Received 29 March 2023; revised 10 June 2023; accepted XX xxxx XXXX; Date of publication XX xxxx XXXX; date of current version 12 July 2023.

Digital Object Identifier 10.1109/OJCSYS.2023.Doi Number

Abstraction-based Planning for Uncertainty-aware Legged Navigation

Jesse Jiang[1] (Student Member, IEEE), Samuel Coogan[1,2] (Senior Member, IEEE), Ye Zhao[3] (Senior Member, IEEE)

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA
[2]School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA
[3]School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA

CORRESPONDING AUTHOR: Jesse Jiang (e-mail: jjiang@gatech.edu)

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

ABSTRACT This paper addresses the problem of temporal-logic-based planning for bipedal robots in uncertain environments. We first propose an Interval Markov Decision Process abstraction of bipedal locomotion (IMDP-BL). Motion perturbations from multiple sources of uncertainty are incorporated into our model using stacked Gaussian process learning in order to achieve formal guarantees on the behavior of the system. We consider tasks which can be specified using Linear Temporal Logic (LTL). Through a product IMDP construction combining the IMDP-BL of the bipedal robot and a Deterministic Rabin Automaton (DRA) of the specifications, we synthesize control policies which allow the robot to safely traverse the environment, iteratively learning the unknown dynamics until the specifications can be satisfied with satisfactory probability. We demonstrate our methods with simulation case studies.

I. Introduction

Legged locomotion and navigation have been extensively studied in recent years due to the accessibility of reliable, highly-agile quadrupedal [1], [2] and bipedal [3], [4] robotic platforms. As compared to more widely received mobile robot and drone platforms, legged robots offer the promise of high mobility in difficult terrain with additional manipulation capabilities for material transportation.

Numerous planning works for legged robots have focused on optimization methods for locally stable control [5], [6], [7]. These methods are primarily concerned with formulating tracking controllers which allow legged robots to track desired motion trajectories formulated by higher-level task planners, especially in the presence of motion perturbations.

In the field of formal methods, there has been much work on using Linear Temporal Logic (LTL) specifications for task and motion planning (TAMP) [8], [9], [10], [11], [12], [13], [14]. Formal methods approaches to TAMP allow for formal guarantees on task satisfaction and offer expressive semantic languages to represent broad classes of TAMP objectives. Many studies have focused on mobile robot navigation in partially observable domains through exploration [15], [16], re-synthesis when encountering unexpected obstacles [17], [18], and receding-horizon planning [19]. However, these existing approaches are better suited for guaranteeing successful navigation and collision avoidance in environments with simple robot dynamics such as point-mass mobile vehicles. Legged navigation in complex environments while incorporating safety-critical locomotion kinematics and dynamics is largely an open research question.

In this work, we apply temporal-logic-based planning techniques onto legged navigation in order to synthesize control policies for legged robots which enforce the satisfaction of complex tasks in uncertain environments. From a formal methods perspective, our work demonstrates the applicability of abstraction-based verification and synthesis techniques onto legged robotic systems with complex dynamics. With respect to the locomotion community, our work advances the use of formal methods as a framework to enable robots to execute a broad class of navigation tasks with formal guarantees on safety and satisfiability.

A. Related Work

High-level task planning for legged robots has not been widely explored, with many works focusing on developing a reduced-order-model-based motion planner which designs optimal locomotion trajectories [20], [21], [22], [23], [24], [25]. One major difficulty in applying high-level task planning approaches to legged locomotion problems is the

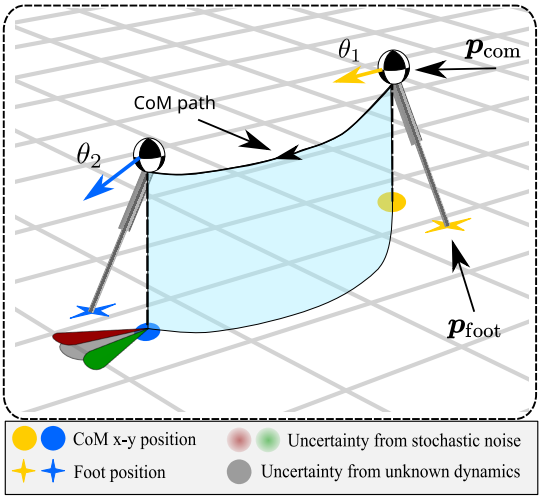


FIGURE 1. Illustration of the PIPM-based IMDP-BL model, depicting a walking single step. The spheres represent the CoM position of the robot at the two discrete apex states, and the arc connecting them shows the continuous trajectory of the CoM during the step. The apex positions are also projected to the $x-y$ plane using the yellow and blue dots, while the foot contact points during the step are depicted using the stars. The heading angle of the robot before and after the step is shown using the arrows coming out of the CoM. The cones in the bottom left corner of the figure illustrate the regions used in transition probability calculations. The GP yaw uncertainty (gray region) is centered on the targeted yaw angle. To calculate the maximum probability that the robot deviates to a yaw angle bin left of the target state, we center the stochastic yaw noise (green region) on the left edge of the GP yaw uncertainty and compute the probability densities. Likewise, the minimum probability of the robot turning further left than intended is computed by centering the stochastic yaw noise on the right edge of the GP yaw uncertainty (red region).

need for reduced-order dynamical models which enable computationally tractable planning while capturing the full-body kinodynamics of the robot [26], [27]. One high-level approach which has seen success is the use of Markov Decision Process (MDP) planners for control policy synthesis [28], [29]. These MDP-based approaches abstract the motion trajectories of the robot by planning using a subset of the robot's states which approximate the low-level states.

In the formal methods community, Interval Markov Decision Process (IMDP) [30] abstraction-based modeling has shown potential for enabling planning for systems with complex dynamics [31], [32], [33], [34], [35]. As opposed to standard MDPs, IMDPs have transition probability intervals, allowing for the modeling of stochastic or uncertain dynamics [36], [37], [38]. Multiple approaches have been proposed in the literature for the problem of control policy synthesis for IMDP-abstracted systems with respect to ω-regular specifications, including partitioning and refinement strategies [39], invariant set computation [40], and two-player games [41]. These approaches address scenarios where the transition probability intervals of the graph remain static. In contrast, in this work we develop graph-based techniques which synthesize control policies to improve transition probability intervals online, thus increasing the probability of satisfying the desired specifications. In particular, we shrink these intervals via integration of Gaussian process (GP) learning [42] of uncertainties into the IMDP as explored in some recent works [43]. GPs have proven to be a popular approach for learning uncertainties in robotic planning problems [44], [45], [46], [47], [48]. IMDPs render a coherent structure to map learned GP high-confidence bounds on the uncertainty into transition probability intervals so that verification and synthesis tasks explicitly consider uncertainty. This methodology combines the formal guarantees associated with abstraction-based planners with the online adaptation capabilities of learning.

