Safe Learning for Uncertainty-Aware Planning via Interval MDP Abstraction

Jesse Jiang, *Student Member, IEEE,* Ye Zhao, *Member, IEEE,* and Samuel Coogan, *Senior Member, IEEE*

***Abstract*—We study the problem of refining satisfiability bounds for partially-known stochastic systems against planning specifications defined using syntactically co-safe Linear Temporal Logic (scLTL). We propose an abstraction-based approach that iteratively generates high-confidence Interval Markov Decision Process (IMDP) abstractions of the system from high-confidence bounds on the unknown component of the dynamics obtained via Gaussian process regression. In particular, we develop a synthesis strategy to sample the unknown dynamics by finding paths which avoid specification-violating states using a product IMDP. We further provide a heuristic to choose among various candidate paths to maximize the information gain. Finally, we propose an iterative algorithm to synthesize a satisfying control policy for the product IMDP system. We demonstrate our work with a case study on mobile robot navigation.**

***Index Terms*—Automata, Hybrid systems, Markov processes, Gaussian process learning**

I. Introduction

ABSTRACTION-based approaches for verification and synthesis of dynamical systems offer computationally tractable methods for accommodating complex specifications [1]. In particular, Interval Markov Decision Processes (IMDP) [2], which allow for an interval of transition probabilities, provide a rich abstraction model for stochastic systems. As compared to stochastic control [3], abstraction-based methods allow for more complex specifications to be considered and have been widely used for hybrid stochastic systems [4].

The transition probability intervals in IMDP abstractions have typically modeled the uncertainty which arises from abstracting the dynamics of continuous states in discrete regions [5]. However, partially-known stochastic systems, which show promise for modeling a wide range of real-world systems, add unknown dynamics which contribute further uncertainty. Previous works model this uncertainty by assuming that some prior data on the dynamics are available [6].

The paper [7] is the first to address the problem of modeling unknown dynamics in stochastic hybrid systems via the use of IMDP abstraction in combination with *Gaussian process* (GP) regression [8]. GP regression can approximate unknown functions with arbitrary accuracy and also provides bounds on the approximation uncertainty [9].

This work was supported in part by the National Science Foundation under grant #1924978.

Jesse Jiang and Samuel Coogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: jjiang@gatech.edu, sam.coogan@gatech.edu). S. Coogan is also with the School of Civil and Environmental Engineering.

Ye Zhao is with the School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ye.zhao@me.gatech.edu).

The main contribution of this work is to develop a method for sampling the unknown dynamics of a stochastic system online in order to reduce abstraction error and increase the probability of satisfying a syntactically co-safe linear temporal logic (scLTL) specification [2].

Our goal is to find a control policy which guarantees the satisfaction of a scLTL specification with sufficient probability. However, we assume a stochastic noise which creates unavoidable perturbation. The system also has unknown dynamics which we estimate with Gaussian processes. This creates an estimation error which increases uncertainty in state transitions and which we aim to reduce by sampling the unknown dynamics. Thus, this paper focuses on the problem of safe learning to allow online exploration rather than a static planning problem using previously collected data samples as in [7].

Our approach is as follows. First, we estimate the unknown dynamics of the system using Gaussian processes and construct a high-confidence IMDP abstraction. We then develop an algorithm for finding *nonviolating cycles* in a product IMDP of the system abstraction combined with a finite automaton of the scLTL specification which allow the dynamics of the system to be sampled without violating the specification. We develop a heuristic for evaluating candidate cycles in order to maximize the uncertainty reduction gained from sampling. Finally, we propose an iterative method to sample the state-space, thereby decreasing the uncertainty of a GP estimation of the unknown dynamics until a satisfying control policy for the system can be synthesized or a terminating condition such as a maximum number of iterations has been reached. We utilize sparse GPs [10] to improve computational efficiency. We demonstrate our method on a case study of robotic motion planning.

II. Problem Setup

Consider a discrete-time, partially-known system

$$x[k+1] = f(x[k]) + u[k] + g(x[k]) + \nu[k] \tag{1}$$

where $x \in X \subseteq \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^n$ is the control action, $f(x)$ is the known dynamics, $g(x)$ is the unknown dynamics to be learned via GP regression, ν is stochastic noise, and time is indexed with brackets. This system has applications in, *e.g.*, biology [11], communications [12], and robotics [13].

Assumption 1: 1) Each dimension $\nu_i[k], i = 1, \ldots, n$ of ν, is an independent, zero mean random variable with stationary, symmetric, and unimodal distribution ρ_{ν_i} and is σ_{ν_i}-sub-Gaussian, *i.e.*, the distribution tail decays at least as fast as a Gaussian random variable with variance $\sigma^2_{\nu_i}$.

2) Given a data set $D = \{(z^j, y^j)\}_{j=1}^m$ where y^j is an observation of $g(z^j)$ perturbed by σ_{ν_i}-sub-Gaussian noise, it is possible to construct an estimate $\hat{g}^D(x)$ of g and bound the estimation error between $g(x)$ and $\hat{g}^D(x)$ by some high-confidence bound $\gamma^D(x)$. Thus,

$$g_-^D(x) = \hat{g}^D(x) - \gamma^D(x), \quad g_+^D(x) = \hat{g}^D(x) + \gamma^D(x) \quad (2)$$

are high-confidence bounds on g, *i.e.*, $g_-^D(x) \leq g(x) \leq g_+^D(x)$ with high confidence. For simplicity, we drop the superscript D when the dataset is clear.

