Signal Temporal Logic Synthesis as Probabilistic Inference

Ki Myung Brian Lee, Chanyeol Yoo and Robert Fitch

Abstract— **We reformulate the signal temporal logic (STL) synthesis problem as a maximum a-posteriori (MAP) inference problem. To this end, we introduce the notion of random STL (RSTL), which extends deterministic STL with random predicates. This new probabilistic extension naturally leads to a synthesis-as-inference approach. The proposed method allows for differentiable, gradient-based synthesis while extending the class of possible uncertain semantics. We demonstrate that the proposed framework scales well with GPU-acceleration, and present realistic applications of uncertain semantics in robotics that involve target tracking and the use of occupancy grids.**

I. Introduction

Temporal logic is a promising tool for robotics applications and explainable AI in that it can be used to represent rich, complex task objectives in the form of human-readable logical specifications. Robot systems equipped with the capability to perform STL synthesis can implement powerful, intuitive command interfaces using STL. We are interested in developing STL synthesis for practical, real-world applications by viewing the synthesis problem as a form of probabilistic inference.

More widespread adoption of formal methods in robot systems, in our view, is limited by two main factors. First, the computational requirements of existing synthesis methods are seen as prohibitive. Second, achieving computational efficiency is seen to compromise semantic expressivity.

The approach we adopt in this paper is to address challenges in both computational efficiency and expressivity by viewing STL synthesis as probabilistic inference. This probabilistic approach fits naturally with robotics perception-action pipelines for important information gathering tasks such as search, target tracking, and mapping.

We first present a probabilistic extension of STL, called random STL (RSTL), that is designed to support synthesis of robot trajectories that satisfy specifications defined over uncertain events in the environment. Formally, RSTL extends STL to include *uncertain* semantic labels. The gain in expressivity is the capacity to specify tasks in a way that facilitates robust behaviour in real-world settings; uncertainty that is inherent to practical environments can be anticipated explicitly instead of handled reactively. Synthesising trajectories is a differentiable problem that largely resembles previously proposed differentiable measures of robustness for deterministic STL synthesis.

We then present three approximate methods for computing the probability of satisfaction of RSTL formulae. Incorporating these methods, we implement synthesis using GPU-accelerated gradient ascent and output the most probable sequence of control actions to satisfy the given specification.

To evaluate our method, we report empirical results that illustrate correctness, convergence, and scalability properties. Further, our method inherits the benefits of gradient-based MPC, including the anytime property and predictable computation time per iteration.

To demonstrate practical use in common robotics scenarios, we provide case studies involving target search and occupancy grids. These examples show that desirable behaviour naturally arises, such as prioritising targets whose location uncertainty is increasing over targets that are physically proximate. They also demonstrate the use of complex predicates such occupancy grids which are prevalent robotics applications. Computational efficiency is shown to parallel recent advances in approximate synthesis for deterministic STL and, importantly, can be further improved through additional GPU hardware to enable development of highly capable robot systems.

We view the main contribution of this work as a step towards the feasibility of temporal logic for robotics in practice through the introduction of a new method for synthesis as probabilistic inference that can accommodate powerful task specifications and that has useful performance characteristics. Our work also provides the basis for interesting theoretical extensions that would allow temporal logic specifications to be integrated with estimation methods and belief-based planning.

II. Related Work

To model tasks in uncertain environments, one approach is to model the robots' dynamics as a discrete Markov decision process (MDP), where each state is assigned semantic labels with corresponding uncertainty [1–4]. A natural task specification tool is linear temporal logic (LTL), given which a product MDP [1] is constructed from an automaton and a sequence of actions are found that maximize the probability of satisfaction. These ideas can be extended to the continuous case by judiciously partitioning the environment [5–7]. However, we find that partitioning is computationally prohibitive for online operations in uncertain environments, because any change in belief about the environment would lead to invalidation and expensive recomputation. Moreover, the construction of a product MDP is a computationally expensive operation, and improving its scalability remains an open problem.

This research is supported by an Australian Government Research Training Program (RTP) Scholarship and the University of Technology Sydney.

Authors are with the University of Technology Sydney, Ultimo, NSW 2006, Australia brian.lee@student.uts.edu.au, {chanyeol.yoo, rfitch}@uts.edu.au

Signal temporal logic (STL) [8] is defined over continuous signals. Unlike in LTL, satisfaction is determined using continuous-valued *robustness* [9]. Existing work uses STL to specify a task defined over a set of deterministic classes of conditions on the environment uncertainty, such as chance constraints or variance limits [10–12]. However, since the robustness evaluation is not differentiable, a common approach is to synthesise solutions using *mixed integer linear programming* (MILP) which scales exponentially with the size of mission horizon [10, 13]. To address the inherent complexity, the robustness metric is approximated to smooth the search space and find a solution using gradient ascent.

III. Problem Formulation

Suppose we have a robot with N-dimensional state $\mathbf{x}_t \in \mathbb{R}^N$ and control actions $\mathbf{u}_t \in \mathbb{U}$, where $\mathbb{U}$ is a continuous set of admissible control actions. The robot's dynamics is uncertain, and is modelled by a discrete-time, continuous-space MDP $\mathcal{P}(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t)$ between t and $t+1$, so that the trajectory distribution over a horizon T is given by:

$$\mathcal{P}(\mathbf{X} \mid \mathbf{U}) = \mathcal{P}(\mathbf{x}_1) \prod_{t=1}^{T} \mathcal{P}(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t), \tag{1}$$

where $\mathbf{X} \equiv \mathbf{x}_1 ... \mathbf{x}_T$ and $\mathbf{U} \equiv \mathbf{u}_1 ... \mathbf{u}_T$.

