Synthesis of Failure-Robust Plans for Multi-Robot Systems under Temporal Logic Specifications

Feifei Huang, Shaoyuan Li and Xiang Yin

***Abstract*— In this study, we address the multi-robot path planning problem for tasks specified by linear temporal logic (LTL) formulae. Unlike existing studies, we take into account the possibility of robot failures, where a failed robot can no longer contribute to the completion of the LTL task. Our objective is to find a *failure-robust* path, which ensures that the LTL task can always be fulfilled, even if a maximum number of robots fail at any point during execution. To achieve this, we extend the mixed-integer linear programming (MILP) approach to the failure-robust setting. To overcome the computational complexity, we identify a fragment of LTL formulae called the free-union-closed LTL, which allows for more scalable synthesis without considering the global combinatorial issue. We present case studies to demonstrate our findings. Our approach provides a novel solution to the problem of multi-robot path planning under robot failures, offering a practical and efficient way to achieve robustness in the face of unforeseen events.**

I. Introduction

Multi-robot systems (MRSs) have been widely used in many different engineering cyber-physical systems such as persistent surveillance [1], [2], underwater and space exploration [3] and multitarget tracking [4]. In the study of MRSs, one of the most fundamental problems is the *path planning* problem that seeks to generate an executable path for each robot in the team [5]. Existing works have focused on finding paths for MRSs to enable reach-avoid navigation [6] or collision avoidance [7], [8]. However, there has been a growing interest in the MRS literature for path planning for high-level specifications [9]–[12].

Formal method provides a promising approach for automatically synthesizing such plans with correctness guarantees. In particular, linear temporal logic (LTL) provides a well-structured and user-friendly manner for specifying the temporal behaviors of the robots [13], [14]. In [15], authors propose a sampling-based algorithm to synthesize optimal plans for MRS. Authors in [16] provide a framework for generating optimal action-level behavior under resource constraints. In [17], task specifications which involve large numbers of discrete locations are considered and an iterator-based planning method is proposed. In [18], the authors develop joint policies for MRS that are robust to potential losses in communication. In [19], LTL path planning under unknown environments is investigated.

This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61833012) and the National Key Research and Development Program of China (2018AAA0101700).

Feifei Huang, Shaoyuan Li and Xiang Yin are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. {huangfeifei,syli,yinxiang}@sjtu.edu.cn.

Most of the existing research on LTL path planning assumes an ideal scenario where each robot operates flawlessly. However, in practical applications, robot *failures* are non-negligible in severe working conditions. For instance, a robot may suffer from permanent damage to its mechanical components or lack of power, rendering it useless in accomplishing the task. In such cases, the LTL task depends on the collaboration of all robots, and a critical robot's failure could potentially jeopardize the entire mission. Therefore, it is crucial to account for the impact of potential failures during the design phase to ensure robustness of the synthesized plans.

In this paper, our objective is to synthesize a set of paths that enable each robot to execute its own plan in a fully distributed manner. However, we consider a scenario where some robots may fail and can no longer contribute to the satisfaction of the overall task. Our goal is to synthesize a *failure-robust* plan for the team of robots, meaning that even in the event of a bounded number of robot failures, the remaining normal robots can still successfully accomplish the LTL task by following the pre-designed plans.

Our approach is to use mixed-integer linear programming (MILP) to encode the path of each robot, as well as the satisfaction of the LTL task. We introduce the concept of an enable sequence, which captures the working status of each robot. However, this basic approach can quickly become computationally intractable, as it requires consideration of all possible combinations of failed robots and failure instants. To mitigate this complexity, we identify a new fragment of LTL formula called *free-union-closed LTL*. Specifically, we show that for LTL tasks with this property, the corresponding planning problem can be solved more efficiently by restricting to the simple case that a robot will fail if and only if it fails initially. This approach significantly reduces the computational complexity of the problem.

There are several related works in the literature that address the issue of robot failures in path planning. For example, [20], [21] considers the target tracking problem for multi-robot systems and assumes that the sensors of the robots may fail due to attacks. The authors in [22] introduce the notion of counting temporal logic, which requires that one task be satisfied by multiple robots, addressing the issue of failure robustness to some extent. In [23], the authors provided a self-diagnositic LTL planning framework such that any failure can be detected within a finite delay. In our previous work [24], we considered a failure-robust reactive synthesis problem for LTL tasks. In this approach, each robot can adjust its plan online depending on the failure

status of other robots. However, this approach essentially requires a global controller or fully communication between each robot, which may not be practical or feasible in some applications. Our current work differs from previous works in that we focus on synthesizing a failure-robust plan for a team of robots that is executed in a fully distributed "open-loop" manner, without relying on a global controller or communication between robots. This approach may be more practical and cost-effective, making it more suitable for real-world applications.

