2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Stochastic Robustness Interval for Motion Planning with Signal Temporal Logic

Roland B. Ilyes, Qi Heng Ho, and Morteza Lahijanian

***Abstract*— In this work, we present a novel robustness measure for continuous-time stochastic trajectories with respect to Signal Temporal Logic (STL) specifications. We show the soundness of the measure and develop a monitor for reasoning about partial trajectories. Using this monitor, we introduce an STL sampling-based motion planning algorithm for robots under uncertainty. Given a minimum robustness requirement, this algorithm finds satisfying motion plans; alternatively, the algorithm also optimizes for the measure. We prove probabilistic completeness and asymptotic optimality of the motion planner with respect to the measure, and demonstrate the effectiveness of our approach on several case studies.**

I. Introduction

In recent years, *Temporal Logics* (TLs) [1] have been increasingly employed to formalize complex robotic tasks. These logics allow precise description of properties over time by combining Boolean logic with temporal operators. A popular choice is *Linear TL* [2], where time is treated as linear and discrete. However, robotic systems operate in continuous time, and their tasks often include properties in dense time. Signal TL (STL) [3] is a variant of TL that provides the means for expressing such tasks with evaluations over continuous signals (trajectories). For motion planning, however, STL introduces both computational and algorithmic challenges precisely due to the same reason that makes it powerful, i.e., reasoning in continuous time. This challenge is exacerbated in real-world robotics, where uncertainty cannot be avoided. Hence, motion planners must reason about the robustness of plans by accounting for uncertainty. This requires a proper notion of a robustness measure. STL does in fact admit such a measure but only for deterministic trajectories [4]. No such a measure is known for stochastic systems. This paper takes on this challenge and aims to develop a stochastic robustness measure for STL with the purpose of using it for efficient motion planning with robustness guarantees.

Consider the following mission for the robot in Fig. 1: *Go to the charger in the next 10 minutes. But, if you traverse a puddle of water, stay away from the charger until you dry off on the carpet within 3 minutes*. Such a specification is easily captured in STL (φ_3 in (9)). Two sample trajectories of an uncertain robot are shown in Fig. 1, where the ellipses represent the 90% confidence contours around the nominal trajectories. Here, both trajectories appear to satisfy the temporal aspects of the specification. However, their spatial robustness is different with regards to their uncertainty. Note

Authors are with the department of Aerospace Engineering Sciences at the University of Colorado Boulder, CO, USA {firstname.lastname}@colorado.edu

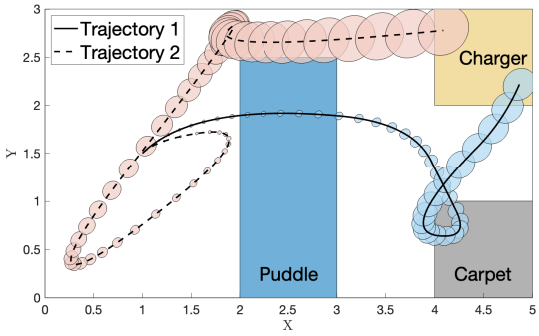


Fig. 1: Two trajectories for φ_3 in (9). Trajectory 1 has a StoRM of 0.96 and Trajectory 2 has a StoRM of 0.71.

that, even though the nominal component of Trajectory 2 gets to the charger without passing through the water puddle, its uncertainty ellipses overlap with the puddle and walls, making it less robust than Trajectory 1 with respect to the task. A capable motion planner must be able to reason about the robustness of these trajectories with respect to both temporal and spatial aspects of the STL task algorithmically.

Much of the literature for control synthesis for STL specifications has been focused on deterministic systems. Success has been found using optimization-based techniques [5]–[8], control barrier function approaches [9], and sampling-based methods [10], [11]. The sampling-based approaches are particularly suitable for robotics applications given their efficiency and scalability. None of those methods, however, account for stochasticity in the robot's dynamics.

Reasoning about systems under uncertainty with continuous-time TL specifications is a rapidly developing topic. Recent works [12]–[17] introduce new TL as extensions to STL that incorporate uncertainty in the logic itself. Nevertheless, they do not define a robustness measure for STL. Other works [13], [18] use STL and reason about the distribution of the STL robustness measure over the realizations of stochastic trajectories. This measure, however, is defined with respect to deterministic trajectories and ignores the knowledge of the system's uncertainty.

In this work, we develop a novel measure of the robustness of continuous-time stochastic trajectories with respect to STL specifications. We refer to it simply as the *Stochastic Robustness Measure* (StoRM). StoRM allows us to quantify how well a stochastic trajectory satisfies a specification. The measure is based on the recursive evaluation of the *Stochastic Robustness Interval* (StoRI) according to the semantics of STL over continuous stochastic trajectories and the probability measure. We also propose a technique to compute the

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

StoRI of partial trajectories, which allows us to reason about (monitor) the satisfaction robustness of the STL properties on an evolving stochastic trajectory. Next, we present a sampling-based algorithm that produces motion plans that can (i) satisfy a user-defined bound on StoRM, or (ii) asympotically optimize for StoRM. We prove the theoretical properties of the algorithm under StoRM constraints, and demonstrate the effectiveness of our measure and planner on a variety of STL formulas in several case studies.

In summary, the contributions of this work are four-fold: (i) a novel measure (StoRM) to quantify the robustness of stochastic trajectories against STL specifications, (ii) a new monitor to compute the robustness of partial trajectories, (iii) a probabilistically-complete planning algorithm that satisfies STL specifications with StoRM constraints for systems under uncertainty and asymptotically optimizes for StoRM, and (iv) a series of case studies and benchmarks that reveal properties of StoRM and performance of the planner.

