Reactive and Risk-Aware Control for Signal Temporal Logic

Lars Lindemann, *Member, IEEE*, George J. Pappas, *Fellow, IEEE*, and Dimos V. Dimarogonas, *Senior Member, IEEE*

***Abstract*—The deployment of autonomous systems in uncertain and dynamic environments has raised fundamental questions. Addressing these is pivotal to build fully autonomous systems and requires a systematic integration of planning and control. We first propose reactive risk signal interval temporal logic (ReRiSITL) as an extension of signal temporal logic (STL) to formulate complex spatiotemporal specifications. Unlike STL, ReRiSITL allows to consider uncontrollable propositions that may model humans as well as random environmental events such as sensor failures. Additionally, ReRiSITL allows to incorporate risk measures, such as (but not limited to) the conditional value-at-risk, to measure the risk of violating certain spatial specifications. Second, we propose an algorithm to check if an ReRiSITL specification is satisfiable. For this purpose, we abstract the ReRiSITL specification into a timed signal transducer and devise a game-based approach. Third, we propose a reactive planning and control framework for dynamical control systems under ReRiSITL specifications.**

***Index Terms*—Formal methods, reactive control, risk-aware control, signal temporal logic (STL), timed automata.**

I. Introduction

TEMPORAL logics allow to express temporal properties in a logical framework providing an expressive specification language. *Signal temporal logic* (STL) is a predicate-based temporal logic that offers many appealing advantages [1]. In particular, STL allows to impose quantitative temporal properties, e.g., combinations of surveillance ("visit regions A, B, and C every 10–60 s"), safety ("always between 5–25 s stay at least 1 m away from D"), and many others. Indeed, there is a rich body of literature on the control of dynamical systems under STL specifications, e.g., [2]–[4].

However, a key obstacle to deploying such control frameworks in real-world settings is to account for uncertain and dynamic environments. In particular, objects of interests may be estimated by simultaneous localization and mapping algorithms and be described as probability distributions, see e.g., [5] and [6], so that one may want to consider risk. Also, random events such as sensor failures or humans requesting assistance play an increasing role. While there has been recent work addressing some of these challenges, e.g., [7]–[10], there exists no reactive and risk-aware planning and control framework with formal correctness guarantees. We claim that no one has rigorously addressed the reactive planning problem for systems under STL specifications. Toward addressing this shortcoming, we leverage ideas from formal methods, risk theory, control theory, game theory, and timed automata theory.

Related Work: For the control under STL specifications, mixed integer linear programs [2], [11], [12] have been presented that encode the STL specification. Nonconvex optimization programs [3], [13] and reinforcement learning approaches [14], [15] have further been proposed and particularly use the quantitative semantics associated with an STL specification [16]. A timed automata-based planning framework has been presented in our previous work [17], where we decompose the STL specification into STL subspecifications. Feedback control laws that implement such STL subspecifications, which are timed transitions, have appeared in [4], [18]–[24].

Linear temporal logic (LTL) is a proposition-based temporal logic, less expressive than STL, that allows to impose qualitative temporal properties. Existing control approaches leverage automata-based synthesis [25]–[27]. Metric interval temporal logic (MITL) is a proposition-based temporal logic with quantitative temporal properties [28], hence more expressive than LTL but less expressive than STL. An MITL specification can be translated into a language equivalent timed automaton [28]. If the accepted language of this automaton is not empty [29], the MITL specification is satisfiable. For pointwise MITL semantics, a tool to perform this translation has been presented in [30]. Pointwise semantics, however, do not guarantee the satisfaction of the MITL specification in continuous time. The procedure of [28], for continuous-time semantics, is complex and not compositional. The results from [31], [32] are more intuitive and present a compositional way to construct a timed signal transducer for an MITL specification. The authors in [33] have proposed a way to control timed automata by reformulating it as a timed two player game, played between controllable (the system) and uncontrollable (environment) events, see also [34]–[36].

Manuscript received 18 December 2020; revised 24 May 2021 and 26 August 2021; accepted 3 October 2021. Date of publication 15 October 2021; date of current version 27 September 2022. This work was supported in part by the Swedish Research Council (VR), in part by the European Research Council (ERC), in part by the Swedish Foundation for Strategic Research (SSF), in part by the Knut and Alice Wallenberg Foundation (KAW), in part by the DARPA Assured Autonomy program, in part by the National Science Foundation (NSF) under Grant CPS 2038873, and in part by the AFOSR under Grant FA9550-19-1-0265. Recommended by Associate Editor C. Mahulea. *(Corresponding author: Lars Lindemann.)*

Lars Lindemann and George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: larsl@seas.upenn.edu; pappasg@seas.upenn.edu).

Dimos V. Dimarogonas is with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 10044 Stockholm, Sweden (e-mail: dimarogonas,dimos@kth.se).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2021.3120681.

Digital Object Identifier 10.1109/TAC.2021.3120681

0018-9286 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF THE MOST IMPORTANT NOTATION USED THROUGHOUT THE ARTICLE

Symbol	Meaning
$\mathcal{F}(\mathcal{X},\mathcal{Y})$	Set of all measurable functions mapping from a set $\mathcal{X}$ into a set $\mathcal{Y}$.
$\boldsymbol{x}$, $\boldsymbol{s}$	The function $\boldsymbol{x}:\mathbb{R}_{\geq 0}\to\mathbb{R}^n$ denotes a deterministic signal, while the element $\boldsymbol{s}\in\mathcal{F}(\mathbb{R}_{\geq 0},\mathbb{B}^{\lvert M^{\text{uc}}\rvert})$ denotes a random signal.
$\boldsymbol{X}$, $\tilde{\boldsymbol{\mu}}$, $\tilde{\Sigma}$	The function $\boldsymbol{X}:\Omega\to\mathbb{R}^{\tilde{n}}$ denotes a random variable with expected value $\tilde{\boldsymbol{\mu}}\in\mathbb{R}^{\tilde{n}}$ and covariance matrix $\tilde{\Sigma}\in\mathbb{R}^{\tilde{n}\times\tilde{n}}$.
h	The function $h:\mathbb{R}^n\times\mathbb{R}^{\tilde{n}}$ denotes predicate functions.
M^{Ri}, M^{uc}, M	M^{Ri}: set of risk predicates, M^{uc}: set of uncontrollable propositions, M: set of risk predicates and uncontrollable propositions.
μ^{Ri}, μ^{uc}	The element $\mu^{\text{Ri}}\in M^{\text{Ri}}$ is a risk predicate, while the element $\mu^{\text{uc}}\in M^{\text{uc}}$ is an uncontrollable proposition.
R, β, γ,	The function $R:\mathcal{F}(\Omega,\mathbb{R})\to\mathbb{R}$ denotes a risk measure, β is a risk level, and γ is a risk threshold.
$(\boldsymbol{x},\boldsymbol{s},\boldsymbol{X},t)\models\phi$	Semantics of an ReRiSITL specification ϕ indicating that $\boldsymbol{x}$, $\boldsymbol{s}$, and $\boldsymbol{X}$ satisfy ϕ at time t.
AP	Set of (atomic) propositions for MITL specifications.
BC	The function BC, e.g., applied as $BC(AP)$, denotes the set of all Boolean combinations over AP.
Tr, Tr^{-1}	The transformation $\varphi=Tr(\phi)$ transforms an ReRiSITL specification ϕ into an MITL specification φ; Tr^{-1} is the inverse.
TST_φ, TST_ϕ	Timed signal transducers for the MITL specification φ and the ReRiSITL specification ϕ.
RA, RA_C, $\overline{RA}_C$	The functions RA, RA_C, $\overline{RA}_C$, e.g., applied as $RA(TST_\phi)$, are different versions of the region automaton of TST_ϕ.
d_p, d_μ	The plan $d_p:\mathbb{R}_{\geq 0}\to BC(AP)$ is constructed for a specification ϕ, d_μ is simply its projection to M via Tr^{-1}.
π, $\hat{\pi}$, W	The functions π and $\hat{\pi}$ are different versions of the controllable predecessor for Algorithm 2 providing the winning condition W.
$\mathfrak{X}_m$, $\mathfrak{X}_m^{\text{EV}}$, $\mathfrak{X}_m^{\text{VaR}}$, $\mathfrak{X}_m^{\text{CVaR}}$	The sets $\mathfrak{X}_m^{\text{EV}}$, $\mathfrak{X}_m^{\text{VaR}}$, $\mathfrak{X}_m^{\text{CVaR}}$ are risk constrained sets that are determinized into the set $\mathfrak{X}_m$.
μ^{det}, M^{det}, $\hat{M}$	The element $\mu^{\text{det}}\in M^{\text{det}}$ is a deterministic predicate; $\hat{M}$ is the set of deterministic predicates and uncontrollable propositions.
TST_θ, TST_θ^{m}	Timed signal transducers for the ReSITL specification θ and the product automaton.

The underlying assumption in these previous works is that the environment is perfectly known. For LTL, this assumption has been relaxed in [5], [6], and [37]. Specifically, the authors in [5] and [6] assumed that the environment is modeled as a semantic map. Target beliefs in surveillance games and Markov decisions process-based approaches are presented in [38] and [39]. Probabilistic computational tree logic and distribution temporal logic [40] account for state distributions and can take chance constraints into account, but only consider qualitative temporal properties and do not consider risk measures [41], [42]. The works in [43] and [44] consider the generalized reactivity (1) fragment, which explicitly accounts for dynamic environments. For STL, the works in [7] and [9] consider chance constraints, whereas [8] and [45] already incorporate risk measures without, however, considering random environmental events. Such events have been considered for STL in [10]. The proposed reactive control strategy in [10] has been evaluated empirically, but without providing formal guarantees. A reactive counter-example-guided framework was proposed in [46] where, however, the risk of violating certain spatial specifications is not considered. Furthermore, only bounded specifications are considered while the STL specification is not allowed to explicitly depend on the environment.

Contributions: In this article, our *first contribution* is to propose reactive risk signal interval temporal logic (ReRiSITL). Compared with STL, ReRiSITL has two distinct features and hence generalizes STL. First, ReRiSITL specifications may contain uncontrollable propositions that allow to model humans, or in general other agents, and environmental events such as sensor failures or communication dropouts. Second, ReRiSITL allows to incorporate risk measures by considering risk predicates so that the risk of violating certain spatial specifications can be taken into account. Such risk predicates can take different risk measures into account, as for instance the conditional value-at-risk (CVaR). Our *second contribution* is an algorithm that allows to check if such an ReRiSITL specification is satisfiable. To do so, we abstract the ReRiSITL specification into a timed signal transducer using and adapting the results from [32] and then following a game-based strategy similarly to [33]. The *third contribution* is a planning and control framework for dynamical control systems under ReRiSITL specifications. The main elements here are a well-defined timed abstraction of the control system that relies on existing feedback control laws as presented in [18]–[24]. We then propose to use a combination of a game-based approach, graph search techniques, and replanning. We remark that our approach is, to the best of our knowledge, the first to incorporate past temporal operators and we hereby establish a connection between monitoring and reactive control.

Structure: Section II presents ReRiSITL and the problem formulation. Section III presents the algorithm to check if an ReRiSITL specification is satisfiable. Sections IV and V propose the planning and control framework for dynamical control systems under ReRiSITL specifications. Simulations and conclusions are provided in Sections VII and VIII.

II. PRELIMINARIES AND PROBLEM FORMULATION

True and false are encoded as $\top:=\infty$ and $\bot:=-\infty$ with $\mathbb{B}:=\{\top,\bot\}$; $\mathbb{R}$, $\mathbb{Q}$, and $\mathbb{N}$ are the real, rational, and natural numbers, respectively, while $\mathbb{R}_{\geq 0}$ ($\mathbb{R}_{>0}$) and $\mathbb{Q}_{\geq 0}$ ($\mathbb{Q}_{>0}$) denote their respective nonnegative (positive) subsets. For $t\in\mathbb{R}_{\geq 0}$ and $I\subseteq\mathbb{R}_{\geq 0}$, let $t\oplus I$ and $t\ominus I$ denote the Minkowski sum and the Minkowski difference of t and I, respectively. For two sets $\mathcal{X}$ and $\mathcal{Y}$, we use the notation $\mathcal{F}(\mathcal{X},\mathcal{Y})$ to denote the set of all measurable functions that map from $\mathcal{X}$ to $\mathcal{Y}$. An element $f\in\mathcal{F}(X,Y)$ is hence a function $f:\mathcal{X}\to\mathcal{Y}$. Let $(\Omega,\mathcal{B}_\Omega,P_\Omega)$ be a probability space, where Ω is the sample space, $\mathcal{B}_\Omega$ is the Borel σ-algebra of Ω, and $P_\Omega:\mathcal{B}_\Omega\to[0,1]$ is a probability measure. A vector of random variables is a measurable function $\boldsymbol{X}:\Omega\to\mathbb{R}^{\tilde{n}}$ defined on a probability space $(\Omega,\mathcal{B}_\Omega,P_\Omega)$, where $\tilde{n}\in\mathbb{N}$. We can associate the probability space $(\mathbb{R}^{\tilde{n}},\mathcal{B}_{\mathbb{R}^{\tilde{n}}},P_{\boldsymbol{X}})$ with $\boldsymbol{X}$, where for $B\in\mathcal{B}_{\mathbb{R}^{\tilde{n}}}$, $P_{\boldsymbol{X}}:\mathcal{B}_{\mathbb{R}^{\tilde{n}}}\to[0,1]$ with $P_{\boldsymbol{X}}(B):=P_\Omega(\boldsymbol{X}^{-1}(B))$ and $\boldsymbol{X}^{-1}(B):=\{\omega\in\Omega\vert\boldsymbol{X}(\omega)\in B\}$. Let $\tilde{\boldsymbol{\mu}}:=EV[\boldsymbol{X}]$ and $\tilde{\Sigma}$ be the expected value and covariance matrix of $\boldsymbol{X}$, respectively, while $\mathcal{N}(\tilde{\boldsymbol{\mu}},\tilde{\Sigma})$ denotes the multivariate normal distribution. We remark that all important symbols that have been or will be introduced in this article are summarized in Table I.

A. Reactive Risk Signal Interval Temporal Logic

To define ReRiSITL, let $h:\mathbb{R}^n\times\mathbb{R}^{\tilde{n}}\to\mathbb{R}$ be a measurable function, referred to as the *predicate function*, where $n,\tilde{n}\in\mathbb{N}$. Let $\boldsymbol{x}:\mathbb{R}_{\geq 0}\to\mathbb{R}^n$ be a deterministic signal and let $\boldsymbol{X}:\Omega\to\mathbb{R}^{\tilde{n}}$ be a vector of random variables defined on the probability

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

space $(\Omega, \mathcal{B}_\Omega, P_\Omega)$.[1] At time t, the probability space $(\mathbb{R}, \mathcal{B}_\mathbb{R}, P_h)$ can be associated with $h(\boldsymbol{x}(t), \boldsymbol{X})$, a random variable, where P_h is derived from the probability space $(\mathbb{R}^{\tilde{n}}, \mathcal{B}_{\mathbb{R}^{\tilde{n}}}, P_{\boldsymbol{X}})$.

We consider *risk predicates* for ReRiSITL based on risk measures as advocated in [41] and [42] toward an axiomatic risk assessment. A *risk measure* $R : \mathcal{F}(\Omega, \mathbb{R}) \to \mathbb{R}$ allows to exclude behavior which is deemed more risky than other behavior. We are interested in $R(-h(\boldsymbol{x}(t), \boldsymbol{X}))$ to argue about the risk of violating $h(\boldsymbol{x}(t), \boldsymbol{X}) \geq 0$. The truth value of a risk predicate $\mu^{\text{Ri}} : \mathbb{R}^n \times \mathbb{R}^{\tilde{n}} \to \mathbb{B}$ at time t is obtained as

$$\mu^{\text{Ri}}(\boldsymbol{x}(t), \boldsymbol{X}) := \begin{cases} \top & \text{if } R(-h(\boldsymbol{x}(t), \boldsymbol{X})) \leq \gamma \\ \bot & \text{otherwise} \end{cases} \tag{1}$$

for a risk threshold $\gamma \in \mathbb{R}$. There are various choices of $R(\cdot)$, see [42] for an overview. We consider the expected value (EV), the VaR, and the CVaR. The expected value of $-h(\boldsymbol{x}(t), \boldsymbol{X})$, denoted by $EV[-h(\boldsymbol{x}(t), \boldsymbol{X})]$, provides a risk neutral risk measure. More risk averse are the VaR and the CVaR as in [41]. The VaR of $-h(\boldsymbol{x}(t), \boldsymbol{X})$ for $\beta \in (0, 1)$ is defined as

$$\text{VaR}_\beta(-h(\boldsymbol{x}(t), \boldsymbol{X})) := \min(d \in \mathbb{R} | P_h(-h(\boldsymbol{x}(t), \boldsymbol{X}) \leq d) \geq \beta)$$

i.e., the worst case $1 - \beta$ probability quantile.

Remark 1: Note that $\text{VaR}_\beta(-h(\boldsymbol{x}(t), \boldsymbol{X})) \leq \gamma$ is equivalent to $P_h(-h(\boldsymbol{x}(t), \boldsymbol{X}) \leq \gamma) \geq \beta$ so that our framework includes chance constraints as for instance used in [9].