B. Contributions

This study proposes a novel integrated planning framework of legged locomotion which fuses high-level formal methods-based control synthesis techniques, learning methods, and low-level dynamics-aware motion plans in order to enable temporal-logic-based planning capabilities for bipedal robots. We first formulate an IMDP model for bipedal locomotion which incorporates the complex dynamics associated with legged locomotion. We then integrate a multi-layer GP learning structure into the IMDP abstraction to allow for learning of correlated robot model and environmental (*i.e.*, terrain) uncertainties. Using this IMDP model, we develop and prove the validity of a planning methodology which allows a robot to iteratively traverse its environment to perform online learning while maintaining safety with respect to complex LTL specifications. Finally, we map our high-level planner onto a low-level full-body kinematics-based bipedal robot model and demonstrate simulation results.

Within our framework of temporal-logic-based planning for legged locomotion, we make the following contributions:

- We develop a novel IMDP abstraction of legged locomotion systems, which possess more complex dynamics than the systems used for the IMDP methodology in our previous work [49]. For high-level planning, we abstract the system using reduced-order Prismatic Inverted Pendulum Model (PIPM) dynamics. We also explicitly consider the effects of low-level full-body kinematic constraints in our IMDP planner, allowing for computationally tractable synthesis of high-level control policies which can be executed on the full-order legged dynamical model.
- We generalize our previous work on IMDP planning in [49] to broaden the language of allowable specifications from the syntactically co-safe fragment of LTL to the entire language of LTL. Using graph-theoretic techniques, we develop theory and algorithms for synthesis of control policies for data sampling which are non-violating with respect to infinite-horizon specifications. We demonstrate the computational tractability of our approach on case studies with state space sizes which are orders of magnitude higher than those in [49].
- We propose a stacked GP learning model for structured learning of the motion uncertainties from modeling and environmental sources inherent to legged locomotion. This allows our planner to compensate for *a priori*

unknown motion perturbations and improve online the probability of satisfying high-level tasks.

II. Background

A. Reduced-order Locomotion Dynamics

We model the rigid body dynamics of a bipedal robot using the prismatic inverted pendulum model (PIPM) [50]. In this model, the mass of the robot is modeled as a single point on the hip of the robot. This point is called the 3D Center of Mass (CoM) and has position $\boldsymbol{p}_{\text{com}} = (x, y, z)^\top$, which are the sagittal, lateral, and vertical coordinates, respectively. The robot also has angular motion with orientation angles $(\phi, \theta, \psi)^\top$ and a foot contact position $\boldsymbol{p}_{\text{foot}} = (x_{\text{foot}}, y_{\text{foot}}, z_{\text{foot}})^\top$. Following the exposition in [51], we take the position and velocity of the CoM to be the system state space $\boldsymbol{\xi} = (\boldsymbol{p}_{\text{com}}^\top, \dot{\boldsymbol{p}}_{\text{com}}^\top)^\top = (x, y, z, \dot{x}, \dot{y}, \dot{z})^\top \in \Xi \subseteq \mathbb{R}^6$, where Ξ is the set of admissible CoM positions and velocities. Figure 1 illustrates these parameters. The second-order CoM dynamics of the legged robot at the q^{th} step are

$$\begin{aligned}
\ddot{x} =& \omega_q^2(x - x_{\text{foot}_q}) - \frac{\omega_q^2}{mg}(\tau_y + b_q\tau_z) \\
\ddot{y} =& \omega_q^2(y - y_{\text{foot}_q}) - \frac{\omega_q^2}{mg}(\tau_x + a_q\tau_z) \\
\ddot{z} =& \, a_q\omega_q^2(x - x_{\text{foot}_q}) + b_q\omega_q^2(y - y_{\text{foot}_q}) \\
& - \frac{\omega_q^2}{mg}(a_q\tau_y + b_q\tau_x + 2a_qb_q\tau_z),
\end{aligned} \tag{1}$$

with m the robot mass, g the gravitational acceleration, ω_q the phase-space asymptotic slope parameter, a_q and b_q the slope coefficients, and c_q the constant bias term. The hybrid control inputs to this system are the discrete foot contact position and the PIPM asymptotic slope ω_q. The derivation of Equation 1 is found in [50]. For hybrid locomotion planning, we introduce *apex* and *keyframe states*:

Definition 1 (Apex State):
The apex state occurs when the CoM sagittal position is equal to the location of the foot contact in the sagittal axis (*i.e.* the point in the walking step when the robot's body is exactly over its foot contact). The coordinates of the CoM at the apex state are $(x_{\text{apex}}, y_{\text{apex}}, z_{\text{apex}})^\top$.

Definition 2 (Locomotion Keyframe State):
A keyframe state of the PIPM is defined as $\boldsymbol{k} = (d, \Delta\theta, \Delta z_{\text{foot}}, v_{\text{apex}}, z_{\text{apex}}) \in \mathcal{K}$, where

- $d := x_{\text{apex,n}} - x_{\text{apex,c}}$ is the walking step length;
- $\Delta\theta := \theta_{\text{apex,n}} - \theta_{\text{apex,c}}$ is the yaw angle change;
- $\Delta z_{\text{foot}} := z_{\text{foot,n}} - z_{\text{foot,c}}$ is the height change;
- v_{apex} is the CoM sagittal apex velocity;
- z_{apex} is the global CoM height at apex.

Summary of the PIPM planning approach: We use the apex states of the robot as discrete-time states for high-level planning. At each apex state, the objective of the discrete-time planner is to target an apex state to reach in the next step. The parameters of the keyframe state correspond to the actions commanded at each apex state. The high-level action $\boldsymbol{a}_{\text{HL}} = (d, \Delta\theta, \Delta z_{\text{foot}}) \in \mathcal{A}_{\text{HL}}$ determines the (x, y, z) coordinates of the next apex state. In order to calculate the corresponding action in the PIPM dynamics, we first need to compute the additional parameters $\boldsymbol{a}_{\text{LL}} = (v_{\text{apex}}, z_{\text{apex}})$ for the action using a low-level motion planner. Together, the keyframe state parameters determine the foot placement of the next walking step. The keyframe state also maps to a corresponding asymptotic slope ω_q required for tracking the continuous trajectory in between apex states.

B. Temporal-Logic-Based Task Planning

For the high-level planning component of our problem, we utilize an Interval Markov Decision Process (IMDP) model:

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.