II. System Setup

Consider a robotic system whose dynamics are described by a linear stochastic differential equation (SDE):

$$dx(t) = (Ax(t) + Bu(t))dt + Gdw(t), \tag{1}$$

where $x \in X \subset \mathbb{R}^n$ is the state, $u \in U \subset \mathbb{R}^m$ is the control input, $w(\cdot)$ is an r-dimensional Wiener process (Brownian motion) with diffusion matrix $Q \in \mathbb{R}^{r\times r}$ representing noise, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $G \in \mathbb{R}^{n\times r}$. The initial state of the robot is $x(0) = x_0$. The solution to the SDE in (1) is a continuous-time Gaussian process [19], i.e.,

$$x(t) \sim b_t = \mathcal{N}(\hat{x}(t), P(t)),$$

where b_t, called the *belief* of $x(t)$, is a normal distribution with mean $\hat{x}(t) \in X$ and covariance $P(t) \in \mathbb{R}^{n\times n}$. The evolution of b_t is governed by:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t), \tag{2}$$

$$\dot{P}(t) = AP(t) + P(t)A^T + GQG^T \tag{3}$$

with initial conditions $\hat{x}(0) = x_0$ and $P(0) = 0$. Note that if, instead of a deterministic initial state, the robot has initial uncertainty described by a Gaussian distribution, only the initial conditions $\hat{x}(0)$ and $P(0)$ change.

Let $T \in \mathbb{R}_{\geq 0}$ be a time duration. Then, given a controller $\mathbf{u} : [0, T] \to U$, a *belief (stochastic) trajectory* $\mathbf{b}$ over time window $[0, T]$ for the robotic system (1) can be computed (predicted) using (2) and (3). An execution of this controller on the robotic system, called a *realization* or *sample* of $\mathbf{b}$, is a *state trajectory* $\mathbf{x}$ over time duration $[0, T]$.

We are interested in properties of System (1) with respect to a set of *linear predicates* defined in state space X. Let $\mathcal{H} = \{h_1, h_2, ..., h_l\}$ be a given set of functions where $h_i : \mathbb{R}^n \to \mathbb{R}$ is a linear function for every $1 \leq i \leq l$. Then, the set of predicates $M = \{\mu_1, ..., \mu_l\}$ is defined on $\mathcal{H}$ such that $\forall i \in \{1, ..., l\}$, the Boolean values of $\mu_i : X \to \{\top, \bot\}$ is determined by the sign of function h_i as:

$$\mu_i(x) = \begin{cases} \top \text{ if } h_i(x) \geq 0 \\ \bot \text{ if } h_i(x) < 0. \end{cases} \tag{4}$$

To express desired properties of the robot with respect to the set of predicates M, we use *signal temporal logic* (STL) [3]. STL is a logic that allows specification of real-time temporal properties, and is therefore well-suited for continuous-time systems such as the one in (1).

Definition 1 (STL Syntax)**.** *The STL Syntax is recursively defined by:*

$$\phi := \top \mid \mu \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathcal{U}_I \phi$$

where $\mu \in M$, and $I = \langle a, b\rangle$ is a time interval with $\langle \in \{(, [\}, \ \rangle \in \{),]\}$, $a, b \in \mathbb{R}_{\geq 0}$ and $a < b < \infty$. Notations $\top$, $\neg$, and $\wedge$ are the Boolean "true," "negation," and "conjunction," respectively, and U denotes the temporal "until" operator.

The temporal operator *eventually* ($\Diamond$) is defined as $\Diamond_I \phi \equiv \top \mathcal{U}_I \phi$ and the operator *globally* is defined as $\Box_I \phi \equiv \neg\Diamond_I \neg\phi$.

Definition 2 (STL Semantics)**.** *The semantics of STL is defined over a state trajectory realization $\mathbf{x}$ at time t as:*

$$\begin{aligned}
(\mathbf{x}, t) &\models \top &&\iff \top \\
(\mathbf{x}, t) &\models \mu &&\iff h(\mathbf{x}(t)) \geq 0 \\
(\mathbf{x}, t) &\models \neg\phi &&\iff (\mathbf{x}, t) \not\models \phi \\
(\mathbf{x}, t) &\models \phi_1 \wedge \phi_2 &&\iff (\mathbf{x}, t) \models \phi_1 \wedge (\mathbf{x}, t) \models \phi_2 \\
(\mathbf{x}, t) &\models \phi_1 \mathcal{U}_{\langle a,b\rangle} \phi_2 &&\iff \exists t' \in \langle t + a, t + b\rangle \ s.t. \ (\mathbf{x}, t') \models \phi_2 \\
&&&\quad \wedge \forall t'' \in [t, t'], \ (\mathbf{x}, t'') \models \phi_1
\end{aligned}$$

where $\models$ denotes satisfaction. A state trajectory $\mathbf{x}$ satisfies an STL formula ϕ if $(\mathbf{x}, 0) \models \phi$.

In addition to the Boolean semantics, STL admits *quantitative* semantics [1]. This is traditionally defined as a robustness metric [4] based on Euclidian distance. This metric is well-defined for deterministic systems. For a stochastic process such as System (1), however, evaluation of the satisfaction of a belief trajectory is not straightforward.

In this work, we aim to develop an appropriate measure of robustness to evaluate the satisfaction of STL formulas by a belief trajectory. Specifically, we are interested in using this measure for robust motion planning for System (1). Therefore, the robustness measure must be appropriate for planning, namely sampling-based motion planning algorithms. This entails that the measure must be able to provide useful information for complete as well as partial trajectories. In the next section, we introduce such a measure. Then, we present a motion planning algorithm that asymptotically optimizes for this measure.

