

# A Stackelberg game approach for signal temporal logic motion planning with uncontrollable agents<sup>☆,☆☆</sup>

Bohan Cui<sup>a</sup>, Xinyi Yu<sup>b</sup>, Alessandro Giua<sup>c</sup>, Xiang Yin<sup>a,\*</sup>

<sup>a</sup> School of Automation and Sensing, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>b</sup> Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

<sup>c</sup> Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari 09123, Italy

## ARTICLE INFO

### Keywords:

Motion planning  
Signal temporal logic  
Stackelberg game  
Control synthesis

## ABSTRACT

In this paper, we investigate the motion planning problem for Signal Temporal Logic (STL) specifications in the presence of uncontrollable agents. Existing works mainly address this problem in a robust control setting by assuming the uncontrollable agents are adversarial and accounting for the worst-case scenario. While this approach ensures safety, it can be overly conservative in scenarios where uncontrollable agents have their own objectives that are not entirely opposed to the system's goals. Motivated by this limitation, we propose a new framework for STL motion planning within the *Stackelberg game* setting. Specifically, we assume that the system controller, acting as the leader, first commits to a plan, after which the uncontrollable agents, acting as followers, take a best response based on the committed plan and their own objectives. Our goal is to synthesize a control sequence for the leader such that, for any rational followers producing a best response, the leader's STL task is guaranteed to be satisfied. We present an effective solution to this problem by transforming it into a single-stage optimization problem and leveraging counter-example guided synthesis techniques. We demonstrate that the proposed approach is sound and identify conditions under which it is optimal. Simulation results are also provided to illustrate the effectiveness of the proposed framework.

## 1. Introduction

Decision making and task planning are fundamental challenges in the design of autonomous systems. In recent years, formal methods have gained significant attention for high-level task planning due to their ability to provide rigorous specifications and guarantees for system correctness [1–3]. Among these methods, temporal logic has emerged as a particularly expressive tool, enabling the description of complex temporal and spatial constraints in a mathematically rigorous yet user-friendly manner. Particularly, Signal Temporal Logic (STL), initially introduced in [4], has been extensively studied as it is well-suited for capturing spatial-temporal requirements of real-valued signals in dynamic and uncertain environments [5–7].

In the context of motion planning for STL specifications, one of the most widely used methods is the optimization-based approach [8,9]. This approach encodes the satisfaction of STL formulae using binary variables and transforms the motion planning problem into a Mixed Integer Programming (MIP) problem. An alternative approach leverages control barrier functions, which

<sup>☆</sup> This article is part of a Special issue entitled: 'TC1.3 Discrete-event and hybrid systems(IFAC WC 2026)' published in Nonlinear Analysis: Hybrid Systems.

<sup>☆☆</sup> This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61833012).

\* Corresponding author.

E-mail addresses: [bohan\\_cui@sjtu.edu.cn](mailto:bohan_cui@sjtu.edu.cn) (B. Cui), [yinxiang@sjtu.edu.cn](mailto:yinxiang@sjtu.edu.cn) (X. Yu), [giua@unica.it](mailto:giua@unica.it) (A. Giua), [xinyi.yu12@usc.edu](mailto:xinyi.yu12@usc.edu) (X. Yin).

provide sufficient conditions to characterize forward invariant sets corresponding to the satisfaction regions of STL formulae [10,11]. These methods primarily address the open-loop control problem, where the goal is to synthesize a single control sequence that satisfies the STL specifications without external disturbances.

When external disturbances or uncertainties are present, feedback control based on online information becomes essential. In the context of STL control synthesis, a direct approach is to integrate open-loop trajectory synthesis methods into the framework of receding horizon control; see, e.g., [12–18]. This involves solving the optimization problem iteratively in real-time, where only the latest control input is applied at each step. Particularly, to ensure the feasibility of recursive computation, it is necessary to solve a robust optimization problem that guarantees the satisfaction of the STL task under all possible disturbances [17,19–21].

The above-mentioned existing works on STL control synthesis in the presence of external inputs essentially fall into the category of *zero-sum games*. In this framework, the controller aims to ensure the satisfaction of the STL formula under *all possible* external inputs, which can be interpreted as a second player whose objective is entirely opposed to the system's STL task. While zero-sum settings are indispensable for purely adversarial environments, they often lead to over-conservatism when uncontrollable agents possess their own rational objectives. For example, consider a two-agent system where only Agent 1 is controllable, and its STL task depends on the joint trajectory. In a zero-sum setting, the controller must plan conservatively to ensure satisfaction against all possible behaviors of Agent 2. However, if Agent 2 has its own (not necessarily conflicting) STL objective, it is more appropriate to account for its rational behavior. This leads naturally to a *non-zero-sum* formulation, where each agent pursues a distinct objective.

**Our Contributions:** Motivated by the above considerations, this paper complements zero-sum approaches by addressing scenarios where the task information of the uncontrollable agent is available through shared protocols or mission announcements within the framework of a Stackelberg game or leader–follower game. Our contributions are summarized as follows:

- First, unlike the scenario in zero-sum game formulations, in which the uncontrollability is modeled as a purely adversarial disturbance drawn from a fixed uncertainty set, we consider and formalize the STL motion planning problem in scenarios where both the controller (leader) and the uncontrollable agent (follower) are task-driven and pursue their own objectives, each specified by an STL formula. These objectives are not necessarily antagonistic, and the follower's task is assumed to be known to the leader. Specifically, the leader first commits to a sequence of control inputs, and the follower then computes a best-response control sequence that optimizes its own task.
- Second, depending on whether the follower's goal is satisfiable under the leader's strategy, we show that this problem can yield two types of solutions: *cooperative solutions*, where the leader anticipates follower behavior and enables satisfaction of both STL goals when possible, and *antagonistic solutions*, where the leader forces the follower to be non-interfering by eliminating the possibility of the follower achieving its STL task. For each type of solution, we propose an effective approach to synthesize optimal control sequences for the leader such that, for any rational follower producing a best response, the leader's STL task is guaranteed to be satisfied.
- Finally, we provide two case studies on robot path planning to illustrate the effectiveness of our proposed framework. Moreover, while our current approach fits an MIP setting, which is poorly scalable, the Stackelberg framework proposed in this paper is also aligned with smooth STL semantics and gradient-based optimization.

**Related Works:** Our work is closely related to hierarchical games within the context of dynamic game theory [22,23] and control synthesis involving uncontrollable agents subject to temporal logic tasks [24–26]. Some existing works have already explored the integration of these two areas. For instance, [27] addresses the minimum violation problem with numeric utilities for stochastic systems within the framework of Stackelberg games. In [28], the authors investigate the use of deception in strategic planning for adversarial environments by introducing a hierarchical hyper-game model. This model enables the leader to develop deceptive strategies to influence the follower's perception, thereby maximizing the likelihood of achieving its own objective. Additionally, [29] studies supervisory control of discrete-event systems within the Stackelberg game framework. However, these works primarily focus on either simple safety specifications or linear temporal logic specifications. To the best of our knowledge, the application of Stackelberg games to STL control synthesis remains unexplored.

**Organizations:** The remainder of this paper is organized as follows. Section 2 presents some necessary preliminaries. In Section 3, we introduce the framework of the leader–follower game and formulate the Stackelberg STL synthesis problem (SPP-STL). Section 4 presents a two-stage synthesis procedure for finding the cooperative and antagonistic solutions to SPP-STL. In Section 5, we conduct two case studies to illustrate the application of our approach. Finally, we conclude the paper and discuss future directions in Section 6.