The CVaR of $-h(\boldsymbol{x}(t), \boldsymbol{X})$ for a risk level β is given by

$$\text{CVaR}_\beta(-h(\boldsymbol{x}(t), \boldsymbol{X})) := EV[-h(\boldsymbol{x}(t), \boldsymbol{X}))| - h(\boldsymbol{x}(t), \boldsymbol{X})) > VaR_\beta(-h(\boldsymbol{x}(t), \boldsymbol{X}))]$$

i.e., the conditional expected value of $-h(\boldsymbol{x}(t), \boldsymbol{X})$ relative to $-h(\boldsymbol{x}(t), \boldsymbol{X})$ being greater than or equal to the VaR. Let now M^{Ri} denote a set of risk predicates.

Let M^{uc} be a set of *uncontrollable propositions* μ^{uc} and $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|})$ be a random Boolean signal corresponding to the truth values of the propositions in M^{uc} over time.[2] Define also the projection of $\boldsymbol{s}$ onto $\mu^{\text{uc}} \in M^{\text{uc}}$ as $\text{proj}_{\mu^{\text{uc}}}(\boldsymbol{s}) : \mathbb{R}_{\geq 0} \to \mathbb{B}$, i.e., the truth value of μ^{uc} over time.

Let $\mu \in M := M^{\text{Ri}} \cup M^{\text{uc}}$ be a risk predicate or an uncontrollable proposition. The syntax of ReRiSITL is

$$\phi \ ::= \ \top \mid \mu \mid \neg\phi \mid \phi' \wedge \phi'' \mid \phi' U_I \phi'' \mid \phi' \underline{U}_I \phi'' \tag{2}$$

where ϕ' and ϕ'' are ReRiSITL formulas and where U_I and $\underline{U}_I$ are the future and past until operators. We restrict the time interval I to belong to the nonnegative rationals, i.e., $I \subseteq \mathbb{Q}_{\geq 0}$. Additionally, we require that I is not a singleton, i.e., I is not allowed to be of the form $I := [a, a]$ for $a \in \mathbb{Q}_{\geq 0}$. Note that the former assumption is not restrictive, while the latter excludes punctuality constraints. We remark that these assumptions are commonly made [28]. Also define $\phi' \vee \phi'' := \neg(\neg\phi' \wedge \neg\phi'')$ (disjunction), $F_I\phi := \top U_I \phi$ (future eventually), $\underline{F}_I\phi := \top \underline{U}_I \phi$ (past eventually), $G_I\phi := \neg F_I \neg\phi$ (future always), $\underline{G}_I\phi := \neg \underline{F}_I \neg\phi$ (past always). We say that an ReRiSITL formula ϕ is in positive normal form if no negation occurs within

[1]We remark that $\boldsymbol{X}$ can be assumed to be a stochastic process $\boldsymbol{X}(t)$. To avoid further technical complexity, this is not followed in this article.

[2]The proposition μ^{uc} is labeled uncontrollable because $\boldsymbol{s}$ is assumed to be a random signal generated by an unknown underlying stochastic process, as highlighted by the notation $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|})$.

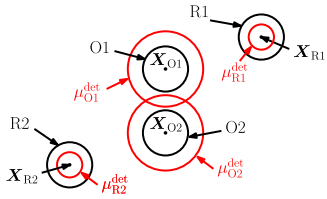


Fig. 1. Overview of the workspace in Example 1.

ϕ [11]. Let $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi$ denote the satisfaction relation as defined next.

Definition 1 (ReRiSITL Semantics): We recursively define the continuous-time semantics of ReRiSITL as

$$\begin{aligned}
(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) &\models \mu^{\text{Ri}} && \text{iff } R(-h(\boldsymbol{x}(t), \boldsymbol{X})) \leq \gamma \\
(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) &\models \mu^{\text{uc}} && \text{iff } \text{proj}_{\mu^{\text{uc}}}(\boldsymbol{s})(t) = \top \\
(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) &\models \neg\phi && \text{iff } \neg((\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi) \\
(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) &\models \phi' \wedge \phi'' && \text{iff } (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi' \\
& && \quad \wedge (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi,'' \\
(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) &\models \phi' U_I \phi'' && \text{iff } \exists t'' \in t \oplus I, \text{ such that} \\
& && \quad (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t'') \models \phi'' \wedge \forall t' \in (t, t''), (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t') \models \phi,' \\
(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) &\models \phi' \underline{U}_I \phi'' && \text{iff } \exists t'' \in t \ominus I, \text{ such that} \\
& && \quad (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t'') \models \phi'' \wedge \forall t' \in (t, t''), (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t') \models \phi'.
\end{aligned}$$

Remark 2: Quantitative semantics can be defined similarly to [16] to determine how well $\boldsymbol{x}$, $\boldsymbol{s}$, and $\boldsymbol{X}$ satisfy ϕ at time t. Here, one has to consider $\gamma - R(-h(\boldsymbol{x}(t), \boldsymbol{X}))$ and $\text{proj}_{\mu^{\text{uc}}}(\boldsymbol{s})(t)$ for risk predicates and uncontrollable propositions, and recursively apply the operations from [16, Def. 10].

Example 1: Consider the workspace in Fig. 1 with regions R1, R2, O1, and O2 described by a normal distribution

$$\boldsymbol{X} := \begin{bmatrix} \boldsymbol{X}_{\text{R1}}^T & \boldsymbol{X}_{\text{R2}}^T & \boldsymbol{X}_{\text{O1}}^T & \boldsymbol{X}_{\text{O2}}^T \end{bmatrix}^T \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\Sigma})$$

with expected value $\tilde{\boldsymbol{\mu}} := [8 \ \ 8 \ \ 2 \ \ 4 \ \ 5 \ \ 7 \ \ 5 \ \ 5]^T$ and covariance $\tilde{\Sigma} := \text{diag}(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$. Consider also the following ReRiSITL specification:

$$\phi := F_{(0,5)} \mu^{\text{Ri}}_{\text{R1}} \wedge G_{[0,\infty)} \left(\mu^{\text{Ri}}_{\text{O1}} \wedge \mu^{\text{Ri}}_{\text{O2}} \wedge \left(\underline{F}_{(0,1)} \mu^{\text{uc}} \Rightarrow F_{(0,3)} \mu^{\text{Ri}}_{\text{R2}} \right) \right)$$

where $\mu^{\text{Ri}}_{\text{R1}}$ and $\mu^{\text{Ri}}_{\text{R2}}$ encode the probability of reaching the regions R1 and R2 using the VaR, $\mu^{\text{Ri}}_{\text{O1}}$ and $\mu^{\text{Ri}}_{\text{O2}}$ encode the risk of colliding with obstacles O1 and O2 using the CVaR, and μ^{uc} is an uncontrollable proposition. The specification ϕ encodes to reach R1 within 5 time units with probability of at least 0.8, while always having a risk of colliding with obstacles O1 and O2 lower than 0. Furthermore, whenever the uncontrollable proposition μ^{uc}, e.g., encoding a human requesting assistance, was true within the last 1 time unit, it should follow that R2 is reached within 3 time units with probability 0.8. We emphasize the use of the past operator $\underline{F}_{(0,1)}$ in ϕ that specifies a form of reactive monitoring. In particular, the predicate functions are

$$h^{\text{Ri}}_{\text{R1}}(\boldsymbol{x}(t), \boldsymbol{X}) := \epsilon - \|\boldsymbol{x}(t) - \boldsymbol{X}_{\text{R1}}\|^2$$

$$h^{\text{Ri}}_{\text{R2}}(\boldsymbol{x}(t), \boldsymbol{X}) := \epsilon - \|\boldsymbol{x}(t) - \boldsymbol{X}_{\text{R2}}\|^2$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

where $\epsilon := 0.5$ and $R_{\mathrm{R1}}(\cdot)$ and $R_{\mathrm{R2}}(\cdot)$ encode the VaR with $\beta_{\mathrm{R1}} = \beta_{\mathrm{R2}} := 0.8$ and $\gamma_{\mathrm{R1}} = \gamma_{\mathrm{R2}} := 0$. Recall that, according to Remark 1, the risk predicate $\mu^{\mathrm{Ri}}_{\mathrm{R1}}$ using VaR encodes the probability that $h^{\mathrm{Ri}}_{\mathrm{R1}}(\boldsymbol{x}(t), \boldsymbol{X}) \geq 0$ is greater than 0.8. Let also

$$h^{\mathrm{Ri}}_{\mathrm{O1}}(\boldsymbol{x}(t), \boldsymbol{X}) := \|\boldsymbol{x}(t) - \boldsymbol{X}_{\mathrm{O1}}\|^2 - \epsilon$$

$$h^{\mathrm{Ri}}_{\mathrm{O2}}(\boldsymbol{x}(t), \boldsymbol{X}) := \|\boldsymbol{x}(t) - \boldsymbol{X}_{\mathrm{O2}}\|^2 - \epsilon$$

where the risk measures $R_{\mathrm{O1}}(\cdot)$ and $R_{\mathrm{O2}}(\cdot)$ encode the CVaR with $\beta_{\mathrm{O1}} = \beta_{\mathrm{O2}} := 0.9$ and $\gamma_{\mathrm{O1}} = \gamma_{\mathrm{O2}} := 0$.

To define satisfiability of an ReRiSITL specification, we need to take into account that propositions in M^{uc} are uncontrollable. We first define what a *nonanticipative strategy* is. A strategy $\boldsymbol{x}_{\mathrm{na}} : \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\mathrm{uc}}|}) \to \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{R}^n)$ is nonanticipative if: For any $t \geq 0$ and for any two signals $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\mathrm{uc}}|})$ with $\boldsymbol{s}(\tau) = \boldsymbol{s}'(\tau)$ for all $\tau \in [0, t]$, it holds that $\boldsymbol{x}_{\mathrm{na}}(\boldsymbol{s})(\tau) = \boldsymbol{x}_{\mathrm{na}}(\boldsymbol{s}')(\tau)$ for all $\tau \in [0, t]$. This means that $\boldsymbol{x}_{\mathrm{na}}(\boldsymbol{s})$ takes, at time t, only current and past values of $\boldsymbol{s}$ into account, i.e., $\boldsymbol{s}(\tau)$, where $\tau \leq t$. This makes sense under the assumption that $\boldsymbol{s}(t)$ can only be observed at time t.

Definition 2 (ReRiSITL Satisfiability): For a given $\boldsymbol{X}$, an ReRiSITL formula ϕ is said to be satisfiable if $\forall \boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\mathrm{uc}}|})$, there exists a nonanticipative strategy $\boldsymbol{x}_{\mathrm{na}} : \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\mathrm{uc}}|}) \to \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{R}^n)$ s.t. $(\boldsymbol{x}_{\mathrm{na}}(\boldsymbol{s}), \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$.

Later in the article, we will replace risk predicates by *deterministic predicates* as originally used in STL. For a given constant $c \in \mathbb{R}$, the truth value of such a deterministic predicate $\mu^{\mathrm{det}} : \mathbb{R}^n \times \mathbb{R}^{\tilde{n}} \to \mathbb{B}$ at time t is obtained as

$$\mu^{\mathrm{det}}(\boldsymbol{x}(t), \tilde{\boldsymbol{\mu}}) := \begin{cases} \top & \text{if } h(\boldsymbol{x}(t), \tilde{\boldsymbol{\mu}}) \geq c \\ \bot & \text{otherwise} \end{cases} \tag{3}$$

where we have replaced $\boldsymbol{X}$ in h by its expected value $\tilde{\boldsymbol{\mu}}$.

If now all risk predicates $\mu^{\mathrm{Ri}} \in M^{\mathrm{Ri}}$ are replaced by deterministic predicates μ^{det}, then ϕ is called a reactive signal interval temporal logic (ReSITL) formula. If uncontrollable propositions μ^{uc} are excluded, i.e., $M^{\mathrm{uc}} = \emptyset$, then ϕ is called a risk signal interval temporal logic (RiSITL) formula. If all risk predicates are replaced by deterministic predicates and $M^{\mathrm{uc}} = \emptyset$, then ϕ reduces to an SITL formula as in [1].

Abbreviation	Features
ReRiSITL	Predicates M^{Ri}, Uncontrollable Propositions M^{uc}
RiSITL	Predicates M^{Ri}
ReSITL	Uncontrollable Propositions M^{uc}
SITL	Deterministic Predicates M^{det} only

B. From MITL to Timed Signal Transducer

We next define MITL [28] which has the advantage that it can be translated into a *timed signal transducer* [32]. We later interpret ReRiSITL formulas as MITL formulas and make use of this translation. Instead of predicates and uncontrollable propositions, MITL considers (controllable) propositions $p \in AP$, where AP is a set of atomic propositions. The MITL syntax is hence

$$\varphi \ ::= \ \top \mid p \mid \neg\varphi \mid \varphi' \wedge \varphi'' \mid \varphi' U_I \varphi'' \mid \varphi' \underline{U}_I \varphi'' \tag{4}$$

where φ' and φ'' are MITL formulas. Let $\boldsymbol{d} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|AP|}$ be a Boolean signal corresponding to truth values of $p \in AP$ over time. Define again the projection of $\boldsymbol{d}$ onto $p \in AP$ as $\mathrm{proj}_p(\boldsymbol{d}) : \mathbb{R}_{\geq 0} \to \mathbb{B}$ and let $(\boldsymbol{d}, t) \models \varphi$ be the satisfaction relation. The continuous-time semantics of an MITL formula [32, Sec. 4] are defined as $(\boldsymbol{d}, t) \models p$ iff $\mathrm{proj}_p(\boldsymbol{d})(t) = \top$ while the other operators are as in Definition 1. An MITL formula φ is satisfiable if $\exists \boldsymbol{d} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|AP|})$, such that $(\boldsymbol{d}, 0) \models \varphi$. Note that the symbols φ and ϕ are used to distinguish between MITL and ReRiSITL formulas, respectively.

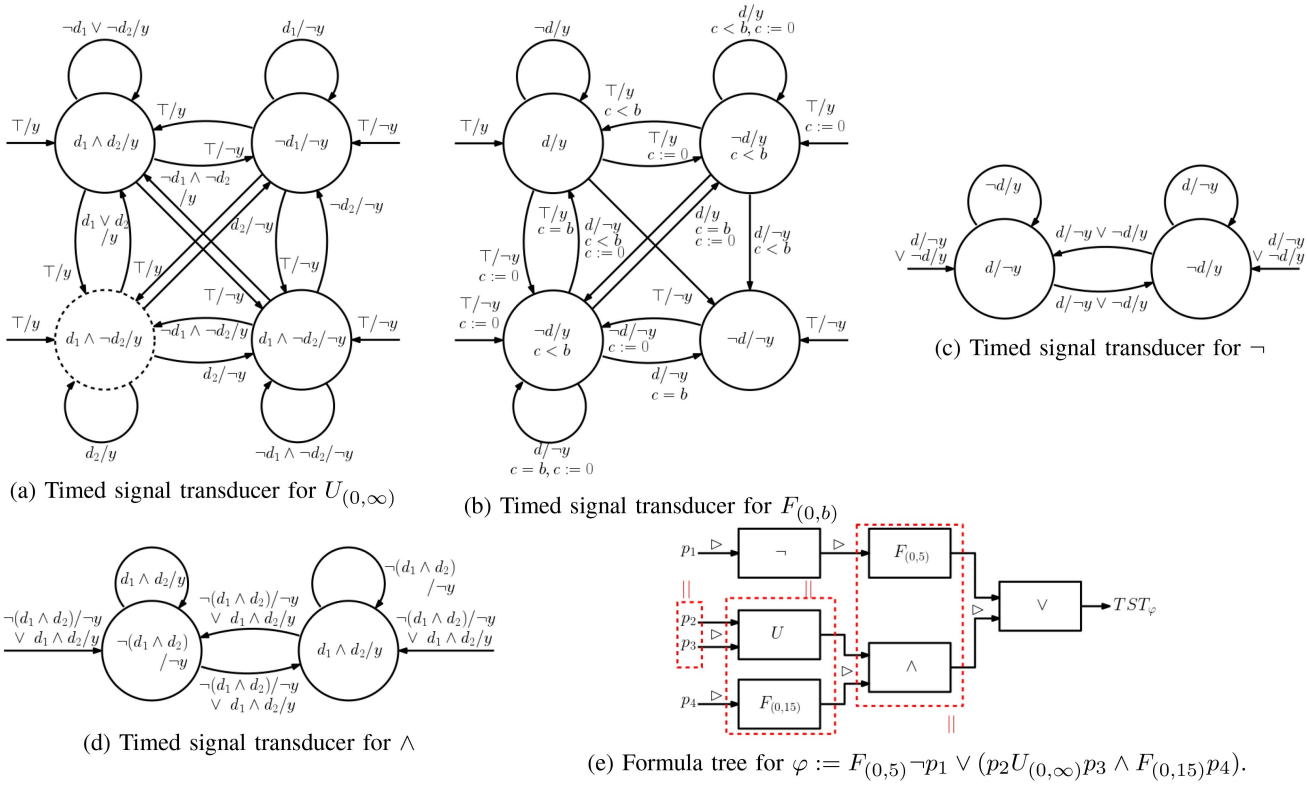
The translation of φ into a timed signal transducer is summarized next [32]. Let $\boldsymbol{c} := [c_1 \quad \dots \quad c_O]^T \in \mathbb{R}^O_{\geq 0}$ be a vector of O clock variables that obey the continuous dynamics $\dot{c}_o(t) := 1$ with $c_o(0) := 0$ for $o \in \{1, \dots, O\}$. Discrete dynamics occur at instantaneous times in form of clock resets. Let $r : \mathbb{R}^O_{\geq 0} \to \mathbb{R}^O_{\geq 0}$ be a reset function, such that $r(\boldsymbol{c}) = \boldsymbol{c}'$ where either $c'_o = c_o$ or $c'_o = 0$. With a slight abuse of notation, we use $r(c_o) = c_o$ and $r(c_0) = 0$. Clocks evolve with time when visiting a state of a timed signal transducer, while clocks may be reset during transitions between states. We define clock constraints as Boolean combinations of conditions of the form $c_o \leq k$ and $c_o \geq k$ for some $k \in \mathbb{Q}_{\geq 0}$. Let $\Phi(\boldsymbol{c})$ denote the set of all clock constraints over clock variables in $\boldsymbol{c}$.

