2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Robustness Measures and Monitors for Time Window Temporal Logic

Ahmad Ahmad[1], Cristian-Ioan Vasile[2], Roberto Tron[1], and Calin Belta[1]

***Abstract*— Temporal logics (TLs) have been widely used to formalize interpretable tasks for cyber-physical systems. Time Window Temporal Logic (TWTL) has been recently proposed as a specification language for dynamical systems. In particular, it can easily express robotic tasks, and it allows for efficient, automata-based verification and synthesis of control policies for such systems. In this paper, we define two quantitative semantics for this logic, and two corresponding monitoring algorithms, which allow for real-time quantification of satisfaction of formulas by trajectories of discrete-time systems. We demonstrate the new semantics and their runtime monitors on numerical examples.**

I. Introduction

Temporal logics (TLs) [1] have been widely used to formulate high-level, expressive specifications for cyber-physical systems. Formal verification and synthesis algorithms have been employed to analyze and control such systems from TL specifications. In particular, Linear Temporal Logic (LTL) [2] has been employed to specify tasks for planning problems [3], [4], [5], [6] and for formal synthesis problems for discrete-time systems [7]. LTL formulas can be translated to automata, which can encode the progress towards task satisfaction. Automata-theoretic tools are typically used with finite abstractions of the system to produce policies that guarantee the satisfaction of tasks, or prove that they cannot be satisfied [3], [7], [8], [9]. Other approaches overcome some scalability issues by sampling-based planning algorithms guided by the specifications automaton, see [5], [6] where the authors use RRT* [10] as the planning algorithm, and in [11], [12] the authors use RRG [10] as the planning algorithm.

Signal Temporal Logic (STL) [13], Metric Temporal Logic (MTL) [14], and Time Window Temporal logic (TWTL) [15], unlike LTL, can express specifications with explicit, concrete-time constraints, e.g., *Perform task A between times t_1 and t_2 ; right after that, spend t_5 time units between times t_3 and t_4 performing task B; and for all times do not perform task C.*

The semantics of both STL and MTL are defined over real-time signals. They both have quantitative semantics, or robustness, which quantifies the degree of satisfaction of a formula by a signal [16], [17]. Most existing works that use STL and MTL for specifications find controllers by maximizing robustness, yielding runs of the system that robustly satisfy the specifications [18], [19], [17], [20], [21]. The work in [22] considers planning for syntactically co-safe LTL using RRT*, in addition to the task specifications, other spatial requirements are expressed using fragment-STL where its robustness is used as the optimality criterion for RRT*. In [23], the authors synthesize controllers for time-critical systems for which they quantify a temporal robustness measure that needs to be optimized. The traditional robustness metric is not differentiable and it is mostly determined by one value of the signal, i.e., it "masks" most of the signal. These issues are addressed by the authors of [24], who introduced an arithmetic and geometric mean (AGM) robustness measure for STL.

TWTL has several advantages over STL, MTL, and other concrete-time TLs. First, its syntax and semantics can express serial tasks in an efficient and explicit way. This is important in many applications, especially in robotics [15]. Second, TWTL formulae can be efficiently translated into automata. The complexity of the translation algorithm is independent of the formula time bounds [15]. This makes this logic suitable for automata-based synthesis and planning problems (see [25] for a planning application). TWTL, however, lacks quantitative semantics that measures the degree of satisfaction or violation of a formula. In this work, we modify the syntax of TWTL and allow it to be defined over predicated regions of the system output space. We define robustness measures to quantify the degree of satisfaction of TWTL formulae, and inspired by [24], we extend the robustness definition to one in which we utilize the notion of AGM robustness. This enables planning and synthesis problems, which we plan to address in future follow-up work.

Our contributions are summarized as follows. First, we adapt the "traditional" quantitative semantics of STL to define a notion of sound robustness metric (Sec. III-A). Second, inspired by the AGM-STL robustness [24], we introduce an AGM robustness measure for TWTL, which is amenable for a wide spectrum of applications (Sec. III-B). Third, given partial runs of the system, i.e. runs with lengths less than the time horizon of a TWTL formula (see Definition 2.1), we tailor the STL robustness interval semantics [26] to monitor the TWTL robustness (Sec. IV). Fourth, we introduce a similar interval sematics to monitor the AGM robustness at runtime. Finally, we validate the proposed robustness measures and their monitors in numerical examples (Sec. V).

This work was partially supported by the MIT Lincoln Laboratory and by the NSF under grants IIS-2024606 and GCR 2219101 at Boston University.

[1] Ahmad Ahmad, Roberto Tron, and Calin Belta ({ahmadgh,tron,cbelta}@bu.edu) are with the Division of System Engineering, Boston University, Boston, MA 02215, USA.

[2] Cristian-Ioan Vasile (cvasile@lehigh.edu) is with Mechanical Engineering and Mechanics Department at Lehigh University, Bethlehem, PA, 18015, USA.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

II. PRELIMINARIES

A. Dynamical System

Consider a discrete-time nonlinear system in the form

$$\begin{aligned} x_{t+\Delta t} &= f(x_t),\ t = t_0, t_0+\Delta t, t_0+2\Delta t \dots \\ o_t &= l(x_t), \end{aligned} \tag{1}$$

where $x \in X \subset \mathbb{R}^d$ is the state taking values in a set X, $\mathbb{R}^d$ is the d-dimensional Euclidean space, $\Delta t \in \mathbb{R}_{>0}$, and $f : \mathbb{R}^d \to \mathbb{R}^d$. o_t is an observable output of the system at time t, and $l(.) : X \to 2^\Pi$ is a labeling function where Π is a set of atomic propositions (tasks) and 2^Π is its power set.