## 2. Preliminaries

### 2.1. System model

We consider discrete-time control systems of the form:

$$x_{t+1} = f(x_t, u_t, w_t)$$

where  $t = 0, 1, \dots$  are the time indices,  $x_t \in X \subseteq \mathbb{R}^n$  is the system state at time  $t$ ,  $u_t \in U \subseteq \mathbb{R}^m$  denotes the bounded input of the system controller at time  $t$  and  $w_t \in W \subseteq \mathbb{R}^e$  denotes the external input. We assume that the initial state is fixed as  $x_0 \in X$ . Given a sequence of control inputs  $\mathbf{u}_{0:N-1} = u_0 u_1 \dots u_{N-1} \in U^N$  and a sequence of external inputs  $\mathbf{w}_{0:N-1} = w_0 w_1 \dots w_{N-1} \in W^N$  with horizon  $N$ , the resulting state sequence of the system is  $\xi_f(x_0, \mathbf{u}_{0:N-1}, \mathbf{w}_{0:N-1}) = \mathbf{x}_{0:N} = x_0 x_1 \dots x_N \in X^{N+1}$ , where for all  $i = 0, 1, \dots, N-1$  we have  $x_{i+1} = f(x_i, u_i, w_i)$ . Hereafter, we will drop the subscript  $0 : N-1$  when the horizon  $N$  is clear from the context.

## 2.2. Signal temporal logic specifications

We consider formal specifications described by signal temporal logic (STL) formulae, whose syntax is as follows

$$\phi ::= \top \mid \pi^\mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathbf{U}_{[a,b]}\phi_2,$$

where  $\top$  is the true predicate,  $\pi^\mu$  is an atomic predicate whose truth value is determined by the sign of its underlying predicate function  $\mu : \mathbb{R}^n \rightarrow \mathbb{R}$ , i.e.,  $\pi^\mu$  is true at time  $t$  if and only if  $\mu(x_t) \geq 0$ . Notations  $\neg$  and  $\wedge$  are the standard Boolean operators “negation” and “conjunction”, respectively, and  $\mathbf{U}_{[a,b]}$  is the temporal operator “until”, where  $a, b \in \mathbb{N}$  and  $a \leq b$ . One can further induce operators “implication” by  $\phi_1 \Rightarrow \phi_2 := \neg\phi_1 \vee \phi_2$ , “disjunction” by  $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$ , “eventually” by  $\mathbf{F}_{[a,b]}\phi := \top \mathbf{U}_{[a,b]}\phi$  and “always” by  $\mathbf{G}_{[a,b]}\phi := \neg\mathbf{F}_{[a,b]}\neg\phi$ .

Given a state sequence  $\mathbf{x}$ , we denote by  $(\mathbf{x}, t) \models \phi$  the satisfaction of STL formula  $\phi$  at time  $t$ . Formally, the Boolean semantics of STL are defined as recursively follows:

$$\begin{aligned} (\mathbf{x}, t) \models \pi^\mu & \Leftrightarrow \mu(x_t) \geq 0 \\ (\mathbf{x}, t) \models \neg\phi & \Leftrightarrow \neg((\mathbf{x}, t) \models \phi) \\ (\mathbf{x}, t) \models \phi_1 \wedge \phi_2 & \Leftrightarrow (\mathbf{x}, t) \models \phi_1 \wedge (\mathbf{x}, t) \models \phi_2 \\ (\mathbf{x}, t) \models \phi_1 \mathbf{U}_{[a,b]}\phi_2 & \Leftrightarrow \exists t' \in [t+a, t+b] : (\mathbf{x}, t') \models \phi_2 \\ & \text{and } \forall t'' \in [t, t'] : (\mathbf{x}, t'') \models \phi_1 \end{aligned}$$

We write  $\mathbf{x} \models \phi$  instead of  $(\mathbf{x}, 0) \models \phi$  for simplicity.

In this paper, we consider bounded-time STL formulae, in which time intervals  $[a, b]$  are bounded. In some cases, it is useful to further consider the quantitative semantics, where the robustness value  $\rho^\phi(\mathbf{x}) \in \mathbb{R}$  is used to quantify the degree to which  $\phi$  is satisfied. The complete robust semantics is defined as follows:

$$\begin{aligned} \rho^{\pi^\mu}(\mathbf{x}, t) &= \mu(x_t) \\ \rho^{\neg\phi}(\mathbf{x}, t) &= -\rho^\phi(\mathbf{x}, t) \\ \rho^{\phi_1 \wedge \phi_2}(\mathbf{x}, t) &= \min(\rho^{\phi_1}(\mathbf{x}, t), \rho^{\phi_2}(\mathbf{x}, t)) \\ \rho^{\phi_1 \mathbf{U}_{[a,b]}\phi_2}(\mathbf{x}, t) &= \max_{t' \in [t+a, t+b]} (\min(\rho^{\phi_2}(\mathbf{x}, t'), \\ & \min_{t'' \in [t, t']} \rho^{\phi_1}(\mathbf{x}, t''))) \end{aligned}$$

The reader is referred to [30] for more details on the computation of the robustness value.

## 3. Problem formulation

### 3.1. Robust STL planning with disturbances

In the literature, the problem of *robust STL motion planning* has been extensively studied. This problem focuses on designing an input sequence from the perspective of the controller that ensures the satisfaction of the STL formula *for all possible* external input sequences. Formally, this problem can be described by the following optimization problem

$$\begin{aligned} & \text{minimize}_{\mathbf{u}} \max_{\mathbf{w}} J(x_0, \mathbf{u}, \mathbf{w}) \\ & \text{subject to } \forall \mathbf{w} \in W^N : \xi_f(x_0, \mathbf{u}, \mathbf{w}) \models \phi, \end{aligned}$$

where  $J : X \times U^N \times W^N \rightarrow \mathbb{R}$  is a generic cost function such as the robust value of the resulting state sequence.

The robust planning problem described above essentially adopts a *zero-sum game* framework, assuming that the external inputs are entirely adversarial. While this assumption is appropriate in scenarios where the external inputs represent disturbances or noise, in many practical applications, where external inputs are not entirely adversarial, but instead originate from another *rational agent* with its own objectives, modeling the interaction as a zero-sum game may lead to *overly conservative* synthesis results.

### 3.2. Stackelberg game formulation for STL planning

In this paper, to capture the rationality of the external uncontrollable agents, we introduce a new formulation for STL planning within the framework of Stackelberg games (or leader–follower games). Specifically, we assume that the overall input at each time step is jointly determined by two agents, referred to as the *leader* and the *follower*. The system dynamics are formalized as follows:

$$x_{t+1} = f(x_t, u_t^L, u_t^F), \quad (1)$$

where  $u_t^L \in U^L \subseteq \mathbb{R}^{m_L}$  and  $u_t^F \in U^F \subseteq \mathbb{R}^{m_F}$  represent the leader’s and the follower’s control inputs at time  $t$ , respectively. Similarly, given the initial state  $x_0 \in X$ , and two sequences of control inputs  $\mathbf{u}^L = u_0^L u_1^L \dots u_{N-1}^L \in (U^L)^N$  and  $\mathbf{u}^F = u_0^F u_1^F \dots u_{N-1}^F \in (U^F)^N$ , the resulting state sequence is  $\xi_f(x_0, \mathbf{u}^L, \mathbf{u}^F) = \mathbf{x} = x_0 x_1 \dots x_N \in X^{N+1}$ , where for each  $i \leq N-1$ , we have  $x_{i+1} = f(x_i, u_i^L, u_i^F)$ .

We assume that both the leader and the follower have their own objectives, represented by STL formulae  $\phi^L$  and  $\phi^F$ , respectively. Note that these two formulae are not necessarily in opposition to each other. Moreover, we assume the leader knows the follower’s STL specification  $\phi^F$ . This reflects a standard Stackelberg game setting where the leader understands the follower’s rationality model. The decision-making process of the two agents proceeds as follows:

- First, the leader selects its control input sequence  $\mathbf{u}^L$  based on its objective  $\phi^L$  and its knowledge of the follower's objective  $\phi^F$ . Once the input sequence of the leader is synthesized, the leader commits to executing this sequence. This decision is then published as public information, making it available to the follower;
- Then, the follower determines its own input sequence  $\mathbf{u}^F$ , optimizing its objective  $\phi^F$  based on the leader's committed actions.

This information structure enables the leader to *anticipate* the follower's reaction and aims to optimize its own objective while considering the follower's likely response. The follower's decisions are then made with full knowledge of the leader's strategy, reflecting the hierarchical nature of the Stackelberg game framework.

Here, we assume that the behavior of the follower is *rational* in the sense that it will not deviate from its best response. This leads to the following definitions.