The robot encounters a finite set of random events $\mathcal{E} = \{E^1, ..., E^M\}$ (e.g., 'object detected'), whose probability of occurrence depends on robot's state $\mathbf{x}_t$ and time t. We are interested in finding control actions $\mathbf{U}^*$ that maximises the probability of satisfying a task Φ defined over $\mathcal{E}$ (e.g. 'detect all objects'). Namely, this is a synthesis problem:

Problem 1 (Synthesis)**.** *Given the uncertain dynamics* (1)*, and a task specification Φ over a set of random events $\mathcal{E}$ with probability of satisfaction $\mathcal{P}((\mathbf{X}, t) \models \Phi)$, find an optimal sequence of controls $\mathbf{U}^*$ such that the trajectory $\mathbf{X}$ maximises the probability of satisfying Φ over time horizon T:*

$$\mathbf{U}^* = \underset{\mathbf{U} \in \mathbb{U}^T}{\operatorname{argmax}} \; \mathcal{P}\left((\mathbf{X}, t) \models \Phi\right), \tag{2}$$

with respect to time $t = 1$.

IV. STL Synthesis as Probabilistic Inference

In this section, we first introduce *random signal temporal logic* (RSTL), a probabilistic extension of STL that allows specification of tasks Φ over random events $\mathcal{E}$. We then present a probabilistic inference formulation of Problem 1.

A. Random STL Formulae

We model the random events $\mathcal{E} = \{E^1, ..., E^M\}$ as (not necessarily independent) Bernoulli random variables that are dependent on robot's state and time, with conditional probability of occurrence:

$$\mathcal{P}(E^i = 1 \mid \mathbf{x}_t, t) = \mathcal{P}^i(\mathbf{x}_t, t). \tag{3}$$

In other words, each E^i is a Bernoulli *random field* over $\mathbb{R}^N \times \mathbb{R}^+$. Given a set of random events $\mathcal{E}$, the syntax of an RSTL formula Φ is given by:

$$\Phi := E \mid \neg\Phi \mid \Phi \wedge \Psi \mid \Phi \mathcal{U}_{[t_1,t_2]} \Psi, \tag{4}$$

where $E \in \mathcal{E}$, and Ψ, Φ are RSTL formulae. $\neg$ is logical negation, $\wedge$ is logical conjunction. $\mathcal{U}$ is the temporal operator 'Until', and $\Phi \mathcal{U}_{[t_1,t_2]} \Psi$ means Φ must hold true between time $[t_1, t_2]$ until Ψ. Other operators such as $\vee$ (disjunction), $\mathcal{F}_{[t_1,t_2]}$ ('in **F**uture', i.e., eventually) and $\mathcal{G}_{[t_1,t_2]}$ ('**G**lobally', i.e., always) can be derived from the syntax the same way as deterministic STL [8]. The events E will be referred to as 'event predicates'.

RSTL is *random* in the sense that, for a given trajectory $\mathbf{X}$, the satisfaction of an RSTL formula Φ is a Bernoulli random event. The probability of satisfaction is the *expected* rate of satisfaction computed over sampled instances of the event predicates $\hat{\mathcal{E}} = \{e_1, ..., e_M\} \sim \mathcal{E}$:

$$\mathcal{P}((\mathbf{X}, t) \models \Phi) = \mathop{\mathbb{E}}_{\hat{\mathcal{E}} \sim \mathcal{E}} \mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi), \tag{5}$$

where $\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi)$ is a deterministic function evaluated recursively as:

$$\begin{aligned}
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, E^i) &\equiv e^i(\mathbf{x}_t, t) \sim \mathcal{P}^i(\mathbf{x}_t, t) \\
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \neg\Phi) &\equiv \neg\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi) \\
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi \wedge \Psi) &\equiv \mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi) \wedge \mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Psi) \\
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi \mathcal{U}_{[t_1,t_2]} \Psi) &\equiv \\
\bigvee_{\tau_1 \in t + [t_1, t_2]} \; \bigwedge_{\tau_2 \in t_1 + [0, \tau_1]} & \mathbf{Sat}(\mathbf{X}, \tau_2, \hat{\mathcal{E}}, \Phi) \vee \mathbf{Sat}(\mathbf{X}, \tau_1, \hat{\mathcal{E}}, \Psi).
\end{aligned} \tag{6}$$

Note that $e^i(\mathbf{x}_t, t) \in \{0, 1\}$ is a sample from E^i at robot state $\mathbf{x}_t$ and time t.

B. STL Synthesis as Inference

Since both task satisfaction and robot dynamics are probabilistic, it is natural to ask if Problem 1 can be solved solely within the realm of probability theory. This is achieved by the control-as-inference paradigm [14, 15], which has been shown to not only encompass existing optimal control problems, but also to enable new approaches. We follow a similar development and present an inference formulation of Problem 1.

In this formulation, the problem is modelled by the joint distribution among task satisfaction, robot trajectory, and control actions:

$$\mathcal{P}(\Phi_t, \mathbf{X}, \mathbf{U}) = \mathcal{P}(\Phi_t \mid \mathbf{X}) \mathcal{P}(\mathbf{X} \mid \mathbf{U}) \mathcal{P}(\mathbf{U}), \tag{7}$$

where $\mathcal{P}(\Phi_t \mid \mathbf{X}) \equiv \mathcal{P}((\mathbf{X}, t) \models \Phi)$ denotes the probability satisfaction of Φ given $\mathbf{X}$ with respect to time t.

