented in the next subsection) to keep the true, disturbed trajectories Δ-close to a nominal trajectory as long as the system's fault status does not change.

We begin by sketching the strategy that achieves specification Φ in Eq. (7). To satisfy Φ, the system can either stay in the healthy mode forever and satisfy φ^{h}, or enter faulty mode at some time t_{o} and start to satisfy φ^{f} from then on. However, since the fault is beyond our control, we can only respond to the fault occurrence passively. In particular, as long as the fault has not been detected yet, there is a chance that the the system is healthy and will be healthy forever. Hence we need to achieve specification φ^{h} for the healthy mode in this case. On the other hand, once the fault is detected, the first half of Φ (i.e., $(\Box\neg\pi^{\mathrm{f}}) \wedge \varphi^{\mathrm{h}}$) can no more be satisfied. Hence the strategy needs to respond to the fault by rendering the system to satisfy φ^{f}.

The above analysis leads to a strategy visualized in Fig. 1 (upper part). Roughly speaking, strategy $\overline{\mathbf{u}}$ should contain two pieces: a sequence $\overline{\mathbf{u}}^{\mathrm{h}}$ (black) that achieves φ^{h} under the healthy dynamics, and a sequence $\overline{\mathbf{u}}^{\mathrm{f}}$ (gray) that achieves φ^{f}

under the faulty dynamics. The two sequences $\overline{\mathbf{u}}^{\mathrm{h}}$ and $\overline{\mathbf{u}}^{\mathrm{f}}$ can have different length (N^{h}, N^{f} respectively), and both of them can be unfolded to obtain infinite sequences (the loops that are to be unfolded are marked with dashed line arrows in Fig. 1). In addition, there should be different control sequences $\overline{\mathbf{u}}^{\mathrm{f}}$ associated with different time instants of detection because our strategy should respond to the fault detected at anytime. We denote each control sequence associated with detection time t_{d} by

$$\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]. \tag{8}$$

Several important remarks on $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$ are made in what follows.

First, it should be noticed that sequence $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$ starts from time t_{d}, but it may correspond to any fault that occurs at $\min\{1, t_{\mathrm{d}} - T\} \leq t_{\mathrm{o}} \leq t_{\mathrm{d}} - 1$, where $\min\{1, t_{\mathrm{d}} - T\}$ is the earliest possible fault occurrence time that associates with t_{d} given T-detectability assumption. All of these fault occurrences associated with the same t_{d} cannot be treated separately because the exact fault occurrence time t_{o} is not known in general. Instead, these different fault occurrences are all controlled with $\overline{\mathbf{u}}^{\mathrm{h}}$ and $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$ in the following way:

- Within the so called "uninformed execution horizon" (i.e., $\min\{1, t_{\mathrm{d}} - T\} \leq t \leq t_{\mathrm{d}} - 1$), we do not know the system is already faulty and have to apply $\overline{\mathbf{u}}^{\mathrm{h}}$ until time $t_{\mathrm{d}} - 1$.
- Starting from time t_{d}, the fault is known and $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$ is applied.

With the above control strategy, each occurrence time t_{o} corresponds to a different trajectory generated under the faulty dynamics, denote by

$$\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}], \tag{9}$$

and our goal is to ensure that $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ can be unfolded into an infinite sequence that satisfies φ^{f} for all t_{d} and $\min\{1, t_{\mathrm{d}} - T\} \leq t_{\mathrm{o}} \leq t_{\mathrm{d}} - 1$.

The family of sequence $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ associated with a fixed t_{d} has a notable property, that is, they all behave approximately the same as one sequence $\overline{\mathbf{x}}^{\mathrm{h}}$ within the uninformed execution horizon, where $\overline{\mathbf{x}}^{\mathrm{h}}$ denotes the sequence generated by $\overline{\mathbf{u}}^{\mathrm{h}}$ under the healthy dynamics. In particular, $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ will be Δ-close to $\overline{\mathbf{x}}^{\mathrm{h}}$ at least until $t_{\mathrm{d}} - 1$. This has to be true given the assumption that the disturbed healthy trajectories is always Δ-close to the nominal trajectory $\overline{\mathbf{x}}^{\mathrm{h}}$ as long as the system is healthy. Consequently, the fault should be detected as long as the real trajectory is outside the Δ-tube of $\overline{\mathbf{x}}^{\mathrm{h}}$. The fact that $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ is close to $\overline{\mathbf{x}}^{\mathrm{h}}$ within the uninformed execution horizon is highlighted with the blue box in the lower part of Fig. 1.