**Definition 1 (Successful Responses).** Suppose the leader's input sequence is committed to be  $\mathbf{u}^L$ . We say follower's input sequence  $\mathbf{u}^F$  is a *successful response* (w.r.t.  $\mathbf{u}^L$  and  $\phi^F$ ) if

$$\xi_f(x_0, \mathbf{u}^L, \mathbf{u}^F) \vDash \phi^F.$$

We denote by  $\mathbf{SR}(\mathbf{u}^L, \phi^F)$  the set of successful responses.

Note that there may be situations where there are no successful responses available, i.e.,  $\mathbf{SR}(\mathbf{u}^L, \phi^F) = \emptyset$ . In such cases, we assume the follower switches to a predefined fallback policy. For the sake of simplicity, we assume that this policy is *non-interfering* in the sense that the input will be consistently zero. However, our approach can be extended directly to an inputs set or even a given control policy.

**Definition 2 (Best Responses).** Suppose that the leader's input sequence is committed to be  $\mathbf{u}^L$ . We say follower's input sequence  $\mathbf{u}^F$  is a *best response* (w.r.t.  $\mathbf{u}^L$  and  $\phi^F$ ) if (i) it is a successful response when  $\mathbf{SR}(\mathbf{u}^L, \phi^F) \neq \emptyset$ ; or (ii) it is the zero sequence  $\mathbf{0} = (0^{m_F})^N$  when  $\mathbf{SR}(\mathbf{u}^L, \phi^F) = \emptyset$ . We denote by  $\mathbf{BR}(\mathbf{u}^L, \phi^F)$  the set of best responses.

Similar to the robust STL synthesis problem, we also consider a generic cost function  $J_S : X^N \times (U^L)^N \times (U^F)^N \rightarrow \mathbb{R}$ , whose value is jointly determined by both the leader's and the follower's control inputs. The control objective is to design an input sequence  $\mathbf{u}^L$  from the leader's perspective such that the system trajectory satisfies  $\phi^L$  under *any* best response of the follower as defined in Definition 2, while minimizing the worst-case cost. This formulation leads to the following Stackelberg STL Planning Problem (SPP-STL):

**Problem 1 (SPP-STL).** Given the system in Eq. (1), the cost function  $J_S$ , the initial state  $x_0$ , the leader's specification  $\phi^L$ , and the follower's specification  $\phi^F$ , synthesize an optimal input sequence for the leader,  $\mathbf{u}^L$ , to minimize the cost function subject to the constraint that the follower takes a best response. Formally, we have

$$\begin{aligned} & \underset{\mathbf{u}^L}{\text{minimize}} && \max_{\mathbf{u}^F \in \mathbf{BR}(\mathbf{u}^L, \phi^F)} J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F) \\ & \text{subject to} && \mathbf{x} \vDash \phi^L \quad \text{for all } \mathbf{u}^F \in \mathbf{BR}(\mathbf{u}^L, \phi^F), \\ & && x_{t+1} = f(x_t, u_t^L, u_t^F), \quad t = 0, 1, \dots, N-1. \end{aligned}$$

Compared to the robust STL planning problem, the main challenge in SPP-STL is that the set of best responses cannot be determined by a single STL formula alone. Instead, the set of best responses must be constructed simultaneously while synthesizing  $\mathbf{u}^L$ . This results in a two-stage optimization problem where the leader's and follower's decisions are inherently coupled.

In our framework, the solutions to the SPP-STL problem can be categorized into two cases, depending on whether the follower's specification is satisfiable under the leader's input sequence. We refer to these as the *cooperative* and *antagonistic* solutions:

- **Cooperative solution:** The leader synthesizes an input sequence  $\mathbf{u}^L$  that enables the follower to achieve its own objective, i.e.,  $\mathbf{SR}(\mathbf{u}^L, \phi^F) \neq \emptyset$ ;
- **Antagonistic solution:** The leader synthesizes an input sequence  $\mathbf{u}^L$  that forces the follower to be non-interfering, i.e.,  $\mathbf{SR}(\mathbf{u}^L, \phi^F) = \emptyset$ .

#### 4. Synthesis of solutions: Two-stage approach

In this section, we present solutions to the SPP-STL for both cooperative and antagonistic cases. Our approach is primarily designed for the cooperative case, which is more challenging due to the need to align the objectives of the leader and the follower. We then demonstrate that the solution approach for the cooperative case can be easily adapted to handle the antagonistic case, where the leader aims to enforce the unsatisfiability of the follower's tasks.

#### 4.1. Synthesis of cooperative solution

We first seek to find a cooperative solution, i.e., to solve the following optimization problem

$$\underset{\mathbf{u}^L}{\text{minimize}} \quad \max_{\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)} J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F) \tag{2a}$$

$$\text{subject to} \quad \mathbf{SR}(\mathbf{u}^L, \phi^F) \neq \emptyset, \tag{2b}$$

$$\mathbf{x} \vDash \phi^L \quad \text{for all} \quad \mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F), \tag{2c}$$

$$x_{t+1} = f(x_t, u_t^L, u_t^F), \quad t = 0, 1, \dots, N - 1.$$

To solve the above optimization problem, motivated by the standard optimization-based approach for STL planning [8,9], we first encode the satisfaction status of the two STL formulae using binary variables  $z^L, z^F \in \{0, 1\}$  such that (i)  $z^L = 1 \Leftrightarrow \mathbf{x} \vDash \phi^L$ , and (ii)  $z^F = 1 \Leftrightarrow \mathbf{x} \vDash \phi^F$ . Specifically, the values of  $z^L$  and  $z^F$  are determined recursively as follows. For each predicate  $\mu$  and each time instant  $t = 0, 1, \dots, N$ , one needs to introduce a binary variable  $z_t^\mu \in \{0, 1\}$  such that

$$\mu(x_t) \leq M z_t^\mu - \epsilon, \quad \text{and} \quad -\mu(x_t) \leq M(1 - z_t^\mu) - \epsilon,$$

where  $M$  and  $\epsilon$  are sufficiently large and sufficiently small positive constants, respectively. These two constraints enforce that  $z_t^\mu = 1$  if and only if  $\mu(x_t) > 0$ . Then, for each sub-formula of  $\phi^L$  or  $\phi^F$ , one needs to introduce additional binary variables to encode its satisfaction, along with new constraints defined by the Boolean and temporal operations. For the sake of simplicity, we omit the intermediate variables and constraints in this discussion; readers are referred to [8] for the detailed encoding process. We denote  $z^L(x_0, \mathbf{u}^L, \mathbf{u}^F)$  and  $z^F(x_0, \mathbf{u}^L, \mathbf{u}^F)$  as the value of binary variables  $z^L$  and  $z^F$  upon state sequence  $\xi_f(x_0, \mathbf{u}^L, \mathbf{u}^F)$ .

To handle the *inner* maximization of  $J_S$ , we introduce an auxiliary variable  $k \in \mathbb{R}$  and rewrite Eq. (2) as follows

$$\underset{k, \mathbf{u}^L, \mathbf{u}^F}{\text{minimize}} \quad k \tag{3a}$$

$$\text{subject to} \quad z^F(x_0, \mathbf{u}^L, \mathbf{u}^F) = 1, \tag{3b}$$

$$J_S(\mathbf{x}', \mathbf{u}^L, \mathbf{u}'^F) \leq k, \quad \forall \mathbf{u}'^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F), \tag{3c}$$

$$z^L(x_0, \mathbf{u}^L, \mathbf{u}'^F) = 1, \quad \forall \mathbf{u}'^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F), \tag{3d}$$

$$x'_0 = x_0,$$

$$x_{t+1} = f(x_t, u_t^L, u_t^F), \quad t = 0, 1, \dots, N - 1,$$

$$x'_{t+1} = f(x'_t, u_t^L, u_t^F), \quad t = 0, 1, \dots, N - 1.$$

Intuitively, constraint (3b) encodes the requirement in constraint (2b) that  $\mathbf{SR}(\mathbf{u}^L, \phi^F) \neq \emptyset$ . Constraints (3c) require  $k$  to be the upper bound of the inner maximization problem, while constraints (3d) encode constraints (2c). By introducing these constraints, we reduce the bi-level optimization problem in Eq. (2) to single-level. Formally, we have the following result.