A state trajectory of system (1) is a sequence of states $\mathbf{x} := x_{t_0} x_{t_0+\Delta t} x_{t_0+2\Delta t} \dots$ that satisfy its dynamics. In this work, an atomic proposition $\pi_A \in \Pi$ takes the Boolean value $\top$ at state $x \in X$ if $o := l(x) \in A$ where $A := \{o | h(o) > \sigma\}$, $h : 2^\Pi \to \mathbb{R}^d$, $\sigma \in \mathbb{R}^d$, and $\bot$ otherwise. $\mathbf{x}$ generates a word, $\mathbf{o} = o_{t_0} o_{t_0+\Delta t} o_{t_0+2\Delta t} \dots \dots$. For $t_1, t_2 \in \mathbb{R}_{\geq 0};\ t_2 := t_1 + n_t \Delta t;$ where $n_t \in \mathbb{N}_{\geq 0}$, we denote the corresponding system trajectory and the generated word, respectively, as the following, $\mathbf{x}_{t_1,t_2} := x_{t_1} x_{t_1+\Delta t} \dots x_{t_2}$ and $\mathbf{o}_{t_1,t_2} := o_{t_1} o_{t_1+\Delta t} \dots o_{t_2}$. For $t \in [t_1, t_2]$, $\mathbf{x}_{t_1,t_2}(t) = x_t$ and $\mathbf{o}_{t_1,t_2}(t) := o_t$.

B. Time Window Temporal Logic

We modify the TWTL syntax in [15] such that the atomic propositions are defined over predicated regions. The TWTL syntax is defined inductively as follows:

$$\phi \ ::= \ H^d \pi_A | H^d \neg \pi_A | \phi_1 \wedge \phi_2 | \phi_1 \vee \phi_2 | \neg \phi | \phi_1 \cdot \phi_2 | [\phi]^{[a,b]} \tag{2}$$

where $\pi_A \in \Pi$ is an atomic proposition defined over the predicated region A; $\neg$, $\wedge$, and $\vee$ are the negation, conjunction, and disjunction Boolean operators, respectively; H^d is the *hold* operator; $\cdot$ is the *concatenation* operator; and $[]^{[a,b]}$ is the *within* operator, where $d, a, b \in \mathbb{Z}_{\geq 0}$ and $a \geq b$.

The Boolean semantics over a word $\mathbf{o}_{t_1,t_2}$ is defined recursively as follows:

$$\begin{aligned}
&\mathbf{o}_{t_1,t_2} \models H^d \pi_A \Leftrightarrow o_t \in A,\ \forall t \in [t_1, t_2] \wedge (t_2 - t_1 \geq d\Delta t) \\
&\mathbf{o}_{t_1,t_2} \models H^d \neg \pi_A \Leftrightarrow o_t \notin A,\ \forall t \in [t_1, t_2] \wedge (t_2 - t_1 \geq d\Delta t) \\
&\mathbf{o}_{t_1,t_2} \models \phi_1 \wedge \phi_2 \Leftrightarrow (\mathbf{o}_{t_1,t_2} \models \phi_1) \wedge (\mathbf{o}_{t_1,t_2} \models \phi_2) \\
&\mathbf{o}_{t_1,t_2} \models \phi_1 \vee \phi_2 \Leftrightarrow (\mathbf{o}_{t_1,t_2} \models \phi_1) \vee (\mathbf{o}_{t_1,t_2} \models \phi_2) \\
&\mathbf{o}_{t_1,t_2} \models \neg \phi \Leftrightarrow \mathbf{o}_{t_1,t_2} \not\models \phi \\
&\mathbf{o}_{t_1,t_2} \models \phi_1 \cdot \phi_2 \Leftrightarrow \exists t = \underset{t \in [t_1,t_2]}{\arg\min} \{\mathbf{o}_{t_1,t} \models \phi_1\} \\
&\qquad \wedge (\mathbf{o}_{t+\Delta t,t_2} \models \phi_2) \\
&\mathbf{o}_{t_1,t_2} \models [\phi]^{[a,b]} \Leftrightarrow \exists t \geq t_1 + a \text{ s.t. } \mathbf{o}_{t,t_1+b} \models \phi \\
&\qquad \wedge (t_2 - t_1 \geq b)
\end{aligned} \tag{3}$$

Definition 2.1 (TWTL Time Horizon [15]): Given ϕ, the time horizon is defined recursively as follows.

$$||\phi|| := \begin{cases} \max(||\phi_1||, ||\phi_2||); & \text{if } \phi \in \{\phi_1 \wedge \phi_2, \phi_1 \vee \phi_2\} \\ ||\phi_1||; & \text{if } \phi = \neg\phi_1 \\ ||\phi_1|| + ||\phi_2|| + \Delta t; & \text{if } \phi = \phi_1 \cdot \phi_2 \\ d\Delta t; & \text{if } \phi = H^d \pi_A \\ b; & \text{if } \phi = [\phi_1]^{[a,b]} \end{cases} \tag{4}$$

III. TWTL QUANTITATIVE SEMANTICS

Inspired by STL robustness [16] and its AGM version [24], we tailor robustness measures for TWTL formulae to reason about the degree of satisfaction of TWTL tasks.