The second remark is about the faults that are detected on the loop. These situation needs to be handled with extra care because it may correspond to multiple fault detections after unfolding the loop. In Fig. 1, for example, if the fault is detected at the black node denoting $\overline{\mathbf{x}}^{\mathrm{h}}(6)$, the detection time $t_{\mathrm{d}} = 6 + ml$ where l is the length of the loop and $m = 0, 1, 2, \ldots$. These cases need to be handled separately. In particular, all the cases with $ml \geq T$ can be treated as one. The number of cases is hence reduced to finitely many and we only need to find $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$ for $1 \leq t_{\mathrm{d}} \leq N^{\mathrm{h}} + T$.

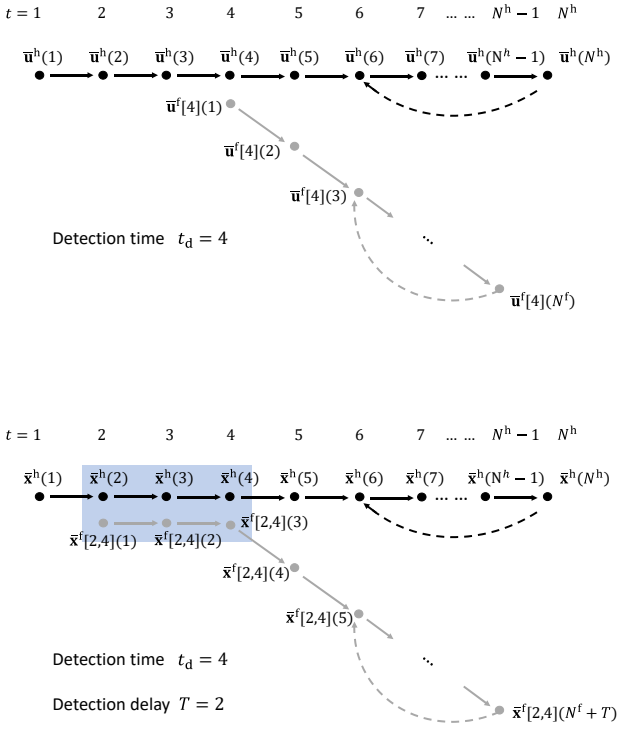

Fig. 1: An Illustration of the fault tolerant path planning strategy (upper) and associated trajectories (lower).

In what follows, we transform the above descriptions of the strategy into MILP formulations.

1) Healthy Mode Path Planning: For the healthy mode control sequence $\overline{\mathbf{u}}^{\mathrm{h}}$ to achieve healthy specification φ^{h} under the healthy dynamics defined by $(A^{\mathrm{h}}, B^{\mathrm{h}}, F^{\mathrm{h}})$, one needs the following constraints:

$$\begin{aligned}
&\exists 1 \leq k^{\mathrm{h}} \leq N^{\mathrm{h}} - 1 : \\
&\quad H_{\varphi^{\mathrm{h}}, k^{\mathrm{h}}, N^{\mathrm{h}}}\left(\overline{\mathbf{x}}^{\mathrm{h}}, \mathbf{b}^{\mathrm{h}}\right) \leq 0 \text{ and} \\
&\quad A^{\mathrm{h}}\overline{\mathbf{x}}^{\mathrm{h}}(N^{\mathrm{h}}) + B^{\mathrm{h}}\overline{\mathbf{u}}^{\mathrm{h}}(N^{\mathrm{h}}) + F^{\mathrm{h}} = \overline{\mathbf{x}}^{\mathrm{h}}(k+1), \\
&\quad \overline{\mathbf{x}}^{\mathrm{h}}(t) = \overline{\mathbf{x}}^{\mathrm{h}}(t - N^{\mathrm{h}} + k), \overline{\mathbf{u}}^{\mathrm{h}}(t) = \overline{\mathbf{u}}^{\mathrm{h}}(t - N^{\mathrm{h}} + k), \\
&\quad t = N^{\mathrm{h}}, \ldots, N^{\mathrm{h}} + T, \qquad (10) \\
&\overline{\mathbf{x}}^{\mathrm{h}}(t+1) = A^{\mathrm{h}}\overline{\mathbf{x}}^{\mathrm{h}}(t) + B^{\mathrm{h}}\overline{\mathbf{u}}^{\mathrm{h}}(t) + F^{\mathrm{h}}, \\
&\quad t = 1, \ldots, N^{\mathrm{h}} - 1, \qquad (11) \\
&\overline{\mathbf{x}}^{\mathrm{h}}(1) = x_{\mathrm{init}}, \qquad (12)
\end{aligned}$$

where $\overline{\mathbf{x}}^{\mathrm{h}}$ is the trajectory generated by the healthy dynamics under $\overline{\mathbf{u}}^{\mathrm{h}}$, $H_{\varphi^{\mathrm{h}}, k^{\mathrm{h}}, N^{\mathrm{h}}}$ is a function, encoding the specification φ^{h} within a horizon of length N^{h}, that is affine in both $\overline{\mathbf{x}}^{\mathrm{h}}$ and auxiliary variables $\mathbf{b}^{\mathrm{h}}$, and x_{init} is the initial state. Note that we extend $\overline{\mathbf{x}}^{\mathrm{h}}$ and $\overline{\mathbf{u}}^{\mathrm{h}}$ by T in Eq. (10) to handle the detection on the loop.