**Proposition 1.** *The optimization problems Eq. (2) and Eq. (3) are equivalent.*

**Proof.** The equivalence is established by transforming the objective function and mapping the constraints between Eq. (2) and Eq. (3).

Consider the objective function of the bilevel problem Eq. (2):

$$\min_{\mathbf{u}^L} \left( \max_{\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)} J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F) \right).$$

Let  $V(\mathbf{u}^L) = \max_{\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)} J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F)$  denote the worst-case cost for a given leader strategy  $\mathbf{u}^L$ . By introducing a scalar auxiliary variable  $k$  (epigraph variable), the minimization of  $V(\mathbf{u}^L)$  can be equivalently written as:

$$\min_{\mathbf{u}^L, k} \quad \text{subject to} \quad V(\mathbf{u}^L) \leq k$$

Note that the constraint  $V(\mathbf{u}^L) \leq k$  holds if and only if the cost function is bounded by  $k$  for all successful responses in the follower's set, i.e.,

$$J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}'^F) \leq k, \quad \forall \mathbf{u}'^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)$$

This corresponds to constraint (3c).

Constraint (2c) requires that  $\mathbf{x} \vDash \phi^L$  for all  $\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)$ . In (3), (3d) enforces  $z^L(x_0, \mathbf{u}^L, \mathbf{u}'^F) = 1$  for all  $\mathbf{u}'^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)$ . Since  $z^L = 1$  is the encoding for  $\mathbf{x} \vDash \phi^L$ , constraints (2c) and (3d) are identical. Constraint (2b) requires  $\mathbf{SR}(\mathbf{u}^L, \phi^F) \neq \emptyset$ . In (3), this is enforced by including a decision variable  $\mathbf{u}^F$  subject to (3b):  $z^F(x_0, \mathbf{u}^L, \mathbf{u}^F) = 1$ . If there exists  $\mathbf{u}^F$  satisfying (3b), then  $\mathbf{SR}(\mathbf{u}^L, \phi^F)$  is non-empty. Conversely, if  $\mathbf{SR}(\mathbf{u}^L, \phi^F) \neq \emptyset$ , there exists a  $\mathbf{u}^F$  for (3b). Thus, the feasible space regarding the existence of a follower strategy is preserved. The proof is now completed. ■

Now we can solve Eq. (2) by instead solving the optimization problem in Eq. (3), where constraints in (3b) and (3d) are binary constraints and can be handled efficiently. However, set  $\mathbf{SR}(\mathbf{u}^L, \phi^F)$  is not necessarily the same for different  $\mathbf{u}^L$ , which makes it difficult to directly verify constraints (3c) and (3d). To tackle this challenge, we take a conservative yet efficient approach by gradually extending the region of  $\mathbf{SR}(\mathbf{u}^L, \phi^F)$ . Specifically, for constraint (3d), we can directly require that  $z^L(x_0, \mathbf{u}^L, \mathbf{u}^F) \geq z^F(x_0, \mathbf{u}^L, \mathbf{u}^F)$  for all  $\mathbf{u}^F \in (U^F)^N$  since  $\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)$  can always be captured by  $z^F(x_0, \mathbf{u}^L, \mathbf{u}^F) = 1$  according to the definitions of  $z^F$  and  $\mathbf{SR}(\mathbf{u}^L, \phi^F)$ . However, in general, constraints (3c) are nonlinear. Therefore, we further introduce a new variable  $\rho^K \in \mathbb{R}$  such that

$$\rho^K \geq 0 \Leftrightarrow [J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F) \leq k] \vee [\mathbf{u}^F \notin \mathbf{SR}(\mathbf{u}^L, \phi^F)].$$

This requirement can be captured by the following set of constraints:

$$\begin{aligned} \rho^K &\geq k - J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F), \rho^K \geq -\rho^{\phi^F}(\mathbf{x}, 0), \\ k - J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F) - bM &\leq \rho^K \leq k - J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F) + bM, \\ -\rho^{\phi^F}(\mathbf{x}, 0) - (1-b)M &\leq \rho^K \leq -\rho^{\phi^F}(\mathbf{x}, 0) + (1-b)M, \end{aligned}$$

where  $b \in \{0, 1\}$  is a binary variable,  $M$  is a sufficiently large positive constant,  $\rho^{\phi^F}(\mathbf{x}, 0)$  is the robustness value of  $\phi^F$  along sequence  $\mathbf{x}$  at time instant 0 [30]. Particularly, for  $\mathbf{x} = \xi_f(x_0, \mathbf{u}^L, \mathbf{u}^F)$  we have  $\rho^{\phi^F}(\mathbf{x}, 0) > 0$  if and only if  $\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)$ . Here, the first two constraints enforce that  $\rho^K \geq \max[k - J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F), -\rho^{\phi^F}(\mathbf{x}, 0)]$ , and the last two constraints enforce that  $\rho^K$  always equals to either  $k - J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F)$  or  $-\rho^{\phi^F}(\mathbf{x}, 0)$ . Therefore, these constraints together enforce that

$$\rho^K = \max[k - J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{u}^F), -\rho^{\phi^F}(\mathbf{x}, 0)]. \tag{4}$$

Analogously, we denote  $\rho^K(k, x_0, \mathbf{u}^L, \mathbf{u}^F)$  as the value of  $\rho^K$  upon  $k$  and  $\mathbf{x} = \xi_f(x_0, \mathbf{u}^L, \mathbf{u}^F)$ .

We finally rewrite the optimization problem as follows

$$\underset{k, \mathbf{u}^L, \mathbf{u}^F}{\text{minimize}} \quad k \tag{5a}$$

$$\text{subject to} \quad z^F(x_0, \mathbf{u}^L, \mathbf{u}^F) = 1, \tag{5b}$$

$$\rho^K(k, x_0, \mathbf{u}^L, \mathbf{u}^F) \geq 0, \forall \mathbf{u}^F \in (U^F)^N, \tag{5c}$$

$$z^L(x_0, \mathbf{u}^L, \mathbf{u}^F) \geq z^F(x_0, \mathbf{u}^L, \mathbf{u}^F), \forall \mathbf{u}^F \in (U^F)^N, \tag{5d}$$

$$x'_0 = x_0,$$

$$x_{t+1} = f(x_t, u_t^L, u_t^F), \quad t = 0, 1, \dots, N-1,$$

$$x'_{t+1} = f(x'_t, u_t^L, u_t^F), \quad t = 0, 1, \dots, N-1.$$

**Proposition 2.** *The optimization problems Eq. (3) and Eq. (5) are equivalent.*

**Proof.** By introducing  $\rho^K$ , we can conclude that constraints (5c) are equivalent to (3c). This is because (3c) is essentially a conditional statement: if  $\mathbf{u}^F \in \mathbf{SR}(\mathbf{u}^L, \phi^F)$ , i.e.,  $\rho^{\phi^F}(\mathbf{x}, 0) > 0$ , then  $J_S(x'_t, \mathbf{u}^L, \mathbf{u}^F) \leq k$ . Logically,  $P \implies Q$  is equivalent to  $\neg P \vee Q$ . Thus, we require either  $-\rho^{\phi^F}(\mathbf{x}, 0) \geq 0$  or  $k - J_S(x'_t, \mathbf{u}^L, \mathbf{u}^F) \geq 0$ . By taking the maximum of these two terms,  $\rho^K \geq 0$  ensures that at least one condition holds. Moreover, according to the encoding of the formula  $\phi^F \Rightarrow \phi^L$ , we have that constraints (5d) are equivalent to (3d). Therefore, Eq. (5) has the same solution as Eq. (3). ■

