Temporal Relaxation of Signal Temporal Logic Specifications for Resilient Control Synthesis

Ali Tevfik Buyukkocak and Derya Aksaray

Abstract—We introduce a metric that can quantify the temporal relaxation of Signal Temporal Logic (STL) specifications and facilitate resilient control synthesis in the face of infeasibilities. The proposed metric quantifies a cumulative notion of relaxation among the subtasks, and minimizing it yields to structural changes in the original STL specification by i) modifying time-intervals, ii) removing subtasks entirely if needed. To this end, we formulate an optimal control problem that extracts state and input sequences by minimally violating the temporal requirements while achieving the desired predicates. We encode this problem in the form of a computationally efficient mixed-integer program. We show some theoretical results on the properties of the new metric. Finally, we present a case study of a robot that minimally violates the time constraints of desired tasks in the face of an infeasibility.

I. INTRODUCTION

Motion planning and control of cyber-physical systems often require the satisfaction of complex tasks with time constraints. One way of expressing such tasks is via temporal logics. In particular, Signal Temporal Logic (STL) [1] is an expressive specification language that can define properties with explicit spatial and time parameters. Different than other temporal logics, STL contains predicates in the form of inequalities and is endowed with a metric called robustness degree that can quantify how well a signal satisfies an STL specification [2]. Such a metric facilitates optimization based control synthesis with STL specifications [3]–[5].

There are two main STL robustness degree metrics, i.e., space and time robustness which capture the effect of shifting the signal in space and time on the satisfaction, respectively [2]. While most of the studies in the literature propose solution methods to optimize space robustness or a variant of it [4]–[11], there are also some works emphasizing the benefits of optimizing time robustness [12]–[14]. Moreover, allowing for negative robustness and maximizing it may provide a notion of minimal violation of an STL specification.

In real-world applications, autonomous systems operate under various disturbances (e.g., internal, external, human-triggered), and there might be cases where the desired STL specification becomes infeasible. In such cases, a resilient control synthesis requires to search for trajectories resulting in minimal violations of the original specification. One common way of achieving this is by relaxing the spatial requirements of the specification (e.g., [15]), such as satisfying $x \geq 4.7$ instead of $x \geq 5$. On the other hand, relaxing the time bounds by strictly enforcing the thresholds over signal values is not investigated much in the STL literature, although some applications (e.g., manipulating objects in fixed regions) would benefit from it. Such a notion of temporal relaxation has been introduced for Time Window Temporal Logic in [16], [17], where an automata-theoretic approach is proposed to shrink or extend the corresponding time intervals of the tasks when needed.

In this paper, we introduce a metric that can quantify the temporal relaxation of STL specifications and facilitate resilient control synthesis in the face of infeasibilities. The proposed metric quantifies a cumulative notion of time relaxation among the subtasks. We propose a mixed-integer encoding for the new metric and formulate a computationally efficient mixed-integer program to minimize it.

This work is closely related to [12], [13], [14], [18], [19]. In [12], the goal is to maximize the time robustness along with the space robustness for a limited family of STL specifications; however, the infeasibility of STL specifications is not discussed. The authors of [13] introduce the mixed-integer encoding of the standard time robustness [2] by counting the consecutive satisfactions and violations. Similarly, time robustness under time shifts in the stochastic signals is assessed along with a risk measure of STL failure in [14]. While maximizing time robustness may lead to satisfaction of the STL specification even when the signal is shifted along the time (e.g., delay in the signal), it may not necessarily result in minimal temporal relaxation of the STL specifications in the presence of violations. This is due to the fact that the quantity of time robustness metric is dominated by the critical violations (i.e., use of min/max functions), which makes it hard to differentiate individual task relaxations. Furthermore, maximizing time robustness especially in the violation cases [12], [13] requires a significant computational effort, which is also illustrated in our case study. Alternatively, repairing STL specifications is also addressed in [18], where an iterative algorithm is proposed to repair infeasible STL specifications by relaxing thresholds over signal values, penalizing relaxations, and determining possible interval modifications based on the relaxed thresholds. However, that work is not equivalent to minimizing temporal relaxation and its iterative nature demands more computational effort. Finally, a notion of temporal relaxation is investigated in [19], where an algorithm is proposed to plan trajectories that satisfy iteratively changing STL specification (i.e., shifting STL). In particular, the original...
STL specification is considered but its time intervals are from the current time instant to the end of the time interval which allows for achieving STL tasks later. However, this work is limited to safety and persistence behaviors. To the best of our knowledge, there is no framework in the literature that minimizes a temporal relaxation metric facilitating both shrinkage and expansion of the time intervals and minimally changing the structure of the STL specification by partially removing subtasks in a computationally efficient manner.

Notation: The set of non-negative (positive) real and integer numbers are denoted by $\mathbb{R}_{\geq 0}$ (positive real numbers) and $\mathbb{Z}_{\geq 0}$ (positive integer numbers), respectively. The set of $n$-dimensional real valued vectors is denoted by $\mathbb{R}^n$. Floor and ceiling functions are designated by $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$, respectively. The set cardinality is represented by $| \cdot |$. Minkowski sum of two sets $A, B \subset \mathbb{Z}$ is denoted by $A \oplus B = \{a + b \mid a \in A, b \in B\}$. With a slight abuse of notation, we also allow $a \oplus B = \{a + b \mid b \in B\}$.