Definition 3 (Timed Signal Transducer [32]): A timed signal transducer is a tuple $\mathrm{TST} := (S, s_0, \Lambda, \Gamma, \boldsymbol{c}, \iota, \Delta, \lambda, \gamma, \mathcal{A})$, where S is a finite set of states, s_0 is the initial state with $s_0 \cap S = \emptyset$, Λ and Γ are a finite sets of input and output variables, respectively, $\iota : S \to \Phi(\boldsymbol{c})$ assigns clock constraints over $\boldsymbol{c}$ to each state, Δ is a transition relation so that $\delta = (s, g, r, s') \in \Delta$ indicates a transition from $s \in S \cup s_0$ to $s' \in S$ satisfying the guard constraint $g \subseteq \Phi(\boldsymbol{c})$ and resetting the clocks according to r; $\lambda : S \cup \Delta \to BC(\Lambda)$ and $\gamma : S \cup \Delta \to BC(\Gamma)$ are input and output labeling functions, where $BC(\Lambda)$ and $BC(\Gamma)$ denote the sets of all Boolean combinations over Λ and Γ, respectively, and $\mathcal{A} \subseteq 2^{S \cup \Delta}$ is a generalized Büchi acceptance condition.

A *run* of a TST over an input signal $\boldsymbol{d} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|\Lambda|}$ is an alternation of time and discrete steps resulting in an output signal $\boldsymbol{y} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|\Gamma|}$. A time step of duration $\tau \in \mathbb{R}_{>0}$ is denoted by $(s, \boldsymbol{c}(t)) \to \tau(s, \boldsymbol{c}(t) + \tau)$ with $\boldsymbol{d}(t + t') \models \lambda(s)$, $\boldsymbol{y}(t + t') \models \gamma(s)$, and $\boldsymbol{c}(t + t') \models \iota(s)$ for each $t' \in (0, \tau)$. A discrete step at time t is denoted by $(s, \boldsymbol{c}(t)) \to \delta(s,' r(\boldsymbol{c}(t)))$ for some transition $\delta = (s, g, r, s') \in \Delta$, such that $\boldsymbol{d}(t) \models \lambda(\delta)$, $\boldsymbol{y}(t) \models \gamma(\delta)$, and $\boldsymbol{c}(t) \models g$. Each run starts with a discrete step from the initial configuration $(s_0, \boldsymbol{c}(0))$. Formally, a run of a TST over $\boldsymbol{d}$ is a sequence $(s_0, \boldsymbol{c}(0)) \to \delta_0(s_1, r_0(\boldsymbol{c}(0))) \to \tau_1(s_1, r_0(\boldsymbol{c}(0)) + \tau_1) \to \delta_1 \dots$. Due to the alternation of time and discrete steps, the signals $\boldsymbol{d}(t)$ and $\boldsymbol{y}(t)$ may be a concatenation of sequences consisting of points and open intervals. We associate a function $q : \mathbb{R}_{\geq 0} \to S \cup \Delta$ with a run as $q(0) := \delta_0$, $q(t) = s_1$ for all $t \in (0, \tau_1), \dots$; $\mathcal{A}$ is a generalized Büchi acceptance condition so that a run over $\boldsymbol{d}(t)$ is *accepting* if, for each $A \in \mathcal{A}$, $\inf(q) \cap A \neq \emptyset$ where $\inf(q)$ contains the states in S that are visited, in q, for an unbounded time duration and transitions in Δ that are taken, in q, infinitely many times. The language of TST is $L(\mathrm{TST}) := \{\boldsymbol{d} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{R}^{|\Lambda|}) \big| \mathrm{TST} \text{ has an accepting run over } \boldsymbol{d}(t)\}$. The synchronous behavior of two timed signal transducers TST_1 and TST_2 is defined by their *synchronous product* $\mathrm{TST}_1 || \mathrm{TST}_2$. The input–output behavior of TST_1 being the input of TST_2 is denoted by their *input–output composition* $\mathrm{TST}_1 \rhd \mathrm{TST}_2$, see [32] and [17, Def. 2 and 3] for definitions.

We can now summarize the procedure of [32]. First, it is shown that every MITL formula φ can be rewritten using only the temporal operators $U_{(0,\infty)}$, $\underline{U}_{(0,\infty)}$, $F_{(0,b)}$, and $\underline{F}_{(0,b)}$ for

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.



(a) Timed signal transducer for $U_{(0,\infty)}$

(b) Timed signal transducer for $F_{(0,b)}$

(c) Timed signal transducer for $\neg$

(d) Timed signal transducer for $\wedge$

(e) Formula tree for $\varphi := F_{(0,5)}\neg p_1 \vee (p_2 U_{(0,\infty)} p_3 \wedge F_{(0,15)} p_4)$.

Fig. 2. Fig. 2(a)–(d) shows timed signal transducers for the basic temporal operators $U_{(0,\infty)}$ and $F_{(0,b)}$ and the Boolean operators $\neg$ and $\wedge$. Note that the variables d, d_1, and d_2 here are used as generic input symbols, while y is a generic output symbol. Fig. 2(e) shows the formula tree for the MITL formula $\varphi := F_{(0,5)}\neg p_1 \vee (p_2 U_{(0,\infty)} p_3 \wedge F_{(0,15)} p_4)$. To construct the timed signal transducer TST_φ for φ from the formula tree, the synchronous product operation $||$ and the input–output composition operation $\triangleright$ need to be applied to the basic timed signal transducers of the blocks in the formula tree as indicated in Fig. 2(e).

rational constants b [32, Proposition 4.5] using the rewriting rules in [32, Lemmas 4.1, 4.2, 4.3, and 4.4]. Second, timed signal transducers for $U_{(0,\infty)}$, $\underline{U}_{(0,\infty)}$, $F_{(0,b)}$, and $\underline{F}_{(0,b)}$ are proposed, see Fig. 2(a) and 2(b) for examples of $\underline{U}_{(0,\infty)}$ and $F_{(0,b)}$. Note that all states and transitions except for the state indicated by the dashed circle in $U_{(0,\infty)}$ are included in $\mathcal{A}$. Timed signal transducers for negations and conjunctions are shown in Fig. 2(c) and (d). Third, the formula tree of an MITL formula φ is constructed as illustrated in Fig. 2(e). Fourth, input–output composition $\triangleright$ and the synchronous product $||$ are used to obtain a timed signal transducer $\text{TST}_\varphi := (S, s_0, \Lambda, \Gamma, \boldsymbol{c}, \iota, \Delta, \lambda, \gamma, \mathcal{A})$ with $\Lambda := AP$ and $\Gamma := \{y\}$; TST_φ has accepting runs over $\boldsymbol{d}$, i.e., $\boldsymbol{d} \in L(\text{TST}_\varphi)$, with $\boldsymbol{y}(0) = \top$ if and only if $(\boldsymbol{d}, 0) \models \varphi$ [32, Th. 6.7]. Note that $\boldsymbol{y}(0) = \top$ (meaning that $\gamma(\delta_0) = y$, where δ_0 is the initial transition) indicates satisfaction of φ at time $t = 0$, while $\boldsymbol{y}(0) = \bot$, i.e., $\gamma(\delta_0) = \neg y$, indicates $(\boldsymbol{d}, 0) \not\models \varphi$.

C. Problem Definition

The first problem is a verification problem to check the satisfiability of an ReRiSITL formula ϕ according to Definition 2.

Problem 1: Given a random variable $\boldsymbol{X}$ and an ReRiSITL formula ϕ as in (2), check whether or not ϕ is satisfiable.

The second problem is a control problem. Let the system

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t)) + g(\boldsymbol{x}(t))\boldsymbol{u},\ \boldsymbol{x}(0) := \boldsymbol{x}_0 \tag{5}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous and where $\boldsymbol{u} \in \mathbb{R}^m$ is a control law.

In this context, $\boldsymbol{X}$ and M^{uc} may model the environment in which the system in (5) operates, e.g., regions of interest and sensor failures can be modeled by $\boldsymbol{X}$ and M^{uc}, respectively. Let now each $\mu_m \in M^{\text{Ri}}$ with $m \in \{1, \ldots, |M^{\text{Ri}}|\}$ be associated with predicate functions $h_m : \mathbb{R}^n \times \mathbb{R}^{\tilde{n}} \to \mathbb{R}$ and risk parameters $R_m(\cdot)$, β_m, and γ_m. For $\mu^{\text{uc}} \in M^{\text{uc}}$, let the truth value of μ^{uc} at time $t \in \mathbb{R}_{\geq 0}$ be captured by $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|})$, i.e., we observe $\text{proj}_{\mu^{\text{uc}}}(\boldsymbol{s})(t)$. Since $\boldsymbol{s}$ is not known beforehand, we assume to observe $\boldsymbol{s}(t)$ at time t.

Problem 2: Given a random variable $\boldsymbol{X}$ and a satisfiable ReRiSITL formula ϕ as in (2), find a nonanticipative strategy $\boldsymbol{u}(\boldsymbol{x}(t), \boldsymbol{s}, t)$ s.t. $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$, where $\boldsymbol{x}$ is the solution to (5) under $\boldsymbol{u}(\boldsymbol{x}(t), \boldsymbol{s}, t)$ and where $\boldsymbol{s}(t)$ is observed at time t.

The next assumption is not explicitly used and needed for our proposed solutions to Problems 1 and 2. We will, however, refer to this assumption in some places to emphasize that computational advantages can be obtained under it.

Assumption 1: The functions $h_m : \mathbb{R}^n \times \mathbb{R}^{\tilde{n}} \to \mathbb{R}$ are linear in its first argument.

III. Satisfiability of ReRiSITL Specifications

In this section, we present a solution to Problem 1. In Sections III-A and III-B, we construct a timed signal transducer TST_ϕ that characterizes all signals $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

$\boldsymbol{s} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\text{uc}}|}$, such that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$. In Section III-C, we consider if, for all $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|})$, there exists a nonanticipative strategy $\boldsymbol{x}_{\text{na}} : \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|}) \to \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{R}^n)$, such that $(\boldsymbol{x}_{\text{na}}(\boldsymbol{s}), \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$, solving Problem 1.

A. From ReRiSITL to Timed Signal Transducer

The first goal is to abstract the ReRiSITL formula ϕ into an MITL formula φ via a transformation $Tr(\cdot)$. Therefore, let us use the notation $\phi(M)$ to make explicit that the ReRiSITL formula ϕ depends on the set of predicates and propositions M. The transformation $Tr(\cdot)$ essentially replaces predicates and uncontrollable propositions M in $\phi(M)$ by a set of propositions AP. For $i \in \{1, \ldots, |M|\}$, associate with each $\mu_i \in M$ a proposition p_i and let $AP := \{p_1, \ldots, p_{|M|}\}$. Let then $\varphi := Tr(\phi(M)) = \phi(AP)$, e.g., $\phi(M) := F_I(\mu_1 \wedge \mu_2)$ becomes $\varphi := \phi(AP) = F_I(p_1 \wedge p_2)$. Let the inverse $Tr^{-1}(\varphi) = Tr^{-1}(Tr(\phi(M))) = \phi(M)$ be obtained by replacing each $p_i \in AP$ in φ with the corresponding $\mu_i \in M$.

Let now $\text{TST}_\varphi := (S, s_0, \Lambda, \Gamma, \boldsymbol{c}, \iota, \Delta, \lambda, \gamma, \mathcal{A})$ be constructed for the MITL formula φ according to Section II-B with $\Lambda := AP$. Since we aim at satisfying the STL formula ϕ, we modify TST_φ by the following operations to account for the error induced by the abstraction from ϕ to φ via Tr.

- [O1] Remove each state $s \in S$ for which there exists no $\boldsymbol{x} \in \mathbb{R}^n$ and no $\boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}$ so that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(s))$.[3] Remove the corresponding s from $\mathcal{A}$. Further remove the corresponding ingoing ($(s,' g, r, s) \in \Delta$ for some $s' \in S$) and outgoing ($(s, g, r, s') \in \Delta$ for some $s' \in S$) transitions.
- [O2] Remove each transition $\delta := (s, g, r, s') \in \Delta$ for which there exists no $\boldsymbol{x} \in \mathbb{R}^n$ and no $\boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}$ so that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(\delta))$. Remove the corresponding δ from $\mathcal{A}$.

The modified TST_φ is denoted by $\text{TST}_\phi := (S^\phi, s_0, \Lambda, \Gamma, \boldsymbol{c}, \iota, \Delta^\phi, \lambda, \gamma, \mathcal{A}^\phi)$ for which naturally $S^\phi \subseteq S$, $\Delta^\phi \subseteq \Delta$, and $\mathcal{A}^\phi \subseteq \mathcal{A}$. Note that it is essential to be able to check if there exists $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}$, such that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(s))$ and $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(\delta))$ in [O1] and [O2], respectively. To do so, techniques as in [47] and summarized in [48, Ch. 2], resulting in nonlinear mixed integer programs, can be employed. Nonlinearity here is in particular induced due to $R(\cdot)$. To address Problem 2 (which will also rely on operations [O1] and [O2]), addressed in Sections IV and V, we will obtain computationally more efficient mixed integer linear programs if Assumption 1 holds.

B. Satisfiability of RiSITL Specifications

To characterize all signals $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\boldsymbol{s} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\text{uc}}|}$ so that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$, we translate TST_ϕ of the previous subsection, which is in essence a timed automaton when removing the output labels, to a region automaton $RA(\text{TST}_\phi)$ [29];[4] $RA(\text{TST}_\phi)$ can be used to check emptiness of TST_ϕ, i.e., to analyze reachability properties of TST_ϕ. Since TST_ϕ has invariants on states $\iota(s)$ and guards g included in transitions $(s, g, r, s') \in \Delta^\phi$, we have to slightly modify the algorithms presented in [29] and [28]. Therefore, we associate a transition relation $\Rightarrow$ over the extended state space $S^\phi \times \mathbb{R}^O_{\geq 0}$.

Definition 4 (Equivalent Transition System of TST_ϕ): Let $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$ be a transition system with $(s, \boldsymbol{c}) \Rightarrow \delta(s,' \boldsymbol{c}')$ if and only if there exist $t' \in \mathbb{R}_{\geq 0}$ and $\delta := (s, g, r, s') \in \Delta^\phi$ so that

1) for all $\tau \in (0, t')$, $\boldsymbol{c} + \tau \models \iota(s)$;
2) it holds that $\boldsymbol{c}' = r(\boldsymbol{c} + t')$ and $\boldsymbol{c} + t' \models g$,

i.e., a combination of time and discrete transitions.

Reachability properties of the infinite state transition system $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$ (and hence of TST_ϕ) can now be analyzed by its finite state region automaton $RA(\text{TST}_\phi)$ that relies on a bisimulation relation $\sim \subseteq \mathbb{R}^O_{\geq 0} \times \mathbb{R}^O_{\geq 0}$ resulting in clock regions. In fact, a clock region is an equivalence class induced by $\sim$. Details are omitted and see [29] for details on the bisimulation $\sim$ and on clock regions. Let α and α' be clock regions and assume $\boldsymbol{c} \in \alpha$ and $\boldsymbol{c}' \in \alpha'$. If $(s, \boldsymbol{c}) \Rightarrow \delta(s,' \boldsymbol{c}')$ and $\boldsymbol{c} \sim \bar{\boldsymbol{c}}$ for some $\bar{\boldsymbol{c}}$, then it holds that there is a $\bar{\boldsymbol{c}}'$ with $\boldsymbol{c}' \sim \bar{\boldsymbol{c}}'$ so that $(s, \bar{\boldsymbol{c}}) \Rightarrow \delta(s,' \bar{\boldsymbol{c}}')$.

Definition 5 (Region Automaton of TST_ϕ): The region automaton $RA(\text{TST}_\phi) := (Q, q_0, \Delta_R, \mathcal{A}_R)$ is the quotient system of $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$ using clock regions as equivalence classes and defined as follows.

1) The states are $q := (s, \alpha)$, where $s \in S^\phi$ and $\alpha \in A$ where A is the set of all clock regions so that $Q := S^\phi \times A$.
2) The initial states are $q_0 := (s_0, \alpha_0) \in Q$, where α_0 is the clock region corresponding to $\boldsymbol{c}(0)$.
3) For $q := (s, \alpha)$ and $q' := (s,' \alpha')$, there is a transition $(q, \delta, q') \in \Delta_R$ if and only if there is a transition $(s, \boldsymbol{c}) \Rightarrow \delta(s,' \boldsymbol{c}')$ for $\boldsymbol{c} \in \alpha$ and $\boldsymbol{c}' \in \alpha'$.
4) $q = (s, \alpha) \in \mathcal{A}_R(i)$ if $s \in \mathcal{A}^\phi(i)$.