III. Stochastic Robustness Measure

A popular method of measuring robustness of belief trajectories in motion planning is based on the notion of chance constraints [20]–[22]. Chance constraints require that the probability of constraint violation (e.g., avoiding obstacles) not exceed some prescribed value. Motion planners typically enforce this chance constraint at each time step.

Such an approach is difficult to extend to STL formulas. This is primarily due to temporal operators and their time

intervals, e.g., the probability of violating $\Diamond_I \phi$ only needs to be below a prescribed value at one time step in I, but it is not clear at which time step to enforce this. Some approaches find success by generating constraints over time windows, and allocating risk of violating the constraints among time steps [12], [13]. However, those methods are limited to reasoning about discrete-time trajectories. This is in conflict with the fundamental idea of STL, which is expressing properties over real-valued continuous-time intervals.

Furthermore, those approaches only provide a *qualitative* (boolean) judgement of a trajectory's satisfaction with respect to a chance constraint. In contrast, we seek to define a measure that provides a *quantitative* judgement of a trajectory's satisfaction. This is analogous to robustness for deterministic STL that provides a quantitative measure beyond the qualitative Boolean semantics. Such a measure can then be extended to reason about partial trajectories, and hence, is advantageous for iterative methods for planning such as sampling-based algorithms. To illustrate what such a measure might convey, consider the following example.

Example 1. Consider the formulas $\phi_1 = \Box_I \mu$ and $\phi_2 = \Diamond_I \mu$. In the case of the $\Box$ operator, which states that a property must hold for all time $t \in I$, a quantitative measure of robustness could be characterized by the point with the lowest probability of satisfying μ in I, i.e., $\min_{t \in I} P(h(\mathbf{x}(t)) \geq 0)$, where h is the linear function that μ is defined on. If the probability of violation at that point is below a certain threshold, then it is also below that threshold at every other point. Similarly, for the $\Diamond$ operator, which states that a property must hold for *a* time point $t \in I$, we could look at the point with the *highest* probability of satisfying μ, i.e., $\max_{t \in \langle a,b \rangle} P(h(\mathbf{x}(t)) \geq 0)$. If the probability of violation at that point is above a certain threshold, then formula ϕ_2 is satisfied. A quantitative measure of stochastic robustness must incorporate this conflicting treatment of temporal operators.

This intuition guides the development of the Stochastic Robustness Interval (StoRI) as defined below.

Definition 3 (Stochastic Robustness Interval)**.** *The Stochastic Robustness Interval (StoRI) of a belief trajectory* $\mathbf{b}$ *over time window* $[0, T]$ *with respect to an STL formula* ϕ *is a functional* $f(\phi, \mathbf{b})$*:*

$$f(\phi, \mathbf{b}) = [f^{\downarrow}(\phi, \mathbf{b}), f^{\uparrow}(\phi, \mathbf{b})]$$

such that $0 \leq f^{\downarrow}(\phi, \mathbf{b}) \leq f^{\uparrow}(\phi, \mathbf{b}) \leq 1$*. For a* $t \in [0, T]$*, let* $\mathbf{b}^t$ *be the time-shifted suffix of trajectory* $\mathbf{b}$ *such that for all* $t' \in [0, T - t]$*,* $\mathbf{b}^t(t') = \mathbf{b}(t + t')$*. Then, lower bound* $f^{\downarrow}(\phi, \mathbf{b})$ *and upper bound* $f^{\uparrow}(\phi, \mathbf{b})$ *are recursively defined by:*

$$\begin{aligned}
&f^{\downarrow}(\top, \mathbf{b}) = f^{\uparrow}(\top, \mathbf{b}) = 1,\\
&f^{\downarrow}(\mu, \mathbf{b}) = f^{\uparrow}(\mu, \mathbf{b}) = P(h(x(0)) \geq 0),\\
&f^{\downarrow}(\neg\phi, \mathbf{b}) = 1 - f^{\uparrow}(\phi, \mathbf{b}),\\
&f^{\uparrow}(\neg\phi, \mathbf{b}) = 1 - f^{\downarrow}(\phi, \mathbf{b}),\\
&f^{\downarrow}(\phi_1 \wedge \phi_2, \mathbf{b}) = \max\left\{f^{\downarrow}(\phi_1, \mathbf{b}) + f^{\downarrow}(\phi_2, \mathbf{b}) - 1, 0\right\},\\
&f^{\uparrow}(\phi_1 \wedge \phi_2, \mathbf{b}) = \min\left\{f^{\uparrow}(\phi_1, \mathbf{b}), f^{\uparrow}(\phi_2, \mathbf{b})\right\},\\
&f^{\downarrow}(\phi_1 U_{\langle a,b \rangle} \phi_2, \mathbf{b}) =\\
&\quad \max_{t \in \langle a,b \rangle} \left\{ \max\left\{f^{\downarrow}(\phi_2, \mathbf{b}^t) + \min_{t' \in [0,t]} f^{\downarrow}(\phi_1, \mathbf{b}^{t'}) - 1, 0\right\}\right\},\\
&f^{\uparrow}(\phi_1 U_{\langle a,b \rangle} \phi_2, \mathbf{b}) =\\
&\quad \max_{t \in \langle a,b \rangle} \left\{ \min\left\{f^{\uparrow}(\phi_2, \mathbf{b}^t), \min_{t' \in [0,t]} f^{\uparrow}(\phi_1, \mathbf{b}^{t'})\right\}\right\}.
\end{aligned}$$

We define the Stochastic Robustness Measure (StoRM) to be the lower bound of StoRI.

Definition 4 (Stochastic Robustness Measure)**.** *The* Stochastic Robustness Measure *(StoRM) of a belief trajectory* $\mathbf{b}$ *with respect to STL formula* ϕ *is the lower bound of the StoRI, i.e,* $f^{\downarrow}(\phi, \mathbf{b})$.