Assumption 2: The state-space X is bounded and is partitioned into hyper-rectangular regions $\{X_q\}_{q\in Q}$ defined as

$$X_q = \{x \mid \ a_q \leq x \leq b_q\} \subset X, \quad (3)$$

where the inequality is taken elementwise for lower and upper bounds $a_q, b_q \in \mathbb{R}^n$ and Q is a finite index set of the regions. Each region has a center $c_q = (a_q + b_q)/2$. Additionally, the system possesses a labeling function L which maps hyper-rectangular regions to observations O.

Define feedback controllers $K_q(\cdot\,; \hat{g}) : X \to X$ as

$$K_q(x; \hat{g}) = c_q - f(x) - \hat{g}(x). \quad (4)$$

The choice $u[k] = K_{q'}(x[k]; \hat{g})$ thus produces a control action which compensates for the known and estimated dynamics to reach the center of region $X_{q'}$, although the actual state update will be perturbed as shown in Figure 1.

Our ultimate goal is to apply a sequence of feedback controllers so that the resulting sequence of observations satisfies a control objective specified as a syntactically co-safe LTL (scLTL) formula over the observations O.

Definition 1 (Syntactically co-safe LTL [2, Def. 2.3]): A *syntactically co-safe linear temporal logic (scLTL)* formula ϕ over a set of observations O is recursively defined as

$$\phi = \top \mid o \mid \neg o \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc \phi \mid \phi_1 \mathcal{U} \phi_2 \mid \Diamond \phi$$

where $o \in O$ is an observation and ϕ, ϕ_1, and ϕ_2 are scLTL formulas. We define the *next* operator $\bigcirc$ as meaning that ϕ will be satisfied in the next state transition, the *until* operator $\mathcal{U}$ as meaning that the system satisfies ϕ_1 until it satisfies ϕ_2, and the *eventually* operator $\Diamond$ as $\top \mathcal{U} \phi$.

ScLTL formulas are characterized by the property that they are satisfied in finite time. It is well-known that scLTL satisfaction can be checked using a finite state automaton:

Definition 2 (Finite State Automaton [2, Def. 2.4]): A *finite state automaton (FSA)* is a tuple $\mathcal{A} = (S, s_0, O, \delta, F)$, where

- S is a finite set of states,
- $s_0 \in S$ is the initial state,
- O is the input alphabet, which corresponds to observations from the scLTL specification,
- $\delta : S \times O \to S$ is a transition function, and
- $F \subseteq S$ is the set of accepting (final) states.

A sequence of inputs (a *word*) from O is said to be accepted by an FSA if it ends in an accepting state. A scLTL formula can always be translated into a FSA that accepts all and only those words satisfying the formula. We use scLTL specifications in this paper because they are well-suited to robotic motion planning tasks which are satisfied in finite time. Additionally, the simpler structure of an FSA as opposed to the Büchi and Rabin automata of general LTL enables the methods we propose below.

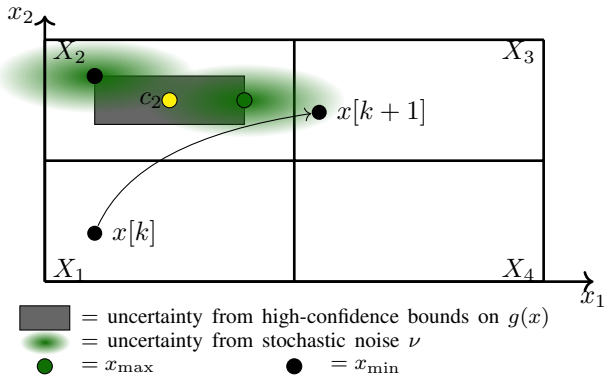

Fig. 1. Feedback controller and calculation of transition probabilities. The controller targets the center of state X_2. The uncertainty in $\hat{g}(x)$ creates a nondeterministic region of transition (brown rectangle). The maximum probability of transitioning to state X_3 is found by centering stochastic noise at the point $x_{\max}$ closest to state X_3 (green point) and calculating the probability that the noise reaches state X_3. The minimum probability of transitioning to state X_3 under this controller is given likewise by centering stochastic noise at the point $x_{\min}$ furthest from X_3 (red point).

Definition 3 (Interval Markov Decision Process): An *Interval Markov Decision Process (IMDP)* is a tuple $\mathcal{I} = (Q, A, \check{T}, \hat{T}, Q_0, O, L)$ where:

- Q is a finite set of states,
- A is a finite set of actions,
- $\check{T}, \hat{T} : Q \times A \times Q' \to [0, 1]$ are lower and upper bounds, respectively, on the transition probability from state $q \in Q$ to state $q' \in Q$ under action $\alpha \in A$,
- $Q_0 \subseteq Q$ is a set of initial states,
- O is a finite set of atomic propositions or observations,
- $L : Q \to O$ is a labeling function.