$A(q)$ denotes the set of actions at q. Moreover, for all $q \in Q$ and all $\alpha \in A(q)$, $\check{T}$ and $\hat{T}$ satisfy

$$\sum_{q' \in Q} \check{T}(q, \alpha, q') \leq 1 \leq \sum_{q' \in Q} \hat{T}(q, \alpha, q').$$

In this work, we consider specifications which can be written using the semantics of Linear Temporal Logic (LTL):

Definition 4 (Linear Temporal Logic [52, Def. 2.1]):
A *linear temporal logic (LTL)* formula ϕ over a set of observations O is recursively defined as

$$\begin{aligned}
\phi =& \top \mid o \mid \neg o \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc \phi \mid \\
& \phi_1 \mathcal{U} \phi_2 \mid \Diamond \phi \mid \Box \phi \mid \phi_1 \to \phi_2 \mid \phi_1 \leftrightarrow \phi_2
\end{aligned}$$

where $o \in O$ is an observation and ϕ, ϕ_1, and ϕ_2 are LTL formulas. We define the *next* operator $\bigcirc$ as satisfying ϕ in the next state transition, the *until* operator $\mathcal{U}$ satisfying ϕ_1 until ϕ_2 is satisfied, the *eventually* operator $\Diamond$ as $\top \mathcal{U} \phi$, and the *always* operator $\Box$ as $\neg\Diamond\neg\top$.

It is well known that the satisfaction of LTL formulas can be checked using deterministic Rabin automata [52, Def. 2.7]:

Definition 5 (Deterministic Rabin Automaton):
A deterministic Rabin automaton (DRA) is a tuple $\mathcal{R} = (S, s_0, O, \delta, F)$, where

- S is a finite set of states,

- $s_0 \subset S$ is a singleton initial state,
- O is the input alphabet, which corresponds to observations from the LTL formula,
- $\delta : S \times O \to 2^S$ is a transition map which is either $\emptyset$ or a singleton for all $s \in S$ and $o \in O$, and
- $F = \{(G_1, B_1), \cdots, (G_n, B_n)\}$, where $G_i, B_i \subseteq S, i = 1, 2, \cdots, n$ is the acceptance condition.

The semantics of a deterministic Rabin automaton are defined over infinite input words in O^ω (the set of infinite sequences of observations). A run of $\mathcal{R}$ over an infinite word $w_O = w_O(1), w_O(2)w_O(3) \cdots \in O^\omega$ is a sequence $w_S(1)w_S(2)w_S(3) \cdots \in S^\omega$, where $w_S(1) = s_0$ and $w_S(k+1) = \delta(w_S(k), w_O(K))$ for all $k \geq 1$.

A run w_S admits a set $\inf(w_S) = \{w_S(i) : \forall m \in \mathbb{N}\ \exists k > m$ s.t. $w_S(k) = w_S(i)\}$, defined as the set of observations in w_S which appear infinitely often.

Then, a run w_S is *accepted* by $\mathcal{R}$ if $\inf(w_S) \cap G_i \neq \emptyset \wedge \inf(w_S) \cap B_i = \emptyset$ for some $i \in \{1, \cdots, n\}$. If w_S is accepted by $\mathcal{R}$, we say that $w_s \models \mathcal{R}$.

C. Gaussian Process Learning

In order to learn the uncertainties present in our system, we will utilize Gaussian process (GP) regression:

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 $D = \{(z^j, y_i^j)\}_{j \in \{1,\ldots,m\}}$, 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 $\sigma^2_{\nu_i}$. 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 $k(z) = [\kappa(z, z^1)\ \kappa(z, z^2) \ldots \kappa(z, z^m)]^T \in \mathbb{R}^m$. 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 $\sigma^2_{g_i,D}$ given by

$$\mu_{g_i,D}(z) = k(z)^T (K + \sigma^2_{\nu_i} I_m)^{-1} Y \quad (2)$$

$$\sigma^2_{g_i,D}(z) = \kappa(z, z) - k(z)^T (K + \sigma^2_{\nu_i} I_m)^{-1} k(z), \quad (3)$$

where I_m is the identity and $Y = \begin{bmatrix} y_i^1 & y_i^2 & \ldots & y_i^m \end{bmatrix}^T$.

In practice, we use a sparse Gaussian process regression approximation [53] to reduce computational complexity.

III. Problem Setup

We now define our planning problem for bipedal robot locomotion. We consider a 3D environment for the bipedal robot in which there exists *a priori* a rectangular partition R on the $x - y$ plane with partition regions $\{R_n\}_{n \in N}$:

$$R_n = \{x, y | a_{x,n} \leq x \leq b_{x,n}, a_{y,n} \leq x \leq b_{y,n}\} \subset R, \quad (4)$$

where $a_{x,n}$ and $a_{y,n}$ are lower bounds on the x and y coordinates, respectively, and $b_{x,n}$ and $b_{y,n}$ are upper bounds. We assume that the bipedal robot can be modeled using the PIPM dynamics as introduced in Subsection IV.A.

Assumption 1:
There exists a model error between the PIPM used for planning and the low-level controller of the bipedal robot which causes deviation from the desired trajectory of motion. Additionally, the terrain elevation of the environment is unknown, which results in a separate motion perturbation. Both sources of uncertainty can be characterized and learned.

Assumption 2:
The bipedal robot experiences a stochastic yaw perturbation ν at each walking step which can be modeled as a zero mean random variable with stationary, symmetric, and unimodal distribution ρ_ν. We further assume a bounded support on the distribution so that the perturbation cannot drive the system into arbitrary regions of the space. In particular, this makes the distribution sub-Gaussian.

In this work, we specifically consider the effect of uncertainties on yaw angle error. Using proprioceptive sensing only for legged robots, the results in [54], [55], [56], [57] proved that the absolute yaw angle is unobservable so that estimates of these quantities will drift over time. To address this issue, we explicitly model yaw uncertainties using GPs.

Our objective is to develop a high-level planning algorithm which considers the robot's full-body kinematics to satisfy a LTL objective in the presence of robot system and environmental uncertainties:

Problem 1:
Design a temporal-logic-based planning algorithm for a bipedal robot to safely traverse its environment and learn system and environmental uncertainties in order to satisfy given LTL specifications with satisfactory probability under worst-case resolutions of the transition probability intervals of the discretized model of the robot.

Subproblem 1.1:
Design an IMDP model of bipedal locomotion based on Prismatic Inverted Pendulum Model dynamics. Integrate Gaussian process learning of system and environmental uncertainties into the bipedal IMDP in order to model and account for motion perturbations.

Subproblem 1.2:
Synthesize a control policy for an IMDP-abstracted system which allows the robot to traverse its environment while maintaining nonzero probability of satisfying given LTL specifications under best-case resolutions of transition probability intervals.

IV. Bipedal IMDP Formulation

In this section, we propose a high-confidence IMDP model of bipedal robot locomotion which incorporates Gaussian process learning of system uncertainties.