II. SIGNAL TEMPORAL LOGIC

Signal Temporal Logic (STL) \([1]\) can express rich properties of time series. In this paper, we specify desired system behaviors with the following expressive STL fragment:\(^1\)

$$\Phi ::= \phi \land \Phi_1 \land \Phi_2 \lor \Phi_1 \lor \Phi_2 \land F_{[a,b]} \Phi \lor G_{[a,b]} \Phi,$$

$$\phi ::= \mu \land \phi_1 \land \phi_2 \lor \phi_1 \lor \phi_2 \land [a,b],$$

(1)

where $\Phi, \phi, \varphi$ are STL specifications and subtasks; $\mu$ is a predicate in an inequality form such as $\mu = p(x) \geq 0$ with the value of a discrete-time signal $x : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}^n$ at time $t$ and a function $p : \mathbb{R}^n \rightarrow \mathbb{R}$. While $(x, t) \models F_{[a,b]} \Phi$ implies that $\Phi$ holds at some time instant within $t \oplus [a,b]$,

$$(x, t) \models F_{[a,b]} \Phi$$

requires $\Phi$ to hold at all time instants within the same time interval, where $a, b \in \mathbb{Z}_{\geq 0}$ are finite time bounds with $b > a$. It is worth noting that the major restriction of the syntax in (1) is not allowing for the conjunction or disjunction of the temporal operators with bare predicates such as $\Phi = F_{[a,b]}(\mu_1 \land F_{[c,d]} \mu_2)$.

STL is endowed with real-valued functions, robustness degree, that are used to quantify the satisfaction of an STL specification $\Phi$ with respect to a signal $(x, t)$. While positive robustness degree indicates the satisfaction of $\Phi$, negative one represents a violation. The space and time robustness functions can be formally defined as below:

**Definition 1.** (STL Space Robustness \([2]\)) Space robustness is a real-valued function $\rho(\cdot) \in \mathbb{R}$ that is used to quantify how much the signal $x$ can be shifted in space such that the specification is still satisfied. It is defined recursively as:

$$\rho(x, \mu, t) := p(x_t),$$

$$\rho(x, \Phi_1 \land \Phi_2, t) := \min \left( \rho(x, \Phi_1, t), \rho(x, \Phi_2, t) \right),$$

$$\rho(x, \Phi_1 \lor \Phi_2, t) := \max \left( \rho(x, \Phi_1, t), \rho(x, \Phi_2, t) \right),$$

$$\rho(x, F_{[a,b]} \Phi, t) := \max \left\{ \rho(x, \Phi, t'), \quad t' \in [a,b] \right\},$$

$$\rho(x, G_{[a,b]} \Phi, t) := \min \left\{ \rho(x, \Phi, t'), \quad t' \in [a,b] \right\}.$$

**Definition 2.** (STL Time Robustness \([2]\)) Right (+) and left (−) time robustness of an STL specification quantify how much the signal $x$ can be shifted either right or left in time such that the specification is still satisfied. Using the characteristic function $\chi_{[a,b]}(\cdot) := \begin{cases} -1, & p(x_t) < 0 \\ +1, & p(x_t) \geq 0 \end{cases}$, the time robustness of $x$ with respect to a predicate $\mu$ can be computed as:

$$\theta^+(x, \mu, t) := \chi_{[a,b]}(\mu, t) \cdot \max \left\{ d \geq 0 \mid \forall t' \in [t, t + d], \chi_{[a,b]}(\mu, t') = \chi_{[a,b]}(\mu, t) \right\},$$

$$\theta^-(x, \mu, t) := \chi_{[a,b]}(\mu, t) \cdot \max \left\{ d \geq 0 \mid \forall t' \in [t - d, t], \chi_{[a,b]}(\mu, t') = \chi_{[a,b]}(\mu, t) \right\},$$

After computing the right/left time robustness of a predicate, the rules in (2) can be used to quantify the overall time robustness of a signal with respect to an STL specification. There also exists a combined notion called space-time robustness, where the inequalities in the characteristic equation are defined based on a desired space robustness threshold \([2]\).

**Example 1.** Consider two signals $x'$ and $x''$ starting from $t = 0$ and illustrated in Fig. 1. Suppose that the specification is $\Phi_1 = G_{[15,60]} x \geq h_1$ meaning that the signal has to satisfy $x_t \geq h_1$ for all $t \in [15, 60]$. While both signals violate $\Phi_1$, $x'$ (blue) stays in the desired region longer than $x''$ (red). However, the (right) time robustness notion cannot differentiate this due to using $\min$ function in its computation as in (2). Specifically, $\theta'^+(x', \Phi_1, 0) = \theta'^+(x'', \Phi_1, 0) = \min_{t \in [15,60]} \theta'^+(x, \Phi_1, 0) \geq h_1$, hence both signals have the same time robustness that is negative due to violation, and its length is illustrated by the purple bar in Fig. 1.

Now, suppose the specification is updated as $\Phi_2 = \Phi_1 \land F_{[75,120]} x \leq h_2$ meaning that the signal needs to satisfy $x_t \leq h_2$ for some time $t \in [75, 120]$ in addition to satisfying $\Phi_1$. Despite the blue signal's satisfaction of the second subtask, the violation of $\Phi_1$ dominates the computation of time robustness of $\Phi_2$. Furthermore, the signal $x''$ slightly violates the second subtask, but the initial violation still dominates. As a result, the violation in the purple area indifferently determines the time robustness score of both signals and specifications with $\theta'^+(x', \Phi_1, 0) = \theta'^+(x'', \Phi_1, 0) = \theta'^+(x', \Phi_2, 0) = \theta'^+(x'', \Phi_2, 0)$. 

