Risk Mitigation for Interval Signal Temporal Logic Monitoring and Synthesis

Luke Baird*†, Andrew Schoer†, Matthew Cleaveland†, Samuel Coogan*, and Kevin Leahy‡

***Abstract*— This letter presents a method to mitigate the risk of violation of a temporal logic specification. Given a time-varying signal with interval-valued uncertainty at each time step, we propose an optimization approach to identify time instances for which a tighter uncertainty bound is required to satisfy the specification, knowledge which can be used to, e.g., focus sensing resources to strategically reduce uncertainty. We demonstrate our method on a simulated unmanned underwater vehicle where GPS calibration is informed by the identified time instances. In contrast to existing methods that solve an uncertainty-aware optimal control program with temporal logic mixed-integer constraints, our proposed interval-tightening approach is several orders of magnitude faster to compute. Additionally, we survey methods to produce interval-valued uncertainty, specifically how probabilistic bounds may translate to confidence intervals about an entire signal.**

***Index Terms*— Uncertain systems, Computational methods, Fault detection**

I. Introduction

VERIFICATION of autonomous systems is critical to incorporating modern control techniques such as learning-based controllers into safety-critical systems. Although perfect performance is impossible to attain, an accurate estimate of performance is informative to an operator about the reliability of the system. That is, a real engineering system always incurs some *risk* of failure. Risk estimation informs the operator of the likelihood that remedial actions will be necessary and the expected frequency and cost of failures. Additionally it is useful to identify likely points of failure within a system.

The principle of focusing engineering effort on anticipated failure points may be applied to the domain of formal verification for autonomous systems with safety encoded by temporal logics. A large margin of safety does not necessarily indicate success if the measurement of the relevant signal is incorrect. On the other hand, a small margin of safety may be sufficient if the signal trace is both precisely and accurately estimated. Temporal logics may be equipped with a quantitative robustness measure providing a margin of satisfaction. Analogously, this robustness is only immediately useful if the signal against which a temporal logic formula is evaluated is perfectly determined. It is helpful as a second-order notion to consider the probability that the reported satisfaction or violation of a temporal logic formula is correct. Additionally, there exist scenarios where violation may be remedied by changing the signal trace at a few points. For instance, if precise position information at a specific point in time would assure that an unmanned underwater vehicle (UUV) completes its mission, this information can improve a resurfacing strategy for GPS position recalibration.

In this work we consider a variant of Signal Temporal Logic (STL) [1] termed Interval Signal Temporal Logic (I-STL) [2] due to the innate margin of satisfaction present with its robust semantics along with sound and efficient overapproximations of interval-valued robustness. I-STL views satisfaction from a worst-case perspective. If its robustness overapproximation contains zero, it is unclear whether the specification will be satisfied or not. Another perspective common in the literature is that of probabilistic satisfaction. There are several variants of STL that encode this, such as the paper [3] which incorporates the probability of satisfaction directly into the STL syntax. Probabilistic confidence intervals about a signal may be propagated through the syntax of STL using I-STL.

In this letter we use an arithmetic geometric mean (AGM) approximation [4] of the $\min$ and $\max$ operators to compute the gradient of robustness with respect to a signal. Although there are several variants of differentiable robustness in the literature with advantages such as soundness [5] and resilience to masking and locality [6], the AGM approximation is sufficient for our purposes since we do not use it for verification. See [7] for a survey of desirable properties of approximate robustness measures for varied applied scenarios.

The work most similar to ours is [8] which uses a risk measure on random variables that obeys axioms from [9]. The paper develops a notion of risk-tightened predicates where predicate robustness is written as the sum of a zero-mean random variable and its expectation. Then, robustness is required to be greater than a risk threshold from the zero-mean random variable in a control synthesis problem. Our method is designed to complement the literature by approaching the problem from a different standpoint, modifying an existing risk-agnostic solution to avoid resolving an integer program. In contrast, [8] resolves the solution in a risk-aware fashion.

The authors are with *Georgia Institute of Technology, †MIT Lincoln Laboratory, ‡Worcester Polytechnic Institute, respectively.
{andrew.schoer,matthew.cleaveland}@ll.mit.edu,
{luke.baird,sam.coogan}@gatech.edu, kleahy@wpi.edu
 The NASA University Leadership Initiative (grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity. Supported in part by the National Science Foundation under award #2333488.

The purpose of this letter is to characterize and mitigate the risk of violating STL formulas in both a monitoring and control synthesis framework using I-STL. Prior works using I-STL treat uncertainty as bounded [10]. This letter is born out of the observation that if either the interval bound is uncertain or if zero lies strictly in the interval-valued robustness, then satisfaction is undetermined. First, this letter surveys methods of constructing confidence intervals about a signal. Second, this letter presents a method to identify points of failure in a signal, identifying the specific points in time when precise knowledge of the signal is necessary to determine satisfaction or violation. Given a trace with interval uncertainty, we compute the interval robustness of the signal. Then, we propose a tightening approach with either a gradient ascent/descent approach or with a mixed-integer linear program (MILP) to shrink the interval until zero no longer lies in the interval using `PyTeLo` [11] and `stlpy` [12], both combined with `npinterval` [13]. Finally, we provide a simulation of a UUV scenario where this interval tightening strategy is used to inform a resurfacing strategy for position estimation.