Here, $\mathcal{P}(\mathbf{U})$ is our prior belief on what the control actions should be, and is representative of the admissible control space $\mathbb{U}$ in the synthesis formulation. For example, if $\mathcal{P}(\mathbf{U})$ is a zero-mean Gaussian prior, it is equivalent to penalising quadratic control cost. The prior can derive from other knowledge, e.g., an imitation-learnt prior as [16] does for optimal control.

A balance between admissibility and probability of satisfaction is captured by the *posterior* probability of control actions given that the task is satisfied:

$$\begin{aligned}\mathcal{P}(\mathbf{U} \mid \Phi_t) &\propto \mathcal{P}(\Phi_t \mid \mathbf{U})\mathcal{P}(\mathbf{U}) \\ &= \mathcal{P}(\mathbf{U}) \int \mathcal{P}(\Phi_t \mid \mathbf{X})\mathcal{P}(\mathbf{X} \mid \mathbf{U}) d\mathbf{X},\end{aligned} \tag{8}$$

We thus pose Problem 1 as a maximum a posteriori (MAP) inference problem:

Problem 2 (Inference)**.** *Given the dynamic model* (1) *and an RSTL task specification* Φ*, find the MAP control actions* $\mathbf{U}^*$ *given the robot's trajectory satisfies* Φ *with respect to* t*:*

$$\mathbf{U}^* = \underset{\mathbf{U}}{\operatorname{argmax}}\, \mathcal{P}(\mathbf{U} \mid \Phi_t). \tag{9}$$

V. Approximate Gradient Ascent

In this section, we first present approximate methods that allow analytical evaluation of (5). We then present a gradient-ascent scheme on these approximate evaluations.

A. Conditional Independence Approximation

To compute (5) analytically, we observe that (5) applies logical operations to samples from Bernoulli random variables. A convenient approximation is the product relation for independent Bernoulli random variables A and B:

$$\mathcal{P}(A \wedge B) = \mathcal{P}(A)\mathcal{P}(B). \tag{10}$$

Technically, the product relation holds true if the operands are *conditionally independent* given $\mathbf{X}$. The conditionally independent (CI)-approximation is defined by one of the authors [17] as follows:

Definition 1 (CI-approximation [17])**.** *Given an RSTL formula* Φ*, the CI-approximation of* $\mathcal{P}(\Phi_t \mid x)$ *is defined by:*

$$\begin{aligned}\mathcal{P}(E_t^i \mid \mathbf{X}) &\equiv \mathcal{P}^i(\mathbf{X}, t) \\ \mathcal{P}(\neg\Phi_t \mid \mathbf{X}) &\equiv 1 - \mathcal{P}(\Phi_t \mid \mathbf{X}) \\ \mathcal{P}(\bigwedge_i \Phi_t^i \mid \mathbf{X}) &\equiv \prod_i \mathcal{P}(\Phi_t^i \mid \mathbf{X}) \\ \mathcal{P}((\mathcal{G}_{[t_1,t_2]}\Phi)_t \mid \mathbf{X}) &\equiv \prod_{\tau \in t+[t_1,t_2]} \mathcal{P}(\Phi_\tau \mid \mathbf{X})\end{aligned} \tag{11}$$

B. Log-odds Transform

The output range for CI-approximation of $\mathcal{P}$ (11) is $[0, 1]$ since it computes probability. This can lead to numerical instability, and gradient ascent often leads to poor convergence. A natural re-parameterisation for Bernoulli random variables is the *log-odds*:

$$\mathcal{L}(A) = \log \frac{\mathcal{P}(A)}{\mathcal{P}(\neg A)}, \tag{12}$$

It can be shown with some algebraic manipulations that re-writing the CI rule for pairwise disjunction $\vee$ in (11) in terms of log-odds leads to:

$$\begin{aligned}\mathcal{L}(A \vee B) &= \log \frac{\mathcal{P}(A)\mathcal{P}(B) + \mathcal{P}(\neg A)\mathcal{P}(B) + \mathcal{P}(A)\mathcal{P}(\neg B)}{\mathcal{P}(\neg A)\mathcal{P}(\neg B)} \\ &= \mathfrak{lse}(\mathcal{L}(A), \mathcal{L}(B), \mathcal{L}(A) + \mathcal{L}(B)),\end{aligned} \tag{13}$$

where A and B are independent Bernoulli random variables, and $\mathfrak{lse}$ is the *log-sum-exp* function:

$$\mathfrak{lse}(\mathcal{L}_1, ..., \mathcal{L}_N) = \log \sum_i \exp \mathcal{L}_i. \tag{14}$$

A series of disjunction operations (13) is then:

$$\mathcal{L}\left(\bigvee_{i \in I} A_i\right) = \log \sum_{J \in 2^I} \exp \sum_{j \in J} \mathcal{L}(A_j), \tag{15}$$

where 2^I denotes the power set of I.