![Fig. 1: Two signals that violate the given STL specifications by different amounts. Green regions are the targets the system must reside in to achieve the specifications. The areas of violation is denoted by the color of the respective signal, while the purple region depicts violation by both signals.](image-url)

Note that the standard space robustness in (2) may be insufficient to differentiate signal behaviors as well. For instance, $\rho(x', \Phi_1, 0) = \rho(x'', \Phi_1, 0) = -\rho^*$. However, this issue is addressed by several studies via modified space robustness metrics and extended STL syntax (e.g., \([6]–[11]\)). Nonetheless, such metrics do not assess satisfaction beyond...
the original time intervals, which motivate us to introduce a temporal relaxation metric for STL specifications.

III. MINIMAL TEMPORAL RELAXATION OF STL SPECIFICATIONS

The spatial relaxation of STL specifications is generally possible by allowing the system to minimally violate its spatial requirements (i.e., maximizing the space robustness while letting it be negative). For example, if the specification is entering a desired region, the spatial relaxation will be approaching the region as close as possible given the time interval. However, spatial relaxation may not be feasible for some missions, e.g., manipulating an object in a fixed region. Therefore, extending or shrinking the time intervals to allow for late or early satisfaction can be practical in some scenarios. Nonetheless, the temporal relaxation of STL is not broadly investigated, and the standard time robustness cannot differentiate partial satisfactions as illustrated in Example 1.

A. Temporal Relaxation of STL

Suppose that a subtask $\phi$ in the STL specification $\Phi$ (as per the syntax in (1)) is defined over a time interval of $I = [a, b]$. For every such subtask, we have the temporally relaxed versions $\phi^\tau = F_{[a, b + \tau]}^{-}[\phi]$ and $\phi^\tau = G_{[a + \tau, b]}^{[\phi]}$ where $\tau_e, \tau_g, \tau_\gamma \in \mathbb{Z}_{\geq 0}$ are temporal relaxation parameters. That is, the system may eventually satisfy $\phi$ within a longer time interval and always satisfy it within a shorter interval, both of which imply the relaxation of the original requirement. Moreover, when multiple temporal operators are nested, the temporal relaxation is enabled for the innermost ones. For example, if the original specification is $\Phi = F_{[c, d]} G_{[a, b]}[\phi]$, then the relaxed version of it is considered as $\Phi^\tau = F_{[c, d]} G_{[a + \tau, b - \tau]}[\phi]$.

Our approach will result in the satisfaction of the predicates and/or their combinations only within the original globally interval $I_G$ (potentially within an interval shorter than the original) and remove the subtask otherwise. On the other hand, the interval of the finally operator $I_F$ is extended, and when not bounded, this extension may yield excessive relaxations, e.g., satisfying $\mu$ at $t = 100$ where the specification is $\phi = F_{[5, 10]}[\mu]$. For this reason, our formulation allows the user to specify acceptable bounds to relax the original interval $I = [a, b]$. Accordingly, the new interval bounding the maximal relaxation is defined as $I_F = (a - \gamma_F \cdot |I_F|, b + \gamma_F \cdot |I_F|)$ where $\gamma_F > 0$ is a user-defined tolerance parameter.

A similar tolerance $0 < \gamma_G \leq 1$ can be defined for the globally as well where the relaxation is allowed over $I_G = [a, b] \setminus \left[ [a, a + \gamma_G \cdot \frac{|I_G|}{2}], (b - \gamma_G \cdot \frac{|I_G|}{2}, b) \right]$. Note that, $I_F$ and $I_G$ are not the relaxed intervals, but they exclusively bound the allowable relaxation according to user preferences.

Next, we define the temporal relaxation metrics that we want to keep minimum. The temporal relaxation metric simply comprises the amount of violation with respect to the maximum allowable relaxation. We define two normalized metrics below to measure the temporal relaxation among the subtasks with finally and globally operators, respectively.

**Definition 3. (Temporal Relaxation Metric)** Given an STL specification with the syntax in (1), the temporal relaxation metrics for the finally and globally operators are defined as:

Finally

$$\tau(x, F_{[a, b]}[\phi], t) := \begin{cases} \max(\gamma_F \cdot |I_F|, 1), & \text{if } \exists t' \in t \odot [a - \gamma_g \cdot |I_F|, b + \gamma_g \cdot |I_F|] \\ s.t. \gamma_g \cdot |I_F| \geq \tau_g, \tau_g \geq 0 \text{ and } (x, t') \models \phi, \text{ otherwise.} \end{cases}$$

Globally

$$\tau(x, G_{[a, b]}[\phi], t) := \begin{cases} \tau_g + \gamma_g \cdot |I_G|, & \text{if } \forall t' \in t \odot [a + \gamma_g \cdot |I_G|, b - \gamma_g \cdot |I_G|] \\ s.t. \gamma_g \cdot |I_G| \geq \tau_g, \tau_g \geq 0 \text{ and } (x, t') \models \phi, \text{ otherwise.} \end{cases}$$

**Proposition 1.** Suppose that $\tau_e, \tau_g \in \mathbb{Z}_{\geq 0}$. Let $\phi = F_{[a, b]}[\phi]$ be a finally subtask endowed with the temporal relaxation metric $\tau(x, F_{[a, b]}[\phi], t)$ defined in (3). When $\tau(x, F_{[a, b]}[\phi], t)$ is minimized, the resulting relaxed interval $[a - \tau_e, b + \tau_e]$ can have at least one of $\tau_e$ and $\tau_g$ equal to zero.