II. Preliminaries

A. System Model

We consider a group of n mobile robots that operate within the same workspace. We denote by $\mathcal{I} = \{1, \ldots, n\}$ the index set for robots. The workspace is represented as a finite collection of distinct regions or states, which we denote as S. The connectivity between these states is captured by a symmetric neighborhood relation, denoted as $\mathcal{N} \subseteq S \times S$. In particular, if $(s, s') \in \mathcal{N}$, we say that states s and s' are neighborhoods. This neighborhood relation dictates that a robot can only move to a neighboring state at any given time instant.

We also assume that each state in the workspace may have certain properties of interest to the user. These properties could be any relevant characteristics, such as resource availability, environmental conditions, or task requirements. Each state may have a unique combination of these properties, making them potentially more or less desirable for the robots to visit or occupy. Let $\mathcal{AP}$ be a finite set of *atomic propositions*. The properties of each state is represented by a labeling function $L : S \to 2^{\mathcal{AP}}$, i.e., for each state $s \in S$, $L(s)$ denotes the set of atomic propositions hold at state s. For a set of states $S' \subseteq S$, we also denote $L(S') = \cup_{s \in S'} L(s)$.

B. Paths and Traces

For each robot $i \in \mathcal{I}$, a *path* is an infinite sequence over the state space satisfying the transition relation. Formally, we say $\mathbf{p}_i = \mathbf{p}_i^0 \mathbf{p}_i^1 \mathbf{p}_i^2 \cdots \in S^\omega$ is path if

$$\mathbf{p}_i^0 = s_i^{int} \text{ and } (\mathbf{p}_i^t, \mathbf{p}_i^{t+1}) \in \mathcal{N}, \forall t \in \mathbb{N}, \tag{1}$$

where s_i^{int} denotes the initial state of the ith robot. The set of all possible paths of the ith robot is denoted as $\mathbf{P}_i$.

We assume that the movement of each robot are fully synchronized, e.g., with a global time clock. Then the collective behavior of the team of n robots is a *joint path*, $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$, where each $\mathbf{p}_i \in \mathbf{P}_i$ is a path for the ith robot. The set of all possible joint paths of the n-robot system satisfying the above conditions is denote by $\mathbf{P}$.

Given that the robots are collaborating to achieve a common task, we define the set of atomic propositions that hold at each instant as the union of all atomic propositions satisfied by each robot. Then given a joint path $\mathbf{p}$, its *trace* is an infinite sequence of atomic propositions defined by $L(\mathbf{p}) = L(\cup_{i \in \mathcal{I}} \mathbf{p}_i^0) L(\cup_{i \in \mathcal{I}} \mathbf{p}_i^1) \cdots \in (2^{\mathcal{AP}})^\omega$.

C. Linear Temporal Logic Task

The desired cooperative task for the team of robots is described by a linear temporal logic (LTL) formula. Formally, the syntax of LTL is defined as follow:

$$\varphi ::= \text{ true } \mid a \in \mathcal{AP} \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2,$$

where $\neg$ and $\wedge$ are Boolean operators "negation" and "conjunction", respectively, and $\bigcirc$ and $\mathbf{U}$ are temporal operators "next" and "until", respectively.

Specifically, an LTL formula belongs to the class of co-safe LTL (scLTL) formulas if it exclusively utilizes the temporal operators $\bigcirc$, $\mathbf{U}$, $\mathbf{F}$, and the negation operator $\neg$ occurs only in front of an atomic proposition [25].

The satisfaction of scLTL formulas is guaranteed in finite time. For any infinite word $\sigma = \sigma_0 \sigma_1 \sigma_2 \cdots \in (2^{\mathcal{AP}})^\omega$, it satisfies a scLTL formula φ if it contains a finite good "prefix" that satisfies φ. We denote by $(\sigma, t) \models \varphi$ if σ satisfies the LTL formula φ at time t. When $t = 0$, we omit instant t and write it as $\sigma \models \varphi$. To provide further details on the semantics of LTL, we refer the reader to the book [26].

III. Failure-Robust LTL Planning

In practice, robot failures may occur, and failed robots cannot contribute to the overall task accomplishment. To address this issue, we present our model for the failure-robust planning problem in this section. Specifically, our model is developed based on the following assumptions:

- A1 Each robot may experience a failure at any point during its execution;
- A2 The failure is considered permanent, meaning that the robot cannot recover from the failure;
- A3 A failed robot is unable to contribute to the satisfaction of atomic propositions for the global task;
- A4 There can be at most $k < n$ robot failures during the entire execution.