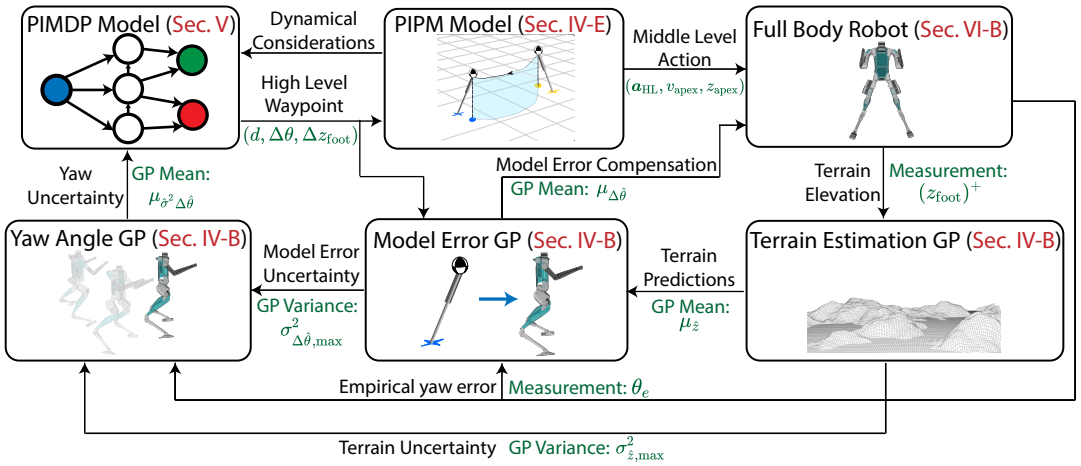


FIGURE 2. **Overall IMDP-BL framework. Planning is done using a product IMDP model which considers dynamical constraints from a PIPM bipedal robot framework. Additionally, the product IMDP relies on GP learning of uncertainties for safe planning with respect to a LTL specification. Control actions from the IMDP planner are input to the PIPM dynamics, where they are mapped to low-level control actions for the physical robot. Trajectory data from the robot executing the control policy is fed back into the stacked GP model, updating the transition probabilities in the product IMDP. Black arrows/text illustrate the flow of information through the system and green text shows the corresponding parameters.**

A. IMDP State Definition

We first define the IMDP abstraction states:

Definition 7 (IMDP Abstraction States):
An IMDP abstraction state $\{X_q\}_{q\in Q}$ is defined as

$$X_q = \{r, \theta, \psi | r \in R_q, a_{\theta,q} \leq \theta < b_{\theta,q}, \psi = \psi_q\} \quad (5)$$

where R_q is a rectangular partition region, $a_{\theta,q}$ and $b_{\theta,q}$ are lower and upper bounds, respectively, on the yaw angle, and $\psi_q \in \{\text{left}, \text{right}\}$ is the left or right foot stance index.

Given the full-body kinematic constraints of the robot, we assume that forward and backward steps are feasible, with a limited turning motion possible in the forward direction. Formally, we have the action set

$$A = \{N, NE, NW, S\}, \quad (6)$$

where N corresponds to a straight forward step, NE is a forward step with a right turn, NW is a forward step with a left turn, and S is a straight backwards step.

We use a high-level waypoint planner δ to map IMDP actions to target IMDP states. We first perform an intermediate mapping from IMDP actions to high-level keyframe state actions $\boldsymbol{a}_{\text{HL}} = (d_\alpha, \Delta\theta_\alpha, \delta z_{\text{foot}})$ as in Definition 2. Then, we map high-level keyframe state actions to their corresponding target IMDP states as follows. Given an initial IMDP state X_q with a geometric center $x_{c,q}, y_{c,q}$, a mean yaw angle $\theta_{c,q}$, and an action $\alpha \in A$, we target the center of the IMDP state X_q^* closest to the desired endpoint, which is calculated as

$$X_{q^*} = \min_{q'\in Q}((x_{q'} - x_q + x_\alpha)^2 + (y_{q'} - y_q + y_\alpha)^2 + (\theta_{c,q'} - \theta_{c,q} + \theta_\alpha)^2)^{1/2}, \quad (7)$$

where $x_\alpha = d_\alpha \cos(\Delta\theta_\alpha + \theta_{c,q})$ and $y_\alpha = d_\alpha \sin(\Delta\theta_\alpha + \theta_{c,q})$ are the x and y components of the desired step, respectively. Once each state-action pair has been associated with a target state during planning time, we implement a controller to execute the appropriate trajectory at runtime.

B. Gaussian Process Formulation

Inspired by stacked Gaussian process learning [58], a hierarchical framework in which output parameters of lower-level GPs are used as inputs to higher-level GPs, we propose a multi-layer GP learning framework to model sources of bipedal locomotion uncertainties.

We first use a Gaussian process $\hat{z}(x, y)$ to model the elevation z of the terrain, which takes as input the x- and y-coordinates of the position. We assume that at each step the robot can measure the terrain elevation at its current CoM position. Therefore, at a given step i we can record the $x-y$ position (x_i, y_i) and the observation z_i.

Next, we model uncertainty in the yaw angle due to model error at each step using Gaussian processes. We first introduce a Gaussian process $\Delta\hat{\theta}(\boldsymbol{a}_{\text{HL}})$, which takes as input the commanded high-level action $\boldsymbol{a}_{\text{HL}} = (d, \Delta\theta, \Delta z_{\text{foot}})$ and outputs a mean expected deviation $\mu_{\Delta\hat{\theta}}$ and a variance $\sigma^2_{\Delta\hat{\theta}}$ from the commanded yaw change. This Gaussian process models the error induced by the low-level dynamics which are not accounted for in the reduced-order PIPM model.

Then, we model the overall expected variance of the yaw angle deviation using another Gaussian process $\hat{\sigma}^2_{\Delta\hat{\theta}}$, which takes as inputs the variance of the terrain uncertainty of the step $\sigma^2_{\hat{z}}$ and the variance of the predicted yaw deviation $\sigma^2_{\Delta\hat{\theta}}$. Intuitively, this Gaussian process captures the range of potential yaw angle deviations from a step, which should increase when there exists more uncertainty in the Gaussian processes used to predict the terrain elevation and yaw angle deviation. Figure 2 shows the structure of the GPs.

Samples for each of the GPs are collected as follows. At each step, the robot collects samples of the terrain elevation $\{(x^+_{\text{foot}}, y^+_{\text{foot}}), (z^+_{\text{foot}})\}$, where $(x^+_{\text{foot}}, y^+_{\text{foot}})$ is the $x-y$ foot stance after the step and (z^+_{foot}) is the corresponding observation of the terrain elevation. This data point is then added to the sample set of the GP $\hat{z}(x, y)$.