Computing log-sum-exp over sum of all subsets is clearly cumbersome. We avoid such computation by using the relationship between elementary symmetric polynomials and monic polynomials. Observe that the summations over $j \in J$ can be taken out of the exponential as products. Then, we have all elementary symmetric polynomials over A_i less 1. We thus arrive at a more compact expression:

$$\mathcal{L}\left(\bigvee_i A_i\right) = \log \left(\prod_i \big(1 + \exp \mathcal{L}(A_i)\big) - 1\right). \tag{16}$$

Now, the CI computation rules in the log-odds domain are given as follows:

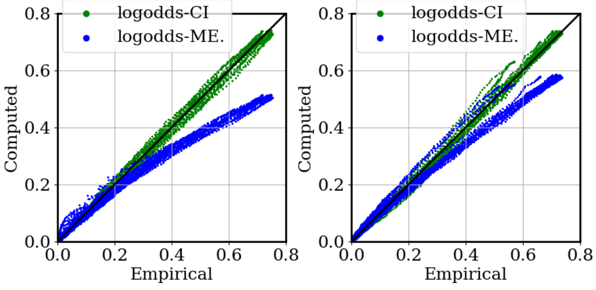
Definition 2 (CI-approximate log-odds)**.** *Given an RSTL formula* Φ*, the CI-approximation of log-odds of satisfaction* $\mathcal{L}(\Phi_t \mid \mathbf{X})$ *is calculated by:*

$$\begin{aligned}\mathcal{L}(E_t^i \mid \mathbf{X}) &\equiv \log \mathcal{P}^i(\mathbf{X}, t) - \log(1 - \mathcal{P}^i(\mathbf{X}, t)) \\ \mathcal{L}(\neg\Phi_t \mid \mathbf{X}) &\equiv -\mathcal{L}(\Phi_t \mid \mathbf{X}) \\ \mathcal{L}\left(\bigvee_i \Phi_t^i \mid \mathbf{X}\right) &\equiv \log \left(\prod_i (1 + \exp \mathcal{L}(\Phi_t^i \mid \mathbf{X})) - 1)\right) \\ \mathcal{L}((\mathcal{F}_{[t_1,t_2]}\Phi)_t \mid \mathbf{X}) &\equiv \log \Big(\prod_{\tau \in t+[t_1,t_2]} \big(1 + \exp \mathcal{L}(\Phi_\tau \mid \mathbf{X}) - 1\Big).\end{aligned} \tag{17}$$

It is interesting to note that the proposed computation rules for probability of satisfaction exhibits strong similarities to existing work on deterministic STL synthesis and model checking [8, 18–20]. In the log-odds domain, certain satisfaction (i.e. probability of 1) translates to ∞, certain dissatisfaction is $-\infty$, and absolute uncertainty (i.e. probability of 0.5) is 0, which are the behaviours of spatial robustness measure for deterministic STL introduced in [8]. Further, the log-sum-exp function has been used in deterministic STL synthesis [18, 19] as a smooth approximation of the maximum function. Finally, A similar expression to (16) was presented in [20] as an alternative robustness measure for deterministic STL. The authors report encouragement of repeated satisfaction, which is consistent with the probability of disjunction increasing with increasing probability of the disjuncts.

Given that the approaches that do not encourage repeated satisfaction [18, 19] still report acceptable results, we consider the following approximation for disjunction:

$$\begin{aligned}\mathcal{L}(A \vee B) &\approx \mathfrak{lse}(\mathcal{L}(A), \mathcal{L}(B))) \\ &= \log \frac{\mathcal{P}(A)\mathcal{P}(\neg B) + \mathcal{P}(\neg A)\mathcal{P}(B)}{\mathcal{P}(\neg A)\mathcal{P}(\neg B)}.\end{aligned} \tag{18}$$



(a) $\mathcal{F}(A) \wedge \mathcal{F}(B)$ (b) $\mathcal{F}(A \wedge \mathcal{F}(B))$

Fig. 1. Comparison of the CI (green) and ME (blue) approximation results against MC estimates ('Empirical'). CI is exact for $\mathcal{F}(A) \wedge \mathcal{F}(B)$, while ME underestimates. For $\mathcal{F}(A \wedge \mathcal{F}(B))$, the error increases, but not significantly. Naive method's result showed numerically insignificant difference to CI, and is omitted.

This ignores repeated satisfaction by omitting the $\mathcal{P}(A)\mathcal{P}(B)$ term from the numerator of (13). Meanwhile, there is a potential numerical benefit that log-sum-exp can be computed numerically stably with the so-called *log-sum-exp trick*, while the product in (16) may underflow. We thus define the *mutually exclusive* (ME) approximation as follows.

Definition 3 (ME approximation)**.** *Given an RSTL formula Φ, the ME approximation of $\mathcal{L}(\Phi_t \mid \mathbf{X})$ is calculated by:*

$$\begin{aligned}\mathcal{L}(E_t^i \mid \mathbf{X}) &\equiv \log \mathcal{P}^i(\mathbf{X}, t) - \log(1 - \mathcal{P}^i(\mathbf{X}, t))\\ \mathcal{L}(\neg\Phi_t \mid \mathbf{X}) &\equiv -\mathcal{L}(\Phi_t \mid \mathbf{X})\\ \mathcal{L}(\Phi_t \vee \Psi_t \mid \mathbf{X}) &\equiv \mathfrak{lse}(\mathcal{L}(\Phi_t \mid x), \mathcal{L}(\Psi_t \mid \mathbf{X}))\\ \mathcal{L}((\mathcal{F}_I\Phi)_t \mid \mathbf{X}) &\equiv \log \sum_{\tau \in t+I} \exp \mathcal{L}(\Phi_\tau \mid \mathbf{X}).\end{aligned} \tag{19}$$