2) Faulty Mode Path Planning: We also require all the possible faulty trajectories $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ to satisfy φ^{f} when they are controlled by the faulty mode control $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$, which leads to the following constraints for $1 \leq t_{\mathrm{d}} \leq N^{\mathrm{h}} + T$ and $\min\{1, t_{\mathrm{d}} - T\} \leq t_{\mathrm{o}} \leq t_{\mathrm{d}} - 1$:

$$\begin{aligned}
&\exists 1 \leq k^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}] \leq N^{\mathrm{f}}: \\
&\quad H_{\varphi^{\mathrm{f}}, k^{\mathrm{f}}, N^{\mathrm{f}}} \left(\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}], \mathbf{b}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]\right) \leq 0 \text{ and} \\
&\quad A^{\mathrm{f}} \overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}](N^{\mathrm{f}}) + B^{\mathrm{f}} \overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}](N^{\mathrm{f}}) + F^{\mathrm{f}} = \overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}](k^{\mathrm{f}} + 1),
\end{aligned} \tag{13}$$

$$\begin{aligned}
&\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}](t+1) = A^{\mathrm{f}} \overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}](t) + B^{\mathrm{f}} \overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}](t) + F^{\mathrm{f}}, \\
&\quad t = T, \ldots, N^{\mathrm{f}} - 1,
\end{aligned} \tag{14}$$

$$\begin{aligned}
&\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}](t) = \overline{\mathbf{x}}^{\mathrm{h}}(t + t_{\mathrm{o}} - 1), \\
&\quad t = 1, \ldots, T, .
\end{aligned} \tag{15}$$

where $H_{\varphi^{\mathrm{f}}, k^{\mathrm{f}}, N^{\mathrm{f}}}$ encodes specification φ^{f} within horizon of length N^{f}. In particular, Eq. (15) requires the first T points in sequence $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ overlaps with the corresponding points in healthy sequence $\overline{\mathbf{x}}^{\mathrm{h}}$. This constraint captures the fact that $\overline{\mathbf{x}}^{\mathrm{h}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ stays close to $\overline{\mathbf{x}}^{\mathrm{h}}$ within the uninformed execution horizon. This constraint hence couples constraints (10)-(12) with constraints (13)-(14).

In summary, let $\overline{\mathbf{u}}^{\mathrm{f}}$ ($\overline{\mathbf{x}}^{\mathrm{f}}$, respectively) be the vector obtained by stacking $\overline{\mathbf{u}}^{\mathrm{f}}[t_{\mathrm{d}}]$ ($\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$, respectively) for all t_{d} ($t_{\mathrm{o}}, t_{\mathrm{d}}$, respectively), the fault tolerant path planning for nominal system with detection delay can be formulated as the following MILP:

$$\begin{aligned}
\text{find} \quad & \overline{\mathbf{x}}^{\mathrm{h}}, \overline{\mathbf{x}}^{\mathrm{f}}, \overline{\mathbf{u}}^{\mathrm{h}}, \overline{\mathbf{u}}^{\mathrm{f}}, \mathbf{b}^{\mathrm{h}}, \mathbf{b}^{\mathrm{f}} \\
\text{s.t.} \quad & \text{Eq. (10)-(15)}, \\
& \overline{\mathbf{x}}^{\mathrm{h}}(t), \overline{\mathbf{x}}^{\mathrm{f}}(t) \in X, \quad \forall t, \\
& \overline{\mathbf{u}}^{\mathrm{h}}(t), \overline{\mathbf{u}}^{\mathrm{f}}(t) \in U, \quad \forall t, \\
& \mathbf{b}^{\mathrm{h}}(t), \mathbf{b}^{\mathrm{f}}(t) \in \{0, 1\}, \quad \forall t.
\end{aligned} \tag{16}$$

Comparing to regular path planning MILP formulation, the fault-tolerant path planning MILP has more constraints and variables. Let $n_{\mathrm{C}}^{\mathrm{f}}$, $n_{\mathrm{B}}^{\mathrm{f}}$ and m^{f} be the number of continuous and binary variables and constraints respectively, which are required to encode the faulty mode LTL specification of a path of length N^{f}, the extra number of continuous and binary variables, and the number of constraints added on top of regular path planning MILP is $\mathcal{O}(N^{\mathrm{h}} n_{\mathrm{C}}^{\mathrm{f}})$, $\mathcal{O}(N^{\mathrm{h}} n_{\mathrm{B}}^{\mathrm{f}})$ and $\mathcal{O}(N^{\mathrm{h}} m^{\mathrm{f}})$ respectively. Particularly, these three quantities do not depend on the detection delay T.