Using standard graph search techniques such as the memory efficient variant of the nested depth first search [49], here adapted to deal with the generalized Büchi acceptance condition as in [50], we may obtain, if existent, and accepting sequence $\mathfrak{q} = (q_0, q_1, \ldots)$ with $q_j := (s_j, \alpha_j)$ and $(q_j, \delta_j, q_{j+1}) \in \Delta_R$ for each $j \in \mathbb{N}$ satisfying the generalized Büchi acceptance condition $\mathcal{A}_R$. In particular, $\mathfrak{q} := (\mathfrak{q}_p, \mathfrak{q}_s^\omega)$ consists of a prefix of length $p+1$ and a suffix of length s, here denoted by $\mathfrak{q}_p := (q_0, \ldots, q_p)$ and $\mathfrak{q}_s := (q_{p+1}, \ldots, q_{p+s})$. Furthermore, we require that $\gamma(\delta_0) = y$ to indicate that we want $(\boldsymbol{d}, 0) \models \varphi$. We next add timings $\bar{\tau} := (\bar{\tau}_p, \bar{\tau}_s^\omega)$ to $\mathfrak{q}$ with $\bar{\tau}_p := (\tau_0 := 0, \ldots, \tau_p)$ and $\bar{\tau}_s := (\tau_{p+1}, \ldots, \tau_{p+s})$, where $\tau_j \in \mathbb{R}_{>0}$ for $j \geq 1$ corresponds to the occurence of δ_j, which happens τ_j time units after the occurrence of δ_{j-1}. We have presented a way to find such $\bar{\tau}$ in [45, Sec. III.C].

By denoting $T_j := \sum_{k=0}^{j} \tau_j$, $\mathfrak{q}$ and $\bar{\tau}$ can be associated with a *plan* given by

$$d_p(t) := \begin{cases} \lambda(\delta_j) & \text{if } t = T_j \\ \lambda(s_j) & \text{if } T_j < t < T_{j+1}. \end{cases} \tag{6}$$

The intuition of a plan $d_p : \mathbb{R}_{\geq 0} \to BC(AP)$ is as follows: A signal $\boldsymbol{d} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|AP|}$ that satisfies the plan d_p also satisfies the MITL specification φ at time $t = 0$, i.e., $\boldsymbol{d}(t) \models d_p(t)$ for all $t \geq 0$ implies that $(d, 0) \models \varphi$.

[3] We use $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(s))$ with a slight abuse of notation instead of $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models Tr^{-1}(\lambda(s))$ since $Tr^{-1}(\lambda(s))$ is a Boolean formula.

[4] We could equivalently use the computationally efficient zone automaton, which is avoided here to keep the discussion in the remainder simple.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

Lemma 1: Given a signal $\boldsymbol{d}: \mathbb{R}_{\geq 0} \to \mathbb{B}^{|AP|}$, there is an accepting run of TST_ϕ over $\boldsymbol{d}(t)$ and $(\boldsymbol{d}, 0) \models \varphi$ if only if there exists a plan $d_p(t)$ so that $\boldsymbol{d}(t) \models d_p(t)$ for all $t \in \mathbb{R}_{\geq 0}$.

Proof: $\Rightarrow$: Departing from TST_ϕ, the infinite state transition system $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$ has, by construction, the same reachable set as TST_ϕ, i.e., the same reachable configurations $(s_0, \boldsymbol{c}(0)), (s_0, r(\boldsymbol{c}(0))), (s_1, r(\boldsymbol{c}(0)) + \tau_1), \ldots$. Since $\sim$ is a bisimulation relation, reachability properties of TST_ϕ can then equivalently be analyzed by considering the finite state transition system $RA(\text{TST}_\phi)$ [29, Lemma 4.13]. If there hence exists an accepting run of TST_ϕ over $\boldsymbol{d}(t)$ and $(\boldsymbol{d}, 0) \models \varphi$, i.e., $\gamma(\delta_0) = y$, the plan $d_p(t)$ can be constructed as described above by obtaining $\mathfrak{q}$ and $\bar{\tau}$ directly from the accepting run of TST_ϕ over $\boldsymbol{d}(t)$. It will, by construction, hold that $\boldsymbol{d}(t) \models d_p(t)$ for all $t \in \mathbb{R}_{\geq 0}$.

$\Leftarrow$: If there exists a plan $d_p(t)$ so that $\boldsymbol{d}(t) \models d_p(t)$ for all $t \in \mathbb{R}_{\geq 0}$, then it follows that TST_ϕ has an accepting run over $\boldsymbol{d}(t)$. This follows by construction of $d_p(t)$, where $\mathfrak{q}$ and $\bar{\tau}$ have been obtained based on $RA(\text{TST}_\phi)$ (as described for the synthesis of $d_p(t)$) and by the bisimulation relation $\sim$. Removing states and transitions from TST_φ according to operations [O1] and [O2] resulting in TST_ϕ only removes behavior from TST_φ (not adding additional behavior), i.e., $L(\text{TST}_\phi) \subseteq L(\text{TST}_\varphi)$, so that, by [32, Th. 6.7], an accepting run of TST_ϕ over $\boldsymbol{d}(t)$ inducing $\boldsymbol{y}(0) = \top$ results in $(\boldsymbol{d}, 0) \models \varphi$.

Note that there may exist an accepting run of TST_φ over $\boldsymbol{d}(t)$ so that $(\boldsymbol{d}, 0) \models \varphi$, while there exists no accepting run of TST_ϕ over $\boldsymbol{d}(t)$ due to operations [O1] and [O2]. We can now associate $d_\mu: \mathbb{R}_{\geq 0} \to BC(M)$ with $d_p(t)$ as

$$d_\mu(t) := Tr^{-1}(d_p(t))$$

and, based on ϕ, state under which conditions $d_p(t)$ exists.

Theorem 1: There exists a plan $d_p(t)$ (and hence a plan $d_\mu(t)$) if and only if there exists $\boldsymbol{x}: \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\boldsymbol{s}: \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\text{uc}}|}$ so that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$.

Proof: $\Rightarrow$: The existence of a plan $d_p(t)$ implies, by Lemma 1, that a signal $\boldsymbol{d}: \mathbb{R}_{\geq 0} \to \mathbb{B}^{|AP|}$ with $\boldsymbol{d}(t) \in d_p(t)$ for all $t \in \mathbb{R}_{\geq 0}$ is such that $(\boldsymbol{d}, 0) \models \varphi$. Operations [O1] and [O2] remove all states s and transitions δ from TST_φ that are infeasible, i.e., for which there exists no $\boldsymbol{x} \in \mathbb{R}^n$ and no $\boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}$ such that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(s))$ and $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(\delta))$, respectively. Recall that the only difference between the semantics of ϕ and φ is the difference in μ_i and p_i, respectively. It follows that, based on the run of TST_φ over $\boldsymbol{d}(t)$, we can construct a signal $\boldsymbol{x}: \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\boldsymbol{s}: \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\text{uc}}|}$ with $(\boldsymbol{x}(t), \boldsymbol{s}(t), \boldsymbol{X}) \models d_\mu(t)$ for all $t \in \mathbb{R}_{\geq 0}$ implying that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$.

$\Leftarrow$: Based on $\boldsymbol{x}(t)$ and $\boldsymbol{s}(t)$, define the signal $\boldsymbol{d}(t) := \begin{bmatrix} h_1^\top(\boldsymbol{x}(t)) & \ldots & h_{|M^{\text{c}}|}^\top(\boldsymbol{x}(t)) & \boldsymbol{s}(t)^T \end{bmatrix}^T$, where $h_m^\top(\boldsymbol{x}) := \top$ if $R_m(h_m(\boldsymbol{x}, \boldsymbol{X})) \leq \gamma_m$ and $h_i^\top(\boldsymbol{x}) := \bot$ otherwise and that is such that $(\boldsymbol{d}, 0) \models \varphi$. Note that $h_m(\boldsymbol{x}, \tilde{\boldsymbol{\mu}})$ is the predicate function associated with μ_m. It follows that $\boldsymbol{d}$ induces an accepting run of TST_ϕ over $\boldsymbol{d}(t)$ since the traversed states and transitions during this run have not been removed by operations [O1] and [O2]. By Lemma 1, it follows that there hence exists a plan $d_p(t)$.

The next two results are straightforward consequences.

Corollary 1: If $\boldsymbol{x}: \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\boldsymbol{s}: \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\text{uc}}|}$ are so that $(\boldsymbol{x}(t), \boldsymbol{s}(t), \boldsymbol{X}) \models d_\mu(t)$ for all $t \in \mathbb{R}_{\geq 0}$, then it follows that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$.

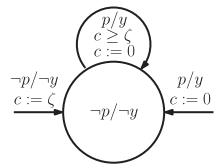


Fig. 3. Timed signal transducer for uncontrollable propositions according to Assumption 2.

Algorithm 1: Algorithm to Check if ϕ is Satisfiable.

1: Obtain the MITL formula $\varphi := Tr(\phi)$.
2: Obtain TST_φ according to Section II-B and where uncontrollable propositions $p_i \in AP$, i.e., p_i with $Tr^{-1}(p_i) \in M \cap M^{\text{uc}}$, are modeled as in Fig. 3.
3: Perform [O1] and [O2] to obtain TST_ϕ.
4: Modify TST_ϕ to avoid Zeno behavior.
5: Translate TST_ϕ into $RA_C(\text{TST}_\phi)$.
6: Translate $RA_C(\text{TST}_\phi)$ into $\overline{RA}_C(\text{TST}_\phi)$.
7: Run Algorithm 2 to obtain W.
8: Check if the conditions in Theorem 2 are satisfied.

Corollary 2: If $M^{\text{uc}} = \emptyset$, i.e., ϕ is an RiSITL formula, then it holds that there exists a plan $d_p(t)$ (and hence a plan $d_\mu(t)$) if and only if ϕ is satisfiable.

C. Satisfiability of ReRiSITL Specifications

The previous results can only be used to check satisfiability of RiSITL. For ReRiSITL specifications ϕ, this requires to check all $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|})$ as in Definition 2. Let us define

$$\boldsymbol{s}^\perp := \begin{bmatrix} \bot & \ldots & \bot \end{bmatrix}^T \in \mathbb{B}^{|M^{\text{uc}}|}$$

and additionally impose the following assumption that all signals $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\text{uc}}|})$ have to satisfy.

Assumption 2: Assume that $\boldsymbol{s}(t) = \boldsymbol{s}^\perp$ for all times except on a set of measure zero, i.e., $\boldsymbol{s}(t) \neq \boldsymbol{s}^\perp$ only for a countable set of times t. There exists a known lower bound $\zeta > 0$ between events $\boldsymbol{s}(t) \neq \boldsymbol{s}^\perp$, i.e., for $\boldsymbol{s}(t') = \boldsymbol{s}(t'') \neq \boldsymbol{s}^\perp$ with $t' \neq t''$, it holds that $|t'' - t'| \geq \zeta$.

Assumption 2 excludes signals $\boldsymbol{s}(t)$ exhibiting Zeno behavior, i.e., infinite changes of $\boldsymbol{s}(t)$ in finite time, and is realistic in the sense that it allows to model instantaneous error signals such as considered for communication dropouts or sensor failures. Assumption 2 is in particular necessary for a game-based approach, see [33]. Furthermore, Assumption 2 is necessary for the replanning procedure in Section V-B.

In Algorithm 1, presented below and explained in the remainder, we summarize the steps to check if ϕ is satisfiable. Line 1 in Algorithm 1 has already been explained, while line 2 is related to Assumption 2. In particular, to model uncontrollable propositions $\mu^{\text{uc}} \in M^{\text{uc}}$ according to Assumption 2, we consider the timed signal transducer in Fig. 3. When constructing TST_φ, we hence model each $p \in AP$ with $\mu^{\text{uc}} = Tr^{-1}(p) \in M^{\text{uc}}$ as in Fig. 3. Line 3 in Algorithm 1 then performs [O1] and [O2] to obtain TST_ϕ.

Within the presented game-based approach, it needs to be ensured that no player (here the two players are the controllable and uncontrollable signals $\boldsymbol{x}$ and $\boldsymbol{s}$) wins by inducing Zeno behavior (see [33] for more intuition). A generic way of avoiding

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

Zeno behavior is to add a clock c to TST_ϕ and add, to each transition, the constraint $c \geq \epsilon$ for a small constant $\epsilon \in \mathbb{Q}_{>0}$ and the reset function $r(c) := 0$. This modification will affect the completeness, but not the soundness of the proposed approach. There are minimally invasive algorithms on how to avoid Zeno behavior, for instance as in [51]. This modification of TST_ϕ is stated in line 4 in Algorithm 1.

Recall from Section II-B that an accepting run in TST_ϕ needs to satisfy the generalized Büchi acceptance condition which implies having infinite length, i.e., the run is not allowed to stop existing. The latter is necessary since we require to be able to extend each finite run in TST_ϕ to an infinite run. Specifically, note that within a state $s \in S^\phi$ in TST_ϕ it may happen that, for some $\boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}$, there exists no $\boldsymbol{x} \in \mathbb{R}^n$, such that a transition can be taken, i.e., there exists no $\delta' := (s, g,' r,' s'') \in \Delta^\phi$, such that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(\delta'))$. This means that there is no continuation of a finite run entering the state s so that the run is not accepting. For instance, in Fig. 2(b) in the bottom right state there exists no transition for $\text{proj}_d(\boldsymbol{d}) = \top$. To account for this, we first modify the infinite state transition system $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$ to $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow_C)$ by separating time and discrete transitions.

Definition 6 (Equivalent Transition System of TST_ϕ): Let $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow_C)$ be a transition system, where $(s, \boldsymbol{c}) \overset{\delta_t}{\Rightarrow}_C (s,' \boldsymbol{c}')$ with $\delta_t \in \{\delta, t\}$ if there is either a *discrete or a time transition* as follows:

1) there is a *discrete transition* $(s, \boldsymbol{c}) \overset{\delta}{\Rightarrow}_C (s,' \boldsymbol{c}')$ if there exists $\delta := (s, g, r, s') \in \Delta^\phi$ so that $\boldsymbol{c}' = r(\boldsymbol{c})$ and $\boldsymbol{c} \models g$;
2) there is a *time transition* $(s, \boldsymbol{c}) \overset{t}{\Rightarrow}_C (s, \boldsymbol{c}')$ if, for all $\tau \in (0, t)$, $\boldsymbol{c} + \tau \models \iota(s)$.

We emphasize that $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow_C)$, $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$, and hence TST_ϕ have the same reachability properties. Let now $RA_C(\text{TST}_\phi) := (Q, q_0, \Delta_R, \mathcal{A}_R)$ denote the region automaton, similar to Definition 5, but now obtained from $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow_C)$ instead of $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$. The translation to $RA_C(\text{TST}_\phi)$ corresponds to line 5 in Algorithm 1.

Definition 7 (Region Automaton of TST_ϕ): The region automaton $RA_C(\text{TST}_\phi) := (Q, q_0, \Delta_R, \mathcal{A}_R)$ is defined as follows.

1) The states are $q := (s, \alpha)$, where $s \in S^\phi$ and $\alpha \in A$ where A is the set of all clock regions so that $Q := S^\phi \times A$.
2) The initial states are $q_0 := (s_0, \alpha_0) \in Q$, where α_0 is the clock region corresponding to $\boldsymbol{c}(0)$.
3) For $q := (s, \alpha)$ and $q' := (s,' \alpha')$, there is a transition $(q, \delta_t, q') \in \Delta_R$, where $\delta_t \in \{\delta, t\}$ if there is
   a) either a discrete transition $(s, \boldsymbol{c}) \overset{\delta}{\Rightarrow}_C (s,' \boldsymbol{c}')$ for $\boldsymbol{c} \in \alpha$ and $\boldsymbol{c}' \in \alpha'$;
   b) or a time transition $(s, \boldsymbol{c}) \overset{t}{\Rightarrow}_C (s,' \boldsymbol{c}')$ for $\boldsymbol{c} \in \alpha$ and $\boldsymbol{c}' \in \alpha'$, where α' is the immediate time successor of α[5].
4) $q = (s, \alpha) \in \mathcal{A}_R(i)$ if $s \in \mathcal{A}^\phi(i)$.

Remark 3: Defining $RA_C(\text{TST}_\phi)$ based on $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow_C)$ by separating discrete and time transitions, and unrolling the time domain as in Definition 7, results in more states compared to $RA(\text{TST}_\phi)$ based on $(S^\phi \times \mathbb{R}^O_{\geq 0}, \Rightarrow)$. This, however, now becomes necessary since uncontrollable signals $\boldsymbol{s}$ may cause undesirable behavior at all times.

[5]See [29, Def. 4.6] for the definition of a time successor. By an "immediate" time successor, we mean that the regions α and α' are connected.

Algorithm 2: Calculation of the Winning Set W.

Input: $\overline{RA}_C(\text{TST}_\phi)$ and $\pi : 2^{\overline{Q}} \to 2^{\overline{Q}}$
Output: W
1: $W_0 := \overline{Q}$
2: **for** $i := 0, 1, \ldots$ until $W_{i+1} = W_i$ **do**
3: $\quad H_0 := \emptyset$
4: $\quad$ **for** $j := 0, 1, \ldots$ until $H_{j+1} = H_j$ **do**
5: $\quad\quad H_{j+1} := \pi(H_j) \cup (\overline{\mathcal{A}}_R \cap \pi(W_i))$
6: $\quad W_{i+1} := H_j$
7: $W := W_i$

Translate now $RA_C(\text{TST}_\phi)$, which is a finite automaton with *generalized* Büchi acceptance condition, into an equivalent finite automaton $\overline{RA}_C(\text{TST}_\phi) := (\overline{Q}, \overline{q}_0, \overline{\Delta}_R, \overline{\mathcal{A}}_R)$ with a Büchi acceptance condition instead, as follows.