The StoRI is an interval that aims to quantify how robustly a belief trajectory $\mathbf{b}$ satisfies an STL formula ϕ. The StoRM of a belief trajectory is the lower bound of its StoRI. A trajectory always satisfies a Boolean $\top$, so the StoRI for $\phi = \top$ is $[1, 1]$ for every trajectory. If $\phi = \mu$ is a linear predicate, both bounds of the StoRI are the probability that the state at time zero $x(0)$ satisfies the linear predicate. Works [20] [23] outline an efficient way to calculate this probability for Gaussian distributions. However, we stress that the StoRI is defined for general belief distributions, and also applies to non-Gaussian beliefs.

The StoRI of the negation of a formula $\neg\phi$ derives from the Unit Measure Axiom of Probability [24]. Note that the lower bound of the StoRI of $\neg\phi$ depends on the upper bound of the StoRI of ϕ, and vice-versa.

The StoRI for a conjunction of two formulas $\phi_1 \wedge \phi_2$ is inspired by the lower and upper bounds on the probability of a conjunction of two events. These are the Boole-Fréchet inequalities for logical conjunction [25], which make no assumptions on the independence on the events, and hence are general. Note that the measure seeks to quantify both upper and lower bounds because the upper bound is conservative when the specification seeks to enforce 'avoidance' of the conjunction, but the lower bound is conservative when it seeks to enforce the 'occupancy' of the conjunction.

The StoRI of $\phi_1 U_{\langle a,b \rangle} \phi_2$ is less straightforward. Intuitively, it aims to report the StoRM at the time $t \in \langle a, b \rangle$ that best balances the point-wise probabilities that ϕ_1 hold for all $t' \in [0, t]$ and the probability that ϕ_2 hold at t.

Note that by following Definition 3, the obtained StoRI for eventually and globally are:

$$f(\Diamond_{\langle a,b \rangle} \phi, \mathbf{b}) = \left[\max_{t \in \langle a,b \rangle} f^{\downarrow}(\phi, \mathbf{b}^t), \max_{t \in \langle a,b \rangle} f^{\uparrow}(\phi, \mathbf{b}^t) \right] \quad (5)$$

$$f(\Box_{\langle a,b \rangle} \phi, \mathbf{b}) = \left[\min_{t \in \langle a,b \rangle} f^{\downarrow}(\phi, \mathbf{b}^t), \min_{t \in \langle a,b \rangle} f^{\uparrow}(\phi, \mathbf{b}^t) \right] \quad (6)$$

These intervals are fully inline with the intuition of the measure as discussed in Example 1.

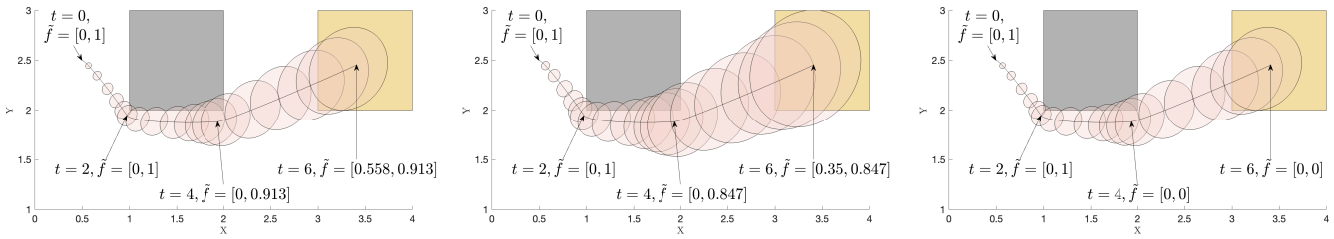


Fig. 2: Sample trajectories that have a StoRI of (left-to-right) $[0.72, 0.91]$, $[0.52, 0.847]$, $[0, 0]$ with respect to φ_1 in (7)

Example 2. Consider the environment in Fig. 2 and formula

$$\varphi_1 = \neg(x \geq 1 \wedge y \geq 2 \wedge x \leq 2) U_I (x \geq 3 \wedge y \geq 2). \quad (7)$$

Fig. 2 shows three belief trajectories, where the ellipses represent the 90% confidence bounds of their uncertainty. The first two trajectories (left and middle) have the same expectation but are subject to different amounts of process noise. When evaluating against φ_1 with time interval $I = [0, 6]$, we see both bounds of the StoRI decrease from $[0.72, 0.91]$ for the first trajectory (left) to $[0.52, 0.85]$ for the the second trajectory that has more uncertainty (middle). This trajectory is less likely to clear the obstacle, and also less likely to arrive in the goal region. The third plot (right) shows the first trajectory but evaluated against φ_1 with smaller interval $I = [0, 4]$. The robot does not arrive to goal until after 5 seconds. Hence, the robot fails to satisfy the specification and its StoRI is $[0, 0]$.

An important property of the StoRI is that a trajectory which has a StoRM of 1 satisfies STL formula ϕ with probability 1, as stated in the following theorem, which illustrates soundness of StoRM.