II. Preliminary Material

A. Notation

Let $\mathbb{N}$ be the natural numbers. Let $[x] = [\underline{x}, \overline{x}]$ where $\underline{x}, \overline{x} \in \mathbb{R}^n$ are the endpoints of an interval. We denote the space of intervals in $\mathbb{R}^n$ by $\mathbb{IR}^n \simeq \mathbb{R}^{2n}$. Then, $[x] \in \mathbb{IR}^n$ or $[x] \subseteq \mathbb{R}^n$. We denote a sequence with subscripts, $x_{0:2} = \{x_0, x_1, x_2\}$. Intervals are extended to functions by inclusion functions [14].

B. Interval Signal Temporal Logic

In STL, the robustness ρ^φ of a specification φ evaluated over a signal x at a time t is a scalar. We often assume $t = 0$ and consider $x \in \mathbb{R}^{n \times \|\varphi\|}$. By contrast, *Interval Signal Temporal Logic* (I-STL) is evaluated over interval signals with quantitative semantics that give an interval-valued robustness. An I-STL specification φ is evaluated over a discrete-time interval signal $[x] \in \mathbb{IR}^{\|\varphi\| \times n}$. Using the minimal inclusion functions $[\min]$ and $[\max]$ given in [2, Proposition 1] we recall the quantitative interval robustness semantics of I-STL.

Definition 1. *(I-STL Quantitative Semantics) The* interval robustness $[\rho]^\varphi$ *of an I-STL specification* φ *evaluated over an interval signal* $[x]$ *at time step* t *is calculated recursively using natural inclusion functions [14] as*

$$
\begin{aligned}
[\rho]^{\Pi}([x],t) &= \mathcal{M}([x]_t), \quad \Pi = (\mathcal{M}([x],t) \subseteq [0,\infty]) \\
[\rho]^{\neg\varphi}([x],t) &= -[\rho]^{\varphi}([x],t) \\
[\rho]^{\varphi\wedge\psi}([x],t) &= [\min]\big([\rho]^{\varphi}([x],t), [\rho]^{\psi}([x],t)\big) \\
[\rho]^{\varphi\vee\psi}([x],t) &= [\max]\big([\rho]^{\varphi}([x],t), [\rho]^{\psi}([x],t)\big) \\
[\rho]^{\Box_{[t_1,t_2]}\varphi}([x],t) &= \min_{t'\in[t+t_1,t+t_2]} \big([\rho]^{\varphi}([x],t')\big) \\
[\rho]^{\Diamond_{[t_1,t_2]}\varphi}([x],t) &= \max_{t'\in[t+t_1,t+t_2]} \big([\rho]^{\varphi}([x],t')\big) \\
[\rho]^{\varphi\mathcal{U}_{[t_1,t_2]}\psi}([x],t) & \\
= \max_{t'\in[t+t_1,t+t_2]} &[\min]\left([\rho]^{\varphi}([x],t'), \min_{t''\in[t+t_1,t']} \big([\rho]^{\psi}([x],t'')\big)\right),
\end{aligned}
\tag{1}
$$

where $\mathcal{M} : \mathbb{IR}^n \to \mathbb{IR}$.

The *horizon* of an I-STL formula is the number of future time steps needed to evaluate the robustness of a signal at the current time step and is given in [1]. $[x] \models \varphi$ means that $[x]$ satisfies φ at time 0 for any realization of φ or $[x]$, *e.g.*, $\underline{\rho}^{\varphi}([x], 0) \geq 0$.

III. Risk Analysis of STL Violation

In this section we survey ways of soundly handling uncertain signals. We discuss propagating probabilistic bounds on a signal through an STL formula, point-wise probabilistic bounds, and limitations of these approaches. The fundamental question we consider is: what happens when zero lies inside of an interval-valued overapproximation of robustness?

An advantage of the interval-based approach is the ease of implementation and computational efficiencies afforded by I-STL. Risk encoded by intervals on state is easily propagated through a specification without an increase in computation time compared to propagating the nominal signal.

A natural way of defining risk is simply the probability of violation of an I-STL formula.

Definition 2 (Risk)**.** *The risk of violation is* $\mathbb{P}([x] \not\models \varphi)$.

This definition of risk shrouds the difficult portions of the problem; namely, how to compute the aforementioned probability. Often $\underline{\rho}^{\varphi}([x], t)$ cannot be treated as a random variable with a known distribution. Rather, information about uncertainty in state or time is present from which risk of violation must then be derived.

