

Time-Robust Control for STL Specifications

Alëna Rodionova, Lars Lindemann, Manfred Morari and George J. Pappas[†]

Abstract—We present a robust control framework for time-critical systems in which satisfying real-time constraints robustly is of utmost importance for the safety of the system. Signal Temporal Logic (STL) provides a formal means to express a large variety of real-time constraints over signals and is suited for planning and control purposes as it allows us to reason about the time robustness of such constraints. The time robustness of STL particularly quantifies the extent to which timing uncertainties can be tolerated without violating real-time specifications. In this paper, we first pose a control problem in which we aim to find an optimal input sequence to a control system that maximizes the time robustness of an STL constraint. We then propose a Mixed Integer Linear Program (MILP) encoding and provide correctness guarantees along with a complexity analysis of the encoding. We also show in two case studies that maximizing STL time robustness allows to account for timing uncertainties of the underlying control system.

Keywords: *Signal Temporal Logic, time robustness, formal control synthesis*

I. INTRODUCTION

Consider the following three time-critical systems: an air traffic control center sequencing arriving airplanes, a group of autonomous robots with the goal to maximize the time within each others communication range, and an automated warehouse in which interleaving processes are required to happen in a pre-specified order. These systems, while being fundamentally different in terms of their dynamics and specifications, share the property that not meeting their real-time constraints can compromise the safety of the system. Due to timing uncertainties, i.e., incorrect clock synchronizations, there may exist various reasons why such real-time constraints are not met by a control system. Hence, a natural objective is to design a control system to be robust to such timing uncertainties and to even maximize this robustness. To formulate real-time constraints, researchers have used real-time temporal logics as a specification formalism such as Metric Temporal Logic (MTL) [1] and Signal Temporal Logic (STL) [2]. In this work, we focus on the time robustness of STL specifications defined in [3] and propose and solve a control problem that aims to maximize the time robustness to make a system resilient to timing uncertainties.

A. Related Work

The real-time temporal logic STL, introduced in [2], is interpreted over real-valued continuous-time signals. Various forms of robust semantics have been proposed for STL

that quantify the extent by which an STL specification is satisfied or violated by a signal. The authors in [4] define the robustness degree as a tube around a signal in which all signals either satisfy or violate the specification at hand. The robustness degree reflects a notion of space robustness. Various other space robustness notions have been proposed in the literature, such as [3] that closely follows [4]. Other space robustness notions are the arithmetic-geometric integral mean robustness [5] and the smooth cumulative robustness [6]. Another space robustness notion, especially tailored for aiding exploration in reinforcement learning problems, has been presented in [7]. A connection between linear, time-invariant filtering and space robustness has been made in [8]. The authors in [3] make a first step towards defining a notion of time robustness that we particularly utilize in this paper. Alternatively, [9] proposes averaged STL and captures a form of time robustness by averaging over time intervals and assigning priorities. This way, similarly to the time robustness in [3], expeditiousness and other modalities can be captured in terms of robustness. All these notions, except for time robustness in [3], primarily focus on space robustness.

Control of dynamical systems under STL specifications has first been considered by the authors in [10] by proposing a Mixed Integer Linear Program (MILP) encoding that allows to maximize the space robustness as defined in [3]. Other optimization-based methods, allowing again the maximization of space robustness, have been proposed in [5], [11], [12] by using smooth approximations of the space robustness notion in [3] that allows to use of off-the-shelf gradient-based solvers. A reinforcement learning approach for multi-agent systems has been presented in [13]. Another direction has been to develop robust feedback control laws that maximize the space robustness of an STL specification [14]. While all these methods have presented compelling results, they focus on forms of space robustness while none of these methods maximize the time robustness of an STL specification. The closest work in terms of control under STL time robustness constraints is [15] where the authors, however, only consider very special instances of STL (eventually and always operators). To the best of our knowledge, the problem of maximizing the time robustness by which a system satisfies an STL specification has neither been formulated nor solved.

B. Contributions and Paper Organization

In this paper, we consider linear discrete-time systems and design controllers for which the time robustness of an STL specification, as presented in [3], is maximized. In particular, we make the following contributions:

[†] The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia PA, USA. {nellro, larsl, morari, pappasg}@seas.upenn.edu.

This work was supported by AFOSR Assured Autonomy.

- 1) We formulate a novel control problem to find an input sequence that maximizes the time robustness by which the system satisfies an STL specification. Thereby, we define a new problem space for the robust control of time-critical systems.
- 2) We extend the theoretical foundation of time robustness and complement the work in [3] with additional theoretical results on soundness of time robustness.
- 3) To solve the optimal control problem, we propose a novel Mixed-Integer Linear Program (MILP) encoding. We provide correctness guarantees and a complexity analysis of the encoding.
- 4) We show how the proposed MILP encoding can be used to perform the time-robust control for unmanned aerial vehicles and multi-agent surveillance.

The remainder of the paper is organized as follows. Sec. II introduces STL with its qualitative and time-robust semantics. In Sec. III we state the control synthesis problem that we aim to solve. We also present the soundness theorem with the proof available in the Appendix Sec. VI-A. In Sec. IV we present the proposed solution using the MILP encoding. Extensive simulations and case studies are presented in Sec. V. Finally, we summarize with conclusions in Sec. VI.

II. SIGNAL TEMPORAL LOGIC (STL) ROBUSTNESS

Let \mathbf{x} be a discrete-time signal $\mathbf{x} : \mathbb{T} \rightarrow X$ such that $\mathbb{T} \subseteq \mathbb{Z}_{\geq 0}$ is the time domain and $X \subseteq \mathbb{R}^n$ is a metric space. We denote a signal state at time step t as $x_t \in X$ and call the set of all signals $\mathbf{x} : \mathbb{T} \rightarrow X$ the *signal space* $X^{\mathbb{T}}$. Let $M = \{\mu_1, \dots, \mu_L\}$ be a set of real-valued linear functions of the state x , $\mu_k(x) : X \rightarrow \mathbb{R}$. For each μ_k its corresponding *predicate* p_k is defined as $p_k := \mu_k(x) \geq 0$. Thus, each predicate defines a set in which p_k holds true, namely p_k defines the set $\{x \in X \mid \mu_k(x) \geq 0\}$ in which p_k is true. All defined predicates construct the set $AP := \{p_1, \dots, p_L\}$. Let interval $I = [a, b] \subset \mathbb{T}$ be a non-empty time interval where $0 \leq a \leq b$. For $t \in \mathbb{T}$, the time interval $[t+a, t+b]$ is denoted as $t+I$. The supremum operator is written \sqcup and infimum is written \sqcap , \top is a Boolean *true*. We interpret $\text{sign}(0) = 1$.

The syntax of Signal Temporal Logic (STL) is defined recursively as follows [2]:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U}_I \varphi_2 \quad (1)$$

where $p \in AP$ is a predicate, \neg and \wedge are the Boolean negation and conjunction, respectively, and \mathcal{U}_I is the Until temporal operator over bounded interval I . The disjunction (\vee) and implication (\implies) are defined as usual. Additional temporal operators Eventually (\diamond) and Always (\square) can be defined as $\diamond_I \varphi = \top \mathcal{U}_I \varphi$ and $\square_I \varphi = \neg \diamond_I \neg \varphi$.