**Proof.** The proof can be found in [21].

For instance, for $\Phi = F_{[5, 10]}[\phi]$, a closest satisfaction such as $(x, 15) \models \phi$ yields the relaxed specification of $\Phi^\tau = F_{[5, 15]}[\phi]$. Alternatively, if $(x, 3) \models \phi$ is the closest satisfaction, then the relaxed specification would be $\Phi^\tau = F_{[3, 10]}[\phi]$. When $\tau_e = \tau_g = 0$, the original specification is satisfied, i.e., $(x, t) \models F_{[a, b]}[\phi]$, without any relaxation, i.e., $\tau(x, F_{[a, b]}[\phi], t) = 0$.

Note that the temporal relaxation metrics in (3) and (4) are normalized and take values continuously within $[0, 1]$. That is, when no relaxation is needed and the original specification is achieved, the relaxation metric is 0 for each operator, and when the allowable relaxation amount is exceeded or the subtask is completely failed, the metric becomes 1 and the subtask is removed. Hence, the aim will be minimizing the temporal relaxation during the satisfaction of an STL specification $\Phi$ in the face of infeasibilities.

Starting from $\tau(x, F_{[a, b]}[\phi], t)$ and $\tau(x, G_{[a, b]}[\phi], t)$ in (3) and (4), we recursively define a general temporal relaxation metric $\tau(x, \Phi, t)$ for the STL specification $\Phi$ as follows:

$$\tau(x, \Phi_1 \land \Phi_2, t) := 2 \tau(x, \Phi_1, t) + \tau(x, \Phi_2, t)$$

$$\tau(x, \Phi_1 \lor \Phi_2, t) := \min \left( \tau(x, \Phi_1, t), \tau(x, \Phi_2, t) \right)$$

$$\tau(x, \Phi_1 \land \Phi_2, t) := \max \left( \tau(x, \Phi_1, t), \tau(x, \Phi_2, t) \right)$$

$$\tau(x, F_{[a, b]}[\Phi], t) := \min_{t' \in t} \tau(x, \Phi, t')$$

By definition, the temporal relaxation is allowed for the innermost temporal STL operators via (3) and (4) when multiple of them are nested. In other words, when $\Phi = F_{[a, b]}[\phi]$ or $\Phi = G_{[a, b]}[\phi]$, i.e., $\Phi = \phi$ as per (1), we use temporal relaxation metrics in (3) and (4), respectively. However, when there are multiple nested temporal operators, e.g., $\Phi = G_{[a, b]} F_{[c, d]}[\phi]$, we use the definitions in (5c) and (5d) for the outer ones together with (3) and (4) inside them.

**Corollary 1.** For a given STL specification $\Phi$ with the syntax in (1), the overall relaxation metric defined according to (5) is bounded such that $0 \leq \tau(x, \Phi, t) \leq 1, \forall t$.

**Proof.** The proof can be found in [21].

It is worth noting that unlike the standard STL quantifying metrics (e.g., [1]), we use min for the disjunction or finally
(in (5b) and (5d)), and max for the globally (in (5c)). This is because the lower temporal relaxation metric implies the closer satisfaction to the original specification. Moreover, the averaging in (5a) enables us to measure collective performance that is not dominated by critical values.

**Proposition 2.** For a given STL specification \( \Phi \) with the syntax in (1), the temporal relaxation metric defined according to (5) is sound in the sense that \( \tau(x, \Phi, t) = 0 \implies (x, t) \models \Phi^\gamma \).

**Proof.** The proof can be found in [21]. □

Proposition 2 along with Corollary 1 can be interpreted as follows: in the best case, the given STL specification is satisfied within the original time intervals with \( \tau(x, \Phi, t) = 0 \). Therefore, an extra performance such as achieving tasks for longer time than required is not demanded unlike the optimization of other quantifying metrics of STL. However, when it is not feasible to achieve \( \Phi \), the minimization of \( \tau(x, \Phi, t) \) leads to the satisfaction of spatial requirements specified in \( \Phi \) with a minimal temporal relaxation. Furthermore, this minimal relaxation may potentially yield structural changes on the \( \Phi \) such as compromising highly demanding subtasks which may jeopardize the success of the others.

**Example 1 (Cont’d).** Consider again the two signals \( x' \) (blue) and \( x'' \) (red) in Fig. 1. Recall that both signals have the same time robustness value for different STL specifications \( \Phi_1 = G_{[15,60]}x \geq h_1 \) and \( \Phi_2 = \Phi_1 \land F_{[75,120]}x \leq h_2 \). We now examine the performance of the new temporal relaxation metric for the same scenario. Let \( I_1 = [15, 60] \) and \( I_2 = [75, 120] \) with \( |I_1| = |I_2| = 46 \). Assume that the tolerance parameters are defined as \( \gamma_F = 1 \) and \( \gamma_G = 1 \), and note the temporal relaxation parameters \( \tau_{\tau'} = \tau_{\tau} = 4 \) from Fig. 1. As both signals violate \( \Phi_1 \), the violation amounts can be calculated using the proposed temporal relaxation metric as \( \tau(x', \Phi_1, t) = 9/46 \) and \( \tau(x'', \Phi_1, t) = (9 + 9)/46 = 18/46 \) implying the temporal relaxation made for \( x' \) is lower than that for \( x'' \). Similarly, when quantifying the violations for \( \Phi_2 \), the ones for \( \Phi_1 \) should neither dominate the computation nor be completely useless, but contribute to the calculation of the cumulative relaxation. Since \( x' \) completely satisfies the finally subtask, the only violation comes from \( \Phi_1 \) but averaged due to total number of two subtasks as \( \tau(x', \Phi_2, t) = 9/246 \). On the other hand, \( x'' \) violates both subtasks leading to \( \tau(x'', \Phi_2, t) = (18 + 4)/46 = 22/46 \). These temporal relaxations for two signals yield the following relaxed specifications: \( \Phi_2' = G_{[15,60]+\tau_{\tau}}x \geq h_1 \land F_{[75,120]}x \leq h_2 \) for \( x' \) requiring less relaxation than that of \( \Phi_2'' = G_{[15,60]+\tau_{\tau}+\tau_{\tau}}x \geq h_1 \land F_{[75,120]+\tau_{\tau}}x \leq h_2 \), the relaxed specification for \( x'' \).