To formalize the above setting, for each robot $i \in \mathcal{I}$, we introduce a binary sequence of the following form

$$\mathbf{e}_i = e_i^0 e_i^1 e_i^2 \cdots \in \{0, 1\}^\omega,$$

which is referred to as the *enable sequence*, to capture the failure status of each robot. Specifically, for each time instant $t \in \mathbb{N}$, $e_i^t = 1$ means that the ith robot is working normally at instant t and $e_i^t = 0$ means that the ith robot has failed at instant t. Furthermore, due to the permanent failure assumption A2, we require that

$$\forall t \in \mathbb{N}, \forall \Delta \in \mathbb{N} : \mathbf{e}_i^t = 0 \Rightarrow \mathbf{e}_i^{t+\Delta} = 0,$$

i.e., whenever robot i fails, it will stay failure from then on.

According to assumption A3, once a robot fails, it can no longer contribute to the overall task. To this end, we use symbol b to represent that a robot is at a "failure state", and we extend the labeling function to domain $S \cup \{b\}$ by adding $L(b) = \emptyset$. Therefore, for any path $\mathbf{p}_i$ of robot i, an enable

sequence $\mathbf{e}_i$ induces a new *failure-enabled path* over $S \cup \{b\}$ denote by $\mathbf{p}_i \otimes \mathbf{e}_i$, where

$$(\mathbf{p}_i \otimes \mathbf{e}_i)^t = \begin{cases} \mathbf{p}_i^t & \text{if } \mathbf{e}_i^t = 1 \\ b & \text{if } \mathbf{e}_i^t = 0 \end{cases}, \quad \forall t \in \mathbb{N}. \tag{2}$$

Similarly, the trace of $\mathbf{p}_i \otimes \mathbf{e}_i$ is $L(\mathbf{p}_i \otimes \mathbf{e}_i)$, where

$$L(\mathbf{p}_i \otimes \mathbf{e}_i)^t = \begin{cases} L(\mathbf{p}_i^t) & \text{if } \mathbf{e}_i^t = 1 \\ \emptyset & \text{if } \mathbf{e}_i^t = 0 \end{cases}, \quad \forall t \in \mathbb{N}. \tag{3}$$

For the entire team of n robots, a *joint enable sequence* is a n-tuple of form

$$\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n),$$

where each $\mathbf{e}_i$ is an enable sequence. Furthermore, by the assumption of bounded number of failure robots in A4, we further require that

$$\sum_{i \in \mathcal{I}} \mathbf{e}_i^t \geq n - k, \forall t \in \mathbb{N}.$$

We denote by $\mathbf{E}_k$ the set of all possible joint enable sequences, where k is the the maximum number of allowed failure robots. Then given a joint path $\mathbf{p} \in \mathbf{P}$ and a joint enable sequence $\mathbf{e} \in \mathbf{E}_k$, the joint failure-enabled path is $\mathbf{p} \otimes \mathbf{e} = (\mathbf{p}_1 \otimes \mathbf{e}_1, \ldots, \mathbf{p}_n \otimes \mathbf{e}_n)$ with trace $L(\mathbf{p} \otimes \mathbf{e})$.

Now, we formulate the failure-robust LTL path planning problem as follows.

Problem 1 (Failure-Robust LTL Planning Problem): For a team of n robots system in a workspace with states S and neighborhood relation $\mathcal{N}$, given an LTL formula φ, a maximum number k of allowed failure robots, find a joint path $\mathbf{p} \in \mathbf{P}$ such that $\forall \mathbf{e} \in \mathbf{E}_k : L(\mathbf{p} \otimes \mathbf{e}) \models \varphi$.

We make the following remarks regarding our setting.

Remark 1: The integer k here represents the maximum number of failures that the team can tolerate. This value can be determined beforehand based on worst-case scenarios or chosen to ensure the team's ability to complete the task under possible failures. It is important to note that k is typically much smaller than the total number n of robots in the team. If k is set to a value that is too large, it may indicate that the hardware of the robots needs to be improved to increase their overall robustness. In such a scenario, it would be more appropriate to focus on improving the hardware rather than relying on robust planning to compensate for a large number of potential failures.