C. Synthesis with Gradient-Ascent

With the probability or log-odds of satisfaction computed, we synthesise a MAP control sequence $\mathbf{U}^*$ that maximises the posterior probability. We use Jensen's inequality to bound the log of posterior probability (9):

$$\log \mathcal{P}(\mathbf{U} \mid \Phi_t) \geq \mathop{\mathbb{E}}_{\mathbf{X} \sim \mathcal{P}(\mathbf{X}|\mathbf{U})} [\log \mathcal{P}(\Phi_t \mid \mathbf{X}))] + \log \mathcal{P}(\mathbf{U}). \tag{20}$$

Subsequently, we maximize the lower bound:

$$\mathbf{U}^* = \underset{\mathbf{U}}{\operatorname{argmax}} \mathop{\mathbb{E}}_{\mathbf{X} \sim \mathcal{P}(\mathbf{X}|\mathbf{U})} [\log \mathcal{P}(\Phi_t \mid \mathbf{U})] + \log \mathcal{P}(\mathbf{U}). \tag{21}$$

The maximisation is done by gradient ascent on (21). Because the expectation in (21) is intractable, we replace it with an empirical mean over a N_s number of trajectory samples, so that the i-th gradient ascent step is:

$$\hat{\mathbf{U}}^{i+1} = \hat{\mathbf{U}}^i + \frac{1}{N_s} \sum_j \frac{\partial}{\partial \hat{\mathbf{U}}^i} [\log \mathcal{P}(\Phi_t \mid \mathbf{X}_{1:T}^j(\hat{\mathbf{U}}^i)) + \log \mathcal{P}(\hat{\mathbf{U}}^i)], \tag{22}$$

where $\hat{\mathbf{U}}^{i+1} = \left[\hat{\mathbf{u}}_1^i ... \hat{\mathbf{u}}_T^i\right]$. Each trajectory sample $\mathbf{X}^j(\hat{\mathbf{U}}^i)$ is obtained by propagating the dynamic model (1) forward in time including actuation uncertainty. Note that, as long as the predicates' distributions and the dynamic model are differentiable, so is (22). For both CI and ME approximations, analytical gradients can be computed easily using autograd frameworks such as Tensorflow [21] or PyTorch [22, 23]. Therefore, we do not present the expressions here.

VI. Empirical Analysis

We evaluate the practical performance characteristics of the proposed method. We first demonstrate that the proposed CI (17) and ME (19) methods reasonably approximate the ground truth (5). We then examine the convergence characteristics of the gradient ascent (22) solution, and demonstrate the computational benefits of GPU acceleration.

A. Quality of Approximation

In this section, we evaluate whether the CI and ME approximations compute the probability of satisfaction accurately. Because the CI computation rule assumes independence amongst operands, we expect it to be exact if 1) all predicates are conditionally independent; 2) each predicate is independent across time and space; and 3) only one operator uses each predicate. For example, assuming 1) and 2) hold, we expect the CI rule to be exact on $\mathcal{F}A \wedge \mathcal{F}B$, but not $\mathcal{F}(A \wedge \mathcal{F}B)$, because the disjuncts of the outer $\mathcal{F}$ operator are not independent. The ME computation rules (19) will not be exact in any case.

We validate these hypotheses by comparing against a 1000-sample Monte Carlo (MC) approximation of RSTL probability of satisfaction (5). We used the trajectories from the first 2000 gradient ascent steps generated from the target search scenario (Fig. 4). For simplicity, we evaluated each predicates' marginal probability independently before sampling, so that the first two conditions of exactness hold.

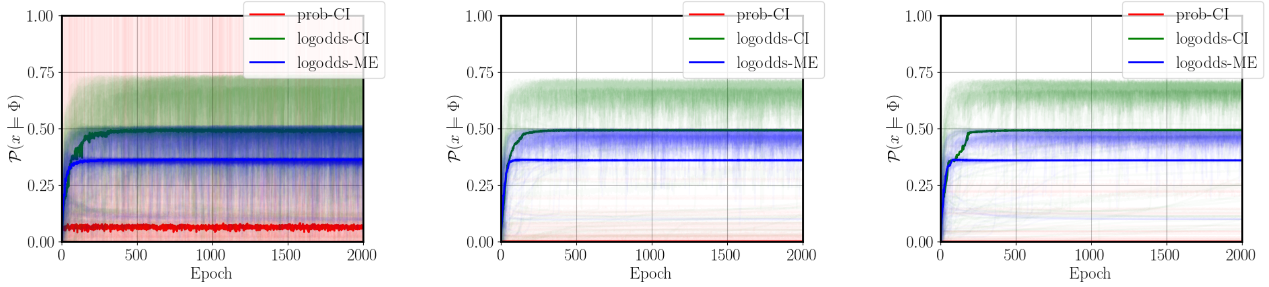
In Fig. 1a, CI (green) and ME (blue) results are compared against the MC estimate for $\mathcal{F}A \wedge \mathcal{F}B$. It can be seen that the CI method matches the MC result as expected, while ME consistently underestimates. This is expected, because ME does not account for multiple satisfaction.

Figure 1b shows comparison for $\mathcal{F}(A \wedge \mathcal{F}B)$. As $\mathcal{F}B$ is double-counted by the outer $\mathcal{F}$, CI and ME tend to overestimate, but not by much. ME continues to underestimate, and CI matches the MC closely, showing that CI and ME are reasonable approximations for practical applications.