A. TWTL Robustness

Definition 3.1: (TWTL Robustness) Given a TWTL formula ϕ and an output word $\mathbf{o}_{t_1,t_2}$ of system (1), we define the robustness degree $\rho(\mathbf{o}_{t_1,t_2}, \phi)$ at time 0, recursively, as follows:

$$\begin{aligned}
&\rho(\mathbf{o}_{t_1,t_2}, H^d \pi_A) := \begin{cases} \min\limits_{t \in [t_1, d+t_1]} h(o_t) & ; (t_2 - t_1 \geq d\Delta t) \\ \rho_\bot & ; \text{otherwise} \end{cases} \\
&\rho(\mathbf{o}_{t_1,t_2}, \phi_1 \wedge \phi_2) := \min\{\rho(\mathbf{o}_{t_1,t_2}, \phi_1), \rho(\mathbf{o}_{t_1,t_2}, \phi_2)\} \\
&\rho(\mathbf{o}_{t_1,t_2}, \phi_1 \vee \phi_2) := \max\{\rho(\mathbf{o}_{t_1,t_2}, \phi_1), \rho(\mathbf{o}_{t_1,t_2}, \phi_2)\} \\
&\rho(\mathbf{o}_{t_1,t_2}, \neg\phi) = -\rho(\mathbf{o}_{t_1,t_2}, \phi) \\
&\rho(\mathbf{o}_{t_1,t_2}, \phi_1 \cdot \phi_2) := \\
&\quad \max_{t \in [t_1, t_2)} \{\min\{\rho(\mathbf{o}_{t_1,t}, \phi_1), \rho(\mathbf{o}_{t+1,t_2}, \phi_2)\}\} \\
&\rho(\mathbf{o}_{t_1,t_2}, [\phi]^{[a,b]}) := \begin{cases} \max\limits_{t \geq t_1 + a} \{\rho(\mathbf{o}_{t,t_1+b}, \phi)\}; (t_2 - t_1 \geq b) \\ \rho_\bot;\ \text{otherwise} \end{cases}
\end{aligned} \tag{5}$$

where $\rho_\bot$ denotes a large negative value that indicates the robustness of Boolean $\bot$.
For $\mathbf{o}_{t_1,t_2}$ and ϕ, the robustness value $\rho(\mathbf{o}_{t+1,t_2}, \phi)$ indicates how far is $\mathbf{o}_{t_1,t_2}$ from the decision boundary of the predicated region of the task. A positive $\rho(\mathbf{o}_{t+1,t_2}, \phi)$ implies the Boolean satisfaction of the task, where the greater the value the more robustly $\rho(\mathbf{o}_{t+1,t_2}, \phi)$ satisfies the task. A similar argument can be made for negative robustness for violation of a task.

Lemma 3.1: TWTL robustness (5) is sound, i.e., the Boolean satisfaction (violation) is implied by a positive (negative) robustness value.

Proof: See the extended version [27]. ■

Example 3.1: Consider a TWTL formula $\phi = [H^6 \pi_A]^{[0,10]}$, where $A = \{o \mid o \geq 4\}$, which reads as: "*Within* time 0 and time 10, *hold* in π_A for 6 time steps"; and three output words $\mathbf{o}_1$, $\mathbf{o}_2$ and $\mathbf{o}_3$, which are depicted as blue, green, and red traces in the top-left figure of Fig. 1, respectively. One can see that $\mathbf{o}_1$ and $\mathbf{o}_2$ satisfy the task specification where $\rho(\mathbf{o}_1, \phi) = \rho(\mathbf{o}_2, \phi) = 0.099$, whereas $\mathbf{o}_2$ violates the task, where $\rho(\mathbf{o}_2, \phi) = -2$.

B. TWTL Arithmetic and Geometric Mean Robustness

TWTL robustness (Definition 3.1) accounts for the most critical points of the system output word, which is necessary for the soundness of the robustness (see Lemma 3.1). However, it may be very pessimistic robustness measure as highlighted in Example 3.2. for instance, the computation of the robustness $\rho(\mathbf{o}_{t_1,t_2}, H^d\pi_A)$ is dominated by the minimum valuation of the predicate function $h(\cdot)$ over the system word $\mathbf{o}$. Moreover, since its computation involves $\min$ and $\max$, it leads to a non-smooth measure which is computationally challenging for heuristic- and gradient-based approaches to maximizes the robustness of the overall task.

To this end, we tailor the notion of AGM robustness [24] to define the AGM robustness measure for TWTL η. As we show in the following, η helps mitigate some of the shortcomings of TWTL robustness ρ and provides a more optimistic, smooth and sound robustness measure for TWTL.

Consider the function $F:\mathbb{R}\rightarrow\mathbb{R}$, and let $[F]_+ := \begin{cases} F;\ F>0 \\ 0;\ \text{otherwise} \end{cases}$ and $[F]_- = -[-F]_+$, where $F = [F]_+ + [F]_-$. We define AGM functions of disjunction and conjunction of $r_i \in \mathbb{R}$, $i=1,\ldots,N$, respectively, as follows.

$$\mathrm{AGM}_\vee(r_1,\ldots,r_N) := \begin{cases} -\sqrt[N]{\prod_{i=1}^{N}(1-r_i)}+1; \\ \quad \text{if } \forall i\in\{1,\ldots,N\}, r_i<0 \\ \frac{1}{N}\sum_{i=1}^{N}[r_i]_+; \quad \text{otherwise} \end{cases} \tag{6}$$

$$\mathrm{AGM}_\wedge(r_1,\ldots,r_N) := \begin{cases} \sqrt[N]{\prod_{i=1}^{N}(1+r_i)}-1; \\ \quad \text{if } \forall i\in\{1,\ldots,N\}, r_i>0 \\ \frac{1}{N}\sum_{i=1}^{N}[r_i]_-; \quad \text{otherwise} \end{cases} \tag{7}$$

Definition 3.2 (Normalized TWTL formulae): Given syntax (2), a normalized TWTL formula ϕ is presented as the formulae defined over normalized atomic propositions $\pi_{A_{\mathrm{norm}}}$, where $A_{\mathrm{norm}} := \{o \mid h_{\mathrm{norm}}(o) > \sigma_n\}$, $h_{\mathrm{norm}} : 2^{\Pi} \rightarrow [-1,1]^d$, and $\sigma_{\mathrm{norm}} \in [-1,1]$.