Formally, the *semantics* of an STL formula φ defines what it means for a system trajectory \mathbf{x} to satisfy φ at time point t , denoted as $(\mathbf{x}, t) \models \varphi$. If satisfaction does not hold it is denoted as $(\mathbf{x}, t) \not\models \varphi$. We will use the characteristic function notation:

Definition 2.1 (STL characteristic function [3]): The characteristic function $\chi_\varphi(\mathbf{x}, t) : X^{\mathbb{T}} \times \mathbb{T} \rightarrow \{-1, +1\}$ of an

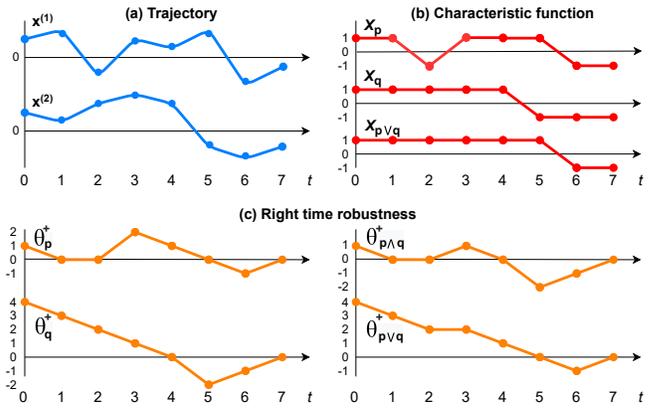


Fig. 1: Evolution of the trajectory (a), characteristic function (b) and time robustness (c) from Example 1.

STL formula φ relative to a trajectory \mathbf{x} at time t is defined inductively as:

$$\begin{aligned} \chi_p(\mathbf{x}, t) &= \text{sign}(\mu(x_t)) \\ \chi_{\neg\varphi}(\mathbf{x}, t) &= -\chi_\varphi(\mathbf{x}, t) \\ \chi_{\varphi_1 \wedge \varphi_2}(\mathbf{x}, t) &= \chi_{\varphi_1}(\mathbf{x}, t) \sqcap \chi_{\varphi_2}(\mathbf{x}, t) \\ \chi_{\varphi_1 \mathcal{U}_I \varphi_2}(\mathbf{x}, t) &= \sqcup_{t' \in t+I} \left(\chi_{\varphi_2}(\mathbf{x}, t') \sqcap \prod_{t'' \in [t, t']} \chi_{\varphi_1}(\mathbf{x}, t'') \right) \end{aligned} \quad (2)$$

The Boolean semantics of STL states that $\chi_\varphi(\mathbf{x}, t) = 1$ when $(\mathbf{x}, t) \models \varphi$, and $\chi_\varphi(\mathbf{x}, t) = -1$ when $(\mathbf{x}, t) \not\models \varphi$. While the Boolean STL semantics shows *whether* a signal \mathbf{x} satisfies a given specification φ at time t or not, STL quantitative semantics, also known as robustness, measures *how much* the signal is satisfying or violating the specification. First notion of a quantitative measure of satisfaction has been presented in [4]. This notion defines a so-called *spatial robustness degree*. Later an alternative notion was presented in [3] where the authors suggested the novel measure of *time robustness*.

Definition 2.2 (Time Robustness [3]): The right and left time robustness of an STL formula φ with respect to a trajectory \mathbf{x} at time t are defined inductively as:

$$\theta_p^+(\mathbf{x}, t) = \chi_p(\mathbf{x}, t) \cdot \max\{\tau \geq 0 : \forall t' \in [t, t+\tau], \chi_p(\mathbf{x}, t') = \chi_p(\mathbf{x}, t)\} \quad (3)$$

$$\theta_p^-(\mathbf{x}, t) = \chi_p(\mathbf{x}, t) \cdot \max\{\tau \geq 0 : \forall t' \in [t-\tau, t], \chi_p(\mathbf{x}, t') = \chi_p(\mathbf{x}, t)\} \quad (4)$$

and then applying to each θ_p^\boxtimes , where $\boxtimes \in \{+, -\}$, the recursive rules of the operators similarly to Def. 2.1 resulting in:

$$\theta_{\neg\varphi}^\boxtimes(\mathbf{x}, t) = -\theta_\varphi^\boxtimes(\mathbf{x}, t) \quad (5)$$

$$\theta_{\varphi_1 \wedge \varphi_2}^\boxtimes(\mathbf{x}, t) = \theta_{\varphi_1}^\boxtimes(\mathbf{x}, t) \sqcap \theta_{\varphi_2}^\boxtimes(\mathbf{x}, t) \quad (6)$$

$$\theta_{\varphi_1 \mathcal{U}_I \varphi_2}^\boxtimes(\mathbf{x}, t) = \sqcup_{t' \in t+I} \left(\theta_{\varphi_2}^\boxtimes(\mathbf{x}, t') \sqcap \prod_{t'' \in [t, t']} \theta_{\varphi_1}^\boxtimes(\mathbf{x}, t'') \right) \quad (7)$$

Example 1: We next illustrate the time robustness notion in Fig. 1. The trajectory $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]^\top$ presented in Fig.1(a) is finite and discrete-time, its state at each time step $t = 0, \dots, 7$ is $x_t = [x_t^{(1)}, x_t^{(2)}]^\top \in \mathbb{R}^2$. Let the two predicates be

$p = x^{(1)} \geq 0$ and $q = x^{(2)} \geq 0$. The characteristic functions for p , q and $\varphi = p \vee q$ are shown in Fig. 1(b). The evolutions of the right time robustness $\theta_\varphi^+(\mathbf{x}, t)$, $t = 0, \dots, 7$ for the above predicates, $\varphi = p \vee q$ and $\varphi = p \wedge q$ are presented in Fig. 1(c). For instance, consider time step $t = 3$. Since $\chi_p(\mathbf{x}, 3) = \chi_p(\mathbf{x}, 4) = \chi_p(\mathbf{x}, 5) = +1$ and $\chi_p(\mathbf{x}, 6) = -1$, by Def. 2.2, $\theta_p^+(\mathbf{x}, 3) = 2$. On the other hand, $\chi_q(\mathbf{x}, 3) = \chi_q(\mathbf{x}, 4) = +1 \neq -1 = \chi_q(\mathbf{x}, 5)$, thus, $\theta_q^+(\mathbf{x}, 3) = 1$. Also, from (6), $\theta_{p \wedge q}^+(\mathbf{x}, 3) = \min(\theta_p^+(\mathbf{x}, 3), \theta_q^+(\mathbf{x}, 3)) = 1$ but $\theta_{p \vee q}^+(\mathbf{x}, 3) = \max(\theta_p^+(\mathbf{x}, 3), \theta_q^+(\mathbf{x}, 3)) = 2$.

A. Soundness of Time Robustness

In this section, we formulate and prove the theorem that states the relationship between the time robustness and Boolean semantics of STL. The following Theorem 2.1 is fundamental to the underlying theory of time robustness and the control synthesis problem defined in the next section, but to the best of our knowledge has never been formally stated in any previously published papers.

Theorem 2.1 (Soundness): For an STL formula φ , trajectory $\mathbf{x} : \mathbb{T} \rightarrow X$, time $t \in \mathbb{T}$ and $\bowtie \in \{+, -\}$ the following results hold:

- 1) $\theta_\varphi^{\bowtie}(\mathbf{x}, t) > 0 \implies \chi_\varphi(\mathbf{x}, t) = +1$
- 2) $\theta_\varphi^{\bowtie}(\mathbf{x}, t) < 0 \implies \chi_\varphi(\mathbf{x}, t) = -1$
- 3) $\chi_\varphi(\mathbf{x}, t) = +1 \implies \theta_\varphi^{\bowtie}(\mathbf{x}, t) \geq 0$
- 4) $\chi_\varphi(\mathbf{x}, t) = -1 \implies \theta_\varphi^{\bowtie}(\mathbf{x}, t) \leq 0$

See the proof in the Appendix Sec. VI-A.