B. Robustification of MILP's Solution via Regulation

From the previous section, we formulate an MILP whose solution leads to a nominal trajectory $\overline{\mathbf{x}}$ that satisfies the specification Φ. In particular, we allow the real trajectory $\mathbf{x}$ to deviate at most Δ from the nominal (i.e., $\|\overline{\mathbf{x}}(t) - \mathbf{x}(t)\| \leq \Delta$) while still satisfying Φ. In this section, we show how to find the minimum margin Δ that can be achieved by a linear regulator and the corresponding regulation gain by solving a quasi-convex optimization problem.

In what follows we consider system

$$x_{t+1} = Ax_t + Bu_t + w_t, \tag{17}$$

where w_t is disturbance satisfying $\|w_t\| \leq d$ for all t, and A, B can refer to the system matrices for the healthy system or the faulty system. Note that the constant offset term F in Eq. (4) is dropped because it only shifts the equilibrium of the system and makes no difference when the regulation is of our concern. We call a system to be nominal if $w_t = 0$ for all t. Given an initial state x_{init} and an open-loop strategy $\overline{\mathbf{u}} = \overline{u}_1 \overline{u}_2 \ldots \overline{u}_N$, the trajectory $\overline{\mathbf{x}} = \overline{x}_1 \overline{x}_2 \ldots \overline{x}_N$ generated by nominal system is governed by

$$\overline{x}_{t+1} = A\overline{x}_t + B\overline{u}_t, \tag{18}$$

$$\overline{x}_1 = x_{\mathrm{init}}. \tag{19}$$

Under the given open-loop strategy, the actual trajectory may deviate from the planned nominal trajectory in the presence of nonzero disturbance w_t. Moreover, such deviation may accumulate with time because there is no feedback. In this work, we introduce feedback to keep the actual trajectory close to the planned nominal trajectory $\overline{\mathbf{x}}$ as time evolves. The block-diagram of the overall hierarchical closed-loop system is shown in Fig. 2. Instead of applying nominal control $\overline{u}_t$ directly to the system, we use

$$u_t = \overline{u}_t + K(\hat{x}_t - \overline{x}_t) \tag{20}$$

where K is the state feedback gain, $\hat{x}_t$ is the estimated state that is assumed to satisfy

$$\|\hat{x}_t - x_t\| \leq E. \tag{21}$$

Our goal is to design feedback gain K, so that the difference between the actual trajectory $\mathbf{x}$ and the planned nominal trajectory $\overline{\mathbf{x}}$ is bounded by a constant Δ over time.

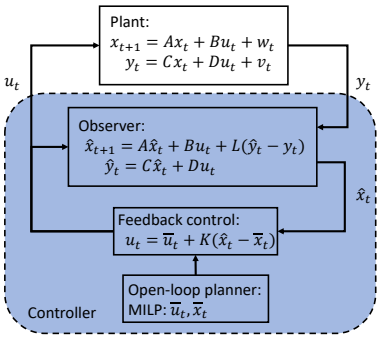

Fig. 2: Block-diagram of the closed-loop system (the extension with output feedback can be found in the Appendix).

The rest of this section focuses on designing K such that the uniform bound on $\|\overline{x}_t - x_t\|$ is minimized. Combining Eq. (17), (18), (20), we have

$$x_{t+1} - \overline{x}_{t+1} = (A + BK)(x_t - \overline{x}_t) + BK(\hat{x}_t - x_t) + w_t, \tag{22}$$

which implies

$$\begin{aligned}\|x_{t+1}-\overline{x}_{t+1}\| \leq & \|(A+BK)(x_t-\overline{x}_t)\| \\ & + \|BK(\hat{x}_t - x_t)\| + \|w_t\| \\ \leq & \|A+BK\|_*\|x_t-\overline{x}_t\| \\ & + \|BK\|_*\|\hat{x}_t - x_t\| + \|w_t\| \\ \leq & \|A+BK\|_*\|x_t-\overline{x}_t\| \\ & + \|BK\|_* E + d, \end{aligned} \tag{23}$$

where d is the bound of $\|w_t\|$ and E is the error bound on the state estimation from Eq. (21). Let Δ be the desired bound on $\|x_t - \overline{x}_t\|$, we require the following recurrence relation:

$$\|x_t - \overline{x}_t\| \leq \Delta \Rightarrow \|x_{t+1} - \overline{x}_{t+1}\| \leq \Delta. \tag{24}$$