Throughout the rest of the paper, we assume that all TWTL formulae are normalized unless explicitly stated otherwise.

Definition 3.3: (TWTL Arithmetic-Geometric Mean Robustness) Given a normalized TWTL formula ϕ, we define the AGM robustness of the output word $\mathbf{o}_{t_1,t_2}$ with respect to ϕ, recursively, using (6) and (7).

$$\begin{aligned} &\eta(\mathbf{o}_{t_1,t_2},\top) := +1 \\ &\eta(\mathbf{o}_{t_1,t_2},\bot) := -1 \\ &\eta(\mathbf{o}_t,\pi_A) := \frac{1}{2}(h(\mathbf{o}_t)-\sigma_n) \\ &\eta(\mathbf{o}_{t_1,t_2},\phi_1\wedge\phi_2) := \mathrm{AGM}_\wedge(\eta(\mathbf{o}_{t_1,t_2},\phi_1),\eta(\mathbf{o}_{t_1,t_2},\phi_2)) \\ &\eta(\mathbf{o}_{t_1,t_2},\phi_1\vee\phi_1) := \mathrm{AGM}_\vee(\eta(\mathbf{o}_{t_1,t_2},\phi_1),\eta(\mathbf{o}_{t_1,t_2},\phi_2)) \\ &\eta(\mathbf{o}_{t_1,t_2},\neg\phi) := -\eta(\mathbf{o}_{t_1,t_2},\phi) \end{aligned} \tag{8}$$

$$\begin{aligned} &\eta(\mathbf{o}_{t_1,t_2},H^d\pi_A) := \begin{cases} \mathrm{AGM}_\wedge(\eta(\mathbf{o}_t,\pi_A)|t\in[t_1,t_1+d\Delta t]);\ \text{if } (t_2-t_1)\geq d \\ -1;\ \text{otherwise} \end{cases} \\ &\eta(\mathbf{o}_{t_1,t_2},[\phi]^{[a,b]}) := \begin{cases} \mathrm{AGM}_\vee(\eta(\mathbf{o}_{t,t_1+b},\phi)|t\in[t_1+a,t_1+b]);\ \text{if } (t_2-t_1)\geq b \\ -1;\ \text{otherwise} \end{cases} \\ &\eta(\mathbf{o}_{t_1,t_2},\phi_1.\phi_2) := \mathrm{AGM}_\vee(\mathrm{AGM}_\wedge(\eta(\mathbf{o}_{t_1,t},\phi_1),\eta(\mathbf{o}_{t+\Delta t,t_2},\phi_2))|t\in[t_1,t_2)) \end{aligned} \tag{9}$$

Theorem 3.1: TWTL AGM robustness, Definition 3.3, is sound. Formally, we have

$$\eta(\mathbf{o}_{t_1,t_2},\phi)>0 \Leftrightarrow \rho(\mathbf{o}_{t_1,t_2},\phi)>0 \implies \mathbf{o}_{t_1,t_2}\models\phi$$
$$\eta(\mathbf{o}_{t_1,t_2},\phi)<0 \Leftrightarrow \rho(\mathbf{o}_{t_1,t_2},\phi)<0 \implies \mathbf{o}_{t_1,t_2}\not\models\phi$$

Proof: See the extended version [27]. ■

Example 3.2: (Continued) Consider the same formula and words of Example 3.1. $\eta(\mathbf{o}_1,\phi)=0.061$, $\eta(\mathbf{o}_2,\phi)=0.010$, where, unlike their ρ value, the AGM robustness measure η rewards words with more satisfying valuations instead of being dominated by the most critical valuations while also preserving the soundness property. The word $\mathbf{o}_1$ (the blue trace in top-left figure of Fig. 1) has more valuations that robustly contribute to satisfying the formula. In $\rho(\mathbf{o}_1,\phi)$, the 4^{th} point of the trace, which is close to lead to violating the task, dominates the robustness computation. Even if we assume that the 4^{th} is 2 (which would lead to violating the task), its η value would be -0.35 that is higher than its corresponding ρ value, -2. The computation of η considers the fact that the trace has promising valuations which contribute to satisfying the task. For $\mathbf{o}_3$ (the red trace in the same figure), on the other hand, $\eta(\mathbf{o}_3,\phi)=-0.61$ which is higher than $\rho(\mathbf{o}_3,\phi) = -2$, is more realistic violation measure given that some valuations of $\mathbf{o}_3$ are close to contributing in satisfying the task.

IV. Runtime Robustness Monitoring

Considering runs of the system with time horizon less than TL specifications time horizon, runtime verification techniques are introduced as light weight algorithms to monitor the satisfaction given such partial runs, see [28] for review on monitoring different TLs. Different techniques are usually utilized for the monitoring task, the work in [29] uses a rewriting technique to monitor the Boolean satisfaction of TWTL. In [26], the authors introduced interval semantics to monitor the robustness degree of STL specifications. The technique considers partial runs, and with the set of all possible completions of the run, it computes the best and worst possible robustness.

In this work, we tailor the STL robustness monitor from [26] to encode a robustness interval $[\rho]$ to monitor TWTL robustness. Similarly, we introduce an AGM robustness interval $[\eta]$ to monitor the AGM robustness. Our monitors are sound, which means the correct (AGM) robustness belongs to the produced interval of the runtime monitor at any time step.

Let us consider some preliminary definitions that we use in our TWTL interval semantics.