Note that equivalence $\theta_\varphi^{\bowtie}(\mathbf{x}, t) \geq 0 \iff \chi_\varphi(\mathbf{x}, t) = +1$ does not hold¹ since when $\theta_\varphi^{\bowtie}(\mathbf{x}, t) = 0$ we cannot determine if the formula is satisfied or violated.

Example 1 (cont.): Take a look again at the trajectory \mathbf{x} shown in Fig. 1. Consider $\varphi = p \vee q$. From Fig 1(c) one can see that $\theta_\varphi^+(\mathbf{x}, 3) = 2 > 0$ and $\theta_\varphi^+(\mathbf{x}, 6) = -1 < 0$. Since $\chi_\varphi(\mathbf{x}, 3) = +1$ then due to Thm. 2.1, $(\mathbf{x}, 3) \models \varphi$. Similarly, $\chi_\varphi(\mathbf{x}, 6) = -1$ leads to $(\mathbf{x}, 6) \not\models \varphi$. By taking a look at Fig. 1(b) one can conclude, that these facts indeed hold. Also note that for time points $t = 5$ and $t = 7$, $\theta_\varphi^+(\mathbf{x}, 5) = \theta_\varphi^+(\mathbf{x}, 7) = 0$ but $\chi_\varphi(\mathbf{x}, 5) = +1$ and $\chi_\varphi(\mathbf{x}, 7) = -1$.

III. TIME-ROBUST STL CONTROL SYNTHESIS

Consider a discrete-time, linear control system:

$$x_{t+1} = Ax_t + Bu_t \quad (8)$$

where $x_t \in X \subseteq \mathbb{R}^n$ is the current state of the system in a bounded domain X , $u_t \in U \subseteq \mathbb{R}^m$ is the current control input, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. The system's initial state x_0 takes values from some initial set $X_0 \subseteq X$. Given an initial state x_0 and a finite control input sequence $\mathbf{u} = (u_0, u_1, \dots, u_{H-1})$ s.t. $u_t \in U$, a *trajectory* of the system is the unique sequence of states $\mathbf{x} = (x_0, x_1, \dots, x_H)$, s.t. $x_t \in X$ and (8) holds. We denote $\mathbb{T} = \{0, 1, \dots, H\}$ to be a finite discrete time domain, where H is a time horizon which is sufficiently large to verify the satisfaction of formula φ .

For time-critical systems one is often not only interested in satisfying an STL specification φ but also in satisfying

¹Equivalence $\theta_\varphi^{\bowtie}(\mathbf{x}, t) \leq 0 \iff \chi_\varphi(\mathbf{x}, t) = -1$ does not hold either, same reasoning applied.

Algorithm 1: The function $(\theta_p^+(\mathbf{x}), \mathcal{P}) = \text{MILP_PREDICATE}(p, \mathbf{x})$

Input: Linear predicate $p := \mu(x) \geq 0$, trajectory \mathbf{x}
Output: Right time robustness $\theta_p^+(\mathbf{x})$ and the set of MILP constraints \mathcal{P}

- 1: Let $z_t \in \mathbb{B}$ be constrained by (9), $t = 0, \dots, H$.
 - 2: Let $\chi_p(\mathbf{x})$ be constrained by (10).
 - 3: Let $c_t^1, c_t^0 \in \mathbb{Z}$ be constrained by (11) and (12).
 - 4: $\theta_p^+(\mathbf{x}) \stackrel{(13)}{=} c_t^1 + c_t^0 - \chi_p(\mathbf{x}, t)$.
 - 5: \mathcal{P} consists of the MILP constraints (9), (10), (11), (12), and (13).
-

φ robustly with respect to the time robustness $\theta_\varphi^{\bowtie}(\mathbf{x}, 0)$. Achieving such robustness is particularly important when the system is subject to timing uncertainties, including agent delays or early starts. Towards this goal, we aim to maximize time robustness $\theta_\varphi^{\bowtie}(\mathbf{x}, 0)$ while imposing a lower bound θ^* on $\theta_\varphi^{\bowtie}(\mathbf{x}, 0)$. In other words, for a given STL specification φ and initial condition $x_0 \in X_0$, we want to find a control input sequence $\mathbf{u}^* = (u_0^*, u_1^*, \dots, u_{H-1}^*)$ such that the corresponding system trajectory \mathbf{x}^* satisfies the specification φ and results in a time robustness $\theta_\varphi^{\bowtie}(\mathbf{x}^*, 0)$ that is maximized and satisfies a minimum required time robustness $\theta^* > 0$. Formally, this can be defined as the following problem.

Problem 1 (Time-Robust STL Control Synthesis): Given an STL specification φ , time horizon H , discrete-time linear control system (8) with initial condition $x_0 \in X_0$ and a lower bound θ^* , solve

$$\begin{aligned} \mathbf{u}^* &= \underset{\mathbf{u}}{\operatorname{argmax}} \quad \theta_\varphi^{\bowtie}(\mathbf{x}, 0) \\ \text{s.t.} \quad &x_{t+1} = Ax_t + Bu_t, \quad u_t \in U, \quad t = 0, \dots, H-1 \\ &x_t \in X, \quad t = 0, \dots, H \\ &\theta_\varphi^{\bowtie}(\mathbf{x}, 0) \geq \theta^* > 0. \end{aligned}$$

Since the robustness function θ_φ^{\bowtie} is neither continuous nor smooth, gradient-based solvers cannot be applied to solve Prob. 1. Even more challenging is the fact that θ_φ^{\bowtie} includes signal shifts according to (3)-(4). Thus, techniques based on smooth approximations [16], [11], non-smooth optimization theory [17] or Monte-Carlo optimization [18] are not applicable either. This motivates the use of Mixed-Integer Linear Programming (MILP) in this work to explicitly encode the signal shifts in (3)-(4). We describe the details of the MILP encoding of Prob. 1 in the next section.

IV. MILP ENCODING OF TIME ROBUSTNESS

In this section, we present the right time robustness encoding. The left time robustness can be encoded analogously with only minor modifications and is hence omitted.

Following Def. 2.2, given the STL formula φ , the time robustness can be computed recursively on the structure of φ . We will start with the main milestone of the overall time robustness MILP encoding, that is the encoding of predicates, i.e. $\theta_p^+(\mathbf{x}, t)$, in Section IV-A. We then briefly describe the encoding of other STL operators in Section IV-B.

A. MILP Encoding of STL Predicates

We propose the idea of using counter variables within an MILP that enumerate the sequence of interest. Such counting idea implemented through the proposed MILP counters described later by (11)-(12) is the key concept behind the presented time robustness MILP encoding. Below we first summarize the result of the encoding for STL predicates, then present an algorithm where we explain the construction in more details and finally, consider the encoding on a particular example.

Proposition 4.1 (MILP encoding of STL predicates): For a linear predicate $p \in AP$ and trajectory \mathbf{x} described by the system constraints (8), the right time robustness sequence $\theta_p^+(\mathbf{x}) = (\theta_p^+(\mathbf{x}, 0), \dots, \theta_p^+(\mathbf{x}, H))$, with $\theta_p^+(\mathbf{x}, t)$ defined by (3), is equivalent to the set of MILP constraints produced by the function $(\theta_p^+(\mathbf{x}), \mathcal{P}) = \text{MILP_PREDICATE}(p, \mathbf{x})$ in Alg. 1.

The proof is by construction and we explain a step-by-step construction of the Alg. 1 as follows.