Theorem 1. *Given belief trajectory* $\mathbf{b}$ *and STL formula* ϕ*, if the Stochastic Robustness Measure* $f^{\downarrow}(\phi, \mathbf{b}) = 1$*, then every realization* $\mathbf{x}$ *of* $\mathbf{b}$ *satisfies the specification, i.e.,* $\mathbf{x} \models \phi$*, except realizations on the set of measure zero.*

Proof Sketch. The proof follows the recursive definition of StoRI, and the fact that $f^{\downarrow}(\phi, \mathbf{b}) \leq f^{\uparrow}(\phi, \mathbf{b})$. If $f^{\downarrow}(\mu, \mathbf{b}) = 1$, this implies that $P(h(\mathbf{x}(0)) \geq 0) = 1$, i.e., almost every realization of the trajectory satisfies the predicate μ. Since ϕ is recursively defined by $\top$ and μ, by the recursive definition of StoRI, we see that $f^{\downarrow}(\phi, \mathbf{b}) = 1$ implies that almost every realization of the trajectory satisfies ϕ. A detailed proof is provided in the extended version of this paper [26]. □

A. StoRI Monitor

StoRI is defined for a given belief trajectory. It does not account for *what could happen if we extend the trajectory.* However, in some cases (such as planning), we are interested in extending trajectories to achieve a higher StoRM. Hence, we need a monitor for StoRI, which assumes the given trajectory is to be extended. The monitor must account for all the possible suffixes of the trajectory and how they might change the StoRM. In this section, we present a monitor for the StoRI with respect to a partial belief trajectory, that acts to bound the achievable StoRI of any extensions of it.

Definition 5 (StoRI Monitor)**.** *The Stochastic Robustness Interval (StoRI) Monitor of a partial belief trajectory* $\mathbf{b}_t$ *over time window* $[0, t]$ *with respect to an STL formula* ϕ *is a functional* $\tilde{f}(\phi, \mathbf{b}_t)$*:*

$$\tilde{f}(\phi, \mathbf{b}_t) = [\tilde{f}^{\downarrow}(\phi, \mathbf{b}_t), \tilde{f}^{\uparrow}(\phi, \mathbf{b}_t)]$$

such that $0 \leq \tilde{f}^{\downarrow}(\phi, \mathbf{b}_t) \leq \tilde{f}^{\uparrow}(\phi, \mathbf{b}_t) \leq 1$*, where*

$$\tilde{f}(\top, \mathbf{b}_t) = f(\top, \mathbf{b}_t)$$
$$\tilde{f}(\mu, \mathbf{b}_t) = f(\mu, \mathbf{b}_t)$$
$$\tilde{f}^{\downarrow}(\neg\phi, \mathbf{b}_t) = 1 - \tilde{f}^{\uparrow}(\phi, \mathbf{b}_t),$$
$$\tilde{f}^{\uparrow}(\neg\phi, \mathbf{b}_t) = 1 - \tilde{f}^{\downarrow}(\phi, \mathbf{b}_t),$$
$$\tilde{f}^{\downarrow}(\phi_1 \wedge \phi_2, \mathbf{b}_t, t) = \max\left\{\tilde{f}^{\downarrow}(\phi_1, \mathbf{b}_t) + \tilde{f}^{\downarrow}(\phi_2, \mathbf{b}_t) - 1, 0\right\},$$
$$\tilde{f}^{\uparrow}(\phi_1 \wedge \phi_2, \mathbf{b}_t, t) = \min\left\{\tilde{f}^{\uparrow}(\phi_1, \mathbf{b}_t), \tilde{f}^{\uparrow}(\phi_2, \mathbf{b}_t)\right\},$$
$$\tilde{f}^{\downarrow}(\phi_1 U_{\langle a,b\rangle} \phi_2, \mathbf{b}_t) = \begin{cases} f^{\downarrow}(\phi_1 U_{\langle a,b\rangle} \phi_2, \mathbf{b}_t) & \text{if } t \geq a \\ 0 & \text{otherwise,} \end{cases}$$
$$\tilde{f}^{\uparrow}(\phi_1 U_{\langle a,b\rangle} \phi_2, \mathbf{b}_t) =$$
$$\begin{cases} f^{\uparrow}(\phi_1 U_{\langle a,b\rangle} \phi_2, \mathbf{b}_t) & \text{if } t \geq b \\ \max\left\{f^{\uparrow}(\phi_1 U_{\langle a,b\rangle} \phi_2, \mathbf{b}_t), \min\limits_{t' \in [0,b]} f^{\uparrow}(\phi_1, \mathbf{b}_t^{t'})\right\} & \text{if } t \in [0, b) \\ 1 & \text{otherwise.} \end{cases}$$

The differences between the StoRI and StoRI Monitor arise in the temporal operators. These operators seek to bound possible future robustness, and also quantify the robustness of the behavior already seen. This is apparent in the StoRI Monitor with respect to the $\Diamond$ and $\Box$ operators:

$$\tilde{f}(\Diamond_{\langle a,b\rangle}\phi, \mathbf{b}_t) = \begin{cases} f(\Diamond_{\langle a,b\rangle}\phi, \mathbf{b}_t) & \text{if } t \geq b \\ \left[\max\limits_{t \in \langle a,b\rangle} f^{\downarrow}(\phi, \mathbf{b}_t^t),\ 1\right] & \text{if } a \leq t < b \\ [0, 1] & \text{if } t < a \end{cases}$$

$$\tilde{f}(\Box_{\langle a,b\rangle}\phi, \mathbf{b}_t) = \begin{cases} f(\Box_{\langle a,b\rangle}\phi, \mathbf{b}_t) & \text{if } t \geq b \\ \left[0,\ \min\limits_{t \in \langle a,b\rangle} f^{\uparrow}(\phi, \mathbf{b}_t^t)\right] & \text{if } a \leq t < b \\ [0, 1] & \text{if } t < a \end{cases}$$

The case when $a \leq t < b$ is of particular interest. In the case of the $\Box$ operator, the StoRI is upper bounded by the best point in the partial trajectory, but is lower bounded by zero to account for possible future violation. In the case of the $\Diamond$ operator, the StoRI is lower bounded by the best point in the partial trajectory, but is upper bounded by one to account for possible future "perfect" satisfaction.