Remark 2: In our previous work [24], a reactive strategy was obtained through a centralized approach, where each robot needed to inquire about the global information of all robots to be aware of their potential failures online. However, in the current work, although the problem still needs to be solved in a global manner, the implementation of the joint plans is fully distributed in nature. This means that each robot will only execute its own path in an open-loop fashion without exchanging any information. This distributed approach is more realistic for severe environments where online communication is either costly or not possible at all.

IV. MILP-Based Failure-Robust Solution

In the field of LTL path planning, an efficient approach is to use mixed-integer linear programming (MILP). This approach was first proposed by [27] for the standard LTL planning problem. The main concept behind this approach is to introduce a set of variables that encode the states of the robots and the satisfaction of the task. In this section, we extend the use of MILP to the failure-robust LTL planning problem.

A. MILP-Based Standard LTL Planning

Encoding Paths: As the task is specified by scLTL, we restrict our attention to paths in finite form

$$\mathbf{p} = \mathbf{p}^0 \mathbf{p}^1 \cdots \mathbf{p}^{h-1}.$$

To solve the programming problem, we fix h and refer to it as the *planning horizon*. In general, h can be an arbitrary integer.

Then we introduce the following variables:

- **state variables**: $\mathbf{p}^0, \mathbf{p}^1, \ldots, \mathbf{p}^{h-1} \in S^n$ representing the joint state of the team of robots within the planning horizon;

State variables should satisfy the **dynamic constraint** in Equation (1).

Encoding LTL Tasks: To encode the LTL task φ, let Φ be the set of all sub-formulae in φ including the atomic propositions. Then for each $\phi \in \Phi$, we introduce a set of

- **LTL variables**: $z_\phi^0, z_\phi^1, \ldots, z_\phi^{h-1} \in \{0, 1\}$ representing the satisfaction of each sub-formula at each instant;

The LTL variables should satisfy the **LTL constraints** defined as follows: for each $t \in \{0, 1, \ldots, h-1\}$, we have

- atomic propositions: if $\phi = a \in \mathcal{AP}$, then

$$z_a^t = 1 \text{ iff } a \in L(\mathbf{p}^t) \tag{4}$$

- negation: if $\phi = \neg\phi'$, then

$$z_\phi^t = 1 \text{ iff } z_{\phi'}^t = 0 \tag{5}$$

- conjunction: if $\phi = \bigwedge_{i=1}^m \phi_i$, then

$$z_\phi^t = 1 \text{ iff } \forall i \in \{1, 2, \cdots m\} : z_{\phi_i}^t = 1 \tag{6}$$

- next: if $\phi = \bigcirc\phi'$, then

$$z_\phi^t = 1 \text{ iff } z_{\phi'}^{t+1} = 1 \tag{7}$$

- until: if $\phi = \phi_1 \mathbf{U} \phi_2$, then

$$z_\phi^t = 1 \text{ iff } z_{\phi_2}^t = 1 \text{ or } z_{\phi_1}^t = 1, z_\phi^{t+1} = 1 \tag{8}$$

The authors are referred to [27] for more details on how to expressed the aforementioned constraints in MILP formulation. It should be noted that due to the finite planning horizon in our study, certain temporal operators that require future LTL variables, such as $\bigcirc$ and $\mathbf{U}$, may not have meaningful interpretations. However, we emphasize that as long as the value of h is set to a sufficiently large value, the individual variable losses will not affect the solution of the problem.

B. Robust constraint

In order to synthesize a failure-robust plan, we need to take into account the effect of enable sequences. It should be noted that the original definition of joint enable sequences is infinite, while in the MILP formulation we restrict our attention to a finite horizon of length h. Therefore, the definition of joint enable sequences also needs to be modified accordingly.

Specifically, for each robot $i \in \mathcal{I}$, we introduce a set of

- **failure-enable variables:** $e_i^0, e_i^1, \ldots, e_i^{h-1} \in \{0,1\}$ representing the failure status of robot i.

Clearly, the failure-enable variables should satisfy the following **permanent failure constraint**

$$\text{if } \mathbf{e}_i^t = 0 \text{ then } \mathbf{e}_i^j = 0, \forall t \leq j \leq h, \tag{9}$$

and the **maximum failure constraint** that $\forall t \in \{0, 1, \ldots, h-1\}$

$$\sum_{i \in \mathcal{I}} e_i^t \geq n - k. \tag{10}$$

Note that Equations (9) and (10) are constraints on the finite joint enable sequences, and these constraints are related with the planning horizon h and the maximum allowed number of failure robots k. Therefore, we define $\mathbf{E}_{k,h}$ as the set of all finite joint enable sequences satisfying the aforementioned constraints.