1) $\overline{Q} := Q \times \{1, \ldots, |\mathcal{A}_R|\}$.
2) $\overline{q}_0 := (q_0, 1)$.
3) $\overline{\Delta}_R := \{((q, i), \delta_t, (q,' j)) | (q, \delta_t, q') \in \Delta_R$ and if $q \in \mathcal{A}_R(i)$, then $j = ((i+1) \mod |\mathcal{A}_R| + 1)$ else $j = i\}$, where $\mathcal{A}_R(i)$ denotes the ith element of $\mathcal{A}_R$.
4) $\overline{\mathcal{A}}_R := (\mathcal{A}_R(1), 1)$.

In particular, the difference is that $\mathcal{A}_R$ consists of several sets $\mathcal{A}_R(i)$ of states, while $\overline{\mathcal{A}}_R$ is a single set of states. By construction, the accepting behavior of $RA_C(\text{TST}_\phi)$ and $\overline{RA}_C(\text{TST}_\phi)$ are the same. This translation corresponds to line 6 in Algorithm 1 and is performed to obtain a simpler acceptance condition that can be expressed as a fixed point expression as we will see below. In fact, a winning condition (for a game played between $\boldsymbol{s}$ and $\boldsymbol{x}$) is that always eventually $\overline{\mathcal{A}}_R$ can be visited by each finite run of $\overline{RA}_C(\text{TST}_\phi)$.

Remark 4: The translation to $\overline{RA}_C(\text{TST}_\phi)$ may induce $|Q| \cdot |\mathcal{A}_R|$ states. One can avoid such a state explosion by neglecting the acceptance condition $\mathcal{A}$ for all timed signal transducers in Fig. 2 except for the until operator in Fig. 2(a), where a Büchi acceptance condition is needed.

In the remainder, we are inspired by the work in [33]. We first introduce the main operator, the controllable predecessor $\pi : 2^{\overline{Q}} \to 2^{\overline{Q}}$. For a certain set $W \subseteq \overline{Q}$, define

$$\pi(W) := \{\overline{q} \in \overline{Q} | \forall \boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}, \exists (\overline{q}, \delta, \overline{q}') \in \overline{\Delta}_R \text{ s.t. } 1)\ \overline{q}' \in W,\ 2)\ \exists \boldsymbol{x} \in \mathbb{R}^n \text{ s.t. } (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}) \models Tr^{-1}(\lambda(\delta))\}.$$

The intuition is that states in $\pi(W)$ will always allow to enforce a transition into W by a suitable $\boldsymbol{x}$ in one step, no matter of the value of $\boldsymbol{s}$. We next present Algorithm 2 to obtain the set W from which we can force to always eventually be within $\overline{\mathcal{A}}_R$. Algorithm 2, called in line 7 in Algorithm 1, differs from the algorithm presented in [33] by the definition of the controllable predecessor $\pi : 2^{\overline{Q}} \to 2^{\overline{Q}}$.

The algorithm starts with $W_0 := \overline{Q}$ (line 1). For this W_0, the inner loop (lines 3–5) calculates all states H_j from which states in $\overline{\mathcal{A}}_R \cap \pi(W_0)$ can be reached, i.e., states in $\overline{\mathcal{A}}_R$ that can be reached and are no deadlock states. For $W_1 := H_j$ (line 6), this inner loop is repeated until eventually obtaining the set of states W that can always eventually be reached.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

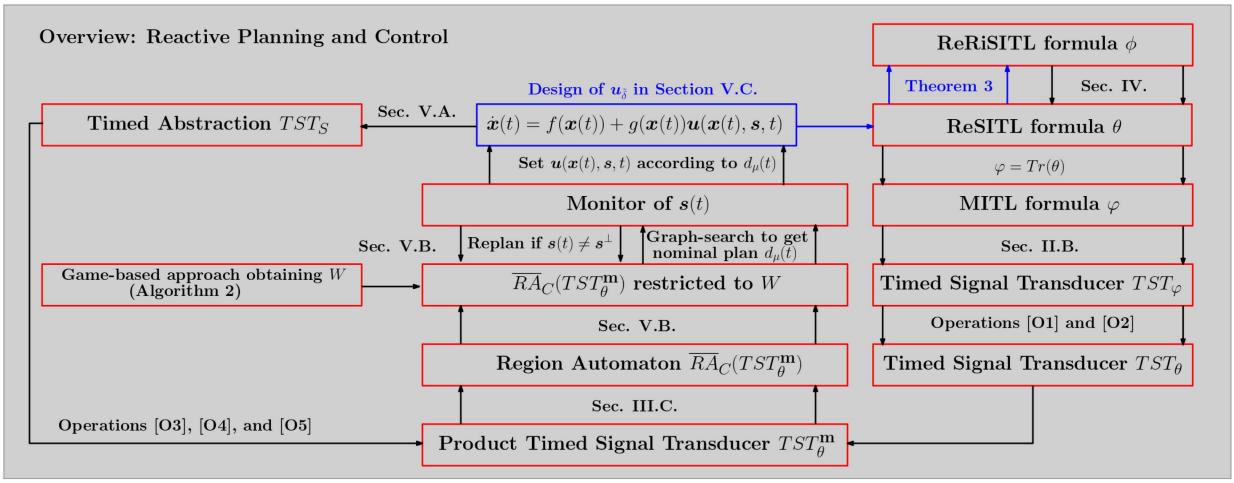


Fig. 4. Overview of the proposed method to reactively plan and control a dynamical system under ReRiSITL Specifications.

The set W tells us if we can let time pass or if a transition according to $\pi(W)$ has to be taken in a particular state. For TST_ϕ restricted to W this means that, at no time, an uncontrollable proposition $\boldsymbol{s}$ can force the system into a state from where the Büchi acceptance condition cannot be satisfied. The operator $\pi(W)$ then determines which $\boldsymbol{x}$ can be selected in case of a particular $\boldsymbol{s}$. Note in particular, as similarly analyzed in [33], that W_i in Algorithm 2 is monotonically decreasing, such that a fixed point, i.e., $W_{i+1} = W_i$, is eventually reached, such that Algorithm 2 terminates in a finite number of steps.

Theorem 2: If $\boldsymbol{s}$ is according to Assumption 2, then it holds that the ReRiSITL formula ϕ is satisfiable if $\overline{q}_0 \in W$ and if there exists $(\overline{q}_0, \delta_0, \overline{q}') \in \overline{\Delta}_R$ with $\gamma(\delta_0) = y$.

Proof: First note that due to the use of the timed signal transducer as in Fig. 3, we account for the form of $\boldsymbol{s}$ as in Assumption 2. Recall also from Theorem 1 that operations [O1] and [O2] restrict the behavior of TST_ϕ to the signals $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\boldsymbol{s} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\mathrm{uc}}|}$ with $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, 0) \models \phi$. Note that $RA_C(\mathrm{TST}_\phi)$ has, by construction, the same reachable set as TST_ϕ. Recall also that $RA_C(\mathrm{TST}_\phi)$ and $\overline{RA}_C(\mathrm{TST}_\phi)$ are equivalent so that reachability properties of TST_ϕ can equivalently be verified on $\overline{RA}_C(\mathrm{TST}_\phi)$. We now need to prove that, for each $\boldsymbol{s} \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathbb{B}^{|M^{\mathrm{uc}}|})$ that satisfies Assumption 2, there is an accepting run in $\overline{RA}_C(\mathrm{TST}_\phi)$ restricted to the states in W that satisfies the Büchi acceptance condition. By Algorithm 2, which is guaranteed to terminate in a finite number of steps, it is ensured that no state in W is a deadlock and can be continued to another state in W. Specifically, it is guaranteed that for each state in W an infinite continuation can be found that satisfies the Büchi acceptance condition, no matter how $\boldsymbol{s}(t)$ behaves. Note also that Zeno winning conditions have been excluded by modifying TST_ϕ to not permit Zeno behavior. Since $\overline{q}_0 \in W$ and since there exists $(\overline{q}_0, \delta_0, \overline{q}') \in \overline{\Delta}_R$ with $\gamma(\delta_0) = \top$, it follows that ϕ is satisfiable in the sense of Definition 2.

Note also that Theorem 2 is sufficient. Necessity does not hold due to the modification of TST_ϕ to avoid Zeno behavior, potentially introducing conservatism.

Finally, we remark that Sections III-A and III-B use graph search techniques, while Section III-C follows a game-based approach. One could argue that only the game-based approach solving Problem 1 is of interest. We have, however, chosen this particular exposition of our results since we will combine graph search techniques with a game-based approach to address Problem 2 in the following Sections IV and V.

IV. From ReRiSITL to ReSITL by Determinizing Risk Predicates

Fig. 4 can be used as a guide in the remainder of the article as it shows an overview of the reactive planning and control strategy that will be presented in Sections IV and V. Starting in the top right box of Fig. 4, this section introduces the idea to determinize risk predicates in M^{Ri} and replace them with deterministic predicates, hence converting the ReRiSITL formula ϕ into an ReSITL formula θ that we then deal with in Section V. We provide conditions under which a certain soundness property holds which ensures that satisfaction of θ implies satisfaction of ϕ. Sections IV-A and IV-B assume that ϕ is in positive normal form. In the end of Section IV-B, we discuss how we can deal with situations where this is not the case.

A. Risk Constrained Sets

In the following two sections, we will define risk-tightened deterministic predicates μ_m^{det} that will replace the risk predicates μ_m^{ri} and allow for the use of existing control methods. Note that $R(-h_m(\boldsymbol{x}, \boldsymbol{X}))$ depends on $\boldsymbol{x} \in \mathbb{R}^n$ (we drop the dependence of $\boldsymbol{x}(t)$ on t in this section for convenience). For given $\beta_m \in (0, 1)$ and $\gamma_m \in \mathbb{R}$, define the sets

$$\mathfrak{X}_m^{\mathrm{EV}}(\gamma_m) := \{\boldsymbol{x} \in \mathfrak{B} | EV[-h_m(\boldsymbol{x}, \boldsymbol{X})] \leq \gamma_m\}$$

$$\mathfrak{X}_m^{\mathrm{VaR}}(\beta_m, \gamma_m) := \{\boldsymbol{x} \in \mathfrak{B} | VaR_{\beta_m}(-h_m(\boldsymbol{x}, \boldsymbol{X})) \leq \gamma_m\}$$

$$\mathfrak{X}_m^{\mathrm{CVaR}}(\beta_m, \gamma_m) := \{\boldsymbol{x} \in \mathfrak{B} | CVaR_{\beta_m}(-h_m(\boldsymbol{x}, \boldsymbol{X})) \leq \gamma_m\}.$$

Note the set $\mathfrak{B} \subseteq \mathbb{R}^n$ that is supposed to be an arbitrarily big compact and convex set as will further be explained in Section V-C. The set $\mathfrak{B}$ can be seen as the workspace that (5) will be forced to remain within. The sets $\mathfrak{X}_m^{\mathrm{EV}}(\gamma_m)$, $\mathfrak{X}_m^{\mathrm{VaR}}(\beta_m, \gamma_m)$, and $\mathfrak{X}_m^{\mathrm{CVaR}}(\beta_m, \gamma_m)$ define all $\boldsymbol{x}$ for which the EV, VaR, and CVaR of $-h_m(\boldsymbol{x}, \boldsymbol{X})$ is less or equal than γ_m, respectively. If these sets are empty, the underlying predicate is not satisfiable. For $c_m \in \mathbb{R}$, which is a design parameter as opposed to β_m and γ_m,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

define

$$\mathfrak{X}_m(c_m) := \{\boldsymbol{x} \in \mathfrak{B} | h_m(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) \geq c_m\}$$

where the mean $\tilde{\boldsymbol{\mu}}$ has been used instead of $\boldsymbol{X}$ to evaluate the predicate function h_m. Note that $\mathfrak{X}_m(c_m)$ is a compact and convex set if Assumption 1 holds. If $\mathfrak{X}_m^{\text{EV}}(\gamma_m) \supseteq \mathfrak{X}_m(c_m)$, then $\boldsymbol{x} \in \mathfrak{X}_m(c_m)$ implies $\boldsymbol{x} \in \mathfrak{X}_m^{\text{EV}}(\gamma_m)$ (similarly for $\mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m)$ and $\mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m)$). This implies that predicates within an ReRiSITL formula ϕ can be determinized by using $h_m(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) \geq c_m$ [recall (3)] instead of $R(-h_m(\boldsymbol{x}, \boldsymbol{X})) \leq \gamma_m$ by conserving an important soundness property (Section IV-B). For given c_m, checking these set inclusions may be nonconvex. As shown in [45, Lemma 1], when $h_m(\boldsymbol{x}, \boldsymbol{X})$ is linear in $\boldsymbol{x}$, this can be checked efficiently since the distribution of $h_m(\boldsymbol{x}, \boldsymbol{X})$ is only shifted.

Lemma 2 [45, Lemma 1]: Assume that $h_m(\boldsymbol{x}, \boldsymbol{X}) = \boldsymbol{v}^T \boldsymbol{x} + h'(\boldsymbol{X})$ for $\boldsymbol{v} \in \mathbb{R}^n$ and for $h' : \mathbb{R}^{\tilde{n}} \to \mathbb{R}$, then

$$\mathfrak{X}_m^{\text{EV}}(\gamma_m) \supseteq \mathfrak{X}_m(c_m) \text{ iff } EV[-h_m(\boldsymbol{x}^*, \boldsymbol{X})] \leq \gamma_m$$

$$\mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m) \supseteq \mathfrak{X}_m(c_m) \text{ iff } \text{VaR}_{\beta_m}(-h_m(\boldsymbol{x}^*, \boldsymbol{X})) \leq \gamma_m$$

$$\mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m) \supseteq \mathfrak{X}_m(c_m) \text{ iff } \text{CVaR}_{\beta_m}(-h_m(\boldsymbol{x}^*, \boldsymbol{X})) \leq \gamma_m$$

where $\boldsymbol{x}^* := \underset{\boldsymbol{x} \in \mathfrak{X}_m(c_m)}{\text{argmin}} \ \boldsymbol{v}^T \boldsymbol{x}$ (a convex problem).

We remark that in particular $\text{VaR}_{\beta_m}(-h_m(\boldsymbol{x}^*, \boldsymbol{X}))$ and $\text{CVaR}_{\beta_m}(-h_m(\boldsymbol{x}^*, \boldsymbol{X}))$ can be efficiently computed [41, Th. 1] and that $\text{VaR}_{\beta}(-h_m(\boldsymbol{x}^*, \boldsymbol{X}))$ is obtained as a byproduct of the calculation of $\text{CVaR}_{\beta}(-h_m(\boldsymbol{x}^*, \boldsymbol{X}))$. If $h_m(\boldsymbol{x}, \boldsymbol{X})$ is nonlinear, we argue that, for some function classes, numerical methods can be used to check these set inclusions, e.g., when $h_m(\boldsymbol{x}, \boldsymbol{X})$ is quadratic in $\boldsymbol{x}$.

B. Converting ReRiSITL Into ReSITL Specifications

Considering the ReRiSITL formula ϕ that consists of the risk predicates $\mu_m \in M^{\text{Ri}}$ with $m \in \{1, \ldots, |M^{\text{Ri}}|\}$, we transform the ReRiSITL formula ϕ into an ReSITL formula θ. In particular, θ is obtained by replacing risk predicates $\mu_m \in M^{\text{Ri}}$ in ϕ by a deterministic predicate μ_m^{det} according to (3). More formally and by denoting $\phi(M^{\text{Ri}}, M^{\text{uc}})$ instead of ϕ to highlight the dependence on risk predicates M^{Ri} and uncontrollable propositions M^{uc}, let $\theta := \phi(M^{\text{det}}, M^{\text{uc}})$, where

$$M^{\text{det}} := \{\mu_1^{\text{det}}, \ldots, \mu_{|M^{\text{Ri}}|}^{\text{det}}\}.$$

Let now $\hat{M} := M^{\text{det}} \cup M^{\text{uc}}$ and let us associate the semantics $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, t) \models \theta$ with an ReSITL formula θ.[6] The next assumption is sufficient to ensure soundness in the sense that $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, t) \models \theta$ implies $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi$.

Assumption 3: For each $m \in \{1, \ldots, |M^{\text{Ri}}|\}$, $\mathfrak{X}_m^{\text{EV}}(\gamma_m) \supseteq \mathfrak{X}_m(c_m)$, $\mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m) \supseteq \mathfrak{X}_m(c_m)$, or $\mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m) \supseteq \mathfrak{X}_m(c_m)$ (depending on the type of predicate).

Example 2: By setting $c := 0.35$ for the VaR predicates and $c := 0.9$ for the CVaR predicates in Example 1, Assumption 3 is satisfied. The red circles in Fig. 1 indicate the obtained deterministic predicates, based on the predicate functions

$$h_{\text{R1}}^{\text{det}}(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) := \epsilon - \|\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_{\text{R1}}\|^2 - 0.35$$

$$h_{\text{R2}}^{\text{det}}(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) := \epsilon - \|\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_{\text{R2}}\|^2 - 0.35$$

$$h_{\text{O1}}^{\text{det}}(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) := \|\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_{\text{O1}}\|^2 - \epsilon - 0.9$$

$$h_{\text{O2}}^{\text{det}}(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) := \|\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_{\text{O2}}\|^2 - \epsilon - 0.9.$$

Passing in between the obstacles O1 and O2 is not possibly due to the uncertainty in $\boldsymbol{X}$ and the risk predicates.

Increasing c_m shrinks the set $\mathfrak{X}_m(c_m)$ so that Assumption 3 (verifiable by Lemma 2) poses a lower bound on c_m.

Theorem 3: Let Assumption 3 hold and ϕ be an ReRiSITL formula in positive normal form. If $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathfrak{B}$ and $\boldsymbol{s} : \mathbb{R}_{\geq 0} \to \mathbb{B}^{|M^{\text{uc}}|}$ are, such that $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, t) \models \theta$, then it follows that $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi$.