The following theorem proves the correctness of StoRI Monitor by showing that it always subsumes StoRI.

Theorem 2. *Let* $\mathbf{b}$ *be a belief trajectory over time window* $[0, T]$ *and* $\mathbf{b}_t$ *be a prefix of* $\mathbf{b}$ *where* $t \in [0, T]$*. Then, given STL formula* ϕ*, the StoRI Monitor for* $\mathbf{b}_t$ *subsumes the StoRI of* $\mathbf{b}$ *for all* $t \in [0, T]$*, i.e.,* $\forall t \in [0, T]$*,* $f(\phi, \mathbf{b}) \subseteq \tilde{f}(\phi, \mathbf{b}_t)$*.*

Proof Sketch. Due to space constraints, we provide the general sketch of this proof. A detailed proof is provided in the extended version of this paper [26]. Without loss of generality, consider $\phi = \psi_1 U_{[a,b]} \psi_2$. Using definitions 5 and 3, the main idea of the proof comes from the fact that the upper bound is a monotonically decreasing function, and that the lower bound is a monotonically increasing function. Additionally, for a complete trajectory $\mathbf{b}$, $\tilde{f}^{\uparrow}(\phi, \mathbf{b}) \geq f^{\uparrow}(\phi, \mathbf{b})$ and $\tilde{f}^{\downarrow}(\phi, \mathbf{b}) \leq f^{\downarrow}(\phi, \mathbf{b})$. □

Example 3. Recall the STL formula in Example 2 and sample trajectories in Fig. 2. The StoRI Monitor of the trajectories at different points in their evolution are shown in the plots in Fig. 2. Note that they subsume the StoRI of the final trajectory and the interval gets smaller with time.

IV. Motion Planning Algorithm

This section presents two sampling-based motion planners that utilize StoRI and the StoRI Monitor. The first planner finds solutions that satisfy a given StoRM constraint, and the second finds solutions that directly optimize for the StoRM.

A. StoRI-$\mathcal{A}$

A kinodynamic sampling-based tree planner $\mathcal{A}$ grows a motion tree in the state space X according to the robot dynamics through sampling and extension procedures. We generalize planner $\mathcal{A}$ to StoRI-$\mathcal{A}$ (Alg. 1) to generate plans for System (1) that are guaranteed to satisfy a given lower bound $\kappa \in [0, 1]$ on StoRM with respect to STL formula ϕ.

Algorithm 1: StoRI-$\mathcal{A}$ $(X, U, \phi, \hat{x}_0, N, \kappa)$

```
P_0 ← 0^{n×n};
𝕍 ← {(x̂_0, P_0)}, 𝔼 ← ∅;
𝔾 ← {𝕍, 𝔼};
for N iterations do
    x̂_rand, t_rand, x̂_near, P_near ← sample(X, 𝕍, T);
    x̂_new, P_new ← Extend(x̂_near, P_near, U);
    if f̃↑(φ, (x̂_0, P_0)...(x̂_new, P_new)) > κ then
        𝕍 ← 𝕍 ∪ {(x̂_new, P_new)} ;
        𝔼 ← 𝔼 ∪ {[(x̂_near, P_near), (x̂_new, P_new)]};
        if f↓(φ, (x̂_0, P_0)...(x̂_new, P_new)) > κ then
            return (x̂_0, P_0)...(x̂_new, P_new);
return ∅
```

The algorithm first initializes the tree $\mathbb{G}$ with the belief of $x(0)$. Each node in this tree is a tuple $(\hat{x}, P)$ of the mean and covariance of the distribution that describes the state. At every iteration, the algorithm samples a random state $\hat{x}_{rand}$ and time t_{rand} and computes the nearest existing node $(\hat{x}_{near}, P_{near})$. t_{rand} is sampled from the time-horizon of the STL formula as defined in [27], and the nearest node is selected using a distance metric that accounts for both state distance $\|\hat{x}_{rand} - \hat{x}_{near}\|$ and time distance $|t_{rand} - t_{near}|$. Second, a random control input $u \in U$ and time duration t is sampled and propagate the system from $(\hat{x}_{near}, P_{near})$ to generate a new belief node $(\hat{x}_{new}, P_{new})$. Third, the StoRI Monitor of the partial belief trajectory $\mathbf{b}_t = (\hat{x}_0, P_0)(\hat{x}_1, P_1) \cdots (\hat{x}_{new}, P_{new})$ is computed for the formula ϕ. If the StoRI Monitor has an upper bound $\tilde{f}^{\uparrow}(\phi, \mathbf{b}_t) \leq \kappa$, $\mathbf{b}_t$ has already violated the STL Specification and the new node is discarded. Otherwise, we add $(\hat{x}_{new}, P_{new})$ and the edge $\big((\hat{x}_{near}, P_{near}), (\hat{x}_{new}, P_{new})\big)$ to $\mathbb{G}$. Finally, $\mathbf{b}_t$ is a solution if the StoRM $f^{\downarrow}(\phi, \mathbf{b}_t) > \kappa$. This process repeats until a solution is found or for a maximum of N iterations.