Finally, by considering the failure-enable variables, the atomic proposition variables should further been restricted by considering both the labeling function and the enable variable. Therefore, we need to replace the constraint on atomic propositions (4) by the following **label-enable constraint**

$$z_a^t = 1 \text{ iff } a \in L(\mathbf{p}^t \otimes \mathbf{e}^t) \tag{11}$$

Given an instance of Problem 1, the following robust-feasibility problem can be formed:

$$\begin{aligned} & \text{Find } \mathbf{p} \in \mathbf{P} \\ s.t. \quad & \text{dynamic constraint } (1) \text{ on } \mathbf{p} \\ & \text{label-enable constraint}(11) \text{ on } \forall a \in \mathcal{AP} \\ & \text{LTL constraints } (5-8) \text{ on } \forall \phi \in \Phi \\ & \text{satisfaction constraint } z_\varphi^0 = 1, \\ & \text{for } \forall \mathbf{e} \in \mathbf{E}_{k,h} \end{aligned}$$

V. Scalable Synthesis for Initial Robustness

In the failure-robust planning problem, we require the planned path $\mathbf{p} \in \mathbf{P}$ to satisfy

$$\forall \mathbf{e} \in \mathbf{E}_k : L(\mathbf{p} \otimes \mathbf{e}) \models \varphi.$$

Hereafter, we refer the above condition to as *global robustness* since the enable sequence set $\mathbf{E}_k$ consider possible failures globally.

Here, we further restrict our attention to a special case where robots are only allowed to fail initially. Furthermore, for the worst-case analysis, we require that there are exactly k failure robots initially. To this end, we define the set of all possible initial-failure enable sequences by

$$\mathbf{E}_k^{int} = \left\{ \mathbf{e} \in \mathbf{E}_k \mid \sum \mathbf{e}_i^0 = n - k \right\}.$$

Then we say that a planned path $\mathbf{p} \in \mathbf{P}$ is *initially-robust* if the LTL task is still fulfilled whenever there are exactly k robots that fail initially, but all remaining robots operate correctly for the entire planning horizon. This means that the planned path is robust to initial robot failures, and we only need to consider all possible initial failure scenarios to ensure robustness.

Definition 5.1 (Initial Robustness): Given an LTL formula φ, a maximum number k of failures, a joint path $\mathbf{p}$ is said to be *initially robust* if $\forall \mathbf{e} \in \mathbf{E}_k^{int} : L(\mathbf{p} \otimes \mathbf{e}) \models \varphi$.

Based on the above definition, we define a new failure-robust planning problem as follows.

Problem 2: For a team of n robots system in a workspace with states S and neighborhood relation $\mathcal{N}$, given an LTL formula φ, a the maximum number k of allowed failure robots, find a joint path $\mathbf{p} \in \mathbf{P}$ such that $\forall \mathbf{e} \in \mathbf{E}_k^{int} : L(\mathbf{p} \otimes \mathbf{e}) \models \varphi\}$.

Note that initial robustness still does not specify which robots are broken, so the planning path should still satisfy the LTL task when any k robots are broken from the start.

Analogously, we can solve a MILP problem to get a initially-robust joint path.

$$\begin{aligned} & \text{Find } \mathbf{p} \in \mathbf{P} \\ s.t. \quad & \text{dynamic constraint } (1) \text{ on } \mathbf{p} \\ & \text{label-enable constraint}(11) \text{ on } \forall a \in \mathcal{AP} \\ & \text{LTL constraints } (5-8) \text{ on } \forall \phi \in \Phi \\ & \text{satisfaction constraint } z_\varphi^0 = 1, \\ & \text{for } \forall \mathbf{e} \in \mathbf{E}_{k,h}^{int} \end{aligned}$$

Here, $\mathbf{E}_{k,h}^{int} \subseteq \mathbf{E}_{k,h}$ is the set of finite initial-failure enable sequences that $\sum \mathbf{e}_i^0 = n - k$.

The above MILP for initial robustness improves the original MILP for global robustness in twofold. Firstly, the MILP for initial robustness considers exact k failures of robots, rather than any subset of k robots. Secondly, and more importantly, the MILP for initial robustness only considers robot failures that occur initially, which effectively avoids the exponential growth of variables that would occur if robot failures were considered over an extended planning horizon h. By limiting the consideration of robot failures to the initial state, the MILP for initial robustness is able to significantly reduce the complexity of the problem and provide a more efficient and manageable solution.