The set of actions A corresponds to the set of all valid feedback controllers for the system. We do not assume that all actions are available at each state. Therefore, each state has a subset $A(q) \subseteq A$ of available actions.

Definition 4 (High-Confidence IMDP Abstraction): Consider stochastic system (1), partitions (3), and the family of feedback controllers (4) where $\hat{g}(x)$ is an estimate of $g(x)$. Further, suppose that $g_-(x)$ and $g_+(x)$ are high-confidence bounds on $\hat{g}(x)$ which satisfy (2). Then, an IMDP $\mathcal{I} = (Q, A, \check{T}, \hat{T}, Q_0, O, L)$ is a *high-confidence IMDP abstraction of* (1), if:

- The set of states Q for the abstraction is the index set of partitions, *i.e.* partition X_q is abstracted as state q, and the set of observations O and labeling function L for the abstraction are the same as for the system,
- For all $q \in Q$, the set of actions $A(q)$ is the set of one-step reachable regions at q under its feedback controllers,
- For all $q \in Q$ and all $\alpha_{q^*} \in A(q)$:

$$\check{T}(q, \alpha_{q^*}, q') \leq \quad (5)$$
$$\min_{x \in X_q} \min_{g_-(x) \leq w \leq g_+(x)} \mathbb{P}_\nu(f(x) + w + K_{q^*}(x; \hat{g}) + \nu \in X_{q'}),$$
$$\hat{T}(q, \alpha_{q^*}, q') \geq \quad (6)$$
$$\max_{x \in X_q} \max_{g_-(x) \leq w \leq g_+(x)} \mathbb{P}_\nu(f(x) + w + K_{q^*}(x; \hat{g}) + \nu \in X_{q'})$$

where $\mathbb{P}_\nu$ denotes probability with respect to ν.

Verification and synthesis problems for IMDP systems evaluated against scLTL specifications are often solved using graph theoretic methods on a product IMDP:

Definition 5 (PIMDP): Let $\mathcal{I} = (Q, A, \check{T}, \hat{T}, Q_0, O, L)$ be an IMDP and $\mathcal{A} = (S, s_0, O, \delta, F)$ be an FSA. The product IMDP (PIMDP) is defined as a tuple $\mathcal{P} = \mathcal{I} \otimes \mathcal{A} = (Q \times S, A, \check{T}', \hat{T}', Q \times s_0, F')$, where

- $\check{T}' : (q, s) \times A \times (q', s') := \check{T}(q, \alpha, q')$ if $s' \in \delta(s, L(q))$ and 0 otherwise
- $\hat{T}' : (q, s) \times A \times (q', s') := \hat{T}(q, \alpha, q')$ if $s' \in \delta(s, L(q))$ and 0 otherwise
- $(q_0, \delta(s_0, L(q_0))) \in (Q \times S)$ is a set of initial states of $\mathcal{I} \otimes \mathcal{A}$, and
- $F' = Q \times F$ is the set of accepting (final) states.

We can now formulate our proposed problem:

Problem 1: Design an iterative algorithm to sample and learn the unknown dynamics of system (1) without violating the scLTL specification ϕ and synthesize a control policy which satisfies ϕ with some desired threshold probability or prove that no such control policy exists.

To solve this problem, we construct a high-confidence IMDP abstraction of the system (1) using a GP estimation of the unknown dynamics. We then formulate a method to sample the state-space without violating the specification, updating the GP estimation until a satisfying control policy can be synthesized.

III. Abstraction of System as IMDP

In this section, we detail our approach to abstracting a system of the form (1) into a high-confidence IMDP.

We first need to determine an approximation of $g(x)$, the unknown dynamics. At each time step of system (1), we know $x[k+1]$, $f(x[k])$, and $u[k]$. Therefore, we can define

$$y[k] = x[k+1] - f(x[k]) - u[k] = g(x[k]) + \nu[k].$$

Then, we construct a Gaussian process estimation $\hat{g}(x)$ for $g(x)$ by considering a dataset of samples $(x[k], y[k])$.

Definition 6 (Gaussian Process Regression): Gaussian Process (GP) regression models a function $g_i : \mathbb{R}^n \to \mathbb{R}$ as a distribution with covariance $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{>0}$. Assume a dataset of m samples $D = \{(z^j, y_i^j)\}_{j \in \{1,\dots,m\}}$, where $z^j \in \mathbb{R}^n$ is the input and y_i^j is an observation of $g_i(z^j)$ under Gaussian noise with variance $\sigma_{\nu_i}^2$. Let $K \in \mathbb{R}^{m \times m}$ be a matrix defined elementwise by $K_{j\ell} = \kappa(z^j, z^\ell)$ and for $z \in \mathbb{R}^n$, let $k(z) = [\kappa(z, z^1)\ \kappa(z, z^2) \dots\ \kappa(z, z^m)]^T \in \mathbb{R}^m$. Then, the predictive distribution of g_i at a test point z is the conditional distribution of g_i given D, which is Gaussian with mean $\mu_{g_i,D}$ and variance $\sigma_{g_i,D}^2$ given by

$$\mu_{g_i,D}(z) = k(z)^T (K + \sigma_{\nu_i}^2 I_m)^{-1} Y \tag{7}$$

$$\sigma_{g_i,D}^2(z) = \kappa(z,z) - k(z)^T (K + \sigma_{\nu_i}^2 I_m)^{-1} k(z), \tag{8}$$

where I_m is the identity and $Y = \begin{bmatrix} y_i^1 & y_i^2 & \dots & y_i^m \end{bmatrix}^T$. In practice, GP regression has a complexity of $O(m^3)$. To mitigate this, we use sparse Gaussian process regression [10]:

Definition 7 (Sparse Gaussian Process Regression): A sparse Gaussian process uses a set $D_\eta = \{(z^j, y_i^j)\}_{j \in \{1,\dots,\eta\}}$ to approximate a GP of a larger dataset D. Given *inducing points* $\{z_j\}_{j \in \{1,\dots,\eta\}}$ with $Y_\eta = \begin{bmatrix} y_i^1 & y_i^2 & \dots & y_i^\eta \end{bmatrix}^T$ and covariance matrix A_η, the predictive distribution of the unknown function g_i has mean μ_{g_i,D_η} and variance σ_{g_i,D_η}^2

$$\mu_{g_i,D_\eta}(z) = k_\eta(z)^T (K_\eta + \sigma_{\nu_i}^2 I_\eta)^{-1} Y_\eta$$
$$\sigma_{g_i,D_\eta}^2(z) = \kappa(z,z) - k_\eta(z)^T K_\eta^{-1} (K_\eta - A_\eta) K_\eta^{-1} k_\eta(z)$$

where $K_\eta \in \mathbb{R}^{\eta \times \eta}$ is a matrix defined elementwise by $K_{\eta,j\ell} = \kappa(z^j, z^\ell)$ for all $z \in D_\eta$. For $z \in \mathbb{R}^n$, let $k_\eta(z) = [\kappa(z, z^1)\ \kappa(z, z^2) \dots\ \kappa(z, z^\eta)]^T \in \mathbb{R}^\eta$. The parameters $\{z^j\}_{j \in \{1,\dots,\eta\}}$, $\{y_i^j\}_{j \in \{1,\dots,\eta\}}$, and A_η are optimized to minimize the Kullback-Leibler divergence (evaluated at the inducing points) between $\mathcal{N}(\mu_{g_i,D_\eta}, \sigma_{g_i,D_\eta}^2)$, the posterior of g_i under the sparse GP; and $p(g_i|Y)$, the posterior of g_i under a GP with the full dataset D. We refer the reader to [10] for a detailed treatment of sparse Gaussian process theory. The computational complexity of sparse GP regression is $O(m\eta^2)$, so by fixing η the algorithm is linear in m. We note that sparse GP regression introduces error into the estimation; however, in practice this error does not affect the validity of our methods. Given some dataset D, we construct an estimation of the unknown dynamics independently in each coordinate and determine high-confidence bounds on the estimation error

$$\hat{g}_i^D(x) := \mu_{g_i,D}(x),$$
$$\gamma_i(x) := \beta \sigma_{g_i,D}(x) \geq |g_i(x) - \hat{g}_i^D(x)|$$

for each $i = 1, \dots, n$. We also determine high-confidence lower and upper bounds on $g(x)$ as

$$g_-(x) = \hat{g}^D(x) - \beta \sigma_{g,D}(x), \quad g_+(x) = \hat{g}^D(x) + \beta \sigma_{g,D}(x)$$

The parameter β is calculated as

$$\beta = \left(\frac{\sigma_\nu}{\sqrt{1 + (2/m)}} (B_i + \sigma_\nu \sqrt{2(\gamma_k^m + 1 + \log \frac{1}{\delta})}) \right) \tag{9}$$

for noise σ_ν-sub-Gaussian, m the number of GP samples, high-confidence parameter δ, information gain constant γ_k^m, and RKHS constant B_i as detailed in Lemma 1, [7]. Note that the same parameter $\beta \sigma_{g,D}$ is used to determine high-confidence bounds on both the estimation error and $g(x)$ itself.

For each region q in the state-space, we select a high-confidence error bound for the unknown dynamics as

$$\gamma_i(q) = \max_{x \in X_q} \gamma_i(x)$$

In practice, we compute this bound by sampling $\gamma_i(x)$ throughout the state-space, introducing a trade-off between approximation error and computation complexity. We now construct transition probability intervals assuming that the high-confidence bounds on unknown dynamics always hold:

Theorem 1 (Construction of Transition Probabilities):