The robot has a target yaw angle $\theta_{c,q'}$ but due to estimation errors and motion perturbation actually reaches a yaw angle of θ^+ after each step. We define a yaw angle error

$$\theta_e = \theta^+ - \theta_{c,q'} \tag{8}$$

and use this error as observations for our remaining GPs as follows. First, we generate samples $(\boldsymbol{a}_{\mathrm{HL}}, \theta_e)$ for the model error Gaussian process $\Delta\hat{\theta}$, where $\boldsymbol{a}_{\mathrm{HL}}$ is the high-level control action used for the step, calculated using the controller in Subsection IV.C. Next, we generate samples $(\sigma^2_{\hat{z}}, \sigma^2_{\Delta\hat{\theta}}, |\theta_e|)$ for the yaw error Gaussian process $\hat{\sigma}^2_{\Delta\hat{\theta}}$ where the inputs are the variances of the terrain and model error GPs used for a step and the value to be predicted is the absolute value of the yaw deviation. Our motivation for using the absolute value of the yaw error is that this captures the total error induced by the model error, terrain uncertainty, and noise, whereas the yaw error itself will not capture the cumulative effect of the the zero-mean stochastic noise.

C. Controller Design

Define a family of controllers $K_q : Z \times X \to \mathcal{A}_{\mathrm{HL}}$ which take as input a current pose $\zeta = (x_{\mathrm{com}}, y_{\mathrm{com}}, z_{\mathrm{foot}}, \theta, \psi) \in Z$. Additionally, we have a target IMDP state $X_{q'}$ with center $(x_{c,q'}, y_{c,q'}, \theta_{c,q'})$ as a second input. The controller then outputs a high-level waypoint-targeting control action $\boldsymbol{a}_{\mathrm{HL}} = (d, \Delta\theta, \Delta z_{\mathrm{foot}})$ as shown in Figure 3.
First, the controller calculates a desired step length d

$$d = \sqrt{(x_{c,q'} - x_{\mathrm{com}})^2 + (y_{c,q'} - y_{\mathrm{com}})^2}. \tag{9}$$

The foot elevation change Δz_{foot} is calculated as

$$\Delta z_{\mathrm{foot}} = \mu_{\hat{z}}(x^+, y^+) - z_{\mathrm{foot}}, \tag{10}$$

where (x^+, y^+) is the desired foot placement which is analytically calculated online at each step, and $\mu_{\hat{z}}$ is the mean of the GP prediction of the foot placement elevation. The yaw angle change $\Delta\theta$ is calculated using the GP prediction $\Delta\hat{\theta}(\boldsymbol{a}_{\mathrm{HL}})$ of the expected yaw angle deviation

$$\Delta\theta = \theta_{c,q'} - \theta - \mu_{\Delta\hat{\theta}}(d, \theta_{c,q'} - \theta, \Delta z_{\mathrm{foot}}), \tag{11}$$

where $\mu_{\Delta\hat{\theta}}$ is the GP estimate of the yaw angle deviation due to model error.

D. IMDP Transition Probability Calculation

We next formulate transition probability bounds for the IMDP abstraction states. Given the controller developed in Subsection IV.C, we separate the IMDP state transition into three components. First, we assume that the rectangular partition region transition is deterministic; that is, given current IMDP state X_q with rectangular partition region R_q and an action which targets IMDP state X_{q^*} with rectangular partition region R_{q^*}, we assume that the actual next IMDP state $X_{q'}$ has rectangular partition region $R_{q'} = R_{q^*}$. Additionally, we assume that the foot stance of the robot alternates between IMDP states, *i.e.* if the current IMDP state X_q has stance $\psi_q = \text{left}$, the next IMDP state $X_{q'}$ always has stance $\psi_{q'} = \text{right}$ and vice versa. Then, the uncertainty in IMDP state transitions arises solely from

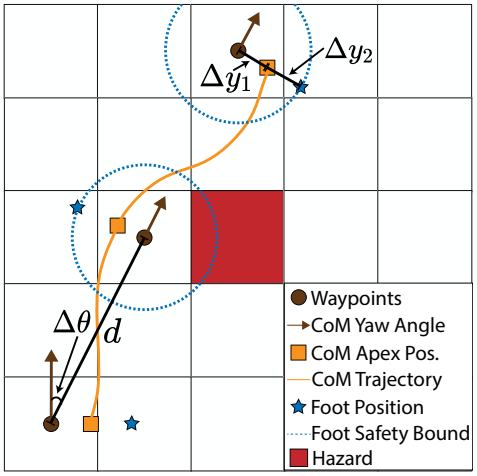


FIGURE 3. Illustration of bipedal robot motion planning. The high-level action components $d, \Delta\theta$ are planned with respect to waypoints (brown dots). The high-level command Δz depends on the elevation change between successive foot placements (blue stars). However, the actual trajectory (orange) of the robot's center of mass follows a sinusoidal curve around the desired waypoints. Additionally, there exists lateral motion deviation at each step. We assume that the lateral deviation between the waypoint and robot apex CoM can be bounded by a parameter $\Delta y_{1,\max}$, and the lateral deviation between the apex CoM and the foot position can be bounded by $\Delta y_{2,\max}$. Thus, we can guarantee that the CoM apex position stays within the target region if the side length of the regions is at least $2\Delta y_{1,\max}$. Similarly, safety with respect to foot positions can be guaranteed for an action if there is no hazard region within a circle of radius $\Delta y_{1,\max} + \Delta y_{2,\max}$ from the center of the target region (blue circle). These conditions will be proved in Theorem 2.

yaw uncertainty. We assume that this yaw uncertainty has two components. First, uncertainty in the GP estimations of the terrain elevation and reduced-order model error induce learnable controller error. We also assume a nonlearnable disturbance modeled as stochastic noise.

Given current IMDP state X_q and an action which targets IMDP state X_{q^*}, we model the yaw uncertainty induced by GP estimation error using the GP $\hat{\sigma}^2_{\Delta\hat{\theta}}$. We first calculate a maximum terrain GP variance $\sigma^2_{\hat{z},\max}$ and a maximum model GP variance $\sigma^2_{\Delta\hat{\theta},\max}$ for the state-action pair as follows. We can conservatively estimate the maximum terrain variance $\sigma^2_{\hat{z},\max}$ under the action as

$$\sigma^2_{\hat{z},\max} = \max_{(x,y)\in R_{q^*}} \sigma^2_{\hat{z}}(x, y) \tag{12}$$

where $\sigma^2_{\hat{z}}(x, y)$ is the variance of the GP prediction of the terrain at a given $x - y$ location.