Eq. (5) further reduces the need to reconstruct the successful response set  $\mathbf{SR}(\mathbf{u}^L, \phi^F)$  compared to Eq. (3). However, constraints (5c) and (5d) are defined on the whole action space  $(U^F)^N$  of the follower, which can be prohibitively large or even infinite and computationally intractable to optimize globally. In practice, rather than seeking a globally optimal solution, we typically prioritize finding an effective one for such a complex optimization problem. To achieve this efficiently, we adapt a counterexample-guided inductive synthesis scheme, as shown in Procedure 1. Specifically, in line 1, we randomly initialize a set of candidate inputs of the follower. Then we compute the optimal solution  $\mathbf{u}^L$  and  $k$  w.r.t.  $\mathbf{U}^{cand}$  in line 3. Next in line 4, we find the counterexample that falsifies the global constraints  $z^L(x_0, \mathbf{u}^L, \mathbf{u}^F) \geq z^F(x_0, \mathbf{u}^L, \mathbf{u}^F)$  or  $\rho^K(k, x_0, \mathbf{u}^L, \mathbf{u}^F) \geq 0$  when the above  $\mathbf{u}^L$  is used. For simplicity, we directly write  $\rho^K(k, x_0, \mathbf{u}^L, \mathbf{u}^F)$  and  $\rho^{\phi^F \Rightarrow \phi^L}(x_0, \mathbf{u}^L, \mathbf{u}^F) = \max[-\rho^{\phi^F}(x_0, \mathbf{u}^L, \mathbf{u}^F), \rho^{\phi^L}(x_0, \mathbf{u}^L, \mathbf{u}^F)]$  as  $\rho^K$  and  $\rho^{\phi^F \Rightarrow \phi^L}$ , respectively. If it does not exist, we can return  $\mathbf{u}^L$  as the result in line 6. Otherwise, we add the counterexample to  $\mathbf{U}^{cand}$  in line 8 and consider them in the next iteration.

**Theorem 1.** *Procedure 1 is sound, in the sense that if it returns  $\mathbf{u}^L \in (U^L)^N$ , then constraints in (5) are all fulfilled.*

**Proof.** We first note that for any  $\mathbf{u}^F \in \mathbf{U}^{cand}$ , we have  $\mathbf{u}^F \in (U^F)^N$ . Therefore, if  $\mathbf{u}^L$  is returned, then we can first conclude that there exists a  $\mathbf{u}^F \in (U^F)^N$ , such that  $z^F = 1$ , which immediately leads that constraint (5b) is satisfied. To prove constraints (5c) and (5d), we suppose that there still exists  $\mathbf{u}^F \in (U^F)^N$  such that  $(\rho^K < 0) \vee (z^L < z^F)$  holds true after  $\mathbf{u}^L$  is returned. Then we can conclude that  $(\rho^K < 0) \vee (\rho^{\phi^F \Rightarrow \phi^L} < 0)$ . And thus  $\min[\rho^K, \rho^{\phi^F \Rightarrow \phi^L}] < 0$ , which is a contradiction to the condition in line 5. Therefore, both constraints (5c) and (5d) are satisfied. ■

---

**Procedure 1:** Cooperative Synthesis

---

**Input:** system dynamic function  $f$ , cost function  $J_S$ , initial state  $x_0$  and specifications  $\phi^L$  and  $\phi^F$

**Output:**  $\mathbf{u}^L$

```

1 Let  $\mathbf{U}^{cand} = \{\text{a set of randomly generated } \mathbf{u}^F\}$ 
2 while True do
3      $k, \mathbf{u}^L \leftarrow$  the solution of Equation (5)
        by changing  $(U^F)^N$  for  $\mathbf{U}^{cand}$ 
4      $\mathbf{u}^F \leftarrow \underset{\mathbf{u}^F}{\operatorname{argmin}} [\rho^K, \rho^{\phi^F \Rightarrow \phi^L}]$ 
5     if  $\min[\rho^K, \rho^{\phi^F \Rightarrow \phi^L}] > 0$  then
6         return  $\mathbf{u}^L$ 
7     else
8          $\mathbf{U}^{cand} \leftarrow \mathbf{U}^{cand} \cup \{\mathbf{u}^F\}$ 

```

---

Note that the random initialization of the set  $\mathbf{U}^{cand}$  only serves to generate an initial candidate response space and does not affect the correctness of the proposed CEGIS-based synthesis procedure, since any missing or unrepresentative responses will be detected and refined through counterexample generation. However, the optimality of the strategy returned by Procedure 1 is still dependent on the initial  $\mathbf{U}^{cand}$ , as the CEGIS loop terminates once a sound solution is found. For instance, if  $\mathbf{U}^{cand}$  contains only a single response  $\mathbf{u}^F$  for which a feasible  $\mathbf{u}^L$  satisfies (5c) and (5d), the algorithm may converge in one iteration without exploring other regions. Thus, global optimality is not strictly guaranteed. In practice, a more informed initialization heuristic, such as exploiting the follower STL specification and system dynamics, may accelerate convergence and improve solution quality, which we view as a promising direction for future work.

#### 4.2. Synthesis of antagonistic solution

In the antagonistic case, the leader must enforce the unsatisfiability of the follower’s task. By assuming that the follower will become non-interfering when its own task is not satisfiable, the leader can set  $\mathbf{u}^F$  to  $\mathbf{0}$  while maximizing its objective function. This leads to the following optimization problem.

$$\underset{\mathbf{u}^L}{\text{minimize}} \quad J_S(\mathbf{x}, \mathbf{u}^L, \mathbf{0}) \tag{6a}$$

$$\text{subject to} \quad \mathbf{SR}(\mathbf{u}^L, \phi^F) = \emptyset, \tag{6b}$$

$$\mathbf{x} \models \phi^L, \tag{6c}$$

$$x_{t+1} = f(x_t, u_t^L, 0^{m_F}), \quad t = 0, 1, \dots, N - 1.$$

To solve this, note that constraints (6b) and (6c) can be directly encoded by  $z^F(x_0, \mathbf{u}^L, \mathbf{u}^F) = 0, \forall \mathbf{u}^F \in (U^F)^N$  and  $z^L(x_0, \mathbf{u}^L, \mathbf{0}) = 1$ . Therefore, we can still use the basic outline of Procedure 1 to compute the leader’s control input sequence  $\mathbf{u}^L$  for the antagonistic case by incorporating the following modifications:

- In line 3, Eq. (5) is changed to Eq. (6) with encoded constraints (6b) and (6c); and
- In line 4 and line 5, the condition for determining counterexamples is changed to  $\max_{\mathbf{u}^F} \rho^{\phi^F} \leq 0$ .

The modified procedure still inherits the counterexample-guided scheme. However, the key distinction lies in the fact that the initialized set  $\mathbf{U}^{cand}$  does not affect the result returned by the procedure since the objective function is always optimized under  $\mathbf{u}^F = \mathbf{0}$ . As a result, given that we can get the optimal solution for the problem in line 3, the modified procedure is not only sound but also optimal for the case of the antagonistic solution.

**Theorem 2.** *In the antagonistic case, if we can find the optimal solution in line 3, then we have that Procedure 1 is both sound and optimal.*

**Proof.** The proof of soundness proceeds similarly to that of Theorem 1. As for completeness, we want to prove that the control input  $\mathbf{u}^L$  returned by Procedure 1 is indeed the optimal solution in the antagonistic case. To prove this, we assume that after  $\mathbf{u}^L$  is returned, there still exists a control input  $\mathbf{u}'^L$  satisfying constraints (6b) and (6c) such that  $J_S(x_0, \mathbf{u}'^L, \mathbf{0}) < J_S(x_0, \mathbf{u}^L, \mathbf{0})$ . From this assumption, we know that for any  $\mathbf{u}^F$  in the current candidate set  $\mathbf{U}^{cand}$ , we also have  $z^F = 0$  under the control input  $\mathbf{u}'^L$ . Therefore, we have  $\mathbf{u}^L = \mathbf{u}'^L$  in line 3, which is a contradiction to the fact that  $\mathbf{u}^L$  is returned. ■

**Remark 1.** We note that although we assume a predefined zero-input for the follower when its task is infeasible for the sake of simplicity, our approach is readily generalized to more complex backup behaviors in two ways:

- If the follower’s behavior is defined by a predefined set of possible inputs when its task is infeasible, the synthesis problem can be reformulated as a min–max optimization problem. In this case, one needs to adjust the constraint (6c) to require the leader to ensure the satisfaction of  $\phi^L$  against all possible inputs, as we presented in 4.1.
- If the follower’s behavior is defined by a specific control policy when its task is infeasible, one can directly substitute its analytical form into the system dynamics and follow the same procedure proposed in this subsection, since the problem still maintains its single-layer optimization structure.