Definition 4.1 (Prefix, Completions): Consider the time horizon $||\phi||$ and output words $\mathbf{o}_{t_1,t'}$ and $\mathbf{o}_{t_1,t_2}$, where $t' < ||\phi||$ and $t_2 \geq ||\phi||$. We denote $\mathbf{o}_{t_1,t'}$ as a prefix of $\mathbf{o}_{t_1,t_2}$ if $\forall t \in [t_1,t'], \mathbf{o}_{t_1,t_2}(t) = \mathbf{o}_{t_1,t'}(t)$; consequently we define a set of all possible completions of a prefix as $\mathfrak{C} := \{\mathbf{o}_{t_1,t_2} \mid \mathbf{o}_{t_1,t'} \text{ is a prefix of } \mathbf{o}_{t_1,t_2}\}$

Definition 4.2 (Arithmetics on interval semantics): Consider the following set of intervals $\mathbf{I} := \{I_i\}_{i=1}^N$, where $I_i := [\underline{I}_i, \bar{I}_i]$ and $\underline{I}_i \leq \bar{I}_i$. We define following arithmetics over $\mathbf{I}$

$$\begin{aligned} \mathbf{max}(\mathbf{I}) &:= [\max(\underline{I}_1,\dots,\underline{I}_N), \max(\bar{I}_1,\dots,\bar{I}_N)], \\ \mathbf{min}(\mathbf{I}) &:= [\min(\underline{I}_1,\dots,\underline{I}_N), \min(\bar{I}_1,\dots,\bar{I}_N)]. \end{aligned} \tag{10}$$

$$\begin{aligned} \mathbf{AGM}_\vee(\mathbf{I}) &:= [\mathrm{AGM}_\vee(\underline{I}_1,\dots,\underline{I}_N), \mathrm{AGM}_\vee(\bar{I}_1,\dots,\bar{I}_N)], \\ \mathbf{AGM}_\wedge(\mathbf{I}) &:= [\mathrm{AGM}_\wedge(\underline{I}_1,\dots,\underline{I}_N), \mathrm{AGM}_\wedge(\bar{I}_1,\dots,\bar{I}_N)]. \end{aligned} \tag{11}$$

The singleton interval $[I, I]$ is denoted by $\{I\}$.

Before introducing the recursive definition of $[\rho]$ and $[\eta]$ we introduce the following definition.

Definition 4.3: (Compact Representation of Set of Intervals) Given words $\mathbf{o}_{t_1,t'}$, $\mathbf{o}_{t,t'}$, $t \in [t_a,t_b]$, and formulae ϕ and ϕ_i, $i = 1,\dots,N$, for $[\rho]$ and $[\eta]$ we denote the following set representation:

$$\begin{aligned} [.]^{i=1,\dots,N}(\mathbf{o}_{t_1,t'},\phi_i) &= \{[.](\mathbf{o}_{t_1,t'},\phi_1),\dots,[.](\mathbf{o}_{t_1,t'},\phi_N)\}, \\ [.]_{t_a:t_b}(\mathbf{o}_{t,t'},\phi) &= \{[.](\mathbf{o}_{t_a,t'},\phi_1),\dots,[.](\mathbf{o}_{t_b,t'},\phi)\}, \end{aligned} \tag{12}$$

where $[.](\mathbf{o_{t_1,t_2}},\phi)$ is the interval semantics defined next.

Consider a TWTL formula ϕ with time horizon $||\phi||$. Given a partial word $\mathbf{o}_{t_1,t'}$, where $t' \leq ||\phi||$, for monitoring the robustness ρ at time t', we define the robustness interval $[\rho]$ recursively as follows.

$$\begin{aligned} &[\rho](\mathbf{o}_{t_1,t_2}, H^d\pi_A) := \begin{cases} \{\rho(\mathbf{o}_{t_1,t_2}, H^d\pi_A)\}; \ (t_2 - t_1 \geq d) \\ [\rho_\perp, \min\limits_{t\in[t_1,d+t_1]} h(o_t)]; \ \text{otherwise} \end{cases} \\ &[\rho](\mathbf{o}_{t_1,t_2}, \phi_1 \wedge \phi_2) := \mathbf{min}([\rho](\mathbf{o}_{t_1,t_2},\phi_1), [\rho](\mathbf{o}_{t_1,t_2},\phi_2)) \\ &[\rho](\mathbf{o}_{t_1,t_2}, \phi_1 \vee \phi_2) := \mathbf{max}([\rho](\mathbf{o}_{t_1,t_2},\phi_1), [\rho](\mathbf{o}_{t_1,t_2},\phi_2)) \\ &[\rho](\mathbf{o}_{t_1,t_2}, \phi_1 \cdot \phi_2) := \\ &\mathbf{max}_{t\in[t_1,t_2)}(\mathbf{min}([\rho](\mathbf{o}_{t_1,t},\phi_1), [\rho](\mathbf{o}_{t+1,t_2},\phi_2))) \\ &[\rho](\mathbf{o}_{t_1,t_2}, [\phi]^{[a,b]}) := \begin{cases} \{\rho(\mathbf{o}_{t_1,t_2}, [\phi]^{[a,b]}\}; \ (t_2 - t_1 \geq b) \\ [\max\limits_{t\geq t_1+a} \{\rho(\mathbf{o}_{t,t_1+b},\phi)\}, \rho_\top]; \\ \quad \text{otherwise} \end{cases} \end{aligned} \tag{13}$$

Considering the same specification and word, for monitoring the AGM robustness η at time t', we define the robustness interval $[\eta]$, recursively, using (14),(15).