Due to the computational efficiency, our general idea is to solve Problem 2 to find an initially-robust solution rather than solving Problem 1. However, our objective is still to require that the synthesized plan is globally-robust. Clearly, if a planning is globally-robust, then it is initially-robust. However, the converse direction is not true in general as shown by the following counter examples.

Example 1: Suppose we have a team of two robots and consider the case of $k = 1$. The global LTL task for the team of robots is given by

$$\varphi = \mathbf{F}(a \wedge \neg b) \vee \mathbf{F}(\neg a \wedge b),$$

where a and b represent two different types of regions. The task requires the robots to eventually visit either region a or region b, but these two regions cannot be visited both. If we only consider initial robustness, then we can design a plan where a and b are visited simultaneously by two robots. However, this plan is not globally-robust since if neither robot fails, the entire task will fail as regions a and b may be visited simultaneously and we can't find a moment that exactly one region is visited.

The above example illustrates why initial robustness does not necessarily imply global robustness. To identify fragments of LTL formulae where initial robustness and global robustness are equivalent, we identify and exclude those scenarios where initial robustness may lead to plans that do not remain robust in the face of potential failures over an extended planning horizon.

A. Free-Union LTL formulae

Although initial robustness and global robustness are not equivalent in general, as we discussed earlier, there are certain fragments of LTL formulae where they are equivalent. In our work, we identify such a fragment of LTL formulae called the *free-union-closed LTL*.

Definition 5.2: Let $\mathbf{w} = w_0 w_1 w_2 \cdots, \mathbf{v} = v_0 v_1 v_2 \cdots \in (2^{\mathcal{AP}})^\omega$ be two infinite words. We say word $\mathbf{w}' = w'_0 w'_1 w'_2 \cdots \in (2^{\mathcal{AP}})^\omega$ is a *free-union* of $\mathbf{w}$ and $\mathbf{v}$ if for some $j \in \mathbb{N}$, we have

$$w'_i = \begin{cases} w_i \cup v_i & \text{if } i < j \\ w_i & \text{if } i \geq j \end{cases} \tag{14}$$

We denote by $\mathbf{w} \biguplus \mathbf{v}$ the set of all free-unions of $\mathbf{w}$ and $\mathbf{v}$.

Intuitively, we can consider $j \in \mathbb{N}$ as the point in time from which the word $\mathbf{v}$ becomes ineffective. Thus, $\mathbf{v}$ can only contribute to the atomic propositions union of $\mathbf{w}$ up to instant $j-1$. It is important to note that the free-union of $\mathbf{w}$ and $\mathbf{v}$ is not unique, as we are free to choose the instant j to be any value from 0 to infinity.

Based on the free-union operator of two words, we introduce the fragment of *free-union-closed LTL*. This fragment ensures that any two words satisfying the LTL formula will have any of their free-unions satisfy the same LTL formula.

Definition 5.3 (Free-Union-Closed LTL): Given an LTL formula φ and let $\mathsf{word}(\varphi) = \{\sigma \in (2^{\mathcal{AP}})^\omega \mid \sigma \models \varphi\}$ be the set of all words satisfying φ. Formula φ is said to be *free-union-closed* if for any $\mathbf{w}, \mathbf{v} \in \mathsf{word}(\varphi)$, we have $\mathbf{w} \biguplus \mathbf{v} \subseteq \mathsf{word}(\varphi)$.

The main result of this section is that if an LTL formula is closed under the free-union operator, then its initial robustness and global robustness are equivalent. Therefore, in order to solve Problem 1, we can solve Problem 2 instead. Due to space constraint, all proofs in this paper are omitted.

Theorem 1: For free-union-closed LTL formula φ, initial robustness of joint path $\mathbf{p}$ is equivalent to global robustness, i.e., $\forall \mathbf{e}' \in \mathbf{E}_k^{int}, L(\mathbf{p} \otimes \mathbf{e}') \models \varphi \Leftrightarrow \forall \mathbf{e} \in \mathbf{E}_k, L(\mathbf{p} \otimes \mathbf{e}) \models \varphi$.

B. Instances of Free-Union-Closed LTL

In this paper, we do not discuss how to systematically check whether or not a given LTL formula is free-union-closed. Instead, we identify a class of scLTL formulae that is free-union-closed by construction. To the end, we start from Boolean expressions and introduce the following definitions.