When no such information about the follower’s fallback behavior is available, our framework reverts to a zero-sum formulation, in which the leader accounts for the worst-case scenario.

**Remark 2.** Although the system model is stated in a general nonlinear form, the proposed synthesis procedure only focuses on systems with linear dynamics in our current experiments. For nonlinear systems, although our approach remains theoretically applicable, the algorithm can be computationally intractable since the encoding would result in a nonlinear program. For computational consideration, a direct extension of our framework is replacing the mixed-integer STL encoding by a smooth robustness surrogate and using gradient-based solvers, which has also been leveraged in standard STL motion planning problems, see, e.g., [31–33]. Extending the Stackelberg setting to this gradient-based framework will preserve the same game structure and the counterexample-guided synthesis scheme in this paper. We consider this a valuable future direction.

### 5. Case studies

In this section, we present simulation results for two case studies. Our approach is implemented in Python 3, and we use Gurobi to solve the optimization problems. All simulations were conducted on a laptop equipped with an Apple M2 CPU and 8 GB of RAM. All source codes are available at <https://github.com/stack-stl/Stackelberg-STL>.

#### 5.1. Double integrator with joint inputs

**System Model:** In this case study, we consider a two-dimensional double integrator system operating in a planar environment, where the acceleration in each dimension is jointly determined by both the leader and the follower. The system dynamics, with a sampling period of 0.5s, is given by

$$x_{k+1} = Ax_k + B(u_k^L + u_k^F),$$

where  $A = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ ,  $B = \begin{bmatrix} 0.125 & 0 \\ 0.5 & 0 \\ 0 & 0.125 \\ 0 & 0.5 \end{bmatrix}$ , and state  $x_k = [x, v_x, y, v_y]^T$  denotes  $x$ -position,  $x$ -velocity,  $y$ -position and

$y$ -velocity, the control input of the leader  $u_k^L = [u_x^L, u_y^L]^T$  and the control input of the follower  $u_k^F = [u_x^F, u_y^F]^T$  affect  $x$ -acceleration and  $y$ -acceleration together by their sum. The physical constraints are  $x \in X = [0, 10] \times [-3, 3] \times [0, 10] \times [-3, 3]$ ,  $u^L \in U^L = [-3, 3]^2$  and  $u^F \in U^F = [-0.01, 0.01]^2$ .

**Planning Objectives:** We assume that the initial state of the robot is  $[2, 0, 6, 0]$ , which is shown as the red point in Fig. 1. The control objective of the leader is to visit region A1 at least once within the time interval 1 to 10 (from 0.5s to 5s), always stay at region A2 during the time interval 14 to 16 (from 7s to 8s), and finally reach region A3 at least once within the time interval 20 to 25 (from 10 s to 12.5s). This objective can be specified by the STL formula

$$\phi^L = \mathbf{F}_{[1,10]}A_1 \wedge \mathbf{G}_{[14,16]}A_2 \wedge \mathbf{F}_{[20,25]}A_3,$$

where  $A_1 = (x \in [8, 10]) \wedge (y \in [8, 10])$ ,  $A_2 = (x \in [1, 4]) \wedge (y \in [1, 4])$  and  $A_3 = (x \in [8, 10]) \wedge (y \in [0, 2])$ . Meanwhile, the control objective of the follower is to visit region B1 at least once between instants 4 to 9 (from 2s to 4.5s) and always stay at region B2 within time interval 12 to 13 (from 6s to 6.5s), which is specified by  $\phi^F = \mathbf{F}_{[4,9]}B_1 \wedge \mathbf{G}_{[12,13]}B_2$ , where  $B_1 = (x \in [1, 3]) \wedge (y \in [6.5, 8])$  and  $B_2 = (x \in [2, 5]) \wedge (y \in [2, 5])$ . The cost function we use is  $J_S(x, u^L, u^F) = 1.6 \times 10^{-7} \times \sum_{i=0}^{T/\phi^L} \|u_i^L\|_2^2 - \rho^{\phi^L}(\xi_f(x_0, u^L, u^F), 0)$ . Note that we use the term  $1.6 \times 10^{-7}$  to match the order of magnitude of the cost of  $u^L$  and the robustness value of  $\phi^L$ .

**Simulation Results and Analysis:** Note that, in this case study, the tasks  $\phi^L$  and  $\phi^F$  are not inherently conflicting. However, since the control input of the follower is restricted to the set  $[-0.01, 0.01]^2$ , the follower can only have a limited influence on the evolution of the system. As a result, whether the task  $\phi^F$  is satisfied is primarily determined by the leader’s actions. This highlights the leader’s dominant role in shaping the system’s behavior in this scenario.

The resulting trajectories for the cooperative case and the antagonistic case are illustrated in Fig. 1 and Fig. 2, respectively. The computation time and the associated cost for each case are summarized in Table 1. Note that the trajectories depicted in the figures are projections onto the first and third dimensions (representing the  $x$ -position and  $y$ -position, respectively). However, the set membership and satisfaction of the STL specifications are still evaluated in the complete 4-dimensional state space, which includes both position and velocity components.

In the cooperative case, the leader must first move to region B1 to collaborate with the follower. Then, it needs to adjust the speed to ensure the entire system remains in region B2 during the time interval  $[12, 13]$ . As a result, the cooperative approach leads to (i) higher control cost and (ii) lower robustness of the formula  $\phi^L$  compared to the antagonistic case. In the antagonistic case,

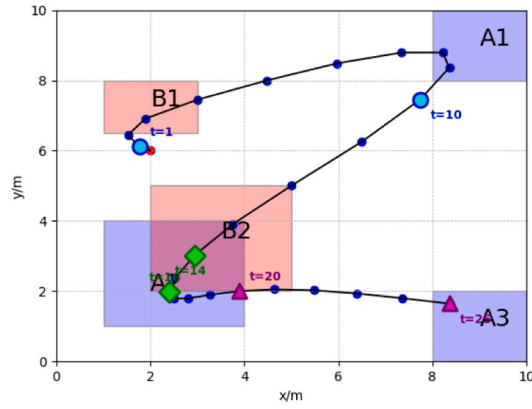


Fig. 1. The single robot trajectory: cooperative case.

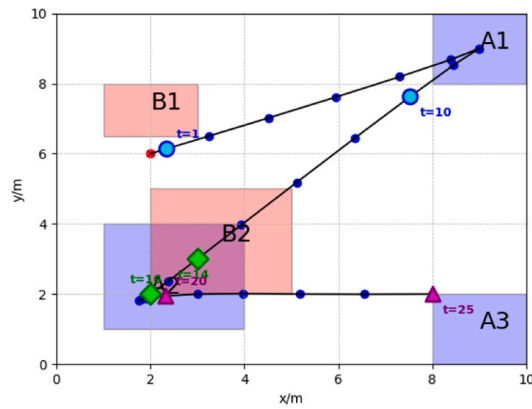


Fig. 2. The single robot trajectory: antagonistic case.

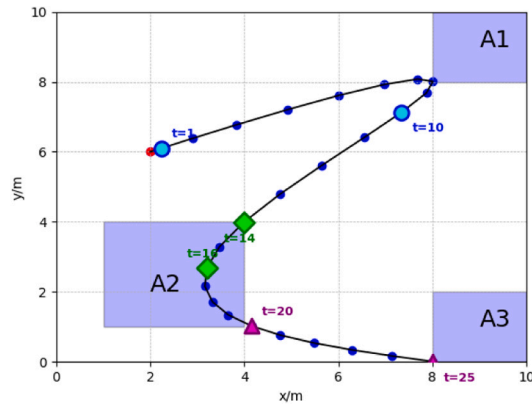


Fig. 3. The single robot trajectory: zero-sum case.

since the follower cannot achieve  $\phi^F$ , it chooses to be non-interfering, and the computation time for this case is shorter than for the cooperative case. Note that the algorithm converges quickly in both cases, and the termination issue did not arise in the considered scenarios.