- 1) First, we construct a binary variable $z_t \in \{0, 1\}$ that corresponds to the Boolean satisfaction of the predicate p by trajectory \mathbf{x} at every time point $t = 0, \dots, H$, i.e. we enforce that $z_t = 1$ if and only if $\mu(x_t) \geq 0$. With an assumption that $\mu(x_t)$ is a linear function of the state, variable z_t can be defined as a set of MILP constraints as follows [19]:

$$\mu(x_t) \leq (M + \varepsilon) \cdot z_t - \varepsilon, \quad \mu(x_t) \geq m \cdot (1 - z_t), \quad (9)$$

where ε is a small positive constant that represents the tolerance, $M = \max_{x \in X} \mu(x)$ and $m = \min_{x \in X} \mu(x)$. By [19], the over-estimate of M and under-estimate of m suffice for the equivalence as well.

- 2) Since $z_t = 1$ if and only if $\mu(x_t) \geq 0$ and by Def. 2.1 $\chi_p(\mathbf{x}, t) = \text{sign} \mu(x_t) \in \{\pm 1\}$, the characteristic function can be encoded as

$$\chi_p(\mathbf{x}, t) = 2z_t - 1. \quad (10)$$

- 3) Recall that

$$\theta_p^+(\mathbf{x}, t) = \chi_p(\mathbf{x}, t) \cdot \max\{\tau \geq 0 : \forall t' \in [t, t + \tau], \chi_p(\mathbf{x}, t') = \chi_p(\mathbf{x}, t)\}.$$

Since $\chi_p(\mathbf{x}, t) \in \{-1, +1\}$ there exist two disjoint possibilities for θ_p^+ which in terms of z_t can be written as:

$$\theta_p^+(\mathbf{x}, t) = \begin{cases} \max\{\tau : \forall t' \in [t, t + \tau], z_{t'} = 1\}, & \text{if } z_t = 1 \\ -\max\{\tau : \forall t' \in [t, t + \tau], z_{t'} = 0\}, & \text{if } z_t = 0 \end{cases}$$

where $\tau \geq 0$. In other words, if at time point t , $z_t = 1$, one can count the maximum number of sequential $z_{t'} = 1$ where $t' > t$ in order to calculate $\theta_p^+(\mathbf{x}, t)$. On the other hand, if $z_t = 0$, then counting 0s to the right and multiplying the final value by -1 will define the time robustness value.

- 4) To implement the counting idea mentioned in the previous step, we construct two variables that count 1s and 0s to the right of t but for $t' \geq t$. Let their

t	0	1	2	3	4	5	6	7	8
z_t	1	1	0	1	1	1	0	0	
$\chi_p(\mathbf{x}, t)$	1	1	-1	1	1	-1	-1		
$c_t^1 = (c_{t+1}^1 + 1) \cdot z_t, c_{H+1}^1 = 0$	2	1	0	3	2	1	0	0	0
$c_t^0 = (c_{t+1}^0 - 1) \cdot (1 - z_t), c_{H+1}^0 = 0$	0	0	-1	0	0	0	-2	-1	0
$\theta_p^+(\mathbf{x}, t) = c_t^1 + c_t^0 - \chi_p(\mathbf{x}, t)$	1	0	0	2	1	0	-1	0	

TABLE I: Estimation of $\theta_p^+(\mathbf{x}, t)$ from Example 1 following Alg. 1.

recursive definitions be as following:

$$c_t^1 = (c_{t+1}^1 + 1) \cdot z_t, \quad c_{H+1}^1 = 0 \quad (11)$$

$$c_t^0 = (c_{t+1}^0 - 1) \cdot (1 - z_t), \quad c_{H+1}^0 = 0 \quad (12)$$

By construction, the counter c_t^1 counts the maximum number of sequential $z_{t'} = 1$ when $t' \geq t$. The second counter c_t^0 counts sequential $z_{t'} = 0$ when $t' \geq t$ and multiplies the final value by -1 . Note that counters are defined backwards: from $t = H + 1$ to $t = 0$, so while the length of trajectory \mathbf{x} is $H + 1$, the lengths of c_t sequences are $H + 2$.

- 5) Time robustness is defined by the sequential $z_{t'}$ when $t' > t$, not when $t' \geq t$. Therefore, to use the counters (11)-(12) to encode the time robustness, the counters should be modified as they must exclude the corresponding counted value at time point t :

$$\tilde{c}_t^1 = c_t^1 - z_t, \quad t = 0, \dots, H$$

$$\tilde{c}_t^0 = c_t^0 + (1 - z_t), \quad t = 0, \dots, H$$

- 6) Using the fact from Step 3) that two possibilities of z_t being 1 or 0 are disjoint (i.e. $\tilde{c}_t^1 \neq 0$ then $\tilde{c}_t^0 = 0$), the right time robustness $\theta_p^+(\mathbf{x}, t)$ is defined as:

$$\begin{aligned} \theta_p^+(\mathbf{x}, t) &= \tilde{c}_t^1 + \tilde{c}_t^0 = c_t^1 + c_t^0 - (2z_t - 1) \\ &= c_t^1 + c_t^0 - \chi_p(\mathbf{x}, t). \end{aligned} \quad (13)$$

Example 1 (cont.): Consider again the signal \mathbf{x} and predicate p shown in Fig. 1. In Table I we consider a step-by-step estimation of the right time robustness for predicate p following the MILP encoding procedure described in Alg.1 and Sec.IV-A. On the other hand, using Def. 2.2, one can check that $\theta_p^+(\mathbf{x})$ is indeed equal to $(1, 0, 0, 2, 1, 0, -1, 0)$. MILP encoding procedure leads to the same result as its estimation by the definition.

Remark. Constraints (11)-(12) are specified using a product of integer and Boolean variables. In Lemma 4.1 below we show that such product can be expressed as MILP constraints.

Lemma 4.1 (If-then-else product construct [20]): Let $b \in \{0, 1\}$ be a Boolean variable and let x be an integer variable such that lower and upper bounds are known constants, $x_l \leq x \leq x_u$. The expression $y = b \cdot x$ can be equivalently expressed as a set of mixed-integer linear constraints as follows:

$$\begin{aligned} x_l b &\leq y \leq x_u b \\ x - x_u(1 - b) &\leq y \leq x - x_l(1 - b). \end{aligned} \quad (14)$$

Mission	# Constraints	# Variables		Computation time (s)		Time Rob., $\theta_\varphi^+(\mathbf{x}, 0)$ (Time units)
		Boolean	Integer	YALMIP	Solver	
$\varphi_1 = \square_{[0,20]}p$, Sec. V	543	71	151	0.26	0.15	29
$\varphi_2 = \diamond_{[0,20]}q$, Sec. V	543	71	119	0.25	0.18	49
$\varphi_3 = \square_{[0,10]} \diamond_{[0,10]}p$, Sec. V	776	182	162	0.27	0.28	39
$\varphi_4 = \varphi_2 \wedge \square_{[0,5]}p$, Sec. V	1061	129	253	0.28	0.23	44
$\varphi_5 = \varphi_2 \wedge \varphi_3$, Sec. V	1324	255	264	0.31	0.24	39
Case study 1, φ_{uav} , Eq.(17)	2447	224	502	0.41	0.34	23
Case study 2, φ_{surv} , Eq.(21)	8794	2750	858	0.82	49.69	14
Case study 2, φ'_{surv} , Eq.(22)	9034	2750	858	0.88	29.08	14
Case study 2, feasibility formulation, $\theta_{\varphi_{surv}}(\mathbf{x}, 0) = 5$, Fig. 5(a)	8795	2750	858	0.81	2.12	5
Case study 2, feasibility formulation, $\theta_{\varphi_{surv}}(\mathbf{x}, 0) = 1$, Fig. 5(b)	8795	2750	858	0.85	1.02	1

TABLE II: Computational complexity report. *Computation time* includes *YALMIP time* (the time used to build the MILP and convert it into appropriate format for the solver) and *Solver time* (the time taken by Gurobi to solve the problem).