The paper [8] presents a systematic development of the notion of risk using axioms applied first in the finance literature [9]. The paper provides desirable properties for a measure of risk such that risk associated with individual predicates and subformulae of an STL formula may be meaningfully combined into a final measure of risk. In a sense, Definition 2 is a subset of the risk proposed in [8] in that the severity of violating any given predicate is not encoded. In this letter, we are concerned with a binary question of satisfaction.

Challenges arise from two scenarios. First, suppose that $\mathbb{P}(x \in [x]) = 1 - \delta$ with $\delta > 0$, a natural case representing a confidence interval. If this holds, then because [2] uses natural inclusion functions yielding sound overapproximations of robustness, we have that $\mathbb{P}(\rho \in [\rho]) \geq 1 - \delta$. However, constructing such confidence intervals along entire trajectories is nontrivial and is often highly conservative. We note that the conservatism from such methods motivates our approach, in that we consider *when* such conservatism matters and pinch the bounds at particular points instead of along an entire trajectory. Conformal prediction [15], [16] and Gaussian processes [17, Theorem 7] are two possible methods for constructing such confidence intervals of trajectories. Conformal prediction considers the question: given a calibration data set of trajectories, what is the probability that a given trajectory lies within a tube generated by the calibration data set? This entire tube may be evaluated over an STL formula by evaluating the vertices of a polytope along the STL formula, or using an interval overapproximation. Gaussian processes can be used

to compute confidence intervals around uncertain trajectories if the underlying statistics are estimable [18]. The paper [17] presents a way to combine this with trajectory prediction from mixed monotone systems theory to generate interval trajectories are varying confidence levels. An alternative approach from approximate semi-infinite programming is to sufficiently sample an uncertainty set to verify non-negative robustness for some risk level [19].

Second, the estimated $[\rho]$ may have zero strictly in the interior the interval. Thus, the question remains of computing the risk of violation. If $\rho \sim \mathcal{U}(\underline{\rho}, \overline{\rho})$ and $\delta \approx 0$, then $\mathbb{P}(\rho^{\varphi}([x], t) > 0) \approx 1 - \frac{-\underline{\rho}}{\overline{\rho}-\underline{\rho}}$. In general, however, ρ will depend upon the distribution of the signal along time, passed through order statistics from the robust semantics recursive formulation.

In discrete time, if pointwise probabilities of $x_k \in [\underline{x}_k, \overline{x}_k]$ are computed, the probability that the entire trajectory lies in its respective bounds is the product of the probabilities. This uncertainty may be directly passed through the STL computation. This method is in general conservative beyond usefulness, but for situations where the vast majority of $\mathbb{P}(x_k \in [x]_k) \approx 1$ except for a few points of interest, the results may approximate reality.

Another risk consideration is *when in time* is signal uncertainty likely to cause a violation. In Example 1, for both monitoring and control synthesis violation or satisfaction is determined by the point in time where the robot drives through the narrow corridor.

Example 1. *Suppose that a robot must navigate from region A to region B through a small, narrow corridor C. The state may have large ≈100% confidence bounds within A and B, but may need to use 50% confidence bounds for three time steps in C, yielding $\mathbb{P}(x \in [x]) \approx .5^3 = .125$. Then, through I-STL (see Section IV-A), $\mathbb{P}(\rho \in [\rho]^{\varphi}([x], t)) \geq .125$.*

Remark 1. *Uncertainty in delays of the system may introduce another degree of risk. For example, it may be that $x_{0:\|\varphi\|} \models \varphi$, but $x_{1:\|\varphi\|+1} \not\models \varphi$. In this work we treat such uncertainty as uncertainty in spatial robustness rather than uncertainty in temporal robustness [20]. A full analysis of the relationship between spatial and temporal robustness is worthy of its own full investigation and is beyond the scope of this letter.*

IV. Risk Mitigation with Tightening Intervals

We present our approach to mitigate risk by estimating where along a state trajectory precise knowledge of the state is necessary to determine satisfaction. We begin by discussing the propagation of trajectory probabilistic bounds through I-STL and highlight the limitations of this approach. Then, we provide an assumption to extract a singular number representing risk from our method. Finally, we construct our interval-tightening algorithm.

A. Confidence Interval Analysis

We consider where interval bounds are given around a signal where the signal is modeled as a sequence of random variables.