**Definition 4. (Time Interval Similarity)** The similarity between two time intervals \( I_1 \) and \( I_2 \) is measured via

\[ S(I_1, I_2) := \frac{|I_1 \cap I_2|}{\max(|I_1|, |I_2|)}. \]

2The reverse \( \tau(x, \Phi, t) = 0 \implies (x, t) \models \Phi \) may not always be true as the satisfaction over the original time intervals implies satisfaction over arbitrarily relaxed time intervals as well with nonzero relaxation parameters according to (3) and (4).

3Note that this similarity measure is a form of Jaccard similarity coefficient when one interval is contained in the other.

**Proposition 3.** Let \( \phi \) be an STL subtask with a single temporal operator over an original interval of \( I = [a, b] \) as in (1), e.g., \( \phi = F_{[a,b]} \varphi \) or \( \phi = G_{[a,b]} \varphi \). Suppose that two feasible minimal relaxations of \( \phi \), \( \tau'(x, \phi, t) \) and \( \tau''(x, \phi, t) \in [0, 1] \), are obtained over the relaxed intervals of \( I' \) and \( I'' \), respectively. If \( \tau'(x, \phi, t) \leq \tau''(x, \phi, t) \) under the same user tolerances \( \gamma_F \) and \( \gamma_G \), then \( S(I', I) \geq S(I'', I) \).

**Proof.** The proof can be found in [21]. □

**Problem 1.** Given an initial state \( x_0 \), an STL specification \( \Phi \), some user-defined relaxation tolerances \( \gamma_F \) and \( \gamma_G \), the minimal temporal relaxation problem for a dynamical system can be formulated as an optimization problem as follows:

\[
\begin{align*}
\mathbf{u}^* &= \arg \min \tau(x, \Phi, 0) \\
\text{s.t.} & \quad x^+ = f(x_t, u_t), \\
& \quad x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}, \quad \forall t,
\end{align*}
\]

where \( f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \) denotes the dynamics of the system; \( \mathcal{X} \) and \( \mathcal{U} \) are the admissible state and control sets, respectively.

If \( \Phi \) can be satisfied by a trajectory starting from the given initial state \( x_0 \), then solving Problem 1 generates a feasible trajectory. Otherwise, solving it results in a trajectory that minimally relaxes the time bounds of \( \Phi \) under the allowable relaxations provided by the user. Such a trajectory is different than the one that can be found by maximizing space robustness (which does not modify the original time bounds) and time robustness.

**B. Temporal Relaxation vs. Time Robustness**

Given an initial state \( x_0 \), if \( \Phi \) cannot be satisfied by a trajectory starting from \( x_0 \), the trajectories found by minimizing temporal relaxation and maximizing time robustness are not the same. In the literature, either right or left time robustness is maximized and addressing both past and future relaxations at the same time is missing. Even if a unified approach can be proposed to account for this, there exists a major difference regarding the computation of time robustness, which uses \( \min/\max \) functions at each level. This leads to the major violation cases to dominate the value of the overall time robustness and cannot differentiate the satisfaction/violation of the other subtasks (as illustrated in Example 1). Furthermore, the time robustness becomes inconclusive when it is equal to zero. This may potentially yield problems in differentiating between a near miss and a tight satisfaction.

For instance, consider \( \phi = F_{[5,10]} \mu \) under two scenarios: either \( (x, 10) \models \mu \) (tight satisfaction) or \( (x, 11) \models \mu \) (near miss). As the (right) time robustness is calculated by \( \theta^+ (x, \phi, 0) = \max_{x \in [5, 10]} \theta^+ (x, \mu, t) = \theta^+ (x, \mu, 10) = 0 \) for both cases, we have the same time robustness score for them. Hence, satisfying \( \phi \) within the specified time interval but at the very last step or violating it by one time step becomes indifferent. Hence, determining the need of relaxing each subtask is not possible by using the notion of time robustness.

Moreover, maximizing time robustness in the presence of a violation leads to checking the overall mission horizon to ensure the satisfaction of the corresponding subtask (or predicate) at least once so that a finite negative time robustness can be obtained. However, this can cause the mandatory
satisfaction instant to be arbitrarily far from the original interval which may not be desirable. On the other hand, our proposed formulation can accommodate user input for maximum relaxations and minimize temporal relaxation under the allowable relaxation tolerances. If a subtask requires relaxation more than the allowable relaxation, then that subtask is removed from the specification. Note that a similar subtask removal by maximizing time robustness might be possible. However, that becomes a more computationally intensive process as it requires to iteratively compute trajectories and their corresponding time robustness values and decide to remove a subtask if the resulting time robustness is smaller than a user-defined threshold. Our proposed approach minimizing temporal relaxation does not require iterative trajectory computation and solves the problem in single shot while still incorporating the user preferences.

IV. Solution Approach

Mixed-integer encoding of STL constraints is a common approach in control synthesis under an STL specification [4]. If the dynamics \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \) is linear (e.g., \( \dot{x}_{t + 1} = A x_t + B u_t \), where \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \) are the system matrices), the mixed-integer encoding of the STL specification \( \Phi \) and the temporal relaxation metric \( \tau(x, \Phi, t) \) renders the optimization problem in (6) a mixed-integer linear program (MILP) in the presence of linear predicates.