We now define a series of parameters in order to calculate the maximum model error GP variance. First, since the controller always seeks to move to the center $x_{c,q'}$ of the target IMDP region $X_{q'}$, lower and upper bounds on the step length input to the model error GP are

$$d_{\min} = \min_{(x,y)\in R_q} \sqrt{(x_{c,q'} - x)^2 + (y_{c,q'} - y)^2} \tag{13}$$

$$d_{\max} = \max_{(x,y)\in R_q} \sqrt{(x_{c,q'} - x)^2 + (y_{c,q'} - y)^2} \tag{14}$$

which are the distances of the closest and furthest points, respectively, in X_q to the center of $X_{q'}$.

Next, we define lower and upper bounds on the yaw angle change inputs to the model error GP as

$$\Delta\theta_{\min} = \min_{a_{\theta,q} \le \theta < b_{\theta,q}} ||\theta_{c,q'} - \theta|| \tag{15}$$

$$\Delta\theta_{\max} = \max_{a_{\theta,q} \le \theta < b_{\theta,q}} ||\theta_{c,q'} - \theta|| \tag{16}$$

where $a_{\theta,q}$ and $b_{\theta,q}$ are lower and upper bounds on the yaw angles in state X_q, and $\theta_{c,q'}$ is the central yaw for state $X_{q'}$. Finally, we define lower and upper bounds on the foot elevation change input to the model error GP as

$$\Delta z_{\text{foot,min}} = \min_{x,y \in R_q} \min_{x',y' \in R_{q'}} ||\hat{z}(x',y') - \hat{z}(x,y)|| \tag{17}$$

$$\Delta z_{\text{foot,max}} = \max_{x,y \in R_q} \max_{x',y' \in R_{q'}} ||\hat{z}(x',y') - \hat{z}(x,y)|| \tag{18}$$

where $\hat{z}(x,y)$ is the mean of the GP prediction of the terrain elevation at point (x,y).
We now calculate the maximum model error GP variance as

$$\sigma^2_{\Delta\hat{\theta},\max} = \max_{d_{\min} \le d \le d_{\max}} \max_{\Delta\theta_{\min} \le \Delta\theta \le \Delta\theta_{\max}} \max_{\Delta z_{\min} \le \Delta z \le \Delta z_{\max}} \sigma^2_{\Delta\hat{\theta}}(d, \Delta\theta, \Delta z) \tag{19}$$

Given maximum terrain and model error GP uncertainties, we then take the corresponding mean prediction of the yaw uncertainty GP $\gamma = \mu_{\hat{\sigma}^2 \Delta\hat{\theta}}(\sigma^2_{\hat{z},\max}, \sigma^2_{\Delta\hat{\theta},\max})$ under these parameters to be a measure of uncertainty in the yaw angle for each state-action pair. We can then bound the yaw angle θ^+ after the action using γ:

$$\theta_{c,q'} - \beta\gamma \le \theta^+ \le \theta_{c,q'} + \beta\gamma, \tag{20}$$

where $\theta_{c,q'}$ is the desired yaw angle after the action is taken and β is a high-confidence interval factor calculated using methods in [43]. We note that although the yaw angle measurements we use to train the GPs are perturbed by the stochastic noise as in Assumption 2, it is established in [57] that we can place high-confidence bounds on the underlying learnable function as in Equation (20). The effect of the noise is to increase the magnitude of the high-confidence interval factor β, thus making the bounds on the underlying function more conservative. However, our approach assumes *a priori* conservative bounds to initialize the GPs, so the calculated bounds always remain conservative and thus the validity of the overall approach is not affected by noise. Practically, the noise perturbation results in the robot having to collect more data samples to generate sufficiently tight bounds.

We can then calculate lower and upper transition probability bounds for the IMDP state-action pair by treating the high-confidence transition region as nondeterministic and centering the stochastic noise at probability minimizing and maximizing points in this region:

Lemma 1 (Yaw-Based Transition Probability Bounds): Consider $X_q, X_{q'}; q, q' \in Q$ and action $\alpha_{q^*} \in A_q$. Lower bound $\check{T}$ and upper bound $\hat{T}$ transition probabilities from X_q to $X_{q'}$ under α_{q^*} are given by

$$\check{T}(X_q, \alpha_{q^*}, X_{q'}) = \int_{a_{\theta,q'}}^{b_{\theta,q'}} \rho_\nu(z - \theta_{\min}(X_q, \alpha_{q^*}, X_{q'}))dz, \tag{21}$$

$$\hat{T}(X_q, \alpha_{q^*}, X_{q'}) = \int_{a_{\theta,q'}}^{b_{\theta,q'}} \rho_\nu(z - \theta_{\max}(X_q, \alpha_{q^*}, X_{q'}))dz, \tag{22}$$

where we recall ρ_{ν_i} is the probability density function of the stochastic noise ν_i and $a_{\theta,q'}$ and $b_{\theta,q'}$ are the lower and upper boundary points for region $X_{q'}$. We define $\theta_{\min}$ and $\theta_{\max}$ as

$$\theta_{\min}(X_q, \alpha_{q^*}, X_{q'}) = \operatorname*{argmax}_{\theta} \theta - \theta_{c,q'} \quad \text{s.t. } \theta_{c,q^*} - \beta\gamma(q) \le \theta \le \theta_{c,q^*} + \beta\gamma(q), \tag{23}$$

$$\theta_{\max}(X_q, \alpha_{q^*}, X_{q'}) = \operatorname*{argmin}_{\theta} \theta - \theta_{c,q'} \quad \text{s.t. } \theta_{c,q^*} - \beta\gamma(q) \le \theta \le \theta_{c,q^*} + \beta\gamma(q), \tag{24}$$

Proof:
The proof for the validity of these transition probability bounds follows from the proof outlined in Theorem 1, [49], as a special one-dimensional case with yaw angles rather than Cartesian coordinates considered. ■

Figure 1 illustrates the nondeterministic transition regions and the stochastic noise centering used in Lemma 1.

E. IMDP for Bipedal Locomotion Definition

With the individual components of the IMDP defined along with the integration of a stacked GP framework, we can now formally define an IMDP abstraction of bipedal locomotion:

Definition 8 (IMDP for Bipedal Locomotion): Consider a bipedal robot modeled according to the PIPM dynamics in 1. An IMDP abstraction of bipedal locomotion (IMDP-BL) modeled using the dynamics in (1) is an IMDP $\mathcal{I} = (Q, A, \check{T}, \hat{T}, Q_0, O, L)$ which satisfies that

- The states of the IMDP are defined as in Definition 7,
- The actions of the IMDP are defined according to Equation 6,
- The lower and upper transition probability bounds are calculated according to Lemma 1.