B. MILP Encoding of STL Operators

Having encoded the time robustness of STL predicates as MILP constraints, the generalization to STL formulas is straight-forward and can use the encoding from [10].

For instance, let $\varphi = \bigwedge_{i=1}^n \varphi_i$ and $\forall i$, $\theta_{\varphi_i}^+(\mathbf{x}, t) = r_i$. Then $\theta_\varphi^+(\mathbf{x}, t) = r$ if and only if:

$$\begin{aligned} r_i - M(1 - b_i) &\leq r \leq r_i, \quad \forall i \in \{1, \dots, n\} \\ \sum_{i=1}^n b_i &= 1 \end{aligned} \quad (15)$$

where $b_i = \{0, 1\}$, $i = \{1, \dots, n\}$ are introduced binary variables and M is a big- M parameter.

For the complete definition of other STL operators, see the quantitative encoding of STL constraints from [10].

The overall MILP encoding framework is summarized in Alg. 2. The function `SYSTEM_CONSTRAINTS`(x_0, \mathbf{u}) defines linear constraints on the decision variable u_t according to (8) and such that $x_t \in X$. The function `MILP_PREDICATE`(p_k, \mathbf{x}) is defined by Alg. 1. We denote all encoded predicates used in φ for all time steps as $\theta_p^+(\mathbf{x})$. The function `MILP_OPERATORS`($\varphi, \mathbf{x}, \theta_p^+(\mathbf{x})$) is defined according to [10, Section IV.D.] and recursively follows the structure of φ . It outputs $\theta_\varphi^+(\mathbf{x}, 0) \in \mathbb{Z}$ and a set of MILP constraints \mathcal{O} . For example, for conjunctions, \mathcal{O} is illustrated in (15).

We summarize the main properties of the above MILP formulation in the following proposition.

- Proposition 4.2:*
- 1) The time-robust STL control synthesis Problem 1 is equivalent to a MILP and can be solved as described in Alg. 2.
 - 2) MILP encoding of $\theta_\varphi^+(\mathbf{x}, t)$ is a function of $O(H \cdot |AP|)$ binary and $O(H \cdot |\varphi|)$ continuous variables, where H is the time horizon, $|\varphi|$ is the number of operators in the formula φ and $|AP|$ is the number of used predicates.

V. EXPERIMENTAL RESULTS

The complete MILP implementation described in this paper together with the code to reproduce all the case studies presented in this section can be found as an open-source project under the following URL: <https://github.com/nellro/time-robust-control>.

All simulations were performed on a computer with an Intel Core i7-9750H 6-core processor and 16GB RAM, running

Ubuntu 18.04. The MILP was implemented in MATLAB using YALMIP [21] with Gurobi 9.1 solver [22].

Table II presents the report on the computational complexity of all simulations described below. It includes the number of generated MILP constraints, number of created binary and integer variables and computation times that include YALMIP and solver times.

To analyze and compare the computational complexity of the two case studies presented below we first provide the results for the following five generic STL formulas:

$$\begin{aligned} \varphi_1 &= \square_{[0,20]}p, & \varphi_2 &= \diamond_{[0,20]}q, \\ \varphi_3 &= \square_{[0,10]} \diamond_{[0,10]}p, & \varphi_4 &= \varphi_2 \wedge \square_{[0,5]}p, & \varphi_5 &= \varphi_2 \wedge \varphi_3 \end{aligned}$$

where $p = x \geq 0.1$, $q = x \leq 0.5$. We used a simple discrete-time linear system $x_{t+1} = u_t$, where $x_t, u_t \in \mathbb{R}$, and set time horizon to $H = 50$. Table II shows that with an increasing complexity of the formula, increases the number of constraints together with the number of created variables. This result is expected due to the previously stated Prop. 4.2. On the other hand, since solvers use various sophisticated heuristics, solver time might not be directly connected to the complexity of the formula. For example, $\varphi_5 = \varphi_2 \wedge \varphi_3$ but the solver time for φ_5 is lower than for its both sub-formulas.

Algorithm 2: MILP encoding of time robustness

Input: Specification φ , initial state x_0 , time horizon H , minimum time robustness θ^*

Output: Control input sequence $\mathbf{u}^* = (u_0^*, \dots, u_{H-1}^*)$

Set $\mathbf{u} = (u_0, \dots, u_{H-1})$ to be the decision variable

$\mathbf{x} = \text{SYSTEM_CONSTRAINTS}(x_0, \mathbf{u})$

for $k = 1, \dots, L$ **do**

$(\theta_{p_k}^+(\mathbf{x}), \mathcal{P}_k) = \text{MILP_PREDICATE}(p_k, \mathbf{x})$ // Sec. IV-A

end

$(\theta_\varphi^+(\mathbf{x}, 0), \mathcal{O}) = \text{MILP_OPERATORS}(\varphi, \mathbf{x}, \theta_p^+(\mathbf{x}))$ // Sec. IV-B

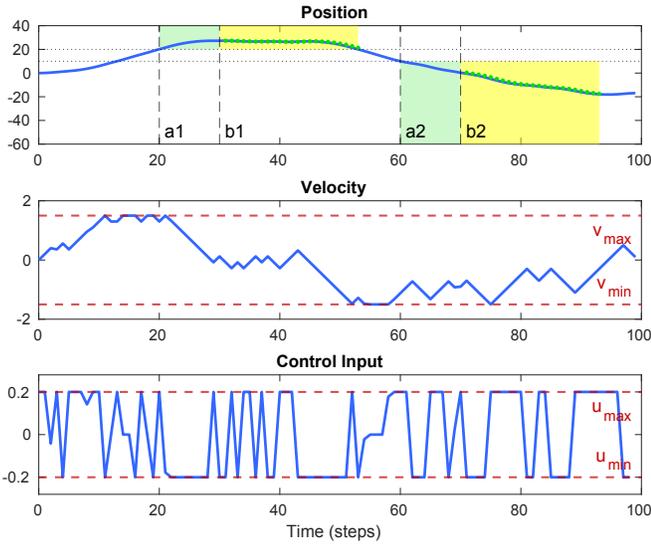
$\mathbf{u}^* = \underset{\mathbf{u}}{\text{argmax}} \theta_\varphi^+(\mathbf{x}, 0)$

s.t. $\theta_\varphi^+(\mathbf{x}, 0) \geq \theta^* > 0$

$\mathbf{x} = \text{SYSTEM_CONSTRAINTS}(x_0, \mathbf{u})$

$u_t \in U$ for $t = 0, \dots, H-1$

$\mathcal{O}, \mathcal{P}_k$, for $k = 1, \dots, L$



(a) UAV position, velocity and control input. Both times, after staying within the green zone from a_i to b_i time steps, UAV continued staying there for 23 more steps (shown as green dots).

Fig. 2: UAV Altitude Control. Found maximum right time robustness is $\theta_{\varphi_{uav}}^+(\mathbf{x}, 0) = 23$ time steps.