Proposition 1. *Let X be a random variable in $\mathbb{R}^n$, $[x] \in \mathbb{IR}^n$, and $\delta \in (0, 1)$. Let $[f] : \mathbb{IR}^n \to \mathbb{IR}^m$ be an inclusion function of a continuous function $f : \mathbb{R}^n \to \mathbb{R}^m$. If $\mathbb{P}(X \in [x]) \geq 1-\delta$, then $\mathbb{P}(f(X) \in [f]([x])) \geq 1 - \delta$.*

Proof. By definition, $f(x) \in [f]([x])$ for any $x \in [x]$ [14]. If f is continuous, we can construct an interval by evaluating $f(x)$ for every $x \in [x]$. The conclusion follows as if some interval $[y] \supseteq [x]$, then $\mathbb{P}(x \in [y]) \geq \mathbb{P}(x \in [x])$. □

Remark 2. *We may apply the above proposition to I-STL robustness over $\mathbb{IR}^{n\times\|\varphi\|}$. Let $X_{0,\dots,\|\varphi\|}$ be a sequence of random variables. Because $[\rho]^{\varphi}([x], 0) : \mathbb{IR}^{n\times\|\varphi\|} \to \mathbb{IR}$ is an inclusion function for the robustness of an STL formula ϕ formed from any realization of φ's predicate inclusion functions [2, Theorem 1], we have that if $\mathbb{P}(X_{0,\dots,\|\varphi\|} \in [x]_{0,\dots,\|\varphi\|}) \geq 1 - \delta$, then $\mathbb{P}(\rho^{\phi}(X, 0) \in [\rho]^{\varphi}([x], 0)) \geq 1 - \delta$.*

Next, we consider the stochastic setting where X is a sequence of random variables. This demands that we compute a sound method of computing (over)approximations of the order statistics to complete the I-STL semantics. However, the bounds quickly become too conservative to be meaningful. Consider $\mathbb{P}(\min(X_1, X_2) \in [x])$. If X_1 and X_2 are independent, the worst case reduces to the probability that $X_1 \in [x]$ and $X_2 \in [x]$, that is, $\geq (1 - \delta)^2$. If X_1 and X_2 are perfectly correlated or anti-correlated , then the probability is bounded below by $1 - \delta$. Over horizon length $\|\varphi\|$, an `eventually` or `always` operator will result in a bound that looks like $(1 - \delta)^{\|\varphi\|}$, similar to Example 1.

B. Constricting Intervals for Risk Estimation

We present our algorithm of a computationally efficient method for risk estimation as follows. In this section we consider state uncertainty as the size of intervals around a nominal trajectory. To recover Definition 2 for risk, we make an assumption regarding a mapping from confidence intervals to a risk estimate.

Assumption 1. *There exists a function $[f] : \mathbb{IR}^n \to [0, 1]$ mapping intervals to probability that the signal $x : \mathbb{R}^+ \to \mathbb{R}^n$ lies in the interval signal $[x] \in \mathbb{IR}^n$.*

Thus, as we iteratively shrink intervals, we can quantify risk. Approximations of $[f]$ can come from conformal prediction or any of the aforementioned methods. For example, we note one control-theoretic example of how an $[f]$ may be computed. Consider the first case study in [17, Section VI]. The interval-valued reachable sets for varying probability levels can be computed in a few milliseconds. The case study uses a probabilistic bound of $\alpha \approx 99\%$ (from 3σ for a Gaussian). This parameter can be tuned to generate intervals of varying widths, approaching zero as $\alpha \to 0$. A simple line search can therefore be used to map trajectory interval uncertainty to probabilistic bounds. [17, Theorem 7] provides a constructive method and conditions for constructing these intervals.

Now, suppose that we are given an interval-valued signal $[x] \in \mathbb{IR}^{n\times\|\varphi\|}$ and a bounded-horizon STL formula φ. Assume that $[x]$ is a large overapproximation of a nominal signal $x \in \mathbb{R}^{n\times\|\varphi\|}$ such that is virtually certain that $x \in$

$[x]$. If $0 \in [\rho]^{\varphi}([x], t)$ when evaluated through I-STL, it is indeterminate whether the true $x \models \varphi$.

We propose two methods to perform interval tightening, one using gradients and the other using a MILP. The former performs gradient ascent on $\underline{x}$ and gradient descent on $\overline{x}$ with a stopping condition that for each, robustness is non-negative. Afterwards, the final interval robustness is checked against $[\underline{x}, \overline{x}]$ to ensure that the solution is valid. The gradient approach is simpler, but like all gradient algorithms struggles with specifications whose robustness functions are nonconvex functions of a signal while the MILP encoding is generally applicable to formulas with affine predicates.

We next address the masking problem detailed as follows. Consider an STL formula $\square_{[0,10]}(x \geq 0)$. If $x = [-1, -1, -1, -10, -1, -1, -1, -1, \dots]$, then the gradient of ρ with respect to x is $[0\ 0\ 0\ 1\ 0\ \dots]$. That is, a singular point masks the contribution of the rest of the signal to robustness. For gradient ascent, it is better to increase the entire signal instead of only a single worst point in time, as eventually all must be ≥ 0 per the STL formula. To handle this we use a smooth approximation of robustness with an arithmetic-geometric mean (AGM) [4]. The smooth approximation of robustness is denoted $\tilde{\rho}$.