B. Convergence

Gradient-based methods cannot guarantee globally optimal solutions unless the objective is convex. We analyse the convergence characteristics of the proposed computation rules in the target search scenario (Fig. 4). We randomly generated 100 initial conditions from the control prior $\mathcal{P}(\mathbf{U})$, and ran the gradient ascent step for 2000 iterations, with $N_s = 1, 50, 100$ number of trajectory samples.

Figure 2 shows the probability of satisfaction over gradient ascent steps for naive CI ((11), red), log-odds CI ((17), green) and log-odds ME ((19), blue) methods with varying number of trajectory samples N_s. It can be seen that while log-odds CI and ME methods find global and local optima, the naive CI method does not find any. The naive CI method's failure is attributed to the computation rules (11) being bounded to



(a) $N_s = 1$ (b) $N_s = 50$ (c) $N_s = 100$

Fig. 2. Comparison of convergence with number of trajectory samples. Solid lines are medians.

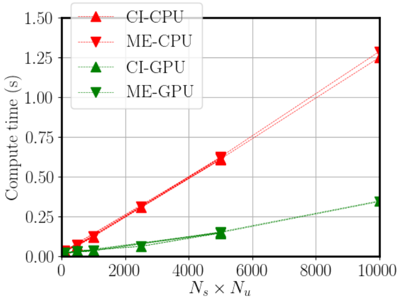


Fig. 3. Comparison of average computation time per gradient ascent step between combinations of CPU (red) and GPU (green) with CI (upward triangle and ME (downward triangle). GPU shows 4-fold improvement in scalability. Variance was in the order of 10^{-4} for all configurations.

$[0, 1]$ range, which causes numerical errors to build up. Note that log-odds ME reports lower probability of satisfaction due to its underestimation property, but we found that the resulting trajectories were still similar.

With increasing N_s, the variance in probability of satisfaction is decreased. In practice, this means the generated plan will more reliably account for state uncertainty, which is useful for, e.g., the collision avoidance scenario in Fig. 5.

C. Computation Time

The results in Sec. VI-B illustrate that it is important to use multiple initial conditions and more trajectory samples to ensure reliable operation. However, this would inevitably increase the computation time as well.

GPU acceleration is a prominent means to circumvent the issue of computation time, but not all algorithms benefit from GPU acceleration. To determine if our proposed methods benefit from GPU acceleration, we compare the computation time per gradient ascent step of our Tensorflow [21] implementation between GPU and CPU. We used all combinations between $N_s = 1, 10, 50, 100$ and $N_u = 1, 10, 50, 100$, and computed the mean over 1000 gradient ascent steps. We used a desktop with CPU (Intel i5-9500) and a GPU (NVIDIA RTX2060) to conduct the experiment.

Figure 3 shows the computation time with varying number of initial conditions N_u and the number of state samples N_s. We found that the total number of samples $N_u \times N_s$ explains all changes. The result shows that the computation time with GPU is significantly lower than that with CPU, and that using a GPU leads to 4-fold improvement in scalability. This demonstrates that the proposed gradient ascent method benefits from GPU acceleration.

VII. Case Studies

We demonstrate two example cases that illustrate the benefit of using RSTL for task specification in uncertain environments. The proposed gradient ascent method was implemented in Tensorflow [21]. For all examples, we consider a robot described by a bicycle dynamic model:

$$\dot{\mathbf{x}}_t = \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} V_t \cos\theta_t \\ V_t \sin\theta_t \\ \omega_t + \epsilon_t \end{bmatrix}, \tag{23}$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_u)$ is white Gaussian noise. The control inputs are $\mathbf{u}_t = \begin{bmatrix} V_t & \omega_t \end{bmatrix}^\top$.

A. 2D Target Search

We consider a 2D target search scenario, where a robot is tasked with detecting possibly moving targets in the environment: *Tom* and *Jerry*. Jerry, as usual, is moving with increasing uncertainty, while Tom is stationary with high certainty. Figure 4 depicts an example where the mean paths for Tom and Jerry are shown in green and red. The growing uncertainty over time is shown around the mean. Note that since Tom (in green) is known to be stationary, its uncertainty does not grow over time. The robot starts at $[0, 0]^\top$.

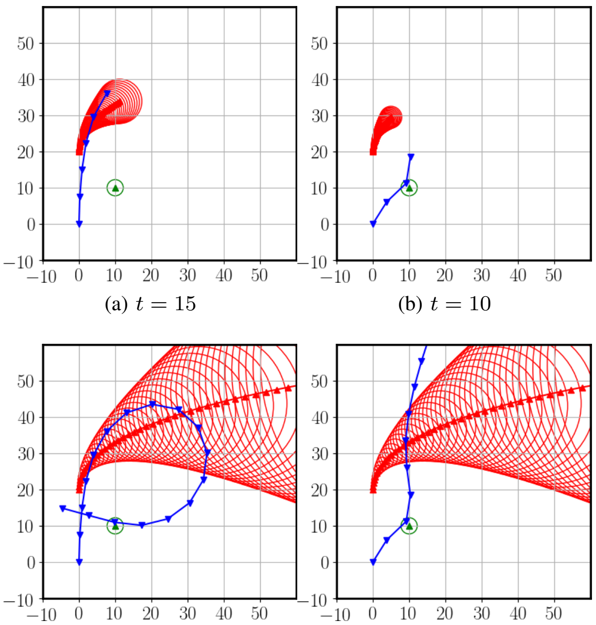
The task of finding Tom and Jerry can be expressed using RSTL as $\Phi = \mathcal{F}(D^{\text{Tom}}) \wedge \mathcal{F}(D^{\text{Jerry}})$ (i.e., 'eventually detect Tom and eventually detect Jerry').