A. Case Study 1: UAV Altitude Control

Consider a one dimensional unmanned aerial vehicle (UAV) which is moving only in the z -axis direction. Its state $x = [z, v_z]^\top \in \mathbb{R}^2$ comprises of altitude z and velocity v_z . Initial UAV position and velocity are assumed to be zero. The UAV discrete-time linear dynamics are defined as:

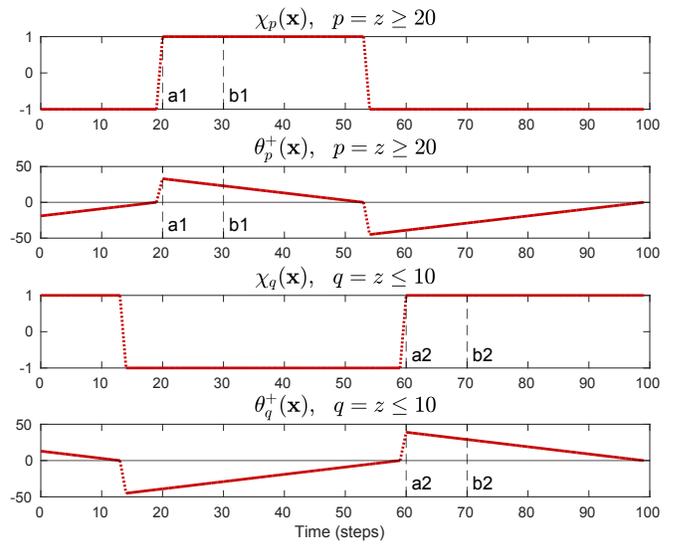
$$x_{t+1} = Ax_t + Bu_t, \quad x_0 = [0, 0]^\top \quad (16)$$

where $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$ and $|u_t| \leq u_{\max} = 0.2$. In this case study, the UAV is tasked to reach and stay above $z \geq 20$ altitude during the time interval $[20, 30]$ and then fly down and stay below $z \leq 10$ during $[60, 70]$ time steps. The UAV should also maintain its velocity within $|v_t| \leq v_{\max} = 1.5$. Time horizon is set to $H = 100$. The mission is captured in the following specification:²

$$\varphi_{uav} = \square_{[20,30]}(z \geq 20) \wedge \square_{[60,70]}(z \leq 10). \quad (17)$$

Results. Solving Prob. 1 for $\bowtie = +$ gives optimal solution $\theta_{\varphi_{uav}}^+(\mathbf{x}, 0) = 23$, see Fig. 2. One can see that even if the UAV started execution of its trajectory earlier by up to 23 time steps, the mission specification would still be satisfied. In Fig. 2(a) one can see that UAV indeed stays above $z = 20$ from 20 to 30 time steps (depicted in green) and then continues being above $z = 20$ for the next 23 time steps (depicted in yellow). On Fig. 2(b) this is seen as $\chi_p(\mathbf{x}, t) = +1$ for all time steps from 20 to 53, where $p = z \geq 20$. For these time steps, $\theta_p^+(\mathbf{x}, t)$ linearly decreases from 33 (when $t = 20$) first to 23 ($t = 30$) and then to 0 ($t = 53$). Which is expected since the right time robustness counts the sequential steps to the future. Above explanation applies analogously to

²Since the control input and velocity constraints specified above are of the form $\square_{[0,H]}p$, their maximum time robustness is 0. Therefore, they were removed from the mission specification φ and were implemented as strict MILP constraints.



(b) Evolution of the characteristic function $\chi(\mathbf{x})$ and the right time robustness $\theta^+(\mathbf{x})$ for predicates p and q , where $p = z \geq 20$ and $q = z \leq 10$.

the second subpart of the formula φ_{uav} when $z \leq 10$. For the computational complexity of this case study see Table II.

B. Case Study 2: Multi-agent Surveillance

We now formalize the case study with multiple agents carrying out a surveillance mission. Consider two identical agents moving in a two-dimensional space, see Fig. 3(a). They are tasked with a surveillance mission of the region of interest while having finite battery lives and specific re-charging schedules. The space consists of two electric agents α_i , $i \in \{1, 2\}$, one region of interest denoted as *Goal* and two charging stations *Charge*⁽ⁱ⁾, $i \in \{1, 2\}$. Let the state of each agent be $x^{(i)} = [x^i, v_x^i, y^i, v_y^i]^\top \in \mathbb{R}^4$ and the control input be $u^{(i)} = [u_x^i, u_y^i]^\top \in \mathbb{R}^2$. We denote the full state of the system as $x = [x^{(1)}, x^{(2)}]^\top$ and full control as $u = [u^{(1)}, u^{(2)}]^\top$. The linear state-space representation of the system is given by:

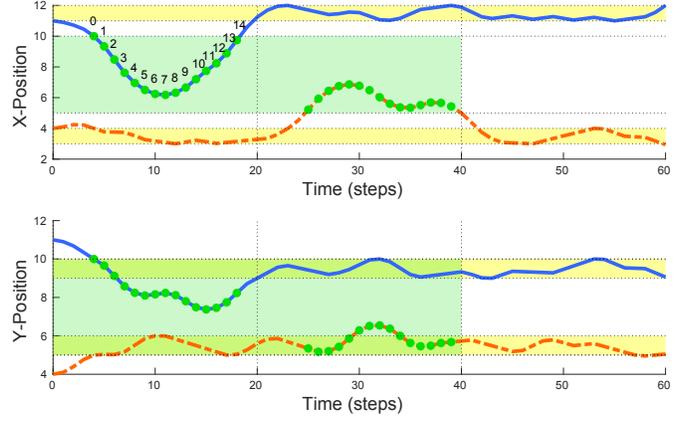
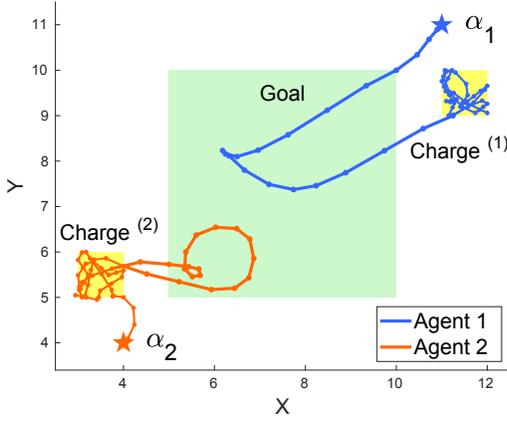
$$x_{t+1} = Ax_t + Bu_t, \quad \|u_t\|_\infty \leq 20, \quad (18)$$

where $A = I_4 \otimes \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}$ and $B = I_4 \otimes \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix}$, with I_n being an identity matrix of dimension $n \times n$ and symbol \otimes denoting the Kronecker product. Initial positions of the agents are $(11, 11)$ and $(4, 4)$, initial velocities are zero. We set time horizon to $H = 60$.

The goal surveillance sub-mission requires the *Goal* region to be visited by at least one of the agents within the first 20 time steps and then within the next 20 time steps. Formally, it is defined as:

$$\varphi_{gs} = \diamond_{[0,20]} [\alpha_1 \in Goal \vee \alpha_2 \in Goal] \wedge \diamond_{[20,40]} [\alpha_1 \in Goal \vee \alpha_2 \in Goal].$$

where notation $\alpha_i \in Z$ denotes more formal $[x^i, y^i]^\top \in Z$, i.e. position of the agent α_i is within the given rectangle Z . This can be defined as a conjunction of four linear predicates $x^i \leq X_{ub}$, $x^i \geq X_{lb}$, $y^i \leq Y_{ub}$, $y^i \geq Y_{lb}$.



(a) Trajectories for two agents generated by solving Prob. 1. Agents perform surveillance and recharging. Initial positions marked by \star .