The IMDP-BL abstracts the dynamics of bipedal robots and also incorporates multi-component GP learning of uncertainties, solving Subproblem 1.1. We note that the reduced-order bipedal PIPM dynamics in (1) are outside of the class of dynamics considered in the previous work [49] for the IMDP-based methodology. Additionally, the IMDP-BL explicitly accounts for low-level full-body kinematics unique to bipedal locomotion such that the proposed IMDP planning method can be realistically implemented on bipedal robots.

Figure 2 shows the structure of the proposed IMDP-BL framework with interconnections between the high-level planning, low-level dynamics, and GP learning components.

V. Control Policy Synthesis

We now turn our attention to the verification and synthesis of control policies for the IMDP-BL against LTL specifications. The structure of our methodology broadly follows our previous work in [49], in which we constructed similar control synthesis techniques for the syntactically cosafe fragment of LTL (scLTL). In this section, we introduce generalized algorithms which are able to accommodate complex specifications from the general class of LTL.

A. Product IMDP

To check LTL specifications against observations from an IMDP abstraction of a bipedal robot, we utilize a product IMDP construction synthesized from the IMDP-BL and a DRA of the LTL specifications, defined as follows:

Definition 9 (PIMDP):
Let $\mathcal{I} = (Q, A, \check{T}, \hat{T}, Q_0, O, L)$ be an IMDP-BL and $\mathcal{A} = (S, s_0, O, \delta, F)$ be an DRA. 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
- $F' = Q \times F = \{Q \times (G_1, B_1), \cdots, Q \times (G_n, B_n)\}$, where $G_i, B_i \subseteq S$ is the ith acceptance condition.

The PIMDP construction encodes both the abstracted dynamics of the robot as well as specification satisfaction.

For LTL specifications, [59, Def. 10.128, Thm. 10.129] show that specification checking on a PIMDP structure reduces to checking reachability of accepting maximal end components, defined as follows:

Definition 10 (End Component [60]):
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 induced by $(\mathcal{T}, Act)$ is strongly connected.

We note that upper bound transition probabilities are used to calculate end components in order to enable all possible edges in the PIMDP. This ensures that the end components are robust to all realizations of the transition probabilities.

Definition 11 (Maximal End Component (MEC) [60]):
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 $(q,s) \in \mathcal{T}$.

Maximal end components are pairwise disjoint, so the number of MECs is bounded above by the number of PIMDP states, thus reducing the computational complexity of specification checking. In order to use graph-theoretic techniques to calculate reachability probabilities, it is necessary to resolve the transition probability bounds in the PIMDP to reduce it to a product MDP. We define adversaries for this purpose:

Definition 12 (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

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

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.

Definition 13 (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.

Then, reachability probabilities in the PIMDP are uniquely determined by the control policy and adversary selected.

B. Graph Pruning

In order to enable online learning of the uncertainties in the system, we require paths through the state-space which the legged robot can take in order to collect samples of the dynamics without violating the specifications of interest. The key idea is that verification of LTL specifications reduces to checking reachability of accepting MECs. Then, states which have zero probability of reaching any accepting end component under a maximizing control policy and maximizing adversary are unsafe, *i.e.* the robot is guaranteed to violate the specification at these states. On the other hand, states which have zero probability of reaching an accepting MEC under a maximizing control policy and minimizing adversary but have nonzero probability under a maximizing adversary are considered safe to sample, as these are states which could benefit most from uncertainty reduction.

Algorithm 1 details our methodology for graph pruning in order to generate a nonviolating sub-graph to sample. First, we assume that MECs corresponding to accepting conditions in the product IMDP have been calculated. The MECs themselves can be calculated using standard algorithms [59, Algorithm 47]. Then, accepting MECs are simply those which contain states in an accepting condition G_i but no states from the corresponding rejecting set B_i. In line 1, we initialize the accepting states of the PIMDP to be those in the union of all of the accepting MECs. In lines 4–14, we prune any state-action pairs which have zero probability of reaching the accepting states under a maximizing adversary.

Algorithm 1: Nonviolating PIMDP Generation

Input: PIMDP $\mathcal{P}$, Accepting MECs $(\mathcal{T}^*, Act^*)$
Output: Nonviolating subset $\mathcal{P}$' of $\mathcal{P}$

1 **Initialize** $G = \{(q,s) \in \mathcal{T}^*\}, R = \{\emptyset\}, U = \{\emptyset\}$;
2 **for** $(q,s) \in \mathcal{P} \setminus G$ **do**
3 **for** $\alpha \in A((q,s))$ **do**
4 **if** $\hat{T}((q,s),\alpha,G) = 0$ **then**
5 Remove α from available actions at (q,s);
6 **end**
7 **end**
8 **if** $A((q,s)) = \emptyset$ **then**
9 $R = R \cup (q,s), U = U \cup (q,s)$;
10 **end**
11 **end**
12 **while** $R \neq \emptyset$ **do**
13 **for** $(q,s) \in \mathcal{P} \setminus U$ **do**
14 **for** $\alpha \in A$ **do**
15 **if** $\hat{T}((q,s),\alpha,U) \neq 0$ **then**
16 Prune action α from state (q,s);
17 **end**
18 **end**
19 **end**
20 $R = \emptyset$;
21 **for** $(q,s) \in \mathcal{P} \setminus U$ **do**
22 **if** $A((q,s)) = \emptyset$ **then**
23 $R = R \cup (q,s), U = U \cup (q,s)$;
24 **end**
25 **end**
26 **end**
27 **return** $\mathcal{P}' = \mathcal{P} \setminus U$

If this pruning leaves any state with no available actions, we add this state to a set of hazard states to be excluded from the nonviolating sub-graph. In lines 15–28, we prune state-action pairs which have nonzero probability under a maximizing adversary to reach any hazard state. Finally, in line 31 we return the remaining state-action pairs as a nonviolating sub-graph.

Once a nonviolating sub-graph has been generated, a control policy to safely sample the state-space can be synthesized using the method detailed in Algorithm 2. For states which are in the nonviolating sub-graph (lines 2–4), we randomly select from the available actions for each state. For states which are outside of the nonviolating subgraph (lines 5–7), we select the action which produces the highest probability under a minimizing adversary of reaching a state in the nonviolating subgraph.

Theorem 1 (Safety of Nonviolating Subgraph): The control policy synthesized using Algorithm 2 from the nonviolating sub-graph output of Algorithm 1 enforces a nonviolating motion plan for a bipedal robot modeled using the IMDP-BL methodology in Section IV.