For comparison, we additionally implemented a standard robust STL planning baseline in which the follower is treated as a purely adversarial input. The resulting trajectory is illustrated in Fig. 3. Under the same system setting and STL specification, although the leader can still satisfy its STL specification, the cost becomes 0.0011 with 1 iteration, which is more conservative than the costs obtained by the proposed Stackelberg formulation in both the cooperative and antagonistic cases. This illustrates that, when the

**Table 1**  
Computational performance of cooperative and antagonistic cases.

Procedure	Computation time	Cost	Iterations
Cooperative	2.7881 s	-0.3613	2
Antagonistic	0.4465 s	-0.9999	1

follower is in fact rational rather than adversarial, explicitly modeling its objective in the Stackelberg framework can substantially reduce the conservatism introduced by the worst-case assumption in zero-sum formulations.

## 5.2. Multi-agent planning problem

**System Model:** In this case study, we consider a team of three identical robots in a common workspace. The dynamics of each robot is modeled by a single integrator on a planar workspace:

$$x_{i+1}^i = Ax_i^i + Bu_i^i$$

where  $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ,  $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ,  $i = 1, 2, 3$ , and state  $x_i^i = [x^i, y^i]^T$  denotes  $x$ -positions and  $y$ -positions for agent  $i$ . The physical constraints are  $x^i \in X^i = [0, 10]^2$ ,  $u^i \in U^i = [-1, 1]^2$  for all  $i = 1, 2, 3$ . All robots start from the same initial state (2, 6).

**Planning Objectives:** Our design objective is to ensure that all three agents can achieve their respective tasks. However, we assume that the designer can only directly control the behavior of Agent 1, while Agents 2 and 3 are rational agents that jointly pursue their own objectives. Consequently, the behavior of Agents 2 and 3 is implicitly influenced by finding a suitable cooperative solution for Agent 1, which aligns their actions with the overall system goals.

The control objective of Agent 1 is given by the following STL formula

$$\phi^L = \mathbf{F}_{[1,10]}A_1 \wedge \mathbf{G}_{[14,16]}A_2 \wedge \mathbf{F}_{[20,25]}A_3,$$

where  $A_1 = (x^1 \in [0, 2]) \wedge (y^1 \in [8, 10])$ ,  $A_2 = (x^1 \in [7, 10]) \wedge (y^1 \in [7, 10])$ , and  $A_3 = (x^1 \in [8, 10]) \wedge (y^1 \in [0, 2])$ . The control objectives of Agents 2 and 3 are as follows. First, during the entire operation horizon  $[0, 25]$ , both Agents 2 and 3 are required to follow the trajectory of Agent 1 by maintaining a distance of less than one unit. This requirement can be formally specified using the following STL formulae:

$$\phi_D^i = \mathbf{G}_{[0,25]}(x^1 - x^i)^2 + (y^1 - y^i)^2 \leq 1, i = 2, 3.$$

In addition, each follower has its own task. Specifically, Agent 2 is required to visit region  $B_1$  at least once within the time interval  $[4, 9]$  and remain within region  $B_2$  during the time interval  $[15, 17]$ . These requirements can be formally specified using the following STL formula:  $\phi^2 = \mathbf{F}_{[4,9]}B_1 \wedge \mathbf{G}_{[15,17]}B_2$ , where  $B_1 = (x^2 \in [1, 3]) \wedge (y^2 \in [6.5, 8])$  and  $B_2 = (x^2 \in [8, 10]) \wedge (y^2 \in [6, 9])$ . For Agent 3, it needs to visit region  $C_1$  at least once within the time interval  $[1, 25]$ , which is specified by STL formula  $\phi^3 = \mathbf{F}_{[1,25]}C_1$ , where  $C_1 = (x^3 \in [3, 6]) \wedge (y^3 \in [3, 6])$ . In summary, the overall STL formula for the followers is given by

$$\phi^F = \phi_D^2 \wedge \phi_D^3 \wedge \phi^2 \wedge \phi^3.$$

In this experiment, we inherit the cost function used in the above experiment  $J_S(\mathbf{x}, \mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3) = 1.6 \times 10^{-7} \times \sum_{i=0}^{T_{\phi^L}} \|u_i^i\|_2^2 - \rho^{\phi^L}(\xi_f(x_0, \mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3), 0)$ . We note that here since we consider a single integrator system, the robustness of  $\phi^L$  generally dominates the value of  $J_S$ .

**Simulation Results and Analysis:** The simulation result of the cooperative solution is shown in Fig. 4, where blue, red and green lines denote the trajectories of Agents 1, 2 and 3, respectively. The computation time for finding the solution is 0.9465s and the value of the optimal cost is  $4.0593e - 6$  with 2 iterations.

Note that, while tasks  $\phi^L$  and  $\phi^F$  do not directly conflict, the mobility of Agents 2 and 3 is constrained by the trajectory of Agent 1, as they need to closely follow the leader's path. Therefore, when synthesizing the trajectory for Agent 1, it cannot pursue its own STL task directly, as doing so would prevent the followers from achieving their reachability tasks while maintaining a close distance. For example, to collaborate with Agent 2, the leader (Agent 1) must first stay close to region  $B_1$  at the start of the execution, and then remain close to region  $B_2$  during the time interval  $[15, 17]$ . While these constraints are not necessary for Agent 1 itself and would decrease the robustness of its task  $\phi^L$ , such sacrifices are essential to ensure the feasibility of the tasks for Agents 2 and 3. In contrast, the zero-sum formulation is unable to capture the cooperative tracking requirements imposed on the followers, and therefore cannot guarantee the satisfaction of the followers' specifications.

Note that our approach for multi-agent systems treats the collective behavior of all agents as an augmented system tuple and solves the resulting optimization problem. Consequently, the computational complexity grows exponentially with the number of agents and the current approach is still limited to small-scale systems.

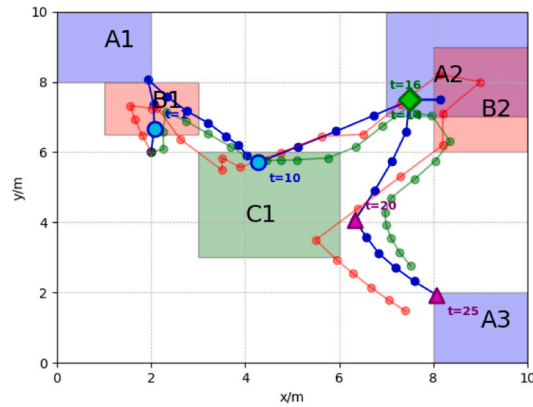


Fig. 4. Trajectories for three robots: cooperative case.

## 6. Conclusion

In this paper, we present a Stackelberg game-theoretical framework for signal temporal logic motion planning in the presence of uncontrollable agents. Compared to robust control approaches, our framework better captures the rationality of uncontrollable agents, providing a more flexible solution for the system controller. This work represents the first step toward STL motion planning within the Stackelberg game framework. However, the current results have certain limitations that we aim to address in future research. First, the current formulation assumes an open-loop control fashion. This setting enables tractable synthesis under STL constraints but limits adaptability to real-time information, particularly in the presence of disturbances. As future work, we plan to extend our framework to incorporate reactive feedback policies that allow agents to adjust their actions based on real-time feedback. Also, our current framework focuses on the Boolean satisfaction of the follower and assumes that the follower does not interfere when the task cannot be satisfied. We intend to expand it to a quantitative setting, where the follower optimizes its STL robustness degree rather than a Boolean satisfaction objective. This quantitative setting may unify the two distinct solution types that arise in the Boolean setting. Finally, our current solution for the multi-agent setting is solved in a centralized fashion. In the future, we aim to develop a decentralized and more scalable setting within this framework, such as modular approaches [34,35] or sampling-based methods [36–38].