$$\begin{aligned} &[\eta](\mathbf{o}_{t_1,t'}, \phi_1 \wedge \phi_2, t') := \mathbf{AGM}_\wedge([\eta](\mathbf{o}_{t_1,t'},\phi_i)^{i=1,2}) \\ &[\eta](\mathbf{o}_{t_1,t'}, \phi_1 \vee \phi_2, t') := \mathbf{AGM}_\vee([\eta](\mathbf{o}_{t_1,t'},\phi_i)^{i=1,2}) \\ &[\eta](\mathbf{o}_{t_1,t'}, H^d\pi_A, t') := [\underline{\eta}, \bar{\eta}] \end{aligned} \tag{14}$$

$$\begin{aligned} &\bar{\eta} := \begin{cases} \sqrt[d+1]{\prod\limits_{t\in[t_1,t']} (1+\eta(\mathbf{o}_t,\pi_A))(1+\eta_{\max})^{|[t',d]|}} - 1; & \text{if } \forall t \in [t_1,t'], \eta(\mathbf{o}_t,\pi_A) > 0 \wedge (t'-t_1) < d \\ \frac{1}{d+1} \sum\limits_{t\in[t_1,t']} [\eta(\mathbf{o}_t,\pi_A)]_-; & \text{if } \exists t \in [t_1,t'], \eta(\mathbf{o}_t,\pi_A) < 0 \wedge (t'-t_1) < d \\ \eta(\mathbf{o}_{t_1,t'}, H^d\pi_A); & \text{If } (t'-t_1) \geq d \end{cases} \\ &\underline{\eta} := \begin{cases} \frac{|(t',d]|}{|[t_1,d]|}\eta_{\min}; & \text{if } \forall t \in [t_1,t'], \eta(\mathbf{o}_t,\pi_A) > 0 \wedge (t'-t_1) < d \\ \frac{1}{d+1}\left(\sum\limits_{t\in[t_1,t']} [\eta(\mathbf{o}_t,\pi_A)]_- + |[t',d]|\eta_{\min} \right); & \text{if } \exists t \in [t_1,t'], \eta(\mathbf{o}_t,\pi_A) < 0 \wedge (t'-t_1) < d \\ \eta(\mathbf{o}_{t_1,t'}, H^d\pi_A); & \text{If } (t'-t_1) \geq d \end{cases} \\ &[\eta](\mathbf{o}_{t_1,t'}, [\phi]^{[a,b]}, t') := \begin{cases} \{\eta(\mathbf{o}_{t_1,t'}, [\phi]^{[a,b]})\}; & \text{if } (t'-t_1) \geq b \\ \mathbf{AGM}_\vee([\eta](\mathbf{o}_{t,t_1+b},\phi)_{t_1+a:t_2}); & \text{otherwise} \end{cases} \\ &[\eta](\mathbf{o}_{t_1,t'}, \phi_1.\phi_2, t') := \mathbf{AGM}_\vee(\mathbf{AGM}_\wedge([\eta](\mathbf{o}_{t_1,t'-\Delta t},\phi_1), [\eta](\mathbf{o}_{t',t'},\phi_2)),\dots,\mathbf{AGM}_\wedge([\eta](\mathbf{o}_{t_1,t_1},\phi_1), [\eta](\mathbf{o}_{t_1+\Delta t,t'},\phi_2))) \end{aligned} \tag{15}$$

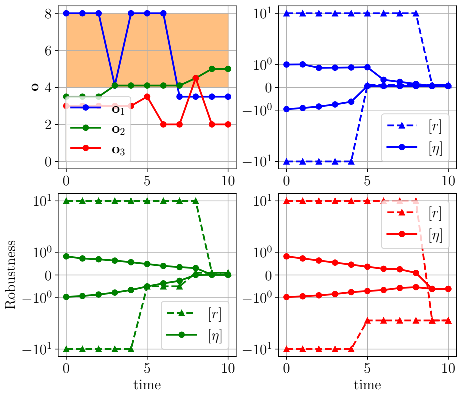

Fig. 1. Demonstration of TWTL robustness and monitoring. (top-left) Depiction of the valuation of words $\mathbf{o}_1$, $\mathbf{o}_2$, and $\mathbf{o}_3$ w.r.t. time. The evolution of $[\rho]$ (dashed-line with triangles) and $[\eta]$ (solid-lines with circles) for words $\mathbf{o}_1$, $\mathbf{o}_2$ (bottom-left), and $\mathbf{o}_3$ are shown in the top-right, bottom-left, and bottom-right figures, respectively.

Theorem 4.1: Consider a TWTL formula ϕ and a partial word $\mathbf{o}_{t_1,t'}$, then for any possible completion word $\mathbf{o}_{t_1,t_2} \in \mathfrak{C}$, $\rho(\mathbf{o}_{t_1,t_2}, H^d\pi_A) \in [\rho](\mathbf{o}_{t_1,t'}, H^d\pi_A)$ and $\eta(\mathbf{o}_{t_1,t_2}, H^d\pi_A) \in [\eta](\mathbf{o}_{t_1,t'}, H^d\pi_A)$.

Proof: See the extended version [27]. ■

Lemma 4.1: (Convergence of Robustness Intervals) Given a TWTL formula ϕ and word $\mathbf{o}_{t_1,t_2}$, where $t_2 \geq ||\phi||$; and for partial words $\mathbf{o}_{t_1,t'_1}$ and $\mathbf{o}_{t_1,t'_2}$, where $t'_1 < t'_2 < t_2$, the following set inclusions hold: $[\rho](\mathbf{o}_{t_1,t'_1}) \subseteq [\rho](\mathbf{o}_{t_1,t'_2})$ and $[\eta](\mathbf{o}_{t_1,t'_1}) \subseteq [\eta](\mathbf{o}_{t_1,t'_2})$. For $||\phi|| \leq t' \leq t_2$ the robustness intervals converge to a singleton which is the true robustness values, i.e., $[\rho](\mathbf{o}_{t_1,t'}) = \{\rho(\mathbf{o}_{t_1,t_2})\}$ and $[\eta](\mathbf{o}_{t_1,t'}) = \{\eta(\mathbf{o}_{t_1,t_2})\}$.