The authors of [13] use the framework in [4] to maximize right time robustness by counting the consecutive satisfaction and violation instances of \( \varphi \) starting from \( t \) toward the future (\( r \)) and past (\( l \)), respectively. If the dynamics \( f \) approach in control synthesis under an STL specification \( \Phi \), the conjunction and disjunction of the predicates, i.e., the 

\[
\begin{align*}
\Phi(x) & = \bigwedge_{i=1}^{m} \varphi_i \\
\Phi(x) & = \bigvee_{i=1}^{m} \varphi_i
\end{align*}
\]

In (1), the conjunction and disjunction of the predicates, i.e., the

\[
\begin{align*}
\varphi(x) & = \bigwedge_{i=1}^{m} \varphi_i \\
\varphi(x) & = \bigvee_{i=1}^{m} \varphi_i
\end{align*}
\]

In (8), the maximum number of consecutive satisfaction and violation instances of \( \varphi \) at future starting from \( t \) for \( t' \geq t \) are defined backward in time, respectively, as below

\[
\begin{align*}
l_t^1(\varphi) & = z_t^1 \cdot (l_{t - 1}^1(\varphi) + 1), \\
l_t^0(\varphi) & = (1 - z_t^1) \cdot (l_{t - 1}^0(\varphi) - 1). \tag{9}
\end{align*}
\]

Similarly, the numbers of consecutive satisfaction and violation instances of \( \varphi \) at future starting from \( t \) for \( t' \geq t \) are defined backward in time, respectively, as below

\[
\begin{align*}
r_t^1(\varphi) & = z_t^1 \cdot (r_{t + 1}^1(\varphi) + 1), \\
r_t^0(\varphi) & = (1 - z_t^1) \cdot (r_{t + 1}^0(\varphi) - 1). \tag{10}
\end{align*}
\]

By construction, \( r_t^1(\varphi) \) and \( l_t^1(\varphi) \) count the maximum numbers of consecutive instances with \( z_t^1 = 1 \) for \( t' \geq t \) and \( t' < t \), respectively. On the other hand, \( r_t^0(\varphi) \) and \( l_t^0(\varphi) \) count the maximum number of consecutive instances with \( z_t^1 = 0 \) and multiply these values by -1.

For the given time intervals \( I_F = I_G = [a, b] \), boundary conditions for the calculations in (9) and (10) are determined based on the type of temporal operator. For \( F \), the calculations in (11) start forward and backward in time with 

\[
l_{t + a - [\gamma_{F} \cdot |I_F|]}(\varphi) = 0 \quad \text{and} \quad r_{t + b + [\gamma_{F} \cdot |I_F|]}(\varphi) = 0,
\]

respectively, where \([\gamma_{F} \cdot |I_F|] \) is the exclusive bound for the allowable relaxation. On the other hand, the boundary conditions for the calculations for \( \text{globally} \) in (12) are started forward and backward in time with 

\[
l_{t + a - 1}^1 = 0 \quad \text{and} \quad r_{t + b + 1}^1 = 0,
\]

respectively.

We continue with the MILP encoding of temporal relaxation metrics for \( F \) and \( \text{globally} \) operators constructed in (3) and (4), respectively.

Finally

\[
\tau(x, F_{[a,b]^{\varphi}}, t) = (z_t^{F_{[a,b]^{\varphi}}} - 1) \cdot \frac{\max([l_{t+a}^{F_{[a,b]^{\varphi}}}(\varphi), r_{t+b}^{F_{[a,b]^{\varphi}}}(\varphi)])}{|I_F|}. \tag{11}
\]

where the variable \( z_t^{F_{[a,b]^{\varphi}}} \) denotes the Boolean satisfaction of the original subtask, i.e., \( z_t^{F_{[a,b]^{\varphi}}} = 1 \iff (x, t) \models F_{[a,b]^{\varphi}} \), and its MILP encoding is similar to disjunction in (8) as \( z_t^{F_{[a,b]^{\varphi}}} = \bigvee_{t = a}^{b} z_t^\varphi \).

Hence, if the \text{finally} subtask is already satisfied there is no need to consider relaxation beyond the original interval or demanding more satisfactory time instants. This way, we keep the original requirement of satisfying \( \varphi \) at any single instant within \([a, b]\) if possible, or at an instant as close as possible to the original interval otherwise.

Globally

\[
\tau(x, G_{[a,b]^{\varphi}}, t) = 1 - z_t^{G_{[a+b-t-\beta]^{\varphi}}} \cdot \left(1 - \frac{|I_G| - l_{t+a+b-\beta}^1(\varphi) - r_{t+a+b}^1(\varphi)}{|I_G|}ight). \tag{12}
\]
where $\beta = \frac{|\gamma G \cdot |IG|}{2 k}$ and the fraction is the ratio of failed time instants to the allowed relaxation. The variable $z_{G}^{(a+b-\beta)}$ denotes the Boolean satisfaction of the subtask under maximum allowable relaxation, and its MILP encoding by conjunction as in (7) is $z_{G}^{(a+b-\beta)} = \bigland_{i=b-a+\beta}^{a+b-\beta} z_{i}^{G}$. 

Therefore, the globally subtask can be satisfied under an allowable relaxation, and otherwise removed. Note that when $\gamma = 1$, the Boolean satisfaction variable becomes redundant as even a single satisfied time instant is an acceptable relaxation for the $\gamma = 1$ case. The encoding in (12) enforces the satisfaction at the middle and possible relaxations at both ends as required by the definition in (4). As $\left[\frac{a+b}{2}\right]$ and $\left[\frac{a+b}{2}\right]$ denote the same time step when $a + b$ is even, for such cases, we consider the satisfaction at only one of these two instants in the numerator of (12) to prevent double counting.