We acknowledge a trade-off where unnecessary tightening may occur for some specifications. For instance, consider the STL formula $\Diamond_{[0,3]}(x \geq 0)$ with $x = [-1, -1, -0.9, -1]$. While using standard robustness would result in increasing only index 2, using AGM robustness will lead to small increases in the other values as well. This is not a novel phenomenon in our work, and other methods that use smooth approximations of robustness have this apparent behavior as well [4], [21]. This slight limitation is acceptable in light of the potential computation speedup from mitigating masking.

We preserve the use of the unapproximated robustness for our algorithm terminal condition as monitoring robustness is fast, *e.g.* [2] reports sub-10 ms evaluation times. This avoids extra tightening from the fact that AGM robustness is sound but not complete. Our algorithm is given in Algorithm 1.

Algorithm 1 Iterative Shrinking of Confidence Intervals

Require: φ, $[x]$, t
Ensure: $\underline{\rho}^{\varphi}([x]^*, t) \geq 0$

$\underline{x}^* \leftarrow \underline{x}$
while $\underline{\rho}(\underline{x}^*, t) < 0$ **do**
  $\underline{x}^* \leftarrow \underline{x}^* + \alpha \nabla_x \tilde{\rho}(x^*, t)$
end while
$\overline{x}^* \leftarrow \overline{x}$
while $\underline{\rho}(\overline{x}^*, t) < 0$ **do**
  $\overline{x}^* \leftarrow \overline{x}^* - \alpha \nabla_x \tilde{\rho}(\overline{x}^*, t)$
end while
if $\underline{\rho}^{\varphi}([x]^*, t) \geq 0$ **then**
  return $[\underline{x}^*, \overline{x}^*]$
else
  return False
end if

We provide one limited example where Algorithm 1 is guaranteed to converge with minor additional engineering effort—directed specifications [22].

Definition 3 (Directed Specification)**.** *Let* $\rho^{\varphi} : \mathbb{R}^{n \times \|\varphi\|} \to \mathbb{R}$ *be the robustness of STL specification* φ*.* φ *is* directed *if* $\nabla_x \rho^{\varphi}(x, t) \geq 0$ *element-wise for all* $x \in \mathbb{R}^n$*.*

Proposition 2. *Let* φ *be a directed specification with continuous predicates. Suppose that there exists some* $\hat{x}$ *such that* $\rho(\hat{x}, t) \geq 0$*. Then, adding the line* $\overline{x}^* \leftarrow \max\{\overline{x}^*, \underline{x}^*\}$ *element-wise will cause Algorithm 1 to successfully terminate.*

Proof. If φ is directed, then increasing the value of x can never decrease $\rho(x, t)$. Noting that ρ is continuous, the gradient algorithms give $\rho^{\varphi}(\underline{x}^*, t) \geq 0$ and $\rho^{\varphi}(\overline{x}^*, t) \geq 0$. We ensure continuity of the interval $[x] = [\underline{x}, \overline{x}]$ by updating $\overline{x}^* \leftarrow \max\{\overline{x}^*, \underline{x}^*\}$ element-wise. Because φ is directed, if $\overline{x}^* \geq \underline{x}^*$, then $\rho(\underline{x}^*, t) \leq \rho(\overline{x}^*, t)$. □

If instead the bounds of $[\rho]^{\varphi}([x], t)$ are not convex functions of a signal or if Algorithm 1 fails, we may use an MILP formulation from [12]. Let x be the signal if there were no uncertainty. We may instead solve,

$$[x]^* \leftarrow \underset{e}{\text{argmax}} \|x + [e]\|_{L_2} \qquad (2)$$
$$\text{s.t. } \underline{\rho}(x + [e], t) \geq 0, \qquad [e] \subseteq [e]_0$$

We pick $[e]_0$ be a maximum interval uncertainty to keep the problem bounded. This is far simpler than I-STL control synthesis problems as there are no dynamics constraints and is demonstrated in the UUV case study in Section V-B.

An advantage of our approach is that it informs the operator where along a plan or signal it is critical to have precise knowledge of state, or to achieve a desired state by a deadline.