Consider $q, q' \in Q$ and action $\alpha_{q^*} \in A(q)$. Lower bound $\check{T}$ and upper bound $\hat{T}$ transition probabilities from q to q' under α_{q^*} are given by

$$\check{T}(q, \alpha_{q^*}, q') = \prod_{i=1}^{n} \int_{a_i'}^{b_i'} \rho_{\nu_i}(z - x_{\min,i}(q, \alpha_{q^*}, q')) dz, \tag{10}$$

$$\hat{T}(q, \alpha_{q^*}, q') = \prod_{i=1}^{n} \int_{a_i'}^{b_i'} \rho_{\nu_i}(z - x_{\max,i}(q, \alpha_{q^*}, q'))dz, \quad (11)$$

where $x_{\min,i}$ is the i-th coordinate of $x_{\min}$ and similarly for $x_{\max,i}$, we recall ρ_{ν_i} is the probability density function of the stochastic noise ν_i, and a' and b' are the lower and upper boundary points for region q'. We define $x_{\min}$ and $x_{\max}$ as

$$x_{\min}(q, \alpha_{q^*}, q') = \underset{x \in X}{\operatorname{argmax}} \; ||x - c_{q'}||_1 \quad (12)$$
$$\text{s.t. } c_{q^*} - \gamma(q) \leq x \leq c_{q^*} + \gamma(q),$$

$$x_{\max}(q, \alpha_{q^*}, q') = \underset{x \in X}{\operatorname{argmin}} \; ||x - c_{q'}||_1 \quad (13)$$
$$\text{s.t. } c_{q^*} - \gamma(q) \leq x \leq c_{q^*} + \gamma(q),$$

where $||\cdot||_1$ is the 1-norm and $\gamma(q)$ is a high-confidence error bound on the unknown dynamics satisfying Assumption 1. Then, the transition probability bounds (10)–(11) satisfy the constraints for high-confidence IMDP abstractions in (5)–(6).

Proof: The righthand side of the bound in equation (5) can be rewritten as

$$\min_{x \in X_q} \min_{\substack{g_-(x) \leq w \\ \leq g_+(x)}} \mathbb{P}_\nu(f(x) + w + K_{q^*}(x; \hat{g}) + \nu \in X_{q'}) \quad (14)$$

$$= \min_{x \in X_q} \min_{g_-(x) \leq w \leq g_+(x)} \mathbb{P}_\nu(c_{q^*} + w - \hat{g}(x) + \nu \in q') \quad (15)$$

$$= \min_{x \in X_q} \min_{-\gamma(x) \leq \omega \leq \gamma(x)} \mathbb{P}_\nu(c_{q^*} + \omega + \nu \in X_{q'}) \quad (16)$$

$$= \min_{x \in X_q} \min_{\substack{-\gamma(x) \leq \omega \\ \leq \gamma(x)}} \prod_{i=1}^{n} \mathbb{P}_{\nu_i}(c_{q^*,i} + \omega_i + \nu_i \in [a_{q',i}, b_{q',i}]) \quad (17)$$

$$= \min_{x \in X_q} \prod_{i=1}^{n} \min_{\substack{-\gamma_i(x) \leq \omega_i \\ \leq \gamma_i(x)}} \mathbb{P}_{\nu_i}(c_{q^*,i} + \omega_i + \nu_i \in [a_{q',i}, b_{q',i}]) \quad (18)$$

$$\geq \prod_{i=1}^{n} \min_{\substack{-\gamma_i(q) \leq \omega_i \\ \leq \gamma_i(q)}} \mathbb{P}_{\nu_i}(c_{q^*,i} + \omega_i + \nu_i \in [a_{q',i}, b_{q',i}]) \quad (19)$$

where (14) is the righthand side of (5); (15) follows after expanding the feedback controller expression $K_{q^*}(x; \hat{g})$ using (4) and simplifying; (16) follows by assumption of high-confidence error bound $\gamma(x)$ and the definition of $g_-(x)$ and $g_+(x)$ from Assumption 1 and taking $\omega = w - \hat{g}(x)$; (17) follows by assumption that each ν_i is independent and $\mathbb{P}_{\nu_i}$ denotes probability with respect to ν_i, where we recall that $a_{q'}$ and $b_{q'}$ are the lower and upper corners of the region $X_{q'}$, and $a_{q',i}$ is the i-th coordinate of $a_{q'}$ and similarly for $c_{q*,i}$ and $b_{q',i}$; (18) follows from the fact that the hyper-rectangular constraint $-\gamma(x) \leq \omega \leq \gamma(x)$ is equivalent to independent constraint $-\gamma_i(x) \leq \omega_i \leq \gamma_i(x)$ along each coordinate; and (19) follows from the definition $\gamma_i(q) = \max_{x \in X_q} \gamma_i(x)$.

Now, because the probability distribution for each random variable ν_i is assumed unimodal and symmetric, $\mathbb{P}_{\nu_i}(c_{q^*,i} + \omega_i + \nu_i \in [a_{q',i}, b_{q',i}])$ is minimized when the distance between $(c_{q^*,i} + \omega_i)$ and the midpoint of $[a_{q',i}, b_{q',i}]$ is maximized, *i.e.*, when $|c_{q^*,i} + \omega_i - c_{q',i}|$ is maximized, subject to the constraint $-\gamma_i(q) \leq \omega_i \leq \gamma_i(q)$. Substituting $x = c_{q^*} + \omega$, and observing that $\|x - c_{q'}\|_1 = \sum_{i=1}^{n} |x_i - c_{q',i}|$, this is exactly the maximizing point specified by $x_{\min}(q, \alpha_{q^*}, q')$ in (12). Thus, the expression in (19) is equivalent to

$$\prod_{i=1}^{n} \mathbb{P}_{\nu_i}(x_{\min,i}(q, \alpha_{q^*}, q') + \nu_i \in [a_{q',i}, b_{q',i}]), \quad (20)$$

which in turn is equivalent to the righthand side of (10), establishing the bound in (5). An analogous argument establishes that (11) satisfies (6). ■

We construct a high-confidence IMDP abstraction of the system using the hyper-rectangular partition regions as states, high-confidence bounds on the unknown dynamics obtained via GP regression, and transition probability intervals calculated using Theorem 1, solving the first part of Problem 1.