Example 4.1: (Continued) Given the TWTL unnormalized formula $\phi = [H^6\pi_A]^{[0,10]}$ with time horizon $||\phi|| = 10$, we demonstrate the proposed runtime monitors by observing the robustness intervals $[\rho]$ and $[\eta]$ of partial words of $\mathbf{o}_1$, $\mathbf{o}_2$, and $\mathbf{o}_3$ (see the top-left figure of Fig.1). Consider the time series $\tau = \{0, \ldots, 10\}$. For each partial word we compute $[\rho]$ and $[\eta]$ at every $t \in \tau$ as we the partial words $\mathbf{o}_1(t)$, $\mathbf{o}_2(t)$, and $\mathbf{o}_3(t)$ become available. The evolution of the intervals $[\rho]$ and $[\eta]$ for $\mathbf{o}_1$, $\mathbf{o}_2$, and $\mathbf{o}_3$ are depicted in the top-right, bottom-left, and bottom-right figures of Fig. 1, respectively, where the evolution of $[\rho]$ is depicted in dashed-lines with triangles and the evolution of $[\eta]$ is depicted in solid-lines with circles. Notice how the intervals become tighter as the partial word grows, where eventually when $t' = ||\phi||$, $[\rho]$ and $[\eta]$ converge to the true ρ and η values, respectively. In this example we consider $\rho_\top = 10$ and $\rho_\perp = -10$; one can notice how the evolution of $[\eta]$ is smoother than the evolution of $[\rho]$, due to using the AGM in the computation of η. Note that we normalize the TWTL formula before computing η robustness values. Thus, in Fig. 1, η stays within $[-1, 1]$.

V. Numerical Example Case Study

We demonstrate the proposed robustness semantics, by monitoring ρ and η for pre-computed runs of a simple planar robot system. Assume the time step, Δt, of the runs is 1. We consider a simple sequential navigation task with deadlines and a safety requirement. In the following unnormalized TWTL formula, we encode the task that reads: *Within time* 0 *and* 8*, visit region* A *and stay there for* 3 *time steps; right after that, within time* 0 *and* 10*, visit region* B *and stay there for* 4 *time steps; right after that, within time* 0 *and* 11*, visit region* C *and stay there for* 3 *time steps; and for all execution time avoid region* O. See the left figure of Fig. 2 for a depiction of the planar regions A, B, C, and O.

$$\phi = \left([H^4\pi_A]^{[0,8]} \cdot [H^4\pi_B]^{[0,10]} \cdot [H^3\pi_C]^{[0,11]}\right) \wedge H^{50}\neg\pi_O \tag{16}$$

The atomic propositions π_A, π_B, π_C, and π_O are defined as predicated regions over the $xy-$plane; where $A := \{(x,y)|1 \leq x \leq 4 \wedge 1 \leq y \leq 4\}$, $B := \{(x,y)|8 \leq x \leq 11 \wedge 3 \leq y \leq\}$, $C := \{(x,y)|1 \leq x \leq 4 \wedge 9 \leq y \leq 12\}$, and $O := \{(x,y)|5 \leq x \leq 7 \wedge 5 \leq x \leq 7\}$.

We monitor two runs of the robot, $\mathbf{o}_1$ and $\mathbf{o}_2$, which are shown as the blue and green traces in the left figure of Fig. 2, respectively. The robustness of $\mathbf{o}_1$ and $\mathbf{o}_2$ are $\rho(\mathbf{o}_1, \phi) = 0.4$ and $\rho(\mathbf{o}_2, \phi) = 0.3$, whereas their AGM robustness are $\eta(\mathbf{o}_1, \phi) = 0.00076$ and $\eta(\mathbf{o}_2, \phi) = 0.00015$. To monitor the robustness measures at runtime, consider the time series $\tau = \{2, 10, 15, 20, 25, 30, 35, 40, 42\}$. For each partial word we compute $[\rho]$ and $[\eta]$ at every $t \in \tau$ as the partial words $\mathbf{o}_1(t)$ and $\mathbf{o}_2(t)$ become available. The valuations of the intervals $[\rho]$ and $[\eta]$ for $\mathbf{o}_1$, and $\mathbf{o}_2$ at every $t \in \tau$ are depicted in the middle, and right figures of Fig. 2, respectively, where $[\rho]$ is depicted in dashed-lines with triangles and the $[\eta]$ is depicted in solid-lines with circles.

Observe that monitoring convergence of $[\eta]$ is smoother than the convergence of $[\rho]$, which would be more useful in some applications. In our future work, we consider incremental planning for TWTL tasks for which we require runtime monitors to use as a heuristic to maximize the satisfaction of the task. We find that aiming to maximize η in planning applications would yield smoother paths, we leave the details of monitoring for planning applications for future work.