After encoding the temporal relaxation metrics of $\tau(x, F_{[a,b]}[\phi], t)$ and $\tau(x, G_{[a,b]}[\phi], t)$, the MILP formulation of $\tau(x, \Phi, t)$ for the overall STL specification $\Phi$ can be done according to the recursive definitions in (5) and using the recipe in [4] for quantitative encoding of min/max functions.

**Algorithm 1: STL Control Synthesis under Minimal Temporal Relaxation using MILP encoding**

input : Initial state $x_{0}$, linear dynamics $f(x, u)$, STL specification $\Phi$, and user tolerance $\gamma_{F}$ and $\gamma_{G}$.

1. Define constraints of the dynamics along the horizon $T$.
2. Formulate MILP constraints on $z_{i}^{G}$ for each subtask $\phi$ over $t \in [0, T]$ (i.e., predicates and their conjunction/disjunction);
3. Formulate STL constraints to define $\tau(x, \Phi, 0)$ built recursively upon the core temporal relaxation metrics $\tau(x, F_{[a,b]}[\phi], t)$ in (3) and $\tau(x, G_{[a,b]}[\phi], t)$ in (4) according to (5);
4. Solve (6) as a MILP to extract state and input sequences;

output : Relaxed STL specification $\Phi^{*}$ with the temporal relaxation amounts on each subtask, state trajectory $x$ and input policy $u^{*}$ that achieve $\Phi^{*}$.

Unlike the time robustness, temporal relaxation metric is not defined throughout the whole mission horizon. In this regard, much less variables are used in MILP encoding compared to time robustness yielding substantially faster solutions in addition to obtaining different signal behaviors. Moreover, counting the violations only for the finally and the satisfactions only for the globally operator within predetermined bounds ($T_{F}$ and $T_{G}$) keeps the number of optimization variables low and contribute to the computational efficiency.

**Theorem 1.** For an STL specification $\Phi$ defined using the syntax in (1) with linear predicates, if any subtask $\Phi_{j}$ in $\Phi = \bigland_{j=1}^{n} \Phi_{j}$ is feasible alone, then Alg. 1 returns $\tau(x, \Phi, t) < 1$ implying that the spatial requirement of at least one subtask is satisfied.

Proof: The proof can be found in [21].

V. CASE STUDIES

We formulate the problem in YALMIP [22] and solve it using Gurobi [23]. A laptop computer with 1.8 GHz, i5 processor is used to run the simulations with the mission horizon of $T = 120$ s. We use $\gamma_{F} = \gamma_{G} = 1$ to bound the relaxations over finally and globally subtasks, respectively.

To illustrate the benefits of the new temporal relaxation metric, here we address control synthesis for an autonomous robot with discrete-time double-integrator dynamics with the state vector $x = [x, v_{x}, y, v_{y}]^{T}$ where $v_{x}$, $v_{y}$ $\in \mathbb{R}$ are the velocities in $x, y \in \mathbb{R}$ directions, respectively; and the input vector $u = [u_{x}, u_{y}]^{T}$ where $u_{x}$, $u_{y}$ $\in \mathbb{R}$ are the accelerations along the given directions with the limit of $|u_{x}|, |u_{y}| \leq 2.2$. The mission scenario requires the robot to visit $R_{A}$ and $R_{B}$ some time within $[32, 42]$ and $[77, 87]$, respectively, and always stay inside $R_{C}$ within $[47, 67]$. These requirements are expressed by the STL specification:

$$\Phi_{\text{case}} = F_{[32, 42]}R_{A} \land F_{[77, 87]}R_{B} \land G_{[47, 67]}R_{C}, \quad (13)$$

where visiting regions can be captured by the conjunction of linear predicates, e.g., $R_{A} = x \geq 4 \land x \leq 8 \land y \geq 6 \land y \leq 10$.

The STL control synthesis problem under $\Phi_{\text{case}}$ is solved by i) minimizing the proposed temporal relaxation metric $\tau(\cdot)$ using Alg. 1 and encoding in Sec. IV, ii) maximizing right $(\theta_{+}(\cdot))$ and left $(\theta_{-}(\cdot))$ time robustness metrics via the approach in [13] for comparison. Results are given in Table I, and realized trajectories with the marked satisfactory instances and the original intervals are presented in Fig. 2. The use of the proposed metric results in smaller temporal relaxation compared to time robustness maximizing trajectories. Furthermore, the less number of constraints and variables in our encoding yields more efficient computation time as well. As a result, the captured signal behaviors via all three approaches in Table I lead the following realized STL specifications with marked relaxations in the time intervals:

$$\Phi_{\text{case}}^{F} = F_{[28, 42]}R_{A} \land F_{[77, 89]}R_{B} \land G_{[50, 65]}R_{C},$$

$$\Phi_{\text{case}}^{\theta_{+}} = F_{[32, 42]}R_{A} \land F_{[77, 95]}R_{B} \land G_{[50, 65]}R_{C},$$

$$\Phi_{\text{case}}^{\theta_{-}} = F_{[21, 42]}R_{A} \land F_{[77, 87]}R_{B} \land G_{[50, 62]}R_{C}.$$
TABLE I: Simulation results and optimization parameters for temporal relaxation $(\tau(x, \Phi_{\text{case}}, 0))$ minimization and standard time robustness $(\theta^{+}(x, \Phi_{\text{case}}, 0))$ and $(\theta^{-}(x, \Phi_{\text{case}}, 0))$ maximization.