Proof: Due to Assumption 3, $\boldsymbol{x} \in \mathfrak{X}_m(c_m)$ implies $\boldsymbol{x} \in \mathfrak{X}_m^{\text{EV}}(\gamma_m)$, $\boldsymbol{x} \in \mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m)$, or $\boldsymbol{x} \in \mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m)$ depending on the type of the predicate m. It is now straightforward to recursively show on the ReRiSITL semantics in Definition 1 that $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, t) \models \theta$ implies $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{X}, t) \models \phi$ when $\boldsymbol{x}(t) \in \mathfrak{B}$, which holds by assumption. This follows since the semantics of ReRiSITL and ReSITL only differ on the predicate level and since negations are excluded since ϕ is in positive normal form.

An important task is to pick the set of c_m. In general, we may induce conservatism since the level sets of $\mathfrak{X}_m(c_m)$ may not be aligned with the level sets of $\mathfrak{X}_m^{\text{EV}}(\gamma_m)$, $\mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m)$, and $\mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m)$. When linearity of $h_m(\boldsymbol{x}, \boldsymbol{X})$ in $\boldsymbol{x}$ holds as in Lemma 2, conservatism can be avoided [45, Lemma 2].

If now, however, ϕ is not in positive normal form, there are two ways how to handle this case. The first way is to find c_m for each $m \in \{1, \ldots, |M^{\text{Ri}}|\}$ according to [45, Lemma 2], i.e., the set inclusion in Assumption 3 is replaced by an equality. More generally, a more elegant way is to bring ϕ into positive normal form, as for instance shown in [11, Proposition 2]. This would lead to a formula ϕ potentially having negations in front of some or all of the predicates, i.e., $\neg\mu_m^{\text{Ri}}$. For those predicates, we redefine the sets $\mathfrak{X}_m^{\text{EV}}(\gamma_m)$, $\mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m)$, and $\mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m)$ as

$$\mathfrak{X}_m^{\text{EV}}(\gamma_m) := \{\boldsymbol{x} \in \mathfrak{B} | EV[-h_m(\boldsymbol{x}, \boldsymbol{X})] > \gamma_m\}$$

$$\mathfrak{X}_m^{\text{VaR}}(\beta_m, \gamma_m) := \{\boldsymbol{x} \in \mathfrak{B} | VaR_{\beta_m}(-h_m(\boldsymbol{x}, \boldsymbol{X})) > \gamma_m\}$$

$$\mathfrak{X}_m^{\text{CVaR}}(\beta_m, \gamma_m) := \{\boldsymbol{x} \in \mathfrak{B} | CVaR_{\beta_m}(-h_m(\boldsymbol{x}, \boldsymbol{X})) > \gamma_m\}.$$

Note that only the sign of the inequality has changed compared to the definition in Section IV-A. For $c_m \in \mathbb{R}$, we then also redefine $\mathfrak{X}_m(c_m)$ as

$$\mathfrak{X}_m(c_m) := \{\boldsymbol{x} \in \mathfrak{B} | h_m(\boldsymbol{x}, \tilde{\boldsymbol{\mu}}) \leq c_m\}.$$

We would now again like to establish the set inclusions as in Assumption 3 by a suitable choice of c_m with these modified definitions. Note that these inclusions can then be similarly checked as in Lemma 2 (just reversing inequalities again).

V. Reactive Planning Under ReSITL Specifications

Following Section IV, we can obtain an ReSITL formula θ from the ReRiSITL formula ϕ. Motivated by the soundness result in Theorem 3, we now propose a reactive planning and control method that leads to a satisfaction of the ReSITL formula θ that consequently leads to the satisfaction of the ReRiSITL formula ϕ (see also the top right box in Fig. 4).

In Section V-A, we abstract the control system in (5) into a timed signal transducer TST_S (top left box in Fig. 4). This

[6] We define $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, t) \models \mu_m^{\text{det}}$ iff $h_m(\boldsymbol{x}(t), \tilde{\boldsymbol{\mu}}) \geq c_m$ using (3) instead of (1), while the other operators follow as in Section II-A.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3: Reactive Planning for ReSITL Formula θ.

1: Obtain the MITL formula $\varphi := Tr(\theta)$.
2: Obtain TST_φ according to Section II-B and where uncontrollable propositions $p_i \in AP$, i.e., p_i with $Tr^{-1}(p_i) \in \hat{M} \cap M^{\text{uc}}$, are modeled as in Fig. 3.
3: Perform [O1] and [O2] to obtain TST_θ.
4: Obtain TST_S according to Section V-A.
5: Perform [O3], [O4], and [O5] to obtain $\text{TST}_\theta^{\text{m}}$.
6: Modify $\text{TST}_\theta^{\text{m}}$ to avoid Zeno behavior.
7: Translate $\text{TST}_\theta^{\text{m}}$ into $RA_C(\text{TST}_\theta^{\text{m}})$.
8: Translate $RA_C(\text{TST}_\theta^{\text{m}})$ into $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$.
9: Run Algorithm 2 with the modified function $\bar{\pi} : 2^{\overline{Q}} \to 2^{\overline{Q}}$ and $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$ as the inputs to obtain W.
10: Calculate the initial plan $d_\mu(t)$ based on $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$ and obtain the associated control law $\boldsymbol{u}(\boldsymbol{x}, t)$ (only possible if the conditions in Theorem 2 are satisfied).
11: **while** $\boldsymbol{s}(t) = \boldsymbol{s}^\perp$ **do**
12: **if** $\boldsymbol{s}(t) \neq \boldsymbol{s}^\perp$ **then**
13: Recalculate $d_p(t)$ and $\boldsymbol{u}(\boldsymbol{x}, t)$
14: Apply $\boldsymbol{u}(\boldsymbol{x}, t)$ to (5)

abstraction is based on the assumption of existing logic-based feedback control laws from Section V-C. We then modify TST_θ into $\text{TST}_\theta^{\text{m}}$ (bottom box in Fig. 4), a product automaton between TST_θ and TST_S that does not induce an exponential state explosion since TST_θ and TST_S "align" in a suitable way due to the particular control laws in Section V-C.[7] In Section V-B, we then present the reactive planning method that consists of a combination of a game-based approach and graph search techniques (boxes in the middle of Fig. 4).

In Algorithm 3 presented below, we summarize the reactive planning algorithm that is presented in this section. In the remainder, we present and explain the steps of Algorithm 3. In line 1, abstract the ReSITL formula $\theta(\hat{M})$ into an MITL formula $\varphi := Tr(\theta(\hat{M})) = \theta(AP)$. Note that we abstract $\theta(\hat{M})$, which depends on deterministic predicates and uncontrollable propositions $\hat{M}$ (recall that $\hat{M} := M^{\text{det}} \cup M^{\text{uc}}$), as opposed to $\phi(M)$ in Section III-A by the transformation $Tr(\cdot)$. Based on φ, construct $\text{TST}_\varphi := (S, s_0, \Lambda, \Gamma, \boldsymbol{c}, \iota, \Delta, \lambda, \gamma, \mathcal{A})$ according to Section II-B (Line 2 in Algorithm 3). We again assume that uncontrollable propositions $p_i \in AP$, i.e., p_i with $Tr^{-1}(p_i) \in \hat{M} \cap M^{\text{uc}}$, are modeled as in Fig. 3. In Line 3, perform operations [O1] and [O2] on TST_φ to obtain the timed signal transducer $\text{TST}_\theta := (S^\theta, s_0, \Lambda, \Gamma, \boldsymbol{c}, \iota, \Delta^\theta, \lambda, \gamma, \mathcal{A}^\theta)$.[8] Note that checking [O1] and [O2] is computationally tractable if Assumption 1 holds due to the determinization in Section IV.

A. Timed Abstraction of the Dynamical Control System

In line 4 of Algorithm 3, we abstract the system in (5) into a timed signal transducer $\text{TST}_S := (\tilde{S}, \tilde{S}_0, \tilde{\Lambda}, \tilde{c}, \tilde{\Delta}, \tilde{\lambda})$ (top left box in Fig. 4). Note the absence of output labels, invariants, and a Büchi acceptance condition, and that $\tilde{c}$ is a scalar. The transition relation $\tilde{\Delta}$ is now based on the ability of the system to switch in finite time, by means of a feedback control law $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ between elements in $Tr^{-1}(BC(\text{TST}_\theta)) \subseteq BC(\tilde{\Lambda})$, where $\tilde{\Lambda} := M$ and $BC(\text{TST}_\theta) := \{z \in BC(AP) | \exists s \in S^\theta, \lambda(s) = z\}$. It is assumed that a library of such logic-based feedback control laws $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ is available, e.g., as presented in Section V-C. Assume that $|\tilde{S}| = |Tr^{-1}(BC(\text{TST}_\theta))|$ and let $\tilde{\lambda} : \tilde{S} \to Tr^{-1}(BC(\text{TST}_\theta))$, where for $\tilde{s}, \tilde{s}'' \in \tilde{S}$ with $\tilde{s}' \neq \tilde{s}''$, it holds that $\tilde{\lambda}(\tilde{s}') \neq \tilde{\lambda}(\tilde{s}'')$ so that each state is uniquely labeled by $\tilde{\lambda}$, i.e., each state indicates exactly one Boolean formula from $Tr^{-1}(BC(\text{TST}_\theta))$. Note that TST_θ and TST_S now "align" in a way that will allow to avoid a state space explosion when forming a product automaton between them. A transition from $\tilde{s}$ to $\tilde{s}'$ is indicated by $(\tilde{s}, \tilde{g}, 0, \tilde{s}') \in \tilde{\Delta}$ where $\tilde{g}$ is a guard that depends on (5). In particular, we assume that $\tilde{g}$ encodes intervals of the form $(C,' C'')$, $[C,' C'')$, $(C,' C'']$, $[C,' C'']$, or conjunctions of them, where $C,' C'' \in \mathbb{Q}_{\geq 0}$ with $C' \leq C''$.

Definition 8 (Transitions in TST_S): There exists a transition $\tilde{\delta} := (\tilde{s}, \tilde{g}, 0, \tilde{s}') \in \tilde{\Delta}$ if, for all $\tau > 0$ with $\tau \models \tilde{g}$ and for all $\boldsymbol{x}_0 \in \mathbb{R}^n$ with $(\boldsymbol{x}_0, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{s})$, there exists a control law $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ so that the solution $\boldsymbol{x}(t)$ to (5) is such that

1) either, for all $t \in [0, \tau)$, $(\boldsymbol{x}(t), \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{s})$ and $(\boldsymbol{x}(\tau), \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{s}')$;
2) or, for all $t \in [0, \tau]$, $(\boldsymbol{x}(t), \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{s})$ and there exists $\tau' > \tau$ such that, for all $t \in (\tau, \tau']$, $(\boldsymbol{x}(\tau'), \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{s}')$;

for which we define $\tilde{\lambda}(\tilde{\delta}) := \tilde{\lambda}(\tilde{s}')$ in the former and $\tilde{\lambda}(\tilde{\delta}) := \tilde{\lambda}(\tilde{s})$ in the latter case.

The two types of transitions in the above definition can be thought of as transitioning into closed and open regions in $\mathbb{R}^n$, respectively. Note that such a control law $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ has to ensure invariance and finite-time reachability properties. Note also that $\boldsymbol{s}^\perp$ is used in Definition 8 since controlled transitions will only happen when all uncontrollable propositions are false. Finally, the set $\tilde{S}_0$ consists of all elements $\tilde{s}_0 \in \tilde{S}$, such that $(\boldsymbol{x}_0, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{s}_0)$.

According to line 5 of Algorithm 3, we next form a product automaton $\text{TST}_\theta^{\text{m}}$ (bottom box in Fig. 4) of TST_θ and TST_S that avoids a state space explosion that is typically the outcome of forming automata products. This follows since each input label of a state or transition in TST_θ corresponds to one state label of TST_S, i.e., TST_θ and TST_S align in a way, so that $\text{TST}_\theta^{\text{m}}$ (defined below and corresponding to the product of TST_θ and TST_S) has no more states than TST_θ. Our approach relies on: 1) The removal of transitions from TST_θ, and 2) constraining guards g of transitions in TST_θ to account for guards $\tilde{g}$ in TST_S. Let us, without loss of generality, assume that each input label of a transition in TST_θ contains every literal from $\hat{M}$ and does not contain any disjunctions.[9]

[O3] For each transition $\delta := (s, g, r, s') \in \Delta^\theta$ for which there exists $\boldsymbol{x} \in \mathbb{R}^n$, such that $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta))$, remove δ if
a) there exists no transition $\tilde{\delta} := (\tilde{s}, \tilde{g}, 0, \tilde{s}') \in \tilde{\Delta}$ with $\lambda(s) = Tr(\tilde{\lambda}(\tilde{s}))$, and $\lambda(s') = Tr(\tilde{\lambda}(\tilde{s}'))$,

[7] TST_θ is a timed signal transducer for θ and constructed in the same way as TST_ϕ was obtained previously for ϕ.

[8] The notation in [O1] and [O2] needs to be slightly modified to account for θ instead of ϕ. In particular, $\boldsymbol{X}$ should be replaced with $\tilde{\boldsymbol{\mu}}$.

[9] Note that each input label of a transition in TST_θ can be converted into full disjunctive normal form. Then, this transition can be split into several transitions, one for each disjunct, where each new input label corresponds to exactly one of the disjuncts.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

and for which $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{\delta})$ implies $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta))$.

Remove the corresponding δ from $\mathcal{A}^\theta$.

We follow two goals with operation [O3]. First, we only consider to remove transitions that are induced by uncontrollable propositions being false, i.e., when $\boldsymbol{s} = \boldsymbol{s}^\perp$. This is important as we would like to keep transitions with $\boldsymbol{s} \neq \boldsymbol{s}^\perp$ for the reactive planning. Note in particular that, if there exists $\boldsymbol{x} \in \mathbb{R}^n$, such that $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta))$, then there exists no $\boldsymbol{x} \in \mathbb{R}^n$, such that $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta))$ for $\boldsymbol{s} \neq \boldsymbol{s}^\perp$. Second, we remove such transitions if there exists no control law $\boldsymbol{u}_{\tilde{\delta}}$ that can simulate the transition in the system (5).

[O4] For each transition $\delta_0 := (s_0, g, r, s') \in \Delta$, remove δ_0 if $(\boldsymbol{x}_0, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \not\models Tr^{-1}(\lambda(s'))$ or if there exists no $\boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}$, such that $(\boldsymbol{x}_0, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}) \not\models Tr^{-1}(\lambda(\delta_0))$. Remove the corresponding δ_0 from $\mathcal{A}^\theta$.

Operation [O4] takes care of the initial condition $\boldsymbol{x}_0$. If s_0 is removed in [O4], the problem is infeasible given the initial condition $\boldsymbol{x}_0$.

Denote next the obtained sets by S^{m}, Δ^{m}, and $\mathcal{A}^{\text{m}}$ for which $S^{\text{m}} \subseteq S^\theta$, $\Delta^{\text{m}} \subseteq \Delta^\theta$, and $\mathcal{A}^{\text{m}} \subseteq \mathcal{A}^\theta$. We further take care of the timings including an additional clock into TST_θ. Therefore, let $\boldsymbol{c}^{\text{m}} := [\boldsymbol{c}^T \quad \tilde{c}]^T$ and perform the operation.

[O5] For each transition $\delta^{\text{m}} := (s, g, r, s') \in \Delta^{\text{m}}$ for which there exists $\boldsymbol{x} \in \mathbb{R}^n$, such that $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta^{\text{m}}))$, let $g^{\text{m}} = g \wedge \tilde{g}$ where $\tilde{\delta} := (\tilde{s}, \tilde{g}, 0, \tilde{s}') \in \tilde{\Delta}$ with $\lambda(s) = Tr(\tilde{\lambda}(\tilde{s}))$, $\lambda(s') = Tr(\tilde{\lambda}(\tilde{s}'))$, and for which $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \tilde{\lambda}(\tilde{\delta})$ implies $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta))$. Replace g and r in δ^{m} with g^{m} and r^{m}, respectively, where r^{m} is obtained in an obvious manner.

We emphasize that adding $\tilde{c}$ and $\tilde{g}$ is crucial to ensure correctness. Let the modified timed signal transducer be denoted by $\text{TST}_\theta^{\text{m}} := (S^{\text{m}}, s_0, \Lambda, \Gamma, \boldsymbol{c}^{\text{m}}, \iota, \Delta^{\text{m}}, \lambda, \gamma, \mathcal{A}^{\text{m}})$ and note that $L(\text{TST}_\theta^{\text{m}}) \subseteq L(\text{TST}_\theta) \subseteq L(\text{TST}_\varphi)$.

Remark 5: The operations [O3]–[O5] result in the timed signal transducer $\text{TST}_\theta^{\text{m}}$ that, by construction, restricts the behavior of TST_θ exactly to the behavior allowed by TST_S and corresponds hence to a product automaton without exhibiting an exponential state space explosion.