(b) Trajectory projection on X-axis and Y-axis. The found robustness value is $\theta_{\varphi_{surv}}^+(\mathbf{x}, 0) = \theta_{\varphi_{gs}}^+(\mathbf{x}, 0) = 14$. We depict the meaning of $\theta_{\varphi_{gs}}^+(\mathbf{x}, 0)$ as green dots along the trajectory.

Fig. 3: Multi-agent surveillance. Goal set is represented in green color, charging zones are in yellow color. Found maximum right time robustness is $\theta_{\varphi_{surv}}^+(\mathbf{x}, 0) = 14$ time steps. Simulation is available at <https://tinyurl.com/multi-surveil>.

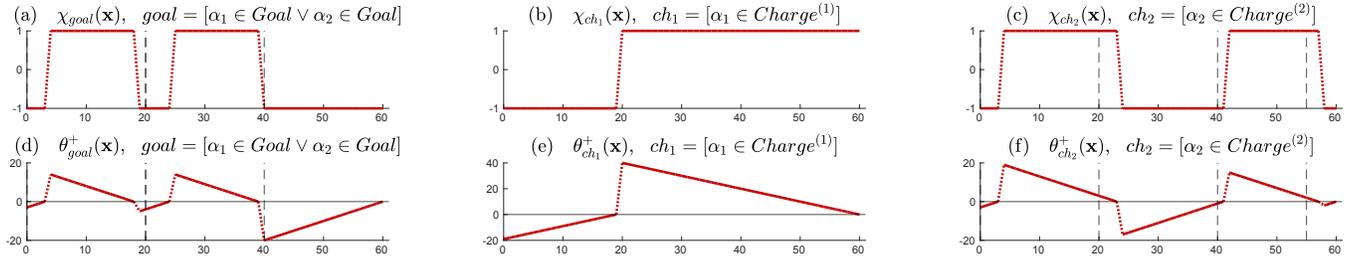


Fig. 4: Multi-agent surveillance. Evolution of the characteristic function $\chi(\mathbf{x})$ and the right time robustness $\theta^+(\mathbf{x})$ for sub-formula *goal* and predicates ch_1, ch_2 . Fig. 4(a),(d): The *goal* is satisfied during two disjoint time intervals, each interval consists of 14 time steps, $\theta_{goal}^+(\mathbf{x}, 4) = \theta_{goal}^+(\mathbf{x}, 25) = 14$. Fig. 4(b),(c),(e),(f): The maximum value of $\theta^+(\mathbf{x})$ for both predicates ch_1 and ch_2 is higher than 14.

The battery life of the first agent is 20 time units and charging takes 20 time units during which the agent should stay at its charging station. Such pattern must be satisfied at all-time within the given time horizon of the formula:

$$\varphi_{ch,1} = \square_{[0,20]} \diamond_{[0,20]} \square_{[0,20]} [\alpha_1 \in Charge^{(1)}]. \quad (19)$$

Second agent must satisfy a less restrictive battery charging schedule which is formally specified as:

$$\begin{aligned} \varphi_{ch,2} = & \diamond_{[0,20]} [\alpha_2 \in Charge^{(2)}] \\ & \wedge \diamond_{[40,55]} [\alpha_2 \in Charge^{(2)}]. \end{aligned} \quad (20)$$

The overall multi-agent surveillance mission is defined as:

$$\varphi_{surv} = \varphi_{gs} \wedge \varphi_{ch,1} \wedge \varphi_{ch,2}. \quad (21)$$

Results. Solving Prob. 1 for $\boxtimes = +$ gives optimal solution $\theta_{\varphi_{surv}}^+(\mathbf{x}, 0) = 14$ time units, see Fig. 3 and Fig. 4. Simulation is available at <https://tinyurl.com/multi-surveil>.

From Fig. 3(b) one can see that Agent 1 surveils the region of interest *Goal* while Agent 2 is in its charging zone $Charge^{(2)}$, and then Agent 1 goes to charging zone $Charge^{(1)}$ while Agent 2 continues to surveil the *Goal*. Table II shows that this case study is the most computationally heavy and requires 2750 Boolean and 858 integer variables. Solver needs 49.69 seconds to solve Prob. 1. For comparison, we also consider a variation of formula φ_{surv} where we require an additional constraint on velocities of both agents:

$$\varphi'_{surv} = \varphi_{surv} \wedge \square_{[0,H]} (\|v\|_{\infty} \leq 4). \quad (22)$$

Such additional constraint did not change the final right time robustness value, though did change the trajectories. Interesting observation is, though φ'_{surv} has an additional constraint on top of φ_{surv} , solver takes only 29.08 seconds to solve it in comparison with 46.69 for φ_{surv} .

We also performed simulations where we solve a variation of Prob. 1 where instead of the robustness maximization we solve a feasibility problem that results in robustness having precisely the desired value. See the resulting trajectories for such feasibility formulation of Prob. 1 in Fig. 5(a) for $\theta_{\varphi_{surv}}^+(\mathbf{x}, 0) = 5$ and Fig. 5(b) for $\theta_{\varphi_{surv}}^+(\mathbf{x}, 0) = 1$. Table II shows that though the size of the problem in terms of number of variables and constraints stays the same, the computation

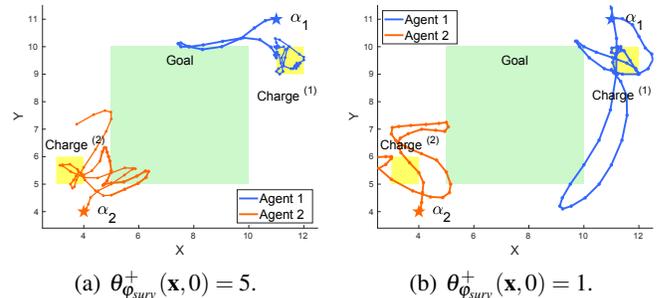


Fig. 5: Solutions to the time robustness feasibility problem instead of the maximization Prob. 1. Simulations are available at <https://tinyurl.com/rob-feasibility>.

time becomes drastically lower (2.12 and 1.02 seconds).

VI. CONCLUSIONS

We proposed a controlling system framework for time-critical systems. In particular, we considered system constraints formulated in Signal Temporal Logic (STL). Our framework is based on the Mixed Integer Linear Program (MILP) encoding that allows to maximize the time robustness of STL constraints while guaranteeing to achieve a desired lower bound of the time robustness. We provided correctness guarantees and a complexity analysis of the encoding and illustrated our theoretical findings in two case studies.

ACKNOWLEDGMENT

The authors would like to thank Yash Vardhan Pant for several insightful discussions about time robustness and the control synthesis problem.