We model the events D^{Tom} and D^{Jerry} as follows. If the location is known, the robot detects Tom and Jerry with likelihood modelled by:

$$\mathcal{P}(D_t \mid \mathbf{x}_t, \mathbf{z}_t) = P_D \exp\left(\frac{||\mathbf{x}_t - \mathbf{z}_t||^2}{2r_D^2}\right), \tag{24}$$

where $\mathbf{z}_t$ is the location of the target, r_D is the radius of detection, and P_D controls the peak.

Tom and Jerry are modelled by a constant acceleration model, which is a linear Gaussian system. The mean $\bar{\mathbf{z}}_t^{a,b}$ and $\Sigma_t^{a,b}$ are propagated given the robots' belief at $t = 0$ using



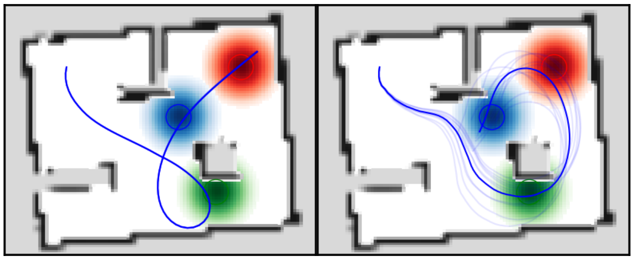
(a) $t = 15$

(b) $t = 10$

(c) $t = 45$

(d) $t = 45$

Fig. 4. A 2D target search scenario. The robot (blue) is tasked with detecting both Tom (Green) and Jerry (Red). The global optimum with $\mathcal{P}(\Phi \mid \mathbf{x}) \approx 0.7$ (left column) is to detect Jerry before uncertainty grows. A local optimum with $\mathcal{P}(\Phi \mid \mathbf{x}) \approx 0.5$ (right column) prefers Tom, who is closer. Circles show 1-covariance bound.



(a) $\sigma_u = 10^{-4}$

(b) $\sigma_u = 0.1$

Fig. 5. Results for complex indoor mission. The robot's (blue) task is a conjunction of 'never visit Rob (red) or Bob (blue) before visiting sanitising station (green), and avoid obstacles (grey colormap)', and 'visit Rob (red) and Bob (blue)'. With higher actuation uncertainty σ_u, the trajectory becomes further from the walls. Solid lines are the robot's nominal trajectory in the absence of noise. Transparent blue lines are the noised samples used during synthesis. The start location is top-left corner.

the standard prediction equations. The marginal probability of detection accounting for their uncertainty is given by:

$$\begin{aligned}\mathcal{P}(D_t \mid \mathbf{x}_t) &= \int \exp\left(\frac{||\mathbf{x}_t - \bar{\mathbf{z}}_t||^2}{2r_d^2}\right)\mathcal{N}(\bar{\mathbf{z}}_t, \Sigma_t^z)d\mathbf{z}_t \\ &= \sqrt{2\pi}^N r_d^N \mathcal{N}(\bar{\mathbf{z}}_t, \Sigma_t^z + r_d^2 I),\end{aligned} \qquad (25)$$

where $\mathcal{N}(\bar{\mathbf{z}}_t, \Sigma_t^z)$ is the multivariate Gaussian PDF.

Figure 4 shows global and local optima found using the log-odds CI computation rule (17). It can be seen that the global optimum (Figs. 4a and 4c) is to detect Jerry first at $t = 15$ (Fig. 4a), and to return to Tom at $t = 45$ (Fig. 4c), while the local optimum is to detect Tom first at $t = 10$ (Fig. 4b) and then Jerry later. This is because the uncertainty of Jerry grows unlike Tom, and the optimal trajectory should detect Jerry first before its uncertainty grows. This demonstrates that RSTL naturally reasons over uncertainty.

B. Complex Missions in an Indoor Environment

Consider a nursing robot in an indoor environment, modelled as an occupancy grid O such that $O_{i,j}$ is the probability of obstacle occupancy. Collision with an obstacle is modelled by an interpolation:

$$\mathcal{P}(O \mid \mathbf{x}_t) = \sum_{i,j} K_{ij}(\mathbf{x}_t)O_{i,j}, \qquad (26)$$

where $K_{ij}(\mathbf{x})$ denotes the interpolant.

The robot cares for two patients, Rob and Bob. The doctor asks the robot to avoid obstacles, and to never visit any of the patients before visiting the sanitising station, which can be written as an RSTL formula:

$$\Phi_1 = (\neg(D^{Rob} \vee D^{Bob})\mathcal{U}D^{San}) \wedge \mathcal{G}(\neg O), \qquad (27)$$

where D^{Rob}, D^{Bob}, and D^{San} are distributed as per (25).