## CRedit authorship contribution statement

**Bohan Cui:** Writing – review & editing, Methodology. **Xinyi Yu:** Writing – original draft. **Alessandro Giua:** Writing – review & editing, Supervision. **Xiang Yin:** Writing – review & editing, Supervision.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Xiang Yin reports financial support was provided by National Natural Science Foundation of China. Xiang Yin reports a relationship with National Natural Science Foundation of China that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

No data was used for the research described in the article.

## References

- [1] H. Kress-Gazit, M. Lahijanian, V. Raman, Synthesis for robots: Guarantees and feedback for robot behavior, *Annu. Rev. Control. Robot. Auton. Syst.* 1 (1) (2018) 211–236.
- [2] C. Belta, S. Sadraddini, Formal methods for control synthesis: An optimization perspective, *Annu. Rev. Control. Robot. Auton. Syst.* 2 (1) (2019) 115–140.
- [3] X. Yin, B. Gao, X. Yu, Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges, *Annu. Rev. Control.* 57 (2024) 100940.
- [4] O. Maler, D. Nickovic, Monitoring temporal properties of continuous signals, in: *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*, 2004, pp. 152–166.
- [5] M. Ma, E. Bartocci, E. Lifland, J. Stankovic, L. Feng, SaSTL: Spatial aggregation signal temporal logic for runtime monitoring in smart cities, in: *ACM/IEEE ICCPS*, 2020, pp. 51–62.

- [6] M. Ma, J. Gao, L. Feng, J. Stankovic, STLnet: Signal temporal logic enforced multivariate recurrent neural networks, *Adv. Neural Inf. Process. Syst.* 33 (2020) 14604–14614.
- [7] G. Silano, T. Baca, R. Penicka, D. Liuzza, M. Saska, Power line inspection tasks with multi-aerial robot systems via signal temporal logic specifications, *IEEE Robot. Autom. Lett.* 6 (2) (2021) 4169–4176.
- [8] V. Raman, A. Donze, M. Maasoumy, R.M. Murray, A. Sangiovanni-Vincentelli, S.A. Seshia, Model predictive control with signal temporal logic specifications, in: *CDC, IEEE*, 2014, pp. 81–87.
- [9] V. Kurtz, H. Lin, Mixed-integer programming for signal temporal logic with fewer binary variables, *IEEE Control. Syst. Lett.* 6 (2022) 2635–2640.
- [10] L. Lindemann, D.V. Dimarogonas, Control barrier functions for signal temporal logic tasks, *IEEE Control. Syst. Lett.* 3 (1) (2018) 96–101.
- [11] W. Xiao, C.A. Belta, C.G. Cassandras, High order control lyapunov-barrier functions for temporal logic specifications, in: *American Control Conference*, 2021, pp. 4886–4891.
- [12] V. Raman, A. Donze, D. Sadigh, R.M. Murray, S.A. Seshia, Reactive synthesis from signal temporal logic specifications, in: *HSCC*, 2015, pp. 239–248.
- [13] L. Lindemann, G.J. Pappas, D.V. Dimarogonas, Reactive and risk-aware control for signal temporal logic, *IEEE Trans. Autom. Control.* 67 (10) (2021) 5262–5277.
- [14] D. Gundana, H. Kress-Gazit, Event-based signal temporal logic synthesis for single and multi-robot tasks, *IEEE Robot. Autom. Lett.* 6 (2) (2021) 3687–3694.
- [15] R.B. Ilyes, Q.H. Ho, M. Lahijanian, Stochastic robustness interval for motion planning with signal temporal logic, in: *ICRA*, 2023, pp. 5716–5722.
- [16] G. Scher, S. Sadraddini, A. Yadin, H. Kress-Gazit, Ensuring reliable robot task performance through probabilistic rare-event verification and synthesis, 2023, *arXiv preprint arXiv:2304.14886*.
- [17] S.S. Farahani, R. Majumdar, V.S. Prabhu, S. Soudjani, Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances, *IEEE Trans. Autom. Control.* 64 (8) (2018) 3324–3331.
- [18] D. Sadigh, A. Kapoor, Safe control under uncertainty with probabilistic signal temporal logic, in: *Proceedings of Robotics: Science and Systems XII*, 2016.
- [19] W. Ren, R. Jungers, Reachability-based control synthesis under signal temporal logic specifications, in: *ACC*, 2022, pp. 2078–2083.
- [20] X. Yu, C. Wang, D. Yuan, S. Li, X. Yin, Model predictive control for signal temporal logic specifications with time interval decomposition, in: *CDC, IEEE*, 2023, pp. 7849–7855.
- [21] E.E. Vlahakis, L. Lindemann, P. Sotasakis, D.V. Dimarogonas, Probabilistic tube-based control synthesis of stochastic multi-agent systems under signal temporal logic, 2024, *arXiv preprint arXiv:2405.02827*.
- [22] T. Başar, G.J. Olsder, *Dynamic Noncooperative Game Theory*, SIAM, 1998.
- [23] M. Zhu, S. Martinez, Stackelberg-game analysis of correlated attacks in cyber-physical systems, in: *ACC*, 2011, pp. 4063–4068.
- [24] A. Ulusoy, C. Belta, Receding horizon temporal logic control in dynamic environments, *Int. J. Robot. Res.* 33 (12) (2014) 1593–1607.
- [25] B. Hoxha, G. Fainekos, Planning in dynamic environments through temporal logic monitoring, in: *Workshops At the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [26] Y. Li, E.M. Shahrivar, J. Liu, Safe linear temporal logic motion planning in dynamic environments, in: *IROS*, 2021, pp. 9818–9825.
- [27] J. Niu, A. Clark, Optimal minimum violation control synthesis of cyber-physical systems under attacks, *IEEE Trans. Autom. Control.* 66 (3) (2020) 995–1008.
- [28] L. Li, H. Ma, A.N. Kulkarni, J. Fu, Dynamic hypergames for synthesis of deceptive strategies with temporal logic objectives, *IEEE Trans. Autom. Sci. Eng.* 20 (1) (2022) 334–345.
- [29] B. Cui, A. Giua, X. Yin, Towards supervisory control theory in tactical environments: A stackelberg game approach, in: *CDC, IEEE*, 2023, pp. 7937–7943.
- [30] A. Donzé, O. Maler, Robust satisfaction of temporal logic over real-valued signals, in: *FORMATS*, 2010, pp. 92–106.
- [31] Y. Gilpin, V. Kurtz, H. Lin, A smooth robustness measure of signal temporal logic for symbolic control, *IEEE Control. Syst. Lett.* 5 (1) (2020) 241–246.
- [32] N. Hashemi, S. Williams, B. Hoxha, D. Prokhorov, G. Fainekos, J. Deshmukh, LB4TL: A smooth semantics for temporal logic to train neural feedback controllers, *IFAC-PapersOnLine* 58 (11) (2024) 183–188.
- [33] S. Han, J. Verhagen, J. Tumova, Exact smooth reformulations for trajectory optimization under signal temporal logic specifications, 2025, *arXiv preprint arXiv:2511.07375*.
- [34] Z. Zhang, S. Haesaert, Modularized control synthesis for complex signal temporal logic specifications, in: *CDC, IEEE*, 2023, pp. 7856–7861.
- [35] M.H.W. Engelaar, Z. Zhang, E.E. Vlahakis, D.V. Dimarogonas, M. Lazar, S. Haesaert, Risk-aware real-time task allocation for stochastic multi-agent systems under STL specifications, in: *CDC, IEEE*, 2024, pp. 8213–8218.
- [36] Y. Meng, C. Fan, Signal temporal logic neural predictive control, *IEEE Robot. Autom. Lett.* 8 (11) (2023) 7719–7726.
- [37] C.-I. Vasile, V. Raman, S. Karaman, Sampling-based synthesis of maximally-satisfying controllers for temporal logic specifications, in: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE*, 2017, pp. 3840–3847.
- [38] X. Yu, Y. Zhao, X. Yin, L. Lindemann, Signal temporal logic control synthesis among uncontrollable dynamic agents with conformal prediction, *Automatica* 183 (2026) 112616.