Remark 3. *The eventually operator* $\Diamond$ *may result in non-uniqueness in solutions. Consider for example* $[x](t) = \{[-1, 1], [-1, 1], [-1, 1]\}$ *with* $\varphi = \Diamond_{[0,2]}(x \geq 0)$*. Increasing the lower bound of* $[x]$ *at any point in time to* $[0, 1]$ *will yield* $\underline{\rho} \geq 0$*. If traditional robustness is used in the gradient ascent algorithm, then numeric idiosyncrasies or* ε*-perturbations may result in any one of these three values becoming* $[0, 1]$*. However, if AGM robustness is used, we will have* $[x] = \{[0, 1], [0, 1], [0, 1]\}$*. The true best case may be represented using a probability, that is, the probability that any of these signals occur combined as disjunction. The point is that disjunction results in non-uniqueness.*

V. Examples

A. Monitoring an Interval-Valued Signal with I-STL

We demonstrate our approach first with an academic example. Consider the following specification: the state x must exceed 1 and dip below -1 every 8 seconds. In STL,

$$\varphi = \square_{[0,10/\Delta t]}\big(\Diamond_{[0,8/\Delta t]}(x \geq 1) \wedge \Diamond_{[0,8/\Delta t]}(x \leq -1)\big). \quad (3)$$

The outer "always" comes from choosing how long to monitor this specification for in time. We select $\Delta t = 0.5s$, thus $\|\varphi\| = 36$. This specification requires that the state periodically increases and decreases, introducing competing components in its predicates.

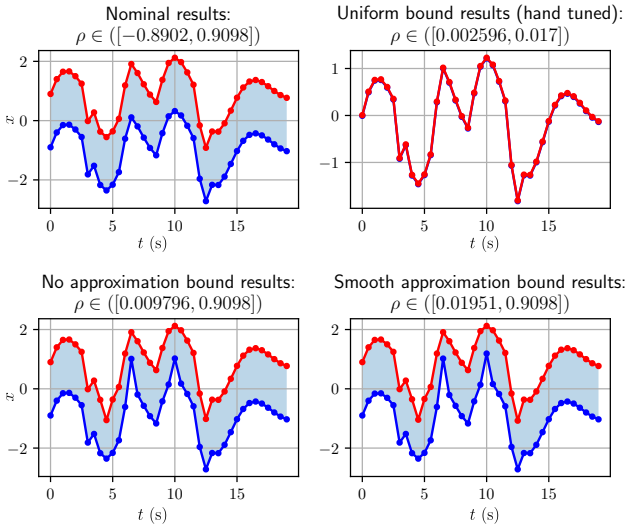

Fig. 1. Top left: unsafe nominal results where the lower bound of robustness is negative for the $100\,\%$ confidence interval. Top right: $20\,\%$ confidence bounds with uniform tightening of the upper and lower constraints by hand. The width is extremely narrow as mostly only the upper bound needed to be changed to achieve positive robustness, but for fixed confidence intervals the lower bound must be changed commensurately. Bottom left: the results of executing Algorithm 1 with nonsmooth traditional robustness. Bottom right: the results of executing Algorithm 1 with AGM robustness. The results in the bottom two figures are nearly identical, but the AGM version executed nearly $40\,\%$ faster. The tightening primarily occurred at three locations: $\boldsymbol{t = 4.5, 6.5, 10}$.

TABLE I

EXECUTION TIME COMPARISON OF THE AGM AND NONSMOOTH TRADITIONAL ROBUSTNESS VERSIONS OF THE ALGORITHM

	Execution Time	Robustness Interval	Probability
Nonsmooth	13.9 s	[.002, 1.232]	.693
AGM	9.1 s	[.001, 1.232]	.693

We first use `stlpy` [12] to generate a nominal trajectory from a double integrator system that satisfies this specification. Then, we introduce a function to translate between interval uncertainty width and probability of satisfaction, that is,

$$[f](\delta) := [-.9\delta^3, .9\delta^3] \tag{4}$$

$$[f]^{\wedge}(\underline{\varepsilon}, \overline{\varepsilon}) = \min\{\sqrt[3]{\underline{\varepsilon}/.9}, \sqrt[3]{\overline{\varepsilon}/.9}\} \tag{5}$$

where $\wedge$ represents the left quasi-inverse. Thus, the 100% confidence interval has an interval diameter of $\varepsilon = 0.9$. We represent the interval state as $[x] = [\mathring{x} - \varepsilon, \mathring{x} + \varepsilon]$ where $\mathring{x}$ is the nominal state. As an example, the I-STL interval robustness for the 100% confidence interval is $[\rho] = [-0.8902, 0.9098]$.

Next, we incorporate the package `npinterval` [13] with `PyTeLo` [11] to implement I-STL for `PyTeLo` along with the AGM robustness from [4]. We use `JAX` [23] to just-in-time compile the gradient ascent and descent routines. `JAX` performs automatic differentiation of both the AGM robustness and traditional robustness of φ yielding significant computational speedup. Then, we execute our algorithm initializing from 100% confidence intervals. Results for the traditional robustness and AGM robustness are listed in Table I. The plots of the final trajectories are given in Figure 1. A hand-tuned result is given corresponding to a 20% confidence interval.

To illustrate the advantage of interval tightening over resolving, we increased all time scales in φ by a factor of five. Now, $\|\varphi\| = 180$. Given an interval signal whose interval robustness contains zero, interval tightening over an interval signal took $357\,\text{s}$. Solving a control problem maximizing a fixed-width uncertainty for a double integrator system using an approach similar to [10] with Gurobi 12.0.1 took $2842\,\text{s}$.