REFERENCES

- [1] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-time systems*, 2(4):255–299, 1990.
- [2] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004.
- [3] A. Donzé and O. Maler. Robust satisfaction of temporal logic over real-valued signals. In *Proceedings of the International Conference on Formal Modeling and Analysis of Timed Systems*, 2010.
- [4] G. Fainekos and G. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 2009.
- [5] N. Mehdipour, C.-I. Vasile, and C. Belta. Average-based robustness for continuous-time signal temporal logic. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5312–5317. IEEE, 2019.
- [6] I. Haghghi, N. Mehdipour, E. Bartocci, and C. Belta. Control from signal temporal logic specifications with smooth cumulative quantitative semantics. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4361–4366. IEEE, 2019.
- [7] P. Varnai and D. V. Dimarogonas. On robustness metrics for learning STL tasks. In *2020 American Control Conference (ACC)*, pages 5394–5399. IEEE, 2020.
- [8] A. Rodionova, E. Bartocci, D. Nickovic, and R. Grosu. Temporal logic as filtering. In *Proceedings of the 19th International Conference on Hybrid Systems: Computing and Control*, pages 11–20, 2016.
- [9] T. Akazaki and I. Hasuo. Time robustness in MTL and expressivity in hybrid system falsification. In *International Conference on Computer Aided Verification*, pages 356–374. Springer, 2015.
- [10] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 81–87. IEEE, 2014.
- [11] Y. V. Pant, H. Abbas, R. A. Quaye, and R. Mangharam. Fly-by-logic: control of multi-drone fleets with temporal logic objectives. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCP)*, pages 186–197. IEEE, 2018.
- [12] Y. Gilpin, V. Kurtz, and H. Lin. A smooth robustness measure of signal temporal logic for symbolic control. *IEEE Control Systems Letters*, 5(1):241–246, 2020.
- [13] D. Muniraj, K. G. Vamvoudakis, and M. Farhood. Enforcing signal temporal logic specifications in multi-agent adversarial environments: A deep q-learning approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4141–4146. IEEE, 2018.
- [14] L. Lindemann and D. V. Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE control systems letters*, 3(1):96–101, 2018.
- [15] Z. Lin and J. S. Baras. Optimization-based motion planning and runtime monitoring for robotic agent with space and time tolerances. In *21st IFAC World Congress*, pages 1900–1905, 2020.

- [16] Y. V. Pant, H. Abbas, and R. Mangharam. Smooth operator: Control using the smooth robustness of temporal logic. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1235–1240. IEEE, 2017.
- [17] H. Abbas and G. Fainekos. Computing descent direction of MTL robustness for non-linear systems. In *American Control Conference (ACC)*, pages 4411–4416, 2013.
- [18] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivančić, and A. Gupta. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 12(2s):1–30, 2013.
- [19] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
- [20] A. Bemporad, F. D. Torrisi, and M. Morari. Discrete-time hybrid modeling and verification of the batch evaporator process benchmark. *European Journal of Control*, 7(4):382–399, 2001.
- [21] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, pages 284–289. IEEE, 2004.
- [22] L. Gurobi Optimization. Gurobi optimizer reference manual, 2021.

APPENDIX

A. Proof of Theorem 2.1

We are going to prove items 1) and 3) for $\boxtimes = +$. Items 2) and 4) for $\boxtimes = +$ as well as all the facts 1)–4) for $\boxtimes = -$ can be proven analogously. The proof is by induction on the structure of φ .

$$1) \theta_{\varphi}^{+}(\mathbf{x}, t) > 0 \implies \chi_{\varphi}(\mathbf{x}, t) = +1.$$

- $\varphi = p$. From the definition of θ_p^{+} , since $\tau \geq 0$ then $\chi_p(\mathbf{x}, t) > 0$ and thus, since $\chi \in \{\pm 1\}$, $\chi_p(\mathbf{x}, t) = +1$.
- $\varphi = \neg\varphi_1$. By definition, $\theta_{\neg\varphi_1}^{+}(\mathbf{x}, t) = -\theta_{\varphi_1}^{+}(\mathbf{x}, t) > 0$. Therefore, $\theta_{\varphi_1}^{+}(\mathbf{x}, t) < 0$ and from the induction hypothesis, $\chi_{\varphi_1}(\mathbf{x}, t) = -1$, thus, $\chi_{\neg\varphi_1}(\mathbf{x}, t) = +1$.
- $\varphi = \varphi_1 \wedge \varphi_2$. Since $\theta_{\varphi_1 \wedge \varphi_2}^{+}(\mathbf{x}, t) = \theta_{\varphi_1}^{+}(\mathbf{x}, t) \cap \theta_{\varphi_2}^{+}(\mathbf{x}, t) > 0$, both terms are positive: $\theta_{\varphi_1}^{+}(\mathbf{x}, t) > 0$ and $\theta_{\varphi_2}^{+}(\mathbf{x}, t) > 0$. From the induction hypothesis, $\chi_{\varphi_1}(\mathbf{x}, t) = \chi_{\varphi_2}(\mathbf{x}, t) = +1$ and thus, $\chi_{\varphi}(\mathbf{x}, t) = \chi_{\varphi_1}(\mathbf{x}, t) \cap \chi_{\varphi_2}(\mathbf{x}, t) = +1$.
- $\varphi = \varphi_1 \mathcal{U} \varphi_2$. Due to the maximum operator in the definition of Until operator, $\exists t' \in t + I$ such that $\theta_{\varphi_2}^{+}(\mathbf{x}, t') \cap \prod_{t'' \in [t, t']} \theta_{\varphi_1}^{+}(\mathbf{x}, t'') > 0$. Now due to the minimum operators, $\theta_{\varphi_2}^{+}(\mathbf{x}, t') > 0$ and $\forall t'' \in [t, t']$, $\theta_{\varphi_1}^{+}(\mathbf{x}, t'') > 0$. Therefore, from the induction hypothesis, $\exists t' \in t + I$, $\chi_{\varphi_2}(\mathbf{x}, t') = +1$ and $\forall t'' \in [t, t']$, $\chi_{\varphi_1}(\mathbf{x}, t'') = +1$. And thus, $\chi_{\varphi}(\mathbf{x}, t) = +1$. This concludes the proof.

$$3) \chi_{\varphi}(\mathbf{x}, t) = +1 \implies \theta_{\varphi}^{+}(\mathbf{x}, t) \geq 0.$$

- $\varphi = p$. $\chi_p(\mathbf{x}, t) = +1$ thus, by def., $\theta_p^{+}(\mathbf{x}, t) \geq 0$.
- $\varphi = \neg\varphi_1$. Since, $\chi_{\neg\varphi_1}(\mathbf{x}, t) = -\chi_{\varphi_1}(\mathbf{x}, t) = +1$, therefore, $\chi_{\varphi_1}(\mathbf{x}, t) = -1$, from the induction hypothesis $\theta_{\varphi_1}^{+} \leq 0$. Since $\theta_{\neg\varphi_1}^{+}(\mathbf{x}, t) = -\theta_{\varphi_1}^{+}(\mathbf{x}, t) \geq 0$.
- $\varphi = \varphi_1 \wedge \varphi_2$. Since $\chi_{\varphi} = \chi_{\varphi_1} \cap \chi_{\varphi_2}(\mathbf{x}, t) = +1$ then both $\chi_{\varphi_1}(\mathbf{x}, t) = +1$ and $\chi_{\varphi_2}(\mathbf{x}, t) = +1$. From induction hypothesis $\theta_{\varphi_1}^{+}(\mathbf{x}, t) \geq 0$ and $\theta_{\varphi_2}^{+}(\mathbf{x}, t) \geq 0$ and thus, $\theta_{\varphi}^{+}(\mathbf{x}, t) = \theta_{\varphi_1}^{+}(\mathbf{x}, t) \cap \theta_{\varphi_2}^{+}(\mathbf{x}, t) \geq 0$.
- $\varphi = \varphi_1 \mathcal{U} \varphi_2$. From the definition of the characteristic function for Until operator, $\exists t' \in t + I$ such that $\chi_{\varphi_2}(\mathbf{x}, t') = +1$ and $\forall t'' \in [t, t']$, $\chi_{\varphi_1}(\mathbf{x}, t'') = +1$. By the induction hypothesis we obtain that $\theta_{\varphi_1}^{+}(\mathbf{x}, t') \geq 0$ and $\theta_{\varphi_1}^{+}(\mathbf{x}, t'') \geq 0$ and thus we conclude that $\theta_{\varphi}^{+}(\mathbf{x}, t) \geq 0$. This concludes the proof.