Eq. (23), (24) hence suggests that

$$\|A+BK\|_* \leq \frac{\Delta - d - \|BK\|_* E}{\Delta}. \tag{25}$$

To minimize Δ, we formulate an optimization problem,

$$\begin{array}{ll} \text{minimize}_{\delta, K} & \delta \\ \text{s.t.} & \|A+BK\|_* \leq \frac{\delta - d - \|BK\|_* E}{\delta} \\ & \delta \geq 0 \end{array}. \tag{P1}$$

Proposition 1: The optimization problem (P1) is equivalent to the following quasi-convex optimization problem:

$$\begin{array}{ll} \text{minimize}_{K} & \frac{d+\|BK\|_* E}{1-\|A+BK\|_*} \\ \text{s.t.} & \|A+BK\|_* \leq 1 \end{array}. \tag{P2}$$

Proof: We first prove the equivalence between the optimization problems (P1) and (P2). Then, we prove that the objective function of the second problem is quasi-convex.

First note that d and E are nonnegative, we hence have

$$\begin{array}{l} \|A+BK\|_* \leq \frac{\delta - d - \|BK\|_* E}{\delta} \\ \delta \geq 0 \end{array} \Leftrightarrow \begin{array}{l} \delta \geq \frac{d+\|BK\|_* E}{1-\|A+BK\|_*} \\ \|A+BK\|_* \leq 1. \end{array} \tag{26}$$

Now let $K^\star, \delta^\star$ be an optimal solution of (P1). By Eq. (26), we know that $K^{\circ\circ} = K^\star$ is feasible for Problem (P2) and leads to an objective value $\frac{d+\|BK^{\circ\circ}\|_* E}{1-\|A+BK^{\circ\circ}\|_*} \leq \delta^\star$. Similarly, let $K^{\star\star}$ be an optimal to Problem (P2), we know that $K^\circ = K^{\star\star}$ and $\delta^\circ = \frac{d+\|BK^\star\|_* E}{1-\|A+BK^\star\|_*}$ are feasible for (P1) and they lead to the same objective value. This hence proves the equivalence between the two problems.

Next, we show that $\frac{d+\|BK\|_* E}{1-\|A+BK\|_*}$ is quasi-convex in K when $\|A+BK\|_* \leq 1$, i.e., $S_s := \left\{K \,\middle|\, \frac{d+\|BK\|_* E}{1-\|A+BK\|_*} \leq s, \|A+BK\|_* \leq 1\right\}$ is a convex set for any s. Without loss of generality, we only need to consider $s \geq 0$ as otherwise $S_s = \emptyset$. In that case,

$$S_s = \left\{ K \;\middle|\; \begin{array}{l} \|BK\|_* E + s\|A+BK\|_* \leq s - d, \\ \|A+BK\|_* \leq 1 \end{array} \right\}. \tag{27}$$

Since constants $s, E \geq 0$, and $\|BK\|_*$, $\|A+BK\|_*$ are convex functions in K, it follows that S_s is a convex set, and this finishes the proof. ∎

We highlight the following three points regarding the above optimization problem. First, a quasi-convex optimization problem can be solved by solving a sequence of convex feasibility problems. The idea is to do a line search on the objective value $f(x)$ and check if $S_s := \{x \text{ feasible} \mid f(x) = s\}$ is empty or not. Since S_s is a convex set by quasi-convexity of f, this can be done relatively efficiently. The detailed algorithm can be found in [2]. Secondly, for the optimization problem (P2) to be feasible, it is necessary (but not sufficient) that pair (A, K) is stabilizable. To see this, consider the equivalent problem in (P1), in which we require $\|A+BK\|_* \leq \frac{\delta - d - \|BK\|_* E}{\delta} < 1$. If (A, B) is not controllable, this constraint can not be satisfied with any gain K. Finally, if the system in Eq. (17) has an output function and the estimated state $\hat{x}_t$ is given by an observer, a similar quasi-convex problem can be derived to minimize the estimation error bound E in Eq. (21). The detailed derivation of the output feedback case can be found in the Appendix.

Several remarks are provided below, regarding the issues when combining the path planning with the regulation.

(i) First, note that we need to design K^{h} for regulating the health dynamics and K^{f} for the faulty dynamics, which leads to Δ^{h} and Δ^{f} margin respectively. The state labeling in the MILP formulation is hence modified with Δ^{h} or Δ^{f} correspondingly.