Now, in addition to the previous command, the doctor asks the robot to visit the two patients:

$$\Phi_2 = \mathcal{F}(D^{Rob}) \wedge \mathcal{F}(D^{Bob}) \wedge \Phi_1. \qquad (28)$$

We created an occupancy grid from a realistic dataset commonly used in perception research [24, 25], and compared the results with low (10^-4rads^{-1}) and high $(\sigma_u = 0.1\text{rad}s^{-1})$ actuation uncertainty. The results are shown in Fig. 5. In both cases, the generated trajectory is correct, visiting the sanitising station first, and then the two patients. Interestingly, the trajectory changes drastically when control noise increases. The path with small control noise in Fig. 5a is aggressively close to the wall, whereas the path with control noise in Fig. 5b is more conservative in that the robot keeps distance away from the wall by manoeuvring around the obstacle. This demonstrates that the proposed probabilistic formulation enables risk-averse behaviour in STL synthesis, a crucial property for practical applications.

VIII. Conclusion

We have presented a probabilistic inference perspective on STL synthesis based on RSTL and corresponding algorithms. Our method exhibits appealing computational and expressivity characteristics that suit practical robotics applications. We anticipate that the inference formulation of STL synthesis presented in this paper will accelerate the development of explainable AI techniques through seamless integration between formal methods and machine learning techniques as did the optimal control-as-inference paradigm [14, 15]. Many avenues of future work arise from our results; one of the most exciting is integration with estimation methods [26–28] that would allow multi-robot systems to augment or replace explicit communication for behaviour coordination with trajectory predictions derived from specifications [29, 30].

References

[1] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of Markov decision processes with linear temporal logic constraints," *IEEE Trans. Automat. Contr.*, vol. 59, no. 5, pp. 1244–1257, 2014.

[2] S. Bharadwaj, M. Ahmadi, T. Tanaka, and U. Topcu, "Transfer entropy in MDPs with temporal logic specifications," in *Proc. of IEEE CDC*, 2018, pp. 4173–4180.

[3] C. Yoo, R. Fitch, and S. Sukkarieh, "Probabilistic temporal logic for motion planning with resource threshold constraints," *Proc. of RSS*, 2012.

[4] ——, "Provably-correct stochastic motion planning with safety constraints," in *Proc. of IEEE ICRA*, 2013, pp. 981–986.

[5] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray, "TuLiP: A software toolbox for receding horizon temporal logic planning," in *Proc. of HSCC*, 2011, pp. 313–314.

[6] C. Yoo, R. Fitch, and S. Sukkarieh, "Online task planning and control for fuel-constrained aerial robots in wind fields," *Int. J. of Robot. Res.*, vol. 35, no. 5, pp. 438–453, 2016.

[7] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Hierarchical planning in time-dependent flow fields for marine robots," in *Proc. of IEEE ICRA*, 2020, pp. 885–891.

[8] A. Donzé, "On signal temporal logic," in *Runtime Verification*, A. Legay and S. Bensalem, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 382–383.

[9] 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 Syst. Des.*, pp. 5–30, 2017.

[10] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Proc. of RSS*, AnnArbor, Michigan, June 2016.

[11] M. Tiger and F. Heintz, "Incremental reasoning in probabilistic signal temporal logic," *Int. J. Approx. Reason.*, vol. 119, pp. 325–352, April 2020.

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

[13] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *Proc. of IEEE CDC*, 2014, pp. 81–87.

[14] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Mach. learn.*, vol. 87, no. 2, pp. 159–182, 2012.

[15] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018.

[16] N. Rhinehart, R. McAllister, and S. Levine, "Deep imitative models for flexible inference, planning, and control," in *Proc. of ICLR*, April 2020.

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

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

[19] I. Haghighi, N. Mehdipour, E. Bartocci, and C. Belta, "Control from signal temporal logic specifications with smooth cumulative quantitative semantics," in *Proc. of IEEE CDC*, 2019, pp. 4361–4366.

[20] N. Mehdipour, C. Vasile, and C. Belta, "Arithmetic-geometric mean robustness for control from signal temporal logic specifications," in *Proc. of IEEE ACC*, 2019, pp. 1690–1695.

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[23] K. Leung, N. Aréchiga, and M. Pavone, "Back-propagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods," in *Algorithmic Foundation of Robotics XIV*, S. M. LaValle, M. Lin, T. Ojala, D. Shell, and J. Yu, Eds. Springer, 2021, vol. 17, pp. 432–449.

[24] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, "Online continuous mapping using Gaussian process implicit surfaces," in *Proc. of IEEE ICRA*, 2019, pp. 6884–6890.

[25] L. Wu, K. M. B. Lee, L. Liu, and T. Vidal-Calleja, "Faithful Euclidean distance field from log-Gaussian process implicit surfaces," *IEEE Robot. and Automat. Lett.*, vol. 6, no. 2, pp. 2461–2468, 2021.

[26] K. M. B. Lee, C. Yoo, B. Hollings, S. Anstee, S. Huang, and R. Fitch, "Online estimation of ocean current from sparse GPS data for underwater vehicles," in *Proc. of IEEE ICRA*, 2019, pp. 3443–3449.

[27] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *Proc. of IROS*, 2015, pp. 5817–5823.

[28] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *Proc. of Int. Conf. on Comput. Vision*, October 2019.

[29] K. M. B. Lee, F. Kong, R. Cannizzaro, J. L. Palmer, D. Johnson, C. Yoo, and R. Fitch, "An upper confidence bound for simultaneous exploration and exploitation in heterogeneous multi-robot systems," in *Proc. of IEEE ICRA*, 2019.

[30] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *The Int. J. of Robot. Res.*, vol. 38, no. 2-3, pp. 316–337, 2019.