Definition 5.4: Let ϕ be a Boolean expression over $\mathcal{AP}$. We say that formula ϕ is

- *monotone* if for any $X_1, X_2 \in 2^{\mathcal{AP}}$, we have $X_1 \models \phi \wedge X_1 \subseteq X_2 \Rightarrow X_2 \models \phi$;
- *closed* if for any $X_1, X_2 \in 2^{\mathcal{AP}}$, we have $X_1 \models \phi \wedge X_2 \models \phi \Rightarrow X_1 \cup X_2 \models \phi$.

The followings are some immediate observations for monotonicity and closedness of Boolean expressions.

- First, we observe that if ϕ is monotone, then it is also closed. But the converse direction is not true in general. For example, for $\phi = \neg a$, it is closed but is not monotone;
- Second, we observe that, if ϕ_1 and ϕ_2 are monotone, their conjunction $\phi_1 \wedge \phi_2$ and disjunction $\phi_1 \vee \phi_2$ are also monotone. However, the negation $\neg\phi_1$ may not be monotone.

Based on the monotonicity and closedness of Boolean expressions, now we show that two commonly used patterns of scLTL formula are free-union-closed, and therefore, the corresponding plan can be efficiently synthesized based on initial robustness.

Proposition 1: Let ϕ be a Boolean expression over $\mathcal{AP}$. Then we have

- if ϕ is monotone, then LTL formulae $\mathbf{F}\phi$ is free-union-closed;
- if ϕ_1 is closed and ϕ_2 is monotone, then LTL formulae $\phi_1 \mathbf{U} \phi_2$ is free-union-closed.

The above provided two patterns of LTL formulae are widely used in describing the behavior of team of robots. For example, $\mathbf{F}\phi$ is a reachability requirement, while $\phi_1 \mathbf{U} \phi_2$ can present a priority requirement.

VI. Case Study

In this section, we present an inspection maintenance task as case study to illustrate the proposed failure-robust LTL planning problem.

We consider a workspace illustrated in Figure 1 and define the set of atomic propositions as $\mathcal{AP} = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, stat\}$. The yellow grids represent seven checkpoints that the team of robots need to inspect, and they are labeled as c_i for $i = 1, 2, \ldots, 7$. The green grids denote maintenance stations of the robots and are represented by the common proposition $stat$. Each robot start from the maintenance stations and the overall task of the robots is to inspect all checkpoints, which can be formally expressed by

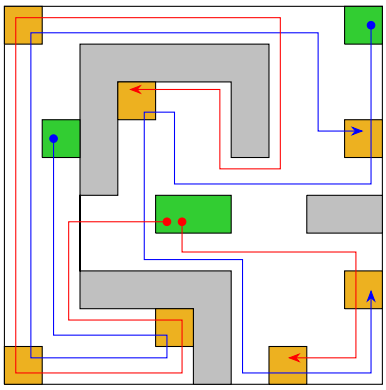

Fig. 1. A failure-robust plan for a 4-robot system to inspect all the seven checkpoints (represented as yellow grids) and return to the maintenance stations (represented as green grids).

the following LTL formula

$$\varphi = \bigwedge_{i=1,\ldots,7} \mathbf{F} c_i$$

By applying the MILP-based approach, we can synthesize a failure-robust plan for $k = 1$ in the workspace shown in Figure 1. The plan, illustrated in the figure, ensures that each checkpoint is inspected by two different robots, thereby ensuring robustness in the case of one possible robot failure during the execution.

VII. Conclusions

This paper presented an approach for synthesizing joint plans for teams of robots based on temporal logic specifications, ensuring robustness against a maximum of k robot failures. We also identified a scalable fragment to improve the scalability of the synthesis procedure. Our work is a first step towards the synthesis of failure-robust plans for LTL tasks. Several future directions can be explored to extend our approach. First, we aim to provide a systematic procedure for verifying whether an arbitrary LTL formula is free-union-closed or not. Second, we plan to identify broader fragments of free-union-closed LTL formulas to increase the applicability of our approach. Finally, we aim to investigate the performance optimization problem in addition to the failure-robustness requirement.

References

[1] Sepehr Seyedi, Yasin Yazicioğlu, and Derya Aksaray. Persistent surveillance with energy-constrained uavs and mobile charging stations. *IFAC-PapersOnLine*, 52(20):193–198, 2019.

[2] Peng Lv, Guangqing Luo, Ziyue Ma, Shaoyuan Li, and Xiang Yin. Optimal multi-robot path planning for cyclic tasks using Petri nets. *Control Engineering Practice*, 2023.

[3] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.

[4] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72, 2013.

[5] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. Path planning and trajectory planning algorithms: A general overview. *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, pages 3–27, 2015.