(ii) The second remark is on splitting the control authority. Let K^{h} (K^{f}, respectively) be the solution of the problem in Eq. (P2) formulated with the healthy (faulty, respectively) dynamics. The control authority required by linear regulation is $U^{\mathrm{h}}_{\mathrm{reg}} = \{u \in \mathbb{R}^m : \|u\| \leq \|K^{\mathrm{h}}\|_* \Delta\}$. Therefore $U^{\mathrm{h}}_{\mathrm{plan}}$, the control authority reserved for path planning, need to be shrunk by $\|K^{\mathrm{h}}\|_* \Delta^{\mathrm{h}}$, i.e., $U^{\mathrm{h}}_{\mathrm{plan}} = U \ominus U^{\mathrm{h}}_{\mathrm{reg}}$. The procedure of splitting U for the faulty mode follows similarly.

(iii) The third remark is about determining the faulty nominal trajectory $\overline{\mathbf{x}}^{\mathrm{f}}$ used in the regulation. Recall that the regulator is in the form of $u_t = \overline{\mathbf{u}}^{\mathrm{f}}(t) + K^{\mathrm{f}}(\mathbf{x}(t) - \overline{\mathbf{x}}^{\mathrm{f}}(r))$ where $\overline{x}_t$ is define by $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ under the faulty mode, but t_{o} is not known. However, we will only switch the regulator gain from K^{h} to K^{f} at time t_{d}, after which $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$ are the same for all t_{o}.

(iv) Finally, note that the true trajectory $\mathbf{x}$ may not be Δ^{h}-close to the healthy nominal trajectory $\overline{\mathbf{x}}^{\mathrm{h}}$ at detection time t_{d}, although this is true for all $t \leq t_{\mathrm{d}} - 1$. Hence extra error is introduced in this last step and need to be added on top of Δ^{h}. This extra error is bounded by

$$\begin{aligned} & \Delta^{\mathrm{h}} + \|A^{\mathrm{h}} - A^{\mathrm{f}}\|_* \|x\| + \|B^{\mathrm{h}} - B^{\mathrm{f}}\|_* \|u\| \\ & + \|F^{\mathrm{h}} - F^{\mathrm{f}}\|. \end{aligned} \tag{28}$$

However, depending on the criticality of the application, this extra error can be neglected if the real-time detection has high enough sampling rate, and thus can report the fault as soon as the Δ margin is violated by a tiny amount.

We now summarize the above construction in Section IV-A and Section IV-B with the following proposition:

Proposition 2: Suppose that the healthy system (and the faulty system, respectively) are regulated by K^{h} (K^{f}, respectively) found by solving problem (P2), around the trajectory obtained by solving the MILP in Eq. (16), then the true trajectory robustly satisfies specification Φ in Eq. (7). This hence solves Problem 1.

V. Numerical Example

We present an example on robot path planning in this section. For simplicity, we assume state feedback in this example. The output feedback version of the problem can be solved using the extension in the Appendix.

The considered system is modeled with a double integrator on the plane. The healthy discrete-time system matrices $A^{\mathrm{h}}, B^{\mathrm{h}}, F^{\mathrm{h}}$ in Eq. (5) are obtained by sampling the following continuous-time system with a sampling rate $\tau = 2$s.

$$A_{\mathrm{c}}^{\mathrm{h}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -20 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -20 \end{bmatrix}, B_{\mathrm{c}}^{\mathrm{h}} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, F_{\mathrm{c}}^{\mathrm{h}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{29}$$

For the faulty system dynamics, we assume that $A^{\mathrm{f}} = A^{\mathrm{h}}$, $B^{\mathrm{f}} = B^{\mathrm{h}}$, but there is a non-zero constant offset term $F^{\mathrm{h}} = [0,\ 1.5,\ 0,\ 0]^T$, resulting in an undesirable drift. Let $x = [x_1, x_2, x_3, x_4]^T$ be the state and $u = [u_1, u_2]$ be the control input, we restrict that $x \in X = [-10, 10] \times [-2, 8] \times [-15, 15] \times [-15, 15]$, and that $u \in U = [-15, 15] \times [-15, 15]$. We also assume that there is an additive disturbance $w \in \mathbb{R}^4$ as in Eq. (17). In particular, disturbance w satisfy $\|w\| \leq 0.25$. We assume that the fault detection delay is bounded by 3 samples (i.e., $T = 3$).