IV. Safe Sampling of PIMDP

A. Probability of Satisfaction Calculation

Given a high-confidence IMDP abstraction of the system and a FSA of a desired scLTL specification, we construct a PIMDP using Definition 5. We first introduce the concept of control policies and adversaries:

Definition 8 (Control Policy): A control policy $\pi \in \Pi$ of a PIMDP is a mapping $(Q \times S)^+ \to A$, where $(Q \times S)^+$ is the set of finite sequences of states of the PIMDP.

Definition 9 (PIMDP Adversary): Given a PIMDP state (q, s) and action α, an adversary $\xi \in \Xi$ is an assignment of transition probabilities T'_ξ to all states (q', s') such that

$$\check{T}'((q,s), \alpha, (q', s')) \leq T'_\xi((q,s), \alpha, (q', s')) \leq \hat{T}'((q,s), \alpha, (q', s')).$$

In particular, we use a *minimizing* adversary, which realizes transition probabilities such that the probability of satisfying the specification is minimal, and a *maximizing* adversary, which maximizes the probability of satisfaction.

To find safe sampling cycles in the PIMDP, we calculate

$$\check{P}_{\max}((q,s) \models \phi) = \max_{\pi \in \Pi} \min_{\xi \in \Xi} P(w \models \phi \mid \pi, \xi, w[0] = (q,s)),$$

which is the probability that a random path w starting at PIMDP state (q, s) satisfies the scLTL specification ϕ under a maximizing control policy π and minimizing adversary ξ.

Additionally, we will also use the best case probability of satisfaction under a maximizing control policy and adversary:

$$\hat{P}_{\max}((q,s) \models \phi) = \max_{\pi \in \Pi} \max_{\xi \in \Xi} P(w_i \models \phi \mid \pi, \xi, w[0] = (q,s))$$

To calculate these probabilities, we use a value iteration method proposed in Section V, [14].

B. Nonviolating Sub-Graph Generation

We note that scLTL specifications may generate FSA states which are absorbing and non-accepting, i.e., it is impossible to satisfy the specification once one of these states is reached. Such states may also exist in PIMDP constructions even without appearing in the corresponding FSA. We define these states as those which have zero probability of satisfying the scLTL specification under any control policy and adversary:

$$\text{Failure States} = \{(q,s) \in Q \times S \mid \hat{P}_{\max}((q,s) \models \phi) = 0\}.$$

We can then define a notion of specification nonviolation:

Definition 10 (Nonviolating PIMDP): A PIMDP $\mathcal{P}$ is *nonviolating* with respect to a scLTL specification ϕ if there exists no failure states in $\mathcal{P}$.

Our algorithm for calculating a nonviolating PIMDP is as follows. We first initialize a set of failure states. Then, we loop through all non-failure states and prune actions which have nonzero upper-bound transition probability to failure states. We check if this pruning has left any states with no available actions, designating these also as failure states to prune. The process continues until no new failure states are found. Our nonviolating sub-graph is the set of all unpruned states with their remaining actions. All algorithm psuedocodes are available in the appendix of an extended version of this paper, posted on arXiv[1].

C. Candidate Cycle Selection

Now that we have a nonviolating sub-graph of our PIMDP, we want to select a path which we can take in order to sample the state-space indefinitely while maximizing the information gain of our Gaussian process. To do this, we first recall the concept of *maximal end components* [15]:

Definition 11 (End Component [15]): An *end component* of a finite PIMDP $\mathcal{P}$ is a pair $(\mathcal{T}, Act)$ with $\mathcal{T} \subseteq (Q \times S)$ and $Act : \mathcal{T} \to A$ such that

- $\emptyset \neq Act(q,s) \subseteq A(q)$ for all states $(q,s) \in \mathcal{T}$,
- $(q,s) \in \mathcal{T}$ and $\alpha \in Act(q,s)$ implies $\{(q',s') \in \mathcal{T} \mid \hat{T}(q,\alpha,q')) > 0, s' \in \delta(s, L(q))\} \subseteq \mathcal{T}$,
- The digraph $G_{(\mathcal{T},Act)}$ induced by $(\mathcal{T}, Act)$ is strongly connected.

Definition 12 (Maximal End Component (MEC) [15]): An end component $(\mathcal{T}, Act)$ of a finite PIMDP $\mathcal{P}$ is *maximal* if there is no end component $(\mathcal{T}^*, Act^*)$ such that $(\mathcal{T}, Act) \neq (\mathcal{T}^*, Act^*)$ and $\mathcal{T} \subseteq \mathcal{T}^*$ and $Act(q,s) \subseteq Act^*(q,s)$ for all $(q,s) \in \mathcal{T}$.