Theorem 3 (Probabilistic Completeness)**.** *Planner StoRI-*$\mathcal{A}$ *in Alg. 1 is probabilistically complete if the underlying planner* $\mathcal{A}$ *is probabilistically complete.*

Proof. StoRI-$\mathcal{A}$ mimics the behavior of planner $\mathcal{A}$, only modifying its validity check. From Theorem 2, this modification only rejects nodes that are guaranteed to violate the STL Specification ϕ. Therefore, if a solution exists, StoRI-$\mathcal{A}$ will find it with probability 1 as $N \to \infty$. □

B. Asymptotically Optimal StoRI-$\mathcal{A}$ (AO-StoRI-$\mathcal{A}$)

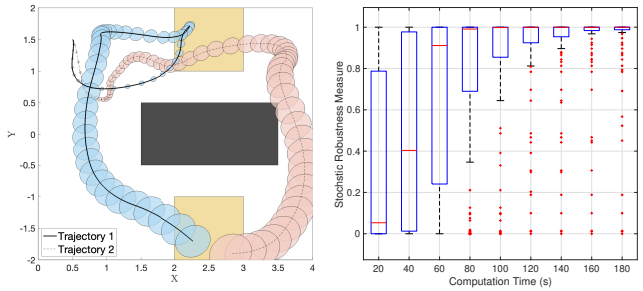
Since the StoRM allows us to compute a quantitative value of robustness, we can also optimize for the StoRM of a belief trajectory in a sampling-based motion planner. This is enabled through the AO-$\mathcal{A}$ meta-algorithm in [28], by repeatedly calling StoRI-$\mathcal{A}$ with increasing κ bounds by setting κ as the StoRM of the previous solution at each iteration [26].

V. Evaluations

We evaluate efficacy and efficiency of the proposed measure and planners subject to a variety of STL specifications. The algorithms are implemented with $\mathcal{A}$ = RRT [29], i.e., StoRI-RRT. The hyperparameters of our algorithm used in each case study are provided in [26]. All algorithms are implemented in the Open Motion Planning Library (OMPL) [30], and computations were performed on 3.9 GHz CPU and 64 GB of RAM.

We considered a noisy second-order unicycle system whose stochastic dynamics are given by: $dx = v\cos(\theta)dt + dw$, $dy = v\sin(\theta)dt + dw$, $d\theta = u_\omega dt + dw$, $dv = u_a dt + dw$ where u_a and u_ω are the acceleration and steering angle inputs. We linearized the dynamics according to the feedback linearization in [31]. We considered environments in Figs. 2, 3a, and 1 respectively with STL formulae φ_1 in (7),

$$\begin{aligned} \varphi_2 &= \neg O U_{[0,10]} A \wedge \neg O U_{[0,10]} B, \\ O &= x \geq 1.5 \wedge x \leq 3.5 \wedge y \geq -0.5 \wedge y \leq 0.5, \\ A &= x \geq 2 \wedge x \leq 3 \wedge y \geq 1, \\ B &= x \geq 2 \wedge x \leq 3 \wedge y \leq -1, \end{aligned} \tag{8}$$



(a) ϕ_2 Trajectories (b) Case Study 3 Benchmark

Fig. 3: Experiment Results. (a) shows two trajectories where $f^{\downarrow}(\varphi_2, \mathbf{b}_1) = 0.98, f^{\downarrow}(\varphi_2, \mathbf{b}_2) = 0.67$, and (b) shows benchmarking results for Case Study 3.

$$\varphi_3 = (\text{Puddle} \rightarrow \neg\text{Charge} U_{[0,3]} \text{Carpet}) U_{[0,10]} \text{Charge}, \quad (9)$$
$$\text{Puddle} = y \leq 2.5 \wedge x \geq 2 \wedge x \leq 3,$$
$$\text{Charge} = y \geq 2 \wedge x \geq 4,$$
$$\text{Carpet} = y \leq 1 \wedge x \geq 4.$$

Here, φ_2 requires avoiding (black) region O until both regions A and B are visited within 10 minutes in Fig. 3a. φ_3 requires the robot to go to the charger within 10 minutes and also that, if it visits the puddle, it must avoid the charger until it visits the carpet within 3 minutes of visiting the puddle in Fig. 1. For all formulas, we also require the robot to remain in the workspace for the duration of the mission.

A. Case Study 1 - Computation Time vs StoRI Threshold

This case study seeks to analyze the relationship between computation time and the StoRM constraint κ. Specifically, we study how StoRI-RRT performs for formula φ_2, in the environment in Fig. 3a. We ran 100 trials, with a maximum computation time of 300 seconds for each trial. Table I reports the computation time and success rate for different κ values. We see that computation time increases and success rate decreases as the κ threshold increases. This is due to the increased difficulty of satisfying the robustness constraint.

TABLE I: Benchmarking Results for Case Study 1

StoRM Threshold κ	Computation Time (s)	Success Rate
0.50	60.10 ± 64.87	97%
0.70	71.37 ± 67.48	97%
0.90	83.30 ± 69.35	90%
0.95	92.48 ± 81.56	86%

Fig. 3a shows two sample trajectories for this specification, where the ellipses represent the 90% confidence bounds of their uncertainty. Trajectory 1 uses $\kappa = 0.9$ and goes through the wider opening on the left. In contrast, Trajectory 2 uses $\kappa = 0.5$ and finds less robust paths that go through the narrow opening on the right.

B. Case Study 2 - Computation Time for Different Formulas

This case study compares computation time for different STL formulas. Table II shows the average computation time of StoRI-RRT for the three formulas and environments over 100 trials with a maximum timeout of 300s. We used StoRM bound $\kappa = 0.9$. We see that, even with a tight StoRM threshold κ, the algorithm finds solutions with a good success rate and computation time. This shows that the algorithm is generally applicable to all STL formulas.

TABLE II: Benchmarking Results for Case Study 2, $\kappa = 0.9$.