The specification is defined by the LTL formula in Eq. (7) with

$$\varphi^{\mathrm{h}} = (\square\neg\pi_{\mathrm{r}}) \wedge (\lozenge\square\pi_{\mathrm{g}}) \wedge (\square\lozenge\pi_{\mathrm{b1}}) \wedge (\square\lozenge\pi_{\mathrm{b2}}), \tag{30}$$

$$\varphi^{\mathrm{f}} = (\square\neg\pi_{\mathrm{r}}) \wedge (\lozenge\square\pi_{\mathrm{g}}). \tag{31}$$

The regions (in x_1–x_3 space) in which each atomic proposition holds are marked in Fig. 3. In particular, the regions for π_{r} and π_{g} are the rectangles with solid boundaries, and the regions associated with π_{b1} (π_{b2}, respectively) are to the left (right, respectively) of the bold blue dashed line.

We first design a linear regulator by solving the quasi-convex optimization problem in (P2). The quasi-convex problem is solved with a standard line-search algorithm [2] that reduces to solving a sequence of convex optimization problems. These convex problems are then solved using CVX [7]. In this example, since $A^{\mathrm{f}} = A^{\mathrm{h}}$, $B^{\mathrm{f}} = B^{\mathrm{h}}$, we only need one regulator gain K, and the extra error introduced by detection delay in Eq. (28) can be bounded by $\|F^{\mathrm{h}} - F^{\mathrm{f}}\|$ (here we assume that the real-time detection has high enough sampling rate so that this extra error can be neglected). The obtained optimal regulator gain leads to a margin $\Delta^{\mathrm{h}} = \Delta^{\mathrm{f}} =: \Delta = 0.4604$. We hence modify the labeling by Δ. In Fig. 3, this modification corresponds to the transparent margin surrounding the rectangles and the thinner dashed lines close to the bold ones. Finally, we shrink the control set U by $\|K\|_* \Delta$, as discussed at the end of Section IV-B in remark (ii).

The fault tolerant path planning is then solved with the MILP formulated in Section IV-A. We solve the MILP with Gurobi [8]. Fig. 3 shows (i) the scenario when the system is always healthy, and (ii) a faulty scenario where the fault happens at time instant $t_{\mathrm{o}} = 1$ and is detected at $t_{\mathrm{d}} = 3$. The black dotted line represents the nominal healthy trajectory $\overline{\mathbf{x}}^{\mathrm{h}}$ and the red dotted line represents the nominal faulty trajectory $\overline{\mathbf{x}}^{\mathrm{f}}[t_{\mathrm{o}}, t_{\mathrm{d}}]$. The dark gray solid curve is the disturbed trajectory assuming that the system remains healthy forever, and the purple solid curve is the disturbed trajectory under the considered faulty scenario. The following observations can be made based on these simulations:

(i) It can be seen that both the disturbed trajectories satisfy the specification corresponding to their mode.

(ii) The two curves stay close until the fault detection, where the nominal trajectory starts to deviate from the healthy trajectory.

(iii) The healthy trajectory (gray) detours to the left more than it requires to satisfy φ^{h}, as it needs to preserve extra "room" for the faulty trajectory (purple) to avoid the red obstacle.

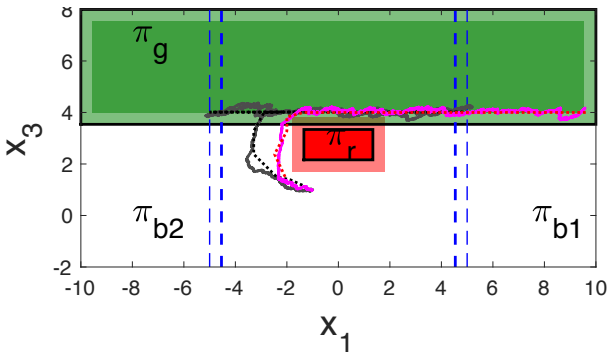

Fig. 3: The planned trajectories (dotted) and the disturbed trajectories (solid) for health mode (black, gray) and faulty mode (red, purple).

VI. Conclusions

In this paper, we presented a hierarchical fault-tolerant control synthesis framework. The framework allows for specifying different requirements for the healthy system and the faulty mode, both of which can be given by arbitrary LTL formulas. We then presented a MILP-based trajectory generation technique for the upper-level that incorporates reactiveness and robustness to faults and detection delays, respectively. For the lower-level, we showed that a feedback regulator can be designed using quasi-convex optimization. Future work will consider handling multiple fault modes and more general dynamics.