PIMDP abstractions have the property that any infinite path will eventually stay in a single MEC. We propose the following heuristic in order to select a MEC to cycle within. First, we calculate $\check{P}_{\max}$ from our initial state to each candidate MEC. We reject any MEC which we cannot reach with probability 1, or, in case no MECs can be reached with probability 1, we immediately select the MEC with the highest reachability probability. If multiple candidate MECs remain, we then calculate the Gaussian process covariance $\kappa(c_q, c_{q^*})$ between the centers of the IMDP states q in each remaining candidate MEC and the accepting IMDP state q^*. We sum the covariances for all states in each MEC and select the MEC with the highest total covariance score, which corresponds to maximum information gain [16], defined as reduction of GP uncertainty at the accepting state. We generate a control policy by selecting the actions at each state which give the maximum probability of reaching the MEC. Once in the MEC, we use a controller which cycles through the available actions.

By applying the algorithms detailed above to calculate a non-violating PIMDP and MEC, we generate a control policy which samples the state-space indefinitely without violating the specification, solving the second part of Problem 1.

[1]https://arxiv.org/abs/2202.01358

D. Iterative Sampling Algorithm

We now detail our complete method to solve Problem 1. Given a scLTL specification ϕ which we want to satisfy with probability P_{sat}, we construct a PIMDP using a high-confidence IMDP abstraction of the system in Eq. (1) and an FSA which models ϕ. Then, we calculate reachability probabilities under a minimizing adversary $\check{P}_{\max}$ from the initial states in the PIMDP to the accepting states. If $\check{P}_{\max} \geq P_{\text{sat}}$, then the control policy selects the actions which produce $\check{P}_{\max}$ at each state and the problem is solved. Otherwise, we calculate a control policy to sample the state-space without violating the specification ϕ using the methods in previous sections. We follow the calculated control policy for a predetermined number of steps and sample the unknown dynamics at each step. We batch update the GP with the data collected, reconstruct transition probability intervals for each state, and recalculate reachability probabilities $\check{P}_{\max}$ for our initial states. If $\check{P}_{\max} \geq P_{\text{sat}}$, a satisfying control policy is found; otherwise, we repeat the process above. Our iterative algorithm ends when $\check{P}_{\max} \geq P_{\text{sat}}$; the GP approximation has low enough uncertainty to know that a successful control policy cannot be synthesized, *i.e.*, when the reachability probability $\hat{P}_{\max}$ under a maximizing adversary is less than the desired P_{sat}; or a maximum number of iterations has been reached.

V. Case Study

Suppose we have a mobile robot in a 2D state-space with position $x \in X := [0,5]^2 \subset \mathbb{R}^2$. The state-space is partitioned into a set of 25 hyper-rectangular regions corresponding to IMDP states. The dynamics of the robot are

$$x[k+1] = x[k] + u[k] + g(x[k]) + v \tag{21}$$

where $g(x)$ models the unknown effect of the slope of the terrain. The control action u is generated by the family of controllers in Section II where the set of available target regions are those left, right, above, or below each region.

Within the state-space, we have one goal region with the atomic proposition `Goal` and a set of hazard regions labeled with `Haz`. These yield the scLTL specification

$$\phi_1 = \neg \texttt{Haz}\ \mathcal{U}\ \texttt{Goal}. \tag{22}$$

An illustration of the state-space is shown in Figure 2. We choose a low-dimensional case study in order to illustrate our methodology. Future works will refine our algorithms on applications with higher-dimensional state-spaces.

The true $g(x)$ is sampled from two randomly generated Gaussian processes (one for each dimension) with bounded support $[-0.4, 0.4]$ and squared exponential kernel κ,

$$\kappa(x, x') = \sigma_g^2 e^{-\frac{(x-x')^2}{2l^2}}. \tag{23}$$

We choose hyperparameters $\sigma_g = 0.45$ and $l = 1.75$.

We estimate the unknown dynamics with two sparse Gaussian processes with the same kernel as the true dynamics. We

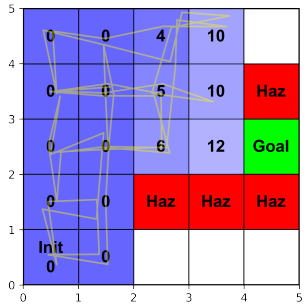

Fig. 2. State-space of the case study. The initial region is labeled with "Init", the (green) target region is labeled with "Goal", and the (red) hazard regions are labeled with "Haz". States that eventually enter the safe cycle are blue, and the number in the region indicates the iteration of the algorithm at which the state enters the safe cycle. States which are not numbered do not enter the safe cycle. The yellow trace is an example of a sampling run.

sample the GPs at 100 points in each region to determine error bounds. We set the number of inducing points $\eta = 250$ and choose our high-confidence-bound parameter $\beta = 2$. Each iteration of the algorithm takes 250 steps, so the total number of data samples m is the number of iterations times 250. Our stochastic noise ν is independently drawn from two truncated Gaussian distributions, one for each dimension, and both with $\sigma_\nu = 0.1$ and bounded support $[-0.2, 0.2]$.