Formula	Computation Time (s)	Success Rate
φ_1	1.98 ± 2.56	100%
φ_2	85.95 ± 82.56	90%
φ_3	44.55 ± 53.89	99%

C. Case Study 3 - Asymptotic Optimality

In this case study, we analyze the relationship between given computation time and the StoRM of the resulting trajectory. Here, we study how AO-StoRI-RRT performs when planning for formula φ_3, in the environment in Fig. 1. Fig. 3b presents the results of 100 trials. It clearly shows that the solutions asymptotically approach an optimal StoRM. Fig. 1 gives two sample trajectories with different StoRMs. Trajectory 2 is found earlier in the optimization, and makes it through the gap. When the planner further optimizes for the StoRM, however, it favors trajectories like Trajectory 1, that enter the puddle and dry off before going to the carpet.

D. Case Study 4 - Simulated Performance

This case study seeks to analyze how the StoRM relates to the *statistical satisfaction rate* (SSR) defined as $\frac{\#\text{satisfying runs}}{\text{total \# runs}}$. We do this by simulating the robot's motion plans from StoRI-RRT. Table III compares the StoRI of the belief trajectories against the SSR of the simulated realizations. Each motion plan is simulated 1000 times. We use the Breach Toolbox [27] to evaluate the realizations' satisfaction. These results show correlation between the StoRM and probability of satisfaction.

TABLE III: Comparison of StoRI against SSR

Formula	$f^{\downarrow}(\phi, \mathbf{b})$ (StoRM)	$f^{\uparrow}(\phi, \mathbf{b})$	Satisfaction Rate
φ_1	0.568	0.568	0.367
φ_1	0.755	0.781	0.767
φ_1	0.986	0.986	0.985
φ_2	0.572	0.589	0.582
φ_2	0.873	0.998	0.754
φ_2	0.912	0.913	0.889
φ_3	0.625	0.677	0.669
φ_3	0.876	0.878	0.872
φ_3	0.989	0.994	0.985

VI. Conclusion and Future Work

This paper proposes a measure, StoRM, for quantifying the robustness of stochastic systems' trajectories with respect to STL specifications. We develop a monitor for this measure that reasons about partial trajectories, and use it in a sampling-based motion planner. We show desirable properties of this measure, that the algorithm is probabilistically complete, and that the algorithm asymptotically optimizes for the StoRM. Emprical evaluation demonstrates the measure and algorithm's effectiveness and utility. For future work, we plan to investigate guiding the growth of the motion tree and incorporating measurement uncertainty in the planner.

REFERENCES

[1] 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, May 2018.

[2] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA: The MIT Press, 2008.

[3] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 152–166.

[4] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2010, pp. 92–106.

[5] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 81–87.

[6] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proceedings of the 18th international conference on hybrid systems: Computation and control*, 2015, pp. 239–248.

[7] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 81–87.

[8] L. Lindemann and D. V. Dimarogonas, "Robust motion planning employing signal temporal logic," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 2950–2955.

[9] ——, "Control barrier functions for signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 1, pp. 96–101, 2018.

[10] 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)*. IEEE, 2017, pp. 3840–3847.

[11] Q. H. Ho, R. B. Ilyes, Z. N. Sunberg, and M. Lahijanian, "Automaton-guided control synthesis for signal temporal logic specifications," *arXiv preprint arXiv:2207.03662*, 2022.

[12] S. Jha, V. Raman, D. Sadigh, and S. A. Seshia, "Safe autonomy under perception uncertainty using chance-constrained temporal logic," *Journal of Automated Reasoning*, vol. 60, no. 1, pp. 43–62, 2018.

[13] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. Soudjani, "Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3324–3331, 2018.

[14] C. Yoo and C. Belta, "Control with probabilistic signal temporal logic," *arXiv preprint arXiv:1510.08474*, 2015.

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

[16] K. M. B. Lee, C. Yoo, and R. Fitch, "Signal temporal logic synthesis as probabilistic inference," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5483–5489.

[17] M. Tiger and F. Heintz, "Incremental reasoning in probabilistic signal temporal logic," *International Journal of Approximate Reasoning*, vol. 119, pp. 325–352, 2020.

[18] E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti, "System design of stochastic models using robustness of temporal properties," *Theoretical Computer Science*, vol. 587, pp. 3–25, 2015.

[19] S. Särkkä and A. Solin, "Applied stochastic differential equations." Cambridge University Press, 2019, vol. 10, ch. 3.

[20] L. Blackmore, H. Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *2006 American Control Conference*. IEEE, 2006, pp. 7–pp.

[21] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, 2010, p. 8160.

[22] Q. H. Ho, Z. N. Sunberg, and M. Lahijanian, "Gaussian belief trees for chance constrained asymptotically optimal motion planning," *arXiv preprint arXiv:2202.12407*, 2022.

[23] E. Pairet, J. D. Hernandez, M. Carreras, Y. Petillot, and M. Lahijanian, "Online mapping and motion planning under uncertainty for safe navigation in unknown environments," *IEEE Transactions on Automation Science and Engineering*, pp. 1–23, 2021.

[24] A. N. Kolmogorov and A. T. Bharucha-Reid, *Foundations of the theory of probability: Second English Edition*. Courier Dover Publications, 2018.

[25] M. Fréchet, "Généralisation du théoreme des probabilités totales," *Fundamenta mathematicae*, vol. 1, no. 25, pp. 379–387, 1935.

[26] R. Ilyes, Q. H. Ho, and M. Lahijanian, "Stochastic robustness interval for motion planning with signal temporal logic," 2022. [Online]. Available: https://arxiv.org/abs/2210.04813

[27] 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.

[28] K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in a state–cost space," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1431–1443, 2016.

[29] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[30] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, https://ompl.kavrakilab.org.

[31] A. De Luca, G. Oriolo, and M. Vendittelli, "Stabilization of the unicycle via dynamic feedback linearization," *IFAC Proceedings Volumes*, vol. 33, no. 27, pp. 687–692, 2000.