B. Reactive Plan Synthesis

Based on $\text{TST}_\theta^{\text{m}}$, let us now present the reactive planning method depicted in the boxes in the middle of Fig. 4. We first derive a *nominal plan* $d_\mu : \mathbb{R}_{\geq 0} \to BC(M)$ from $\text{TST}_\theta^{\text{m}}$ based on the assumption that $\boldsymbol{s}(t) = \boldsymbol{s}^\perp$ for all $t \in \mathbb{R}_{\geq 0}$. This plan is executed until $\boldsymbol{s}(t_{\text{replan}}) \neq \boldsymbol{s}^\perp$ for some $t_{\text{replan}} \in \mathbb{R}_{\geq 0}$, the moment when *reactive and online replanning* is needed. In line 6 of Algorithm 3, let $\text{TST}_\theta^{\text{m}}$ again be modified to not exhibit Zeno behavior and let $\overline{RA}_C(\text{TST}_\theta^{\text{m}}) := (\overline{Q}, \overline{q}_0, \overline{\Delta}_R, \overline{\mathcal{A}}_R)$ be the region automaton of $\text{TST}_\theta^{\text{m}}$ based on $(S^{\text{m}} \times \mathbb{R}^O_{\geq 0}, \Rightarrow_C)$ and Definitions 6 and 7 (lines 7 and 8 of Algorithm 3).[10] Replanning may now require to take, at an unknown time instant t_{replan}, a transition that is not contained within the nominal plan. Those instances may possibly require an infeasible discontinuity in the physical state $\boldsymbol{x}$ that we need to rule out.

[10]Definitions 6 and 7 need to be altered to account for using $\text{TST}_\theta^{\text{m}}$ instead of TST_ϕ in an obvious manner.

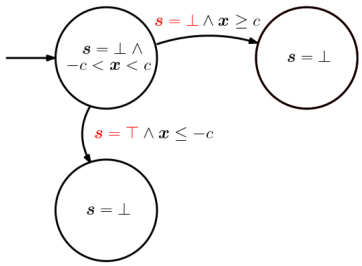


Fig. 5. Illustration of Example 3 and why a modified definition of $\pi(W)$ is needed to avoid discontinuities in $\boldsymbol{x}(t)$.

Example 3: To illustrate the aforementioned issue, consider Fig. 5. For the top left state, there exist two transitions to the top right and the bottom left state. Assume the former transition can be realized by the control law $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$. Starting from the top left state, the initial plan will consider the transition with $\boldsymbol{s} = \bot$ to the top right state implying that $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ is used until time $t = t_{\text{replan}}$, such that $0 < \boldsymbol{x}(t_{\text{replan}}) < c$. After replanning, however, the other transition with $\boldsymbol{s} = \top$ to the bottom left state has to instantaneously be taken requiring to immediately achieve $\boldsymbol{x} \leq -c$. Such a discontinuity in $\boldsymbol{x}(t)$ is not realizable in (5) that only admits continuous $\boldsymbol{x}(t)$.

One way of dealing with this issue is to modify the predecessor operator. Recall therefore that a state $\overline{q} \in \overline{Q}$ in $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$ consists of the elements $\overline{q} := (s, \alpha, i) \in Q \subseteq S^{\text{m}} \times A \times \{1, \ldots, |\mathcal{A}^{\text{m}}|\}$ and redefine now $\pi(W)$ to

$$\hat{\pi}(W) := \{\overline{q} \in \overline{Q} | \forall \boldsymbol{s} \in \mathbb{B}^{|M^{\text{uc}}|}, \exists (\overline{q}, \delta, \overline{q}') \in \overline{\Delta}_R \text{ s.t.}$$

$$1)\ \overline{q}' \in W,\ 2)\ \forall \boldsymbol{x} \in \mathbb{R}^n \text{ s.t. } (\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(s))$$

$$(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(\delta)) \text{ and } (\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\lambda(s'))\}.$$

The second condition in $\hat{\pi}(W)$ now additionally ensures that all $\boldsymbol{x}$ that satisfy the state label of s also satisfy the state labels of the transition δ as well as the next state s'. As a consequence, an instantaneous transition from $\overline{q}$ to $\overline{q}'$ due to $\boldsymbol{s}(t) \neq \boldsymbol{s}^\perp$ can happen without requiring that $\boldsymbol{x}(t)$ is discontinuous. We emphasize, again, that this condition is necessary with respect to the solutions to (5). Let W be obtained from Algorithm 2 with $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$ and $\hat{\pi} : 2^{\overline{Q}} \to 2^{\overline{Q}}$ as the input (line 9 of Algorithm 3).

Initial Plan Synthesis: For line 10 in Algorithm 3, let $d_p(t)$, as opposed to Section III-B, now be obtained from $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$ as follows. We find, using graph search techniques, a sequence $\mathfrak{q} := (\overline{q}_0, \overline{q}_1, \ldots) := (\mathfrak{q}_p, \mathfrak{q}_s^\omega)$ satisfying the Büchi acceptance condition $\overline{\mathcal{A}}_R$ with $\overline{q}_j \in \overline{Q} \cap W$ for each $j \in \mathbb{N}$ and where $(\overline{q}_j, \delta_{t,j}, \overline{q}_{j+1}) \in \overline{\Delta}_R$ so that, for each $\delta_{t,j}$, there exists $\boldsymbol{x} \in \mathbb{R}^n$, such that $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\delta_{t,j})$. Additionally and for the initial transition δ_0, we again require that $\gamma(\delta_0) = y$ to indicate $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, 0) \models \theta$. Note in particular the restriction to $\overline{q}_j \in \overline{Q} \cap W$ which will allow to replan if $\boldsymbol{s}(t_{\text{replan}}) \neq \boldsymbol{s}^\perp$ for some $t_{\text{replan}} \in \mathbb{R}_{\geq 0}$. We again find timings $\bar{\tau} := (\tau_0, \tau_1, \ldots) := (\bar{\tau}_p, \bar{\tau}_s^\omega)$ that are associated with $\mathfrak{q}$. Such a plan $d_p(t)$ is guaranteed to exist if the conditions in Theorem 2 are satisfied. Recall that $T_j := \sum_{k=0}^{j} \tau_j$, and define

$$d_p(t) := \begin{cases} \lambda(\delta_{t,j}) & \text{if } t = T_j \\ \lambda(s_j) & \text{if } T_j < t < T_{j+1}. \end{cases} \tag{7}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

We can now define the control law $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$ based on the plan $d_\mu(t) := Tr^{-1}(d_p(t))$. Recall therefore that each transition $\delta_{t,j}$ is associated, when projecting back to TST_S, with a control law $\boldsymbol{u}_{\tilde{\delta}_{t,j}}(\boldsymbol{x}, t)$ as explained in Section V-A. Recall the definition of T_j and let

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t) := \begin{cases} \boldsymbol{u}_{\tilde{\delta}_1}(\boldsymbol{x}, t) & \text{for } t \in [0, T_1) \\ \boldsymbol{u}_{\tilde{\delta}_{j+1}}(\boldsymbol{x}, t - T_j) & \text{for } t \in (T_j, T_{j+1}) \text{ with } j \geq 2 \end{cases}$$

and, for $t = T_j$ with $j \geq 2$, let

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, T_j) := \begin{cases} \boldsymbol{u}_{\tilde{\delta}_{j+1}}(\boldsymbol{x}, 0) & \text{if } \tilde{\lambda}(\tilde{s}_{j+1}) = d_\mu(T_j) \\ \boldsymbol{u}_{\tilde{\delta}_j}(\boldsymbol{x}, \tau_j) & \text{if } \tilde{\lambda}(\tilde{s}_j) = d_\mu(T_j). \end{cases}$$

Note that $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, T_j)$ in particular accounts for the two types of transitions in Definition 8.

Corollary 3: Assume that $\boldsymbol{s}(t) = \boldsymbol{s}^\perp$ for all $t \in \mathbb{R}_{\geq 0}$, $\overline{q}_0 \in W$, and there exists $(\overline{q}_0, \delta_0, \overline{q}') \in \Delta_R$ with $\gamma(\delta_0) = y$, then $d_p(t)$ as in (7) exists and $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$ results in $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, 0) \models \theta$.

Proof: Similar to Theorem 2 and by the construction of $\text{TST}_\theta^{\text{m}}$, it follows that θ is satisfiable given that $\overline{q}_0 \in W$ and that there exists $(\overline{q}_0, \delta_0, \overline{q}') \in \Delta_R$ with $\gamma(\delta_0) = y$. It directly follows that, in this case, a plan $d_p(t)$ exists. Note next that by construction of TST_S and $\text{TST}_\theta^{\text{m}}$, each transition δ in $\text{TST}_\theta^{\text{m}}$ can be realized in (5) by an associated control law $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$. By the construction of the plan $d_p(t)$ and the associated control law $\boldsymbol{u}(\boldsymbol{x}, t)$, it follows trivially that $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$, build from a sequence of such $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$, results in $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, 0) \models \theta$.

Reactive and Online Replanning: If hence $\boldsymbol{s}(t) = \boldsymbol{s}^\perp$ for all $t \in \mathbb{R}_{\geq 0}$, there is nothing left to do and we apply $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$ as in line 14 of Algorithm 3. If, however, $\boldsymbol{s}(t_{\text{replan}}) \neq \boldsymbol{s}^\perp$ for some $t_{\text{replan}} \in \mathbb{R}_{\geq 0}$, we need to replan and update our plan $d_\mu(t)$ that may be violated by this particular $\boldsymbol{s}(t_{\text{replan}})$ (lines 12 and 13 in Algorithm 3). Assume that, at time t_{replan}, the system is in state $\overline{q}_{j^*}$. We then find an updated sequence $\mathfrak{q}_{\text{replan}} := (\overline{q}_{j^*}, \overline{q}_{j^*+1}, \ldots)$ satisfying the Büchi acceptance condition $\overline{\mathcal{A}}_R$ again with $\overline{q}_j \in \overline{Q} \cap W$ for each $j > j^*$ and where $(\overline{q}_j, \delta_{t,j}, \overline{q}_{j^*+1}) \in \overline{\Delta}_R$ so that 1) $(\boldsymbol{x}(t_{\text{replan}}), \boldsymbol{s}(t_{\text{replan}}), \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\delta_{t,j^*})$, and 2) for each $\delta_{t,j}$ with $j > j^*$ there exists $\boldsymbol{x} \in \mathbb{R}^n$, such that $(\boldsymbol{x}, \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models Tr^{-1}(\delta_{t,j})$. If $t_{\text{replan}} = 0$, it is additionally required that $\gamma(\delta_{t,j^*}) = y$. We again find timings $\bar{\tau} := (\tau_{j^*}, \tau_{j^*+1}, \ldots)$ that are associated with $\mathfrak{q}_{\text{replan}}$. Based on this updated sequence, we recalculate $d_p(t)$ in (7) and $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$ in an obvious manner.

Theorem 4: Assume that $\boldsymbol{s}(t)$ is according to Assumption 2, $\overline{q}_0 \in W$, and there exists $(\overline{q}_0, \delta_0, \overline{q}') \in \Delta_R$ with $\gamma(\delta_0) = y$, then finding an initial plan $d_p(t)$ and updating $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$ in the previously described manner in case that $\boldsymbol{s}(t_{\text{replan}}) \neq \boldsymbol{s}^\perp$ results in $(\boldsymbol{x}, \boldsymbol{s}, \tilde{\boldsymbol{\mu}}, 0) \models \theta$.

Proof: The assumptions that $\overline{q}_0 \in W$ and that there exists $(\overline{q}_0, \delta_0, \overline{q}') \in \Delta_R$ with $\gamma(\delta_0) = y$, again guarantee that there exists an initial plan $d_p(t)$. Due to the properties of W and given that $\boldsymbol{s}(t)$ is according to Assumption 2, it holds that a new plan and an updated $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, t)$ can always be found whenever $\boldsymbol{s}(t_{\text{replan}}) \neq \boldsymbol{s}^\perp$. Each such instantaneous transition is well defined in the sense of not requiring a discontinuity in $\boldsymbol{x}(t)$ due to the modified definition of $\pi(W)$.

We remark that Assumption 2 is not only necessary for the game-based approach in Algorithm 2 but that the assumption is also necessary to be able to replan. Without Assumption 2, there is no information about the value of $\boldsymbol{s}(t)$ shortly after t_{replan}. By Assumption 2, there follows an open time interval in which $\boldsymbol{s}(t) = \boldsymbol{s}^\perp$ after t_{replan} so that a next state can be selected whose state label is satisfied by $\boldsymbol{s}^\perp$. Further note that Assumption 2 effectively poses an upper bound on the frequency of times that replanning is initiated.

To conclude this section, we note that a combination of graph search techniques and a game-based approach has been presented. The game-based approach ensures that it is always possible to make progress toward satisfying the Büchi acceptance condition by ruling out "bad" transitions, while graph search techniques actually enforce this progress.

C. Feedback Control Under STL Specifications

In this section, we discuss the control laws $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ that are supposed to achieve the transitions $\tilde{\delta} := (\tilde{s}, \tilde{g}, 0, \tilde{s}') \in \tilde{\Delta}$ in Definition 8 for the timed abstraction TST_S. In particular, such transitions can be captured by the STL formulas

$$G_{[0,\tau)}\mu_{\text{inv}}(\boldsymbol{x}) \wedge F_\tau \mu_{\text{reach}}(\boldsymbol{x}) \wedge G_{[0,\tau]}\mu_{\text{ws}}(\boldsymbol{x}) \tag{8}$$

$$G_{[0,\tau]}\mu_{\text{inv}}(\boldsymbol{x}) \wedge G_{(\tau,\tau']}\mu_{\text{reach}}(\boldsymbol{x}) \wedge G_{[0,\tau']}\mu_{\text{ws}}(\boldsymbol{x}) \tag{9}$$

where $\mu_{\text{inv}}(\boldsymbol{x}) := \tilde{\lambda}(\tilde{s})$ and $\mu_{\text{reach}}(\boldsymbol{x}) := \tilde{\lambda}(\tilde{s}')$ are deterministic predicates as in (3), and where $\tau \in \mathbb{R}_{>0}$ with $\tau \models \tilde{g}$, while $\mu_{\text{ws}}(\boldsymbol{x})$ encodes a compact set $\mathfrak{B}$ according to Section IV; $\mathfrak{B}$ can be any compact set, typically the workspace. With $\mu_{\text{inv}}(\boldsymbol{x})$, $\mu_{\text{reach}}(\boldsymbol{x})$, and $\mu_{\text{ws}}(\boldsymbol{x})$, we can now associate predicate functions $h_{\text{inv}}(\boldsymbol{x})$, $h_{\text{reach}}(\boldsymbol{x})$, and $h_{\text{ws}}(\boldsymbol{x})$.

There is a plethora of recent works that have addressed the problem of controlling systems as in (5) under spatio-temporal constraints as in (8) or (9). In particular [4] and [20] address the control problem by time-varying control-barrier functions and fixed time control Lyapunov functions, respectively. For robotic specific problem setups, funnel control laws to solve (8) or (9) have also appeared in [18], while optimization-based methods are presented in [19]. Another approach, relying on time-varying vector fields, has appeared in [24]. We are, purposefully and with respect to page limitations, not presenting a specific type of feedback control law here and emphasize that our proposed reactive planning method is agnostic to feedback control laws that can achieve the STL specification as in (8) or (9). Note that the previously mentioned works pose certain assumptions on the systems dynamics in (5) as well as on the form of $h_{\text{inv}}(\boldsymbol{x})$, $h_{\text{reach}}(\boldsymbol{x})$, and $h_{\text{ws}}(\boldsymbol{x})$. We remark that controlling systems under timed specifications of the type in (8) or (9) has recently attracted interest in the research community so that we expect more progress in this respect.

VI. Completeness and Complexity

In summary, the presented framework consists of: 1) Translating the ReRiSITL specification ϕ into a ReSITL specification θ in Section IV, and 2) reactive planning under this ReSITL specifications θ in Section V, as summarized in Algorithm 3. The framework is sound in the sense of Theorems 3 and 4, but not necessarily complete, i.e., there may exist a solution even though we may not find it. There are three reasons for such conservatism. First, the translation from the ReRiSITL specification ϕ to the ReSITL specification θ may induce conservatism as discussed in Section IV. Second, in line 6 of Algorithm 3, we need to modify $\text{TST}_\theta^{\text{m}}$ to avoid Zeno behavior. This operation potentially induces conservatism that can, however, be reduced as also discussed previously. Third, the construction of nonlinear control laws,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

presented in Section V-C, may introduce conservatism. This is inherent in nonlinear control and we do not view this as a drawback of our method.

The presented framework consists of several computationally expensive operations. Fortunately, these operations can be performed offline. We focus on space complexity. First, the translation from the MITL formula φ to the timed signal transducer TST_φ induces $O(|\varphi|M)$ clocks and $2^{O(|\varphi|M)}$ states, where $|\varphi|$ denotes the complexity of φ and M is related to the length of the maximum time interval in φ (see [32, Th. 6.7]). Operations [O1] and [O2], which transform TST_φ into TST_θ, ease the complexity by removing a considerable number of states and transitions from TST_φ. An exact number is in general not quantifiable as those removals depend on predicate dependencies in the specification θ. Operations [O3] and [O4] further remove states and transitions from TST_θ to obtain the product automaton $\text{TST}_\theta^{\text{m}}$. Note that we obtain computational benefits over existing methods that would induce additional $O(|S^\theta||\tilde{S}|)$ states. The operation $RA_C(\text{TST}_\theta^{\text{m}})$ results in an automaton with $O(|S^{\text{m}}|\text{len}(\boldsymbol{c}_m))$ states, where $\text{len}(\boldsymbol{c}_m)$ denotes the length of clock constraints in $\text{TST}_\theta^{\text{m}}$ (see [29, Sec. 4.3]). The translation from $RA_C(\text{TST}_\theta^{\text{m}})$ to $\overline{RA}_C(\text{TST}_\theta^{\text{m}})$ results in an automaton with $O(|Q||\mathcal{A}_R|)$ states, which can considerably be reduced as discussed in Remark 4. The time complexity of Algorithm 2 and graph search techniques to find a plan $d_p(t)$ follows standard arguments. Operations [O1], [O2], [O3], and [O4] involve solving nonlinear mixed integer programs, and in particular mixed integer linear programs when Assumption 1 holds.