[6] Rupak Majumdar, Kaushik Mallik, Mahmoud Salamati, Sadegh Soudjani, and Mehrdad Zareian. Symbolic reach-avoid control of multi-agent systems. In *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, pages 209–220, 2021.

[7] Suiyi He, Jun Zeng, Bike Zhang, and Koushil Sreenath. Rule-based safety-critical control design using control barrier functions with application to autonomous lane change. In *2021 American Control Conference*, pages 178–185. IEEE, 2021.

[8] Chengyang Peng, Octavian Donca, and Ayonga Hereid. Safe path planning for polynomial shape obstacles via control barrier functions and logistic regression. *arXiv preprint arXiv:2210.03704*, 2022.

[9] Zhiyu Liu, Jin Dai, Bo Wu, and Hai Lin. Communication-aware motion planning for multi-agent systems from signal temporal logic specifications. In *2017 American Control Conference*, pages 2516–2521. IEEE, 2017.

[10] Xinyi Yu, Xiang Yin, Shaoyuan Li, and Zhaojian Li. Security-preserving multi-agent coordination for complex temporal logic tasks. *Control Engineering Practice*, 123:105130, 2022.

[11] Dhaval Gujarathi and Indranil Saha. Mt*: Multi-robot path planning for temporal logic specifications. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 13692–13699. IEEE, 2022.

[12] Weijie Shi, Zhou He, Ziyue Ma, Ning Rang, and Xiang Yin. Security-preserving multi-robot path planning for Boolean specification tasks using labeled Petri nets. *IEEE Control Systems Letters*, 2023.

[13] Shuo Yang, Xiang Yin, Shaoyuan Li, and Majid Zamani. Secure-by-construction optimal path planning for linear temporal logic tasks. In *59th IEEE Conference on Decision and Control*, pages 4460–4466, 2020.

[14] Bohan Cui, Keyi Zhu, Shaoyuan Li, and Xiang Yin. Security-aware reinforcement learning under linear temporal logic specifications. In *IEEE International Conference on Robotics and Automation*, pages 12367–12373, 2023.

[15] Yiannis Kantaros and Michael M Zavlanos. Sampling-based optimal control synthesis for multirobot systems under global temporal tasks. *IEEE Transactions on Automatic Control*, 64(5):1916–1931, 2018.

[16] Philipp Schillinger, Mathias Bürger, and Dimos V Dimarogonas. Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems. *The international journal of robotics research*, 37(7):818–838, 2018.

[17] Sebastián A Zudaire, Martin Garrett, and Sebastián Uchite. Iterator-based temporal logic task planning. In *2020 IEEE International Conference on Robotics and Automation*, pages 11472–11478. IEEE, 2020.

[18] Mustafa O Karabag, Cyrus Neary, and Ufuk Topcu. Planning not to talk: Multiagent systems that are robust to communication loss. *arXiv preprint arXiv:2201.06619*, 2022.

[19] Jianing Zhao, Keyi Zhu, Shaoyuan Li, and Xiang Yin. To explore or not to explore: Regret-based LTL planning in partially-known environments. In *22nd IFAC World Congress*, 2023.

[20] Lifeng Zhou, Vasileios Tzoumas, George J Pappas, and Pratap Tokekar. Resilient active target tracking with multiple robots. *IEEE Robotics and Automation Letters*, 4(1):129–136, 2018.

[21] Lifeng Zhou, Vasileios Tzoumas, George J Pappas, and Pratap Tokekar. Distributed attack-robust submodular maximization for multirobot planning. *IEEE Transactions on Robotics*, 38(5):3097–3112, 2022.

[22] Yunus Emre Sahin, Petter Nilsson, and Necmiye Ozay. Multirobot coordination with counting temporal logics. *IEEE Transactions on Robotics*, 36(4):1189–1206, 2019.

[23] Jianing Zhao, Shuqi Wang, and Xiang Yin. Failure-aware self-diagnostic task planning under temporal logic specifications. In *22nd IFAC World Congress*, 2023.

[24] Feifei Huang, Xiang Yin, and Shaoyuan Li. Failure-robust multi-robot tasks planning under linear temporal logic specifications. In *2022 13th Asian Control Conference*, pages 1052–1059. IEEE, 2022.

[25] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. *Formal methods for discrete-time dynamical systems*, volume 15. Springer, 2017.

[26] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[27] Yunus Emre Sahin, Petter Nilsson, and Necmiye Ozay. Provably-correct coordination of large collections of agents with counting temporal logic constraints. In *Proceedings of the 8th International Conference on Cyber-Physical Systems*, pages 249–258, 2017.