VI. Conclusion and Future Work

Given the richness of Time Window Temporal Logic as a specification language for dynamical systems, we introduce two quantitative semantics to measure the robustness of TWTL formulae. In the first measure, which we call TWTL robustness, we introduce a distance measure between the system run and the formula satisfaction decision boundary based on the most critical values of the run. In the second measure, which we call AGM TWTL robustness, we quantify the satisfaction degree by another distance measure using the arithmetic and geometric mean of the system run values based on some rules that guarantee the soundness of the measure. In planning applications, AGM TWTL robustness enjoys a key advantage over the first one, in that it gives more reward to values that contribute to the formula satisfaction whereas TWTL robustness is dominated by the most critical values. We plan to demonstrate this advantage in

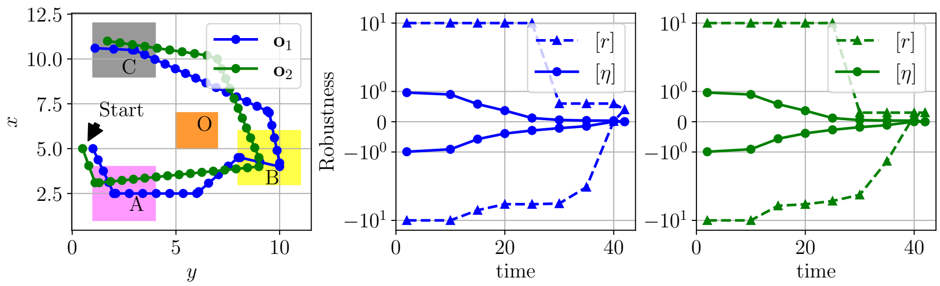

Fig. 2. Demonstration of monitoring the TWTL robustness and AGM TWTL robustness of precomputed planar robot runs, $\mathbf{o}_1$ and $\mathbf{o}_1$, against satisfying formula (16). Words $\mathbf{o}_1$ and $\mathbf{o}_2$ in an $xy-$ planar environment are shown in the blue and green traces in the left figure, respectively. The valuations of the intervals $[\rho]$ and $[\eta]$ for $\mathbf{o}_1$, and $\mathbf{o}_2$ at every $t \in \{2, 10, 15, 20, 25, 30, 35, 40, 42\}$ are depicted in the middle, and right figures, respectively, where $[\rho]$ is depicted in dashed-lines with triangles and the $[\eta]$ is depicted in solid-lines with circles.

future follow-up work. Given partial runs of the system, we develop runtime monitors that produce interval bounds on the quantitative semantics. We demonstrate the introduced quantitative semantics by monitoring TWTL robustness and AGM TWTL robustness of precomputed planar robot runs against some TWTL specifications.

REFERENCES

[1] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[2] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. ieee, 1977, pp. 46–57.

[3] E. Plaku and S. Karaman, "Motion planning with temporal-logic specifications: Progress and challenges," *AI communications*, vol. 29, no. 1, pp. 151–162, 2016.

[4] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for robots: Guarantees and feedback for robot behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 211–236, 2018.

[5] X. Luo, Y. Kantaros, and M. M. Zavlanos, "An abstraction-free method for multirobot temporal logic optimal control synthesis," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1487–1507, 2021.

[6] Y. Kantaros and M. M. Zavlanos, "Stylus*: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 812–836, 2020.

[7] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 15.

[8] X. Li, C.-I. Vasile, and C. Belta, "Reinforcement learning with temporal logic rewards," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3834–3839.

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

[10] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. [Online]. Available: https://doi.org/10.1177/0278364911406761

[11] K. Grover, F. dos Santos Barbosa, J. Tumova, and J. Kretınsky, "Semantic abstraction-guided motion planningfor scltl missions in unknown environments," in *Robotics: Science and Systems*, 2021.

[12] C. I. Vasile, X. Li, and C. Belta, "Reactive sampling-based path planning with temporal logic specifications," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 1002–1028, 2020. [Online]. Available: https://doi.org/10.1177/0278364920918919

[13] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems: Joint International Conferences on Formal Modeling and Analysis of Timed Systmes, FORMATS 2004, and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004. Proceedings*. Springer, 2004, pp. 152–166.

[14] R. Koymans, "Specifying real-time properties with metric temporal logic," *Real-time systems*, vol. 2, no. 4, pp. 255–299, 1990.

[15] C.-I. Vasile, D. Aksaray, and C. Belta, "Time window temporal logic," *Theoretical Computer Science*, vol. 691, pp. 27–54, 2017.

[16] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Formal Modeling and Analysis of Timed Systems: 8th International Conference, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings 8*. Springer, 2010, pp. 92–106.

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

[18] D. Aksaray, A. Jones, Z. Kong, M. Schwager, and C. Belta, "Q-learning for robust satisfaction of signal temporal logic specifications," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6565–6570.

[19] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 772–779.

[20] C.-I. Vasile, V. Raman, and S. Karaman, "Sampling-based synthesis of maximally-satisfying controllers for temporal logic specifications," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3840–3847.

[21] K. Leung, N. Aréchiga, and M. Pavone, "Backpropagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods," *The International Journal of Robotics Research*, p. 02783649221082115, 2020.

[22] J. Karlsson, F. S. Barbosa, and J. Tumova, "Sampling-based motion planning with temporal logic missions and spatial preferences," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 537–15 543, 2020.

[23] A. Rodionova, L. Lindemann, M. Morari, and G. J. Pappas, "Time-robust control for stl specifications," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 572–579.

[24] N. Mehdipour, C.-I. Vasile, and C. Belta, "Arithmetic-geometric mean robustness for control from signal temporal logic specifications," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 1690–1695.

[25] F. Penedo, C.-I. Vasile, and C. Belta, "Language-guided sampling-based planning using temporal relaxation," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 128–143.

[26] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia, "Robust online monitoring of signal temporal logic," *Formal Methods in System Design*, vol. 51, no. 1, pp. 5–30, 2017.

[27] A. Ahmad, C.-I. Vasile, R. Tron, and C. Belta, "Robustness measures and monitors for time window temporal logic," *arXiv preprint arXiv:2304.06645*, 2023.

[28] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan, "Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications," *Lectures on Runtime Verification: Introductory and Advanced Topics*, pp. 135–175, 2018.

[29] E. Bonnah and K. A. Hoque, "Runtime monitoring of time window temporal logic," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5888–5895, 2022.