VII. Simulations

We consider a unicycle model with dynamics $\dot{\boldsymbol{z}} = f(\boldsymbol{z}) + g(\boldsymbol{z})\boldsymbol{u}$ and state $\boldsymbol{z} := \begin{bmatrix}\boldsymbol{x}^T & x_a\end{bmatrix}^T := \begin{bmatrix}x_x & x_y & x_a\end{bmatrix}^T$ to model the two-dimensional position and orientation, respectively. Here, $\boldsymbol{u} := \begin{bmatrix}v & \omega\end{bmatrix}^T$ contains the translational and rotational control inputs. In particular, let $f(\boldsymbol{x}) := 0.5 \cdot \begin{bmatrix}-\text{sat}(x_x) & -\text{sat}(x_y) & 0\end{bmatrix}^T$, where $\text{sat}(x) = x$ if $|x| \leq 1$ and $\text{sat}(x) = 1$ otherwise, and let $g(\boldsymbol{x}) := \begin{bmatrix}\cos(x_a) & 0\\ \sin(x_a) & 0\\ 0 & 1\end{bmatrix}$. To obtain $\boldsymbol{u}(\boldsymbol{z}, t)$, we use here the time-varying control barrier functions from [4]. In particular, time-varying control barrier functions adapted for nonholonmic systems from [45] are used for which no knowledge of $f(\boldsymbol{z})$ is required.

For this system, the imposed ReRiSITL specification ϕ is the one given in Example 1. The specification ϕ is rich enough to illustrate all theoretical findings (i.e., how to deal with risk predicates, uncontrollable propositions, and past temporal operators) and yet basic enough to explain all subtleties of ϕ and the reactive and risk-aware control sythesis.

Recall the determinization of risk predicates according to Section IV in Example 2 resulting in the ReSITL specification

$$\theta := Tr(\phi) = F_{(0,5)}\mu_{\text{R1}}^{\text{det}} \wedge G_{[0,\infty)}\left(\mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \left(\underline{F}_{(0,1)}\mu^{\text{uc}} \Rightarrow F_{(0,3)}\mu_{\text{R2}}^{\text{det}}\right)\right)$$

for which initially $(\boldsymbol{x}(0), \boldsymbol{s}^\perp, \tilde{\boldsymbol{\mu}}) \models \neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}}$ is assumed. For the construction of TST_S in Section V-A, we assume that we have control laws $\boldsymbol{u}_{\tilde{\delta}}(\boldsymbol{x}, t)$ that can accomplish each transition $\tilde{\delta}$ as per Definition 8 with $\tilde{g} := [1, \infty)$.

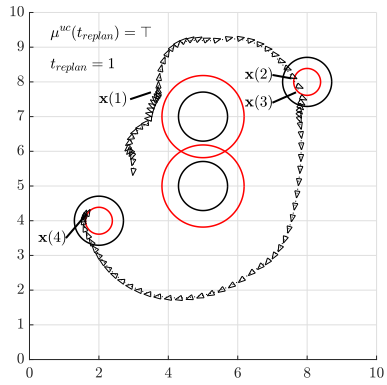


Fig. 6. Unicycle model for the ReRiSITL specification ϕ with $\zeta := 5$ and when an uncontrollable event occurs.

Setting 1: With respect to Assumption 2, we first assume that $\zeta := 1$. Recall that ζ determines the frequency by which the uncontrollable event μ^{uc} may occur. In this case, the set W does not contain the element $\overline{q}_0$, i.e., $\overline{q}_0 \notin W$, so that by Theorem 4 no plan $d_\mu(t)$ is found that satisfies θ and consequently ϕ. Note that this follows mainly since $\boldsymbol{s}(t) = \text{proj}_{\mu^{\text{uc}}}(s)(t) = \top$ may occur within ζ time unit intervals implying that, in the worst case, $\mu_{\text{R2}}^{\text{Ch}}$ should always be true so that there is no time to satisfy $\mu_{\text{R1}}^{\text{Ch}}$.

Setting 2: By increasing ζ, the frequency by which the uncontrollable event μ^{uc} may occur is decreased. We set $\zeta := 5$ and now observe that $\overline{q}_0 \in W$. The synthesized initial plan $d_\mu(t)$ is as follows:

$$d_\mu(t) := \begin{cases}\neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in (0, 4)\\ \mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in [4, 5.7]\\ \neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in (5.7, \infty).\end{cases}$$

However, now assume that $\boldsymbol{s}(1) = \text{proj}_{\mu^{\text{uc}}}(1) = \top$ so that at $t_{\text{replan}} = 1$ replanning is needed. Our revised plan then is

$$d_\mu(t) := \begin{cases}\neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in (0, 1)\\ \neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \mu^{\text{uc}} & t = 1\\ \neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in (1, 2)\\ \mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in [2, 3)\\ \neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \neg\mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in [3, 4)\\ \neg\mu_{\text{R1}}^{\text{det}} \wedge \mu_{\text{O1}}^{\text{det}} \wedge \mu_{\text{O2}}^{\text{det}} \wedge \mu_{\text{R2}}^{\text{det}} \wedge \neg\mu^{\text{uc}} & t \in [4, \infty)\end{cases}$$

i.e., to prepone satisfying $\mu_{\text{R1}}^{\text{det}}$ and to satisfy $\mu_{\text{R2}}^{\text{det}}$ right after and within 3 time units from when μ^{uc} happened. The simulation results for this case are depicted in Fig. 6.

Simulations were performed on a 1.4-GHz quad-core Intel Core i5 with 8 GB RAM. Construction of TST_θ and $\overline{RA}_C(\text{TST}_\theta)$ took 2.5 and 78.5 s, respectively, while Algorithm 2 and the graph search took 130 and 15.5 s, respectively. All MATLAB implementations and an animation of the simulation can be found in [52].

VIII. Conclusion

This article has presented ReRiSITL as a significant extension of STL. ReRiSITL additionally allows to consider the risk of not satisfying an ReRiSITL specification as well as allowing to consider environmental events such as sensor failures. We

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

have then proposed an algorithm to check if such an ReRiSITL specification is satisfiable. Lastly, we have proposed a reactive planning and control framework for dynamical systems under ReRiSITL specifications by combining a game-based approach with graph search techniques.

Acknowledgment

The authors would like to thank Prof. A. Girard for providing useful feedback in stating Defintion 2.

References

[1] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Proc. Int. Conf. FORMATS FTRTFT*, Grenoble, France, 2004, pp. 152–166.

[2] 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 *Proc. Conf. Decis. Control*, Los Angeles, CA, USA, 2014, pp. 81–87.

[3] Y. V. Pant, H. Abbas, R. A. Quaye, and R. Mangharam, "Fly-by-logic: Control of multi-drone fleets with temporal logic objectives," in *Proc. Int. Conf. Cyber- Phys. Syst.*, Porto, Portugal, 2018, pp. 186–197.

[4] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 96–101, Jan. 2019.

[5] Y. Kantaros and G. Pappas, "Optimal temporal logic planning for multi-robot systems in uncertain semantic maps," in *Proc. Int. Conf. Intel. Robots Syst.*, Macau, Hong Kong, 2019, pp. 4127–4132.

[6] J. Fu, N. Atanasov, U. Topcu, and G. J. Pappas, "Optimal temporal logic planning in probabilistic semantic maps," in *Proc. Int. Conf. Robot. Autom.*, Stockholm, Sweden, 2016, pp. 3690–3697.

[7] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. Soudjani, "Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3324–3331, Aug. 2019.

[8] S. Safaoui, L. Lindemann, D. V. Dimarogonas, I. Shames, and T. H. Summers, "Control design for risk-based signal temporal logic specifications," *IEEE Control Syst. Lett.*, vol. 4, no. 4, pp. 1000–1005, Oct. 2020.

[9] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Proc. Robotics: Sci. Syst.*, 2016. [Online]. Available: http://www.roboticsproceedings.org/rss12/p17.html

[10] D. Gundana and H. Kress-Gazit, "Event-based signal temporal logic synthesis for single and multi-robot tasks," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3687–3694, 2021.

[11] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, 2015, pp. 772–779.

[12] C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 2, pp. 115–140, 2019.

[13] N. Mehdipour, C.-I. Vasile, and C. Belta, "Arithmetic-geometric mean robustness for control from signal temporal logic specifications," in *Proc. Am. Control Conf.*, 2019, pp. 1690–1695.

[14] P. Varnai and D. V. Dimarogonas, "Prescribed performance control guided policy improvement for satisfying signal temporal logic tasks," in *Proc. Amer. Control Conf.*, 2019, pp. 286–291.

[15] D. Muniraj, K. G. Vamvoudakis, and M. Farhood, "Enforcing signal temporal logic specifications in multi-agent adversarial environments: A deep q-learning approach," in *Proc. Conf. Decis. Control*, 2018, pp. 4141–4146.

[16] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theor. Computer Sci.*, vol. 410, no. 42, pp. 4262–4291, 2009.

[17] L. Lindemann and D. V. Dimarogonas, "Efficient automata-based planning and control under spatio-temporal logic specifications," in *Proc. Amer Control Conf.*, 2020, pp. 4707–4714.

[18] C. K. Verginis and D. V. Dimarogonas, "Timed abstractions for distributed cooperative manipulation," *Auton. Robots*, vol. 42, no. 4, pp. 781–799, 2018.

[19] U. A. Fiaz and J. S. Baras, "Fast, composable rescue mission planning for UAVs using metric temporal logic," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15404–15411, 2020.

[20] K. Garg and D. Panagou, "Control-Lyapunov and control-barrier functions based quadratic program for spatio-temporal specifications," in *Proc. Conf. Decis. Control*, Nice, France, 2019, pp. 1422–1429.

[21] G. Yang, C. Belta, and R. Tron, "Continuous-time signal temporal logic planning with control barrier functions," in *Proc. Amer. Control Conf.*, 2020, pp. 4612–4618.

[22] B. Ramasubramanian, L. Niu, A. Clark, L. Bushnell, and R. Poovendran, "Linear temporal logic satisfaction in adversarial environments using secure control barrier certificates," in *Proc. Int. Conf. Decis. Game Theory Secur.*, Springer, 2019, pp. 385–403.

[23] L. Niu and A. Clark, "Control barrier functions for abstraction-free control synthesis under temporal logic constraints," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 816–823.

[24] C. N. Mavridis, C. Vrohidis, J. S. Baras, and K. J. Kyriakopoulos, "Robot navigation under MITL constraints using time-dependent vector field based control," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 232–237.

[25] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.

[26] Y. Kantaros and M. M. Zavlanos, "Sampling-based optimal control synthesis for multirobot systems under global temporal tasks," *IEEE Trans. Autom. Control*, vol. 64, no. 5, pp. 1916–1931, May 2019.

[27] Y. E. Sahin, P. Nilsson, and N. Ozay, "Synchronous and asynchronous multi-agent coordination with cLTL constraints," in *Proc. Conf. Decis. Control*, 2017, pp. 335–342.

[28] R. Alur and T. A. Henzinger, "The benefits of relaxing punctuality," *J. ACM*, vol. 43, no. 1, pp. 116–146, 1996.

[29] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994.

[30] T. Brihaye, G. Geeraerts, H.-M. Ho, and B. Monmege, "Mighty L: A compositional translation from MITL to timed automata," in *Proc. Int. Conf. Comp. Aided Verification*, Heidelberg, Germany, 2017, pp. 421–440.

[31] O. Maler, D. Nickovic, and A. Pnueli, "From MITL to timed automata," in *Proc. Int. Conf. Formal Model. Anal. Timed Syst.*, Paris, France, 2006, pp. 274–289.

[32] T. Ferrére, O. Maler, D. Ničković, and A. Pnueli, "From real-time logic to timed automata," *J. ACM (JACM)*, vol. 66, no. 3, p. 19, 2019.

[33] O. Maler, A. Pnueli, and J. Sifakis, "On the synthesis of discrete controllers for timed systems," in *Proc. Annu. Symp. Theor. Aspects Comput. Sci.*, Springer, 1995, pp. 229–242.

[34] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis, "Controller synthesis for timed automata," *IFAC Proc. Volumes*, vol. 31, no. 18, pp. 447–452, 1998.

[35] E. Asarin, O. Maler, and A. Pnueli, "Symbolic controller synthesis for discrete and timed systems," in *Proc. Int. Hybrid Syst. Workshop*, 1994, pp. 1–20.

[36] E. Asarin and O. Maler, "As soon as possible: Time optimal control for timed automata," in *Proc. Int. Workshop Hybrid Syst.: Comput. Control*, Springer, 1999, pp. 19–30.

[37] M. Lahijanian, M. R. Maly, D. Fried, L. E. Kavraki, H. Kress-Gazit, and M. Y. Vardi, "Iterative temporal planning in uncertain environments with partial satisfaction guarantees," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 583–599, Jun. 2016.

[38] S. Bharadwaj, R. Dimitrova, and U. Topcu, "Synthesis of surveillance strategies via belief abstraction," in *Proc. Conf. Decis. Control*, 2018, pp. 4159–4166.

[39] M. Guo and M. M. Zavlanos, "Probabilistic motion planning under temporal tasks and soft constraints," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4051–4066, Dec. 2018.

[40] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications," in *Proc. Conf. Decis. Control*, 2016, pp. 7419–7424.

[41] R. T. Rockafellar *et al.*, "Optimization of conditional value-at-risk," *J. Risk*, vol. 2, pp. 21–42, 2000.

[42] A. Majumdar and M. Pavone, "How should a robot assess risk? Towards an axiomatic theory of risk in robotics," in *Robotics Research*. Berlin, Germany: Springer, 2020, pp. 75–84.

[43] N. Piterman, A. Pnueli, and Y. Sa'ar, "Synthesis of reactive (1) designs," in *Proc. Int. Workshop VMCAI*, 2006, pp. 364–380.

[44] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1370–1381, Dec. 2009.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.

[45] L. Lindemann, G. J. Pappas, and D. V. Dimarogonas, "Control barrier functions for nonholonomic systems under risk signal temporal logic specifications," in *Proc. Conf. Decis. Control*, 2020, pp. 1422–1428.
[46] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proc. 18th Int. Conf. Hybrid Syst.: Comput. Control*, 2015, pp. 239–248.
[47] H. P. Williams, *Model Building in Mathematical Programming*, 5th ed. Hoboken, NJ, USA: Wiley, 2013.
[48] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
[49] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis, "Memory-efficient algorithms for the verification of temporal properties," *Formal Methods Syst. Des.*, vol. 1, no. 2/3, pp. 275–288, 1992.
[50] H. Tauriainen, "Nested emptiness search for generalized Büchi automata," *Fundamenta Informaticae*, vol. 70, no. 1/2, pp. 127–154, 2006.
[51] R. Gómez and H. Bowman, "Efficient detection of Zeno runs in timed automata," in *Proc. Int. Conf. Formal Model. Anal. Timed Syst.*, 2007, pp. 195–210.
[52] L. Lindemann, "Code: Reactive timed automata-based planning," [Online]. Available: https://github.com/Lindemann1989/Reactive-and-Risk-Aware-Control-for-Signal-Temporal-Logic.git



Lars Lindemann (Member, IEEE) received the B.Sc. degree in electrical and information engineering and the second B.Sc. degree in engineering management from Christian-Albrechts-University (CAU), Kiel, Germany, in 2014, the M.Sc. degree in systems, control and robotics and the Ph.D. degree in electrical engineering from KTH Royal Institute of Technology, Stockholm, Sweden, in 2016 and 2020.

He is currently a Postdoctoral Researcher with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He is a Member of the GRASP Lab. His research interests include control theory, formal methods, machine learning, multiagent systems, and autonomous systems.

Dr. Lindemann was the recipient of the Outstanding Student Paper Award at the 58th IEEE Conference on Decision and Control and was a Best Student Paper Award Finalist at the 2018 American Control Conference.



George J. Pappas (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998.

He is currently the UPS Foundation Professor and Chair of the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA, where he also holds a secondary appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics. He is a Member of the GRASP Lab and the PRECISE Center. He had previously served as the Deputy Dean for Research with the School of Engineering and Applied Science. His research interests include control theory and, in particular, hybrid systems, embedded systems, cyber-physical systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks.

Dr. Pappas was the recipient of various awards, such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, the Hugo Schuck Best Paper Award, the George H. Heilmeier Award, the National Science Foundation PECASE Award, and numerous best student papers awards at American Control Conference, IEEE Conference on Decision and Control, and International Conference on Cyber-Physical Systems.



Dimos V. Dimarogonas (Senior Member, IEEE) was born in Athens, Greece, in 1978. He received the diploma in electrical and computer engineering and the Ph.D. degree in mechanical engineering from National Technical University of Athens (NTUA), Athens, Greece, in 2001 and 2007.

Between 2007 and 2010, he held Postdoctoral positions with the KTH Royal Institute of Technology, Stockholm, Sweden, Department of Automatic Control and MIT, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH. His research interests include multi-agent systems, hybrid systems and control, robot navigation and manipulation, human-robot-interaction, and networked control.

Dr. Dimarogonas was the recipient of the ERC Starting Grant, in 2014, the ERC Consolidator Grant, in 2019, and the Knut och Alice Wallenberg Academy Fellowship, in 2015. He serves on the Editorial Board of *Automatica* and the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:45:31 UTC from IEEE Xplore. Restrictions apply.