References

[1] C. Baier and J. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] J. W. Burdick, N. du Toit, A. Howard, C. Looman, J. Ma, R. M. Murray, and T. Wongpiromsarn. Sensing, navigation and reasoning technologies for the darpa urban challenge. Technical report, California Inst. of Technology, Pasadena, Jet Propulsion Laboratory, 2007.
[4] G. Fainekos, A. Girard, H. Kress-Gazit, and G. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, 2009.
[5] D. Frick, T. A. Wood, G. Ulli, and M. Kamgarpour. Robust control policies given formal specifications in uncertain environments. *IEEE control systems letters*, 1(1):20–25, 2017.
[6] A. Girault and É. Rutten. Automating the addition of fault tolerance with discrete controller synthesis. *Formal Methods in System Design*, 35(2):190, 2009.
[7] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.
[8] I. Gurobi Optimization. Gurobi optimizer reference manual. *URL http://www. gurobi. com*, 2015.
[9] F. Harirchi and N. Ozay. Guaranteed model-based fault detection in cyber-physical systems: a model invalidation approach. *Automatica*, 2018. to appear.
[10] J. Liu and N. Ozay. Finite abstractions with robustness margins for temporal logic-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 22:1–15, 2016.
[11] J. Liu, N. Ozay, U. Topcu, and R. M. Murray. Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Transactions on Automatic Control*, 58(7):1771–1785, 2013.
[12] O. Mickelin, N. Ozay, and R. M. Murray. Synthesis of correct-by-construction control protocols for hybrid systems using partial state information. In *Proc. of ACC*, pages 2305–2311, 2014.
[13] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 239–248. ACM, 2015.
[14] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone. Robust online motion planning via contraction theory and convex optimization. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5883–5890. IEEE, 2017.
[15] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.
[16] E. M. Wolff, U. Topcu, and R. M. Murray. Optimization-based trajectory generation with linear temporal logic specifications. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5319–5325. IEEE, 2014.
[17] L. Yang and N. Ozay. Provably-correct fault tolerant control with delayed information. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017.
[18] L. Yang, N. Ozay, and A. Karnik. Synthesis of fault tolerant switching protocols for vehicle engine thermal management. In *American Control Conference (ACC)*, 2016.

APPENDIX

A similar quasi-convex problem can be derived to minimize the estimation error bound E in Eq. (21), when the system in Eq. (17) has an output function

$$y_t = Cx_t + Du_t + v_t. \tag{32}$$

where v_t is measurement noise satisfying $\|v_t\| \leq e$ for all t. In this case, the estimated state $\hat{x}_t$ is given by an observer:

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + L(\hat{y}_t - y_t), \tag{33}$$

$$\hat{y}_t = C\hat{x}_t + Du_t, \tag{34}$$

where L is the observer gain. To find L such that $\|\hat{x}_t - x_t\|$ is bounded by a number E that is as small as possible, we derive a recurrence relation on $\|\hat{x}_t - x_t\|$ that is similar to Eq. (24). Combining Eq. (17), (32), (33), (34), we have

$$\hat{x}_{t+1} - x_{t+1} = (A + LC)(\hat{x}_t - x_t) - w_t - Lv_t, \tag{35}$$

which gives

$$\begin{aligned}\|\hat{x}_{t+1} - x_{t+1}\| &\leq \|(A + LC)(\hat{x}_t - x_t)\| + \|w_t\| + \|Lv_t\| \\ &\leq \|A + LC\|_*\|\hat{x}_t - x_t\| + \|w_t\| + \|L\|_*\|v_t\| \\ &\leq \|A + LC\|_*\|\hat{x}_t - x_t\| + d + \|L\|_* e, \end{aligned} \tag{36}$$

where d and e are the bounds on the norm of disturbance w_t and noise v_t respectively. Let E be the desired bound on $\|\hat{x}_t - x_t\|$, we need

$$\|\hat{x}_t - x_t\| \leq E \Rightarrow \|\hat{x}_{t+1} - x_{t+1}\| \leq E. \tag{37}$$

Substituting Eq. (36) into recurrence relation in Eq. (37) suggests that

$$\|A + LC\|_* \leq \frac{E - d - \|L\|_* e}{E}. \tag{38}$$

To minimize E, we formulate the following optimization problem:

$$\begin{aligned} \text{minimize}_{\varepsilon, L} \quad & \varepsilon \\ \text{s.t.} \quad & \|A + LC\|_* \leq \tfrac{\varepsilon - d - \|L\|_* e}{\varepsilon} \\ & \varepsilon \geq 0 \end{aligned} . \tag{39}$$

With a similar proof as that of Proposition 1, one can show that the problem in Eq. (39) is equivalent to the following quasi-convex problem:

$$\begin{aligned} \text{minimize}_{L} \quad & \tfrac{d + \|L\|_* e}{1 - \|A + LC\|_*} \\ \text{s.t.} \quad & \|A + LC\|_* \leq 1 \end{aligned} . \tag{40}$$

The problem in Eq. (40) is feasible only if (A, C) is detectable.