Algorithm 2: Nonviolating Control Policy Synthesis

Input: PIMDP $\mathcal{P}$, Nonviolating subgraph $\mathcal{P}'$ output by Algorithm 1
Output: Nonviolating control policy π'

1 **Initialize** $G = \{(q,s,a) \in \mathcal{P}'\}$;
2 **for** $\{(q,s,a) \in G\}$ **do**
3 $\pi'(q,s) = \text{random}\{a | (q,s,a) \in G\}$;
4 **end**
5 **for** $\{(q,s) | (q,s,a) \notin G \ \forall a \in A\}$ **do**
6 $\pi'(q,s) = \text{argmax}_{a \in A} \max_{(q',s') \in G} \check{T}((q,s),\alpha,(q',s'))$;
7 **end**
8 **return** π'

Proof:
By construction, Algorithm 1 produces as output a subgraph $\mathcal{P}'$ in which every state has nonzero probability of reaching an accepting maximal end component of the specifications. Therefore, executing a control policy generated by Algorithm 2, which performs control policy synthesis using the nonviolating sub-graph from Algorithm 1, ensures that the robot will always traverse states which have nonzero probability of satisfying the LTL specifications. Thus, the robot can traverse the state space using this control policy indefinitely while remaining in nonviolating regions, proving the safety of the control policy generated from Algorithms 1 and 2. ■

Therefore, executing Algorithms 1 and 2 on a bipedal PIMDP produces a control policy which allows the robot to traverse its state space indefinitely while guaranteeing safety with respect to LTL specifications, solving Subproblem 1.2.

VI. Bipedal Abstraction-based Planning

A. Task Planning Algorithm

Given the IMDP-BL model developed in Section IV and the LTL control policy synthesis framework proposed in Section V, we can now formulate a complete algorithm for bipedal task planning for safe online learning with respect to complex LTL specifications, detailed in Algorithm 3.

Given a biped with PIPM dynamics and LTL specification ϕ with desired probability of satisfaction P_{sat}, we first construct an IMDP-BL of the bipedal dynamics (initializing the GP models of the system and environmental uncertainties with conservative bounds) and a DRA of the LTL specification ϕ. We then construct a PIMDP and calculate minimum and maximum probabilities of satisfaction from the initial state (lines 1–3). If the initial minimum probability of satisfaction is too small (line 4), we next check the maximal probability of satisfaction. If this probability is less than P_{sat}, we know that the LTL specification cannot be satisfied with sufficient probability regardless of how well we learn the uncertainties, so the algorithm is terminated (lines 5–7). Otherwise, we use Algorithm 1 and Algorithm 2 to calculate and traverse a nonviolating PIMDP, collecting

Algorithm 3: Synthesis Algorithm

Input: Bipedal PIPM (1), LTL specification ϕ, P_{sat}
Output: Satisfying control policy $\pi^{\dagger}$

1 **Construct** IMDP-BL $\mathcal{I}$ from System (1), DRA $\mathcal{R}$ from ϕ, PIMDP $\mathcal{P}$ from $\mathcal{I}$ and $\mathcal{R}$, initial GP $\hat{g}(x)$;
2 Calculate $\check{P}_{\max}((q_0, \delta(q_0, s_0)) \models \phi)$ for initial state;
3 Calculate $\hat{P}_{\max}((q_0, \delta(q_0, s_0)) \models \phi)$ for initial state;
4 **while** *(*$\check{P}_{\max} < P_{\text{sat}}$*)* ***and*** *(*Count< MaxBatches*)* **do**
5 **if** $\hat{P}_{\max} < P_{\text{sat}}$ **then**
6 **return** *No satisfying control policy exists;*
7 **end**
8 Find nonviolating PIMDP $\mathcal{P}'$ using Algorithm 1;
9 Find MEC to cycle in with corresponding control policy π^* using Algorithm 2;
10 **for** *NumSteps* **do**
11 Take action $\pi^*(q)$ at current state q;
12 Sample terrain z_{foot}, yaw error θ_e;
13 **end**
14 Reconstruct GPs using collected samples;
15 Recalculate transition probability intervals for $\mathcal{P}$;
16 Recalculate $\check{P}_{\max}((q_0, \delta(q_0, s_0)) \models \phi)$;
17 Recalculate $\hat{P}_{\max}((q_0, \delta(q_0, s_0)) \models \phi)$;
18 **end**
19 **if** $\check{P}_{\max} \geq P_{\text{sat}}$ **then**
20 **return** *Control policy*
 $\pi^{\dagger} = \underset{\alpha \in A(q)}{\text{argmax}}\ \check{P}_{\max}((q, s) \models \phi)\ \forall (q, s) \in \mathcal{P}$;
21 **end**

a batch of data on the system dynamics for GP retraining (lines 8–14). Finally, we recalculate the PIMDP transition probabilities based on the new GPs (lines 16–18) and recheck against the desired probability of satisfaction. If we have reached the desired threshold, we output a final control policy which maximizes the probability of satisfying the specification under a minimizing adversary from all states (lines 19–20); else, we repeat the sampling process. The algorithm terminates once a satisfying control policy is found, the nonexistence of a satisfying control policy has been proved, or a maximum number of batches has been reached. Algorithm 3 provides a complete framework for the bipedal task planning problem with online learning of uncertainties, solving Problem 1.

B. Locomotion-Specific Modeling Considerations

In order to apply our planning method to bipedal robots, additional leg kinematic considerations (*e.g.*, full-body kinematics) must be taken. Thus, we must solve the problem of converting high-level waypoints produced by the PIMDP task planner into motion plans via the PIPM. Using an optimization-based approach as in [51, Algorithm 1], we first solve for the PIPM keyframe state $v_{\text{apex}}, z_{\text{apex}}$ in the motion planner (see Definition 2). This gives us middle-level waypoints from which we can interpolate a trajectory of robot's CoM motion. These middle-level waypoints can then be translated into low-level joint motion plans.

We next map the low-level kinematic constraints associated with legged locomotion to the high-level PIMDP task planner. First, we consider the problem of lateral CoM motion deviation. The nonperiodic gaits we design result in sinusoidal center-of-mass trajectories, *i.e.*, the CoM motion wiggles between the desired CoM waypoint and the actual foot contact position. To ensure accurate waypoint tracking, we follow the results in [51, Prop. V.1] and assume a bound $\Delta y_{1,\max}$ on the maximum lateral deviation between the target waypoint and the CoM apex position. Similarly, we assume a bound $\Delta y_{2,\max}$ on the maximum lateral deviation between the CoM and foot apex positions. Figure 3 illustrates these parameters. These global bounds can be enforced with appropriate selection of v_{apex} ([51, Algorithm 1]) and a restriction of the possible set of motions (which we do by restricting the set of IMDP-BL actions to be finite).

Second, we need to enforce CoM and foot position constraints in order to ensure the validity of our high-level PIMDP planner with respect to low-level