B. Unmanned Underwater Vehicle with Limited GPS

Consider an unmanned underwater vehicle (UUV) that is exploring various regions, capturing image data and monitoring the environment, inspired by [24]. For a given planned underwater trajectory, we consider two sources of error: tracking error and pose estimation error. For the tracking error, we assume a fixed worst-case tracking error achievable by some tracking controller. The pose estimation error experiences linear drift over time since the last GPS calibration. To recalibrate, the UUV must resurface to collect a GPS reading. However, while the UUV resurfaces, it is not executing its primary mission to monitor the underwater environment. Thus, we plan resurface events using our interval tightening algorithm.

We setup our problem as follows. The trajectory for the UUV is planned using a double integrator system with energy loss, $\dot{p}_j = v_j$ and $\dot{v}_j = -\alpha v_j + u_j$, for $j \in \{x, y\}$ and $\alpha = 0.1$. We are interested in exploring three regions labeled R_1, R_2, and R_3, visualized in Figure 2. Our mission is specified as follows: eventually in the first 25 minutes, monitor each region for 5 minute blocks. This is written in STL as

$$\begin{aligned}\varphi = &\Diamond_{[0,25]}\Box_{[0,5]}(p \in R_1) \wedge \Diamond_{[0,25]}\Box_{[0,5]}(p \in R_2) \\ &\wedge \Diamond_{[0,25]}\Box_{[0,5]}(p \in R_3)\end{aligned} \tag{6}$$

We use a discretization rate of $0.25\,\text{min}$. We assume a fixed tracking error of $1\,\text{m}$ and a state estimation error of $\left(0.03\frac{m}{s}\right) t_{\text{GPS}}$, where t_{GPS} is the time since the last resurface.

First, we use a naïve approach where a worst-case measurement error is assumed that is too large, *i.e.*, resurfacing occurs periodically, but it is not planned for. That is, we start with a nominal signal x and add a constant interval $[e] = [-12, 12]m$ to it, yielding an initial interval robustness of $[-9.5, 14.5]$. This is used to initialize the interval tightening algorithm.

Next, we apply Algorithm 1 with the MILP formulation (2). We note where the boxes were tightened, and use these points to inform our resurfacing strategy. The reconstructed interval trajectory from the points of resurfacing is given in the left three panes of Figure 2, with an interval robustness of $[0, 10]$.

Finally, for a comparison, we solve a control synthesis problem from scratch accounting for uncertainty,

$$\begin{aligned}&\max \underline{\rho}([p], 0) \\ \text{s.t. } &\begin{bmatrix} p_j^{(k+1)} \\ v_j^{(k+1)} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1-\alpha \end{bmatrix} \begin{bmatrix} p_j^{(k)} \\ v_j^{(k)} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} u_j, \\ &\qquad k = 0, \ldots, \|\varphi\|,\ j \in \{x, y\} \\ &\underline{\rho}([p], 0) \geq 0, \qquad [p] = p + [e]\end{aligned} \tag{7}$$

with a periodic resurfacing strategy that defines the interval error signal $[e]$ for the optimization program (7). The results are plotted in the rightmost pane of Figure 2. Both methods resulted in 19 resurfaces, but our method took $0.9\,\text{s}$ while resolving the control synthesis problem took $1634\,\text{s}$. This

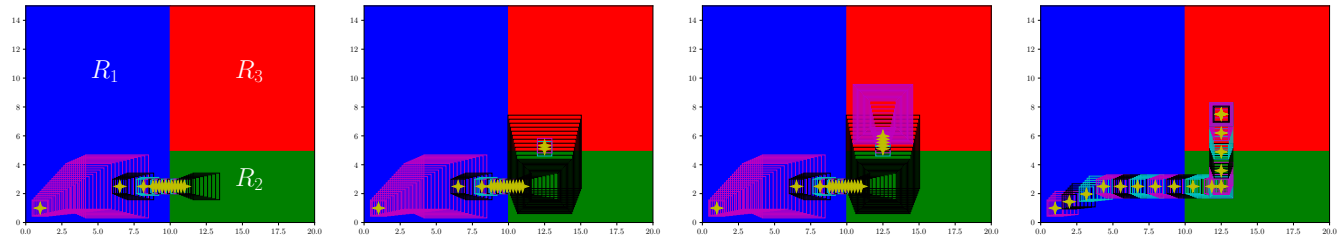

Fig. 2. From left to right plotted $\boldsymbol{p_x}$ vs $\boldsymbol{p_y}$: interval tightening results evolving over time 1/2 way through the run, 3/4 through, completed, and comparison with a periodic resurfacing strategy with a resynthesized path plan. Yellow stars represent points of resurfacing with the uncertainty boxes changing colors after each resurface. Resolving a control synthesis problem results in a periodic resurface strategy. The periodic comparison resurfacing strategy does not take into account more difficult portions of the formula such as when transitions are made between regions. Unlike the periodic resurface strategy, our method prescribes resurfacing only near boundary transitions, accepting large uncertainty inside individual regions.