We next apply the iterative algorithm described in Section IV-D, setting the desired probability of satisfying the specification to 1. Our algorithm successfully finds a satisfying feedback control strategy in an average of 15 iterations (calculated over 10 runs). The algorithm is implemented in Python on a 2.5 GHz Intel Core i9 machine with 16 GB of RAM and a Nvidia RTX 3060 GPU, and requires on average 1 minute 14 seconds to complete.

Figure 2 depicts the expansion of the safe cycle used to sample the state-space. Initially, only the left two columns of states are safe and reachable. As the algorithm progresses, more states and actions are added to the safe cycle, moving the system closer to the goal until the unknown dynamics can be estimated with enough certainty to achieve a probability of satisfying the specification of 1.

The left plot in Figure 3 depicts the total transition probability uncertainty for the system after each iteration

$$T_{\text{unc,total}} = \sum_{q \in Q} \sum_{\alpha \in A(q)} \sum_{q' \in Q} \hat{T}(q, \alpha, q') - \check{T}(q, \alpha, q'). \quad (24)$$

The right plot in Figure 3 shows the probability of satisfying the specification after each iteration.

VI. Conclusion

In this work, we developed a method to safely learn unknown dynamics for a system motivated by the robotic motion-planning problem. Our approach uses an IMDP abstraction of the system and a finite state automaton of scLTL specifications. We designed an algorithm for finding nonviolating paths within a product IMDP construction which can be used to sample the state-space and construct a Gaussian process approximation of unknown dynamics. We then detailed an algorithm to iteratively sample the state-space to improve the probability of satisfying a desired specification and demonstrated its use with a case study of robot navigation. Our approach can be used with any system for which a high-confidence IMDP abstraction can be constructed as well as any objective which can be written as a scLTL specification. Future work will apply these methods to models of bipedal walking robots utilizing region-based motion planning [13].

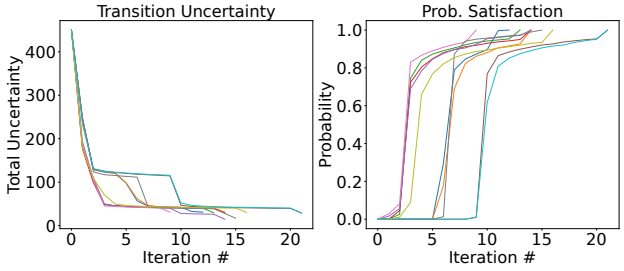

Fig. 3. The left plot shows the total uncertainty in transition probability intervals after each iteration of the algorithm, and the right plot shows the probability of satisfying the specification after each iteration. Results are plotted over 10 runs of the algorithm. The uncertainty decreases as more data samples are collected, and likewise the probability of satisfaction increases once the safe cycle has expanded close enough to the goal.

References

[1] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[2] C. Belta, B. Yordanov, and E. Göl, *Formal Methods for Discrete-Time Dynamical Systems*, ser. Studies in Systems, Decision and Control. Springer International Publishing, 2017.

[3] C. Mark and S. Liu, "Stochastic MPC with distributionally robust chance constraints," *IFAC*, vol. 53, no. 2, pp. 7136–7141, 2020.

[4] A. Lavaei, S. Soudjani, A. Abate, and M. Zamani, "Automated verification and synthesis of stochastic hybrid systems: A survey," 2021.

[5] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 524–541, Jun. 2003.

[6] M. Ahmadi, A. Israel, and U. Topcu, "Safety assessemt based on physically-viable data-driven models," in *56th IEEE CDC*, Dec. 2017, pp. 6409–6414.

[7] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Strategy synthesis for partially-known switched stochastic systems," in *Proceedings of HSCC '21*, pp. 1–11.

[8] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in neural information processing systems 8*. MIT press, 1996, pp. 514–520.

[9] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with Gaussian processes," in *53rd IEEE CDC*, Dec. 2014, pp. 1424–1431.

[10] F. Leibfried, V. Dutordoir, S. John, and N. Durrande, "A tutorial on sparse gaussian processes and variational inference," 2021, arXiv: 2012.13962.

[11] A. A. Julius, A. Halasz, M. S. Sakar, H. Rubin, V. Kumar, and G. J. Pappas, "Stochastic modeling and control of biological systems: The lactose regulation system of *Escherichia Coli*," *IEEE Transactions on Automatic Control*, vol. 53, no. Special Issue, pp. 51–65, 2008.

[12] E. Altman, T. Başar, and R. Srikant, "Congestion control as a stochastic control problem with action delays," *Automatica*, vol. 35, no. 12, pp. 1937–1950, 1999.

[13] A. Shamsah, J. Warnke, Z. Gu, and Y. Zhao, "Integrated Task and Motion Planning for Safe Legged Navigation in Partially Observable Environments," 2021, arXiv: 2110.12097.

[14] M. Lahijanian, S. B. Andersson, and C. Belta, "Formal Verification and Synthesis for Discrete-Time Stochastic Systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 8, pp. 2031–2045, Aug. 2015.

[15] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[16] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, p. 3250–3265, May 2012.