| Optimized Metric | # Constraints | # Variables | Computation Time [s] | Temporal Relaxation $\in [0, 1]$ |
|------------------|---------------|-------------|----------------------|----------------------------------|
| $\tau(x, \Phi_{\text{case}}, 0)$ | 5505 | 310 1821 | 1.66 | 39.2 | 0.277 |
| $\theta^{+}(x, \Phi_{\text{case}}, 0)$ | 8199 | 970 1861 | 1.98 | 1118.6 | 0.465 |
| $\theta^{-}(x, \Phi_{\text{case}}, 0)$ | 8175 | 964 1861 | 2.32 | 632.9 | 0.461 |

VI. CONCLUSIONS

We introduce a metric that quantifies temporal relaxation of STL specifications. We propose a mixed-integer encoding for temporal relaxation metric and formulate an optimization problem to minimize it. We compare the behavior obtained by minimizing temporal relaxation with the ones obtained by maximizing time robustness. We demonstrate that the proposed formulation is computationally efficient and leads to the satisfaction of STL specifications by minimally modifying the time intervals under the allowable relaxation limits.

REFERENCES

[1] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in Proc. Formal Techn., Modelling and Anal. of Timed and Fault-Tolerant Syst., 2004, pp. 152–166.
[2] A. Donzé and O. Maler, “Robust satisfaction of temporal logic over real-valued signals,” in Int. Conf. on Formal Modeling and Anal. of Timed Syst., 2010, pp. 92–106.
[3] S. Karaman, R. Sanfelice, and E. Frazzoli, “Optimal control of mixed logical dynamical systems with linear temporal logic specifications,” in Conf. Decis. Control, 2008, pp. 2117–2122.
[4] V. Raman, A. Donzé, M. Maassoumy, R. Murray, A. Sangiovanni-Vincentelli, and S. Seshia, “Model predictive control with stl specifications,” in Conf. Decis. and Cont. IEEE, 2014, pp. 81–87.
[5] Y. V. Pant, H. Abbas, and R. Mangharam, “Smooth operator: Control using the smooth robustness of temporal logic,” in Conf. on Control Tech. and Applications, 2017, pp. 1235–1240.
[6] T. Akazaki and I. Hasuo, “Time robustness in MTL and expressivity in hybrid system falsification,” in Int. Conf. on Computer Aided Verification, 2015, pp. 356–374.
[7] A. Rodionova, E. Bartocci, D. Nickovic, and R. Grosu, “Temporal logic as filtering,” in Proc. Int. Conf. on Hybrid Systems: Computation and Control, 2016, pp. 11–20.
[8] L. Lindemann and D. V. Dimarogonas, “Robust control for signal temporal logic specifications using discrete average space robustness,” Automatica, vol. 101, pp. 377–387, 2019.
[9] I. Haghighi, N. Mehdipour, E. Bartocci, and C. Belta, “Control from stl specifications with smooth cumulative quantitative semantics,” in Conf. on Decis. and Cont., 2019, pp. 4361–4366.
[10] N. Mehdipour, C.-I. Vasilie, and C. Belta, “Arithmetic-geometric mean robustness for control from temporal logic specifications,” in American Control Conference, 2019, pp. 1690–1695.
[11] A. T. Buyukkocak, D. Aksaray, and Y. Yuzcuoglu, “Control synthesis using stl specifications with integral and derivative predicates,” in 2021 American Control Conf. (ACC). IEEE, 2021, pp. 8743–8747.
[12] Z. Lin and J. S. Baras, “Optimization-based motion planning and runtime monitoring for robotic agent with space and time tolerances,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 1874–1879, 2020.
[13] 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.
[14] L. Lindemann, A. Rodionova, and G. J. Pappas, “Temporal robustness of stochastic signals,” arXiv preprint arXiv:2202.02553, 2022.
[15] N. Mehdipour, C.-I. Vasilie, and C. Belta, “Specifying user preferences using weighted signal temporal logic,” IEEE Contr. Syst. Letters, vol. 5, no. 6, pp. 2006–2111, 2020.
[16] D. Aksaray, C.-I. Vasilie, and C. Belta, “Dynamic routing of energy-aware vehicles with temporal logic constraints,” in 2016 IEEE Int. Conf. on Robotics and Automation (ICRA), 2016, pp. 3141–3146.
[17] C.-I. Vasilie, D. Aksaray, and C. Belta, “Time window temporal logic,” Theoretical Computer Science, vol. 691, pp. 27–54, 2017.
[18] S. Ghosh, D. Sadigh, P. Nuzzo, V. Raman, A. Donzé, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia, “Diagnosis and repair for synthesis from stl specifications,” in Proc. of the 19th Int. Conf. on Hybrid Syst.: Comp. and Control, 2016, pp. 31–40.
[19] D. Aksaray, “Resilient satisfaction of persistent and safety specifications by autonomous systems,” in AIAA Scitech Forum, 2021, p. 1124.
[20] J. Ouaknine and J. Worrell, “Some recent results in mtl,” in Int. Conf. on Formal Modeling and Anal. of Timed Syst., 2008, pp. 1–13.
[21] A. T. Buyukkocak and D. Aksaray, “Temporal relaxation of signal temporal logic specifications for resilient control synthesis,” arXiv preprint arXiv:2208.08384, 2022.
[22] J. Löfberg, “Yalmip : A toolbox for modeling and optimization in matlab,” in In Proc. of the CACSD Conf. Taipei, Taiwan, 2004.
[23] Gurobi-Optimization, “Gurobi Optimizer Ref. Manual,” 2022.