highlights the computational advantages of avoiding resolving the control synthesis optimization program and adapting the resurfacing strategy to an existing solution. The number of resurfaces is sensitive to parameters and the precise φ but empirically the interval tightening method often requires fewer resurfaces than a fixed periodic resurfacing strategy and never vice-versa. Repeating the mission three times and employing interval tightening results in 38 resurfaces and a solve time of $23.9\,\mathrm{s}$ while periodic resurfacing used 52 resurfaces[1].

VI. Conclusion

This letter surveys risk of failure and identifies likely points of failure in the context of satisfying I-STL formulas. Given a signal, we consider approaches to determine from probabilistic uncertainty bounds the probability that a signal satisfies an I-STL formula. We then present an interval tightening algorithm that identifies where if more precise information were available satisfaction could be certified. This is demonstrated in a UUV case study informing a resurfacing strategy. Future work includes developing bounds on the probability of satisfaction using random variables. By defining a partial order on the space of random variables, one can treat the robustness of an I-STL formula as a random variable and bound it with two other random variables. An additional line of research is investigating the interaction between spatial and temporal robustness in the context of risk.

References

[1] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pp. 152–166, Springer, 2004.

[2] L. Baird, A. Harapanahalli, and S. Coogan, "Interval signal temporal logic from natural inclusion functions," *IEEE Control Systems Letters*, vol. 7, pp. 3555–3560, 2023.

[3] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Proceedings of Robotics: Science and Systems XII*, 2016.

[4] Y. Gilpin, V. Kurtz, and H. Lin, "A smooth robustness measure of signal temporal logic for symbolic control," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 241–246, 2020.

[5] N. Mehdipour, C. I. Vasile, and C. Belta, "Generalized mean robustness for signal temporal logic," *IEEE Transactions on Automatic Control*, 2024.

[6] S. Uzun, P. Elango, P. L. Garoche, and B. Açikmeşe, "Optimization with temporal and logical specifications via generalized mean-based smooth robustness measures," *arXiv preprint arXiv:2405.10996*, 2024.

[1]The code for these plots may be found at `https://github.com/gtfactslab/RiskTightening_LCSS2024`

[7] P. Varnai and D. V. Dimarogonas, "On robustness metrics for learning stl tasks," in *2020 American Control Conference (ACC)*, pp. 5394–5399, IEEE, 2020.

[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 Systems Letters*, vol. 4, no. 4, pp. 1000–1005, 2020.

[9] P. Artzner, F. Delbaen, J. M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical finance*, vol. 9, no. 3, pp. 203–228, 1999.

[10] L. Baird and S. Coogan, "Interval signal temporal logic for robust optimal control," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 5197–5202, IEEE, 2024.

[11] G. A. Cardona, K. Leahy, M. Mann, and C. I. Vasile, "A flexible and efficient temporal logic tool for python: Pytelo," *arXiv preprint arXiv:2310.08714*, 2023.

[12] V. Kurtz and H. Lin, "Mixed-integer programming for signal temporal logic with fewer binary variables," *IEEE Control Systems Letters*, vol. 6, pp. 2635–2640, 2022.

[13] A. Harapanahalli, S. Jafarpour, and S. Coogan, "A toolbox for fast interval arithmetic in numpy with an application to formal verification of neural network controlled systems," *arXiv preprint arXiv:2306.15340*, 2023.

[14] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer London, 2001.

[15] L. Lindemann, X. Qin, J. V. Deshmukh, and G. J. Pappas, "Conformal prediction for stl runtime verification," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pp. 142–153, 2023.

[16] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[17] M. Cao, M. Bloch, and S. Coogan, "Efficient learning of hyperrectangular invariant sets using gaussian processes," *IEEE Open Journal of Control Systems*, vol. 1, pp. 223–236, 2022.

[18] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*, pp. 63–71, Springer, 2003.

[19] T. Alamo, R. Tempo, and E. F. Camacho, "Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2545–2559, 2009.

[20] 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)*, pp. 572–579, IEEE, 2021.

[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*, vol. 42, no. 6, pp. 356–370, 2023.

[22] E. S. Kim, M. Arcak, and S. A. Seshia, "Directed specifications and assumption mining for monotone dynamical systems," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pp. 21–30, 2016.

[23] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[24] M. Cleaveland, E. Yel, Y. Kantaros, I. Lee, and N. Bezzo, "Learning enabled fast planning and control in dynamic environments with intermittent information," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10290–10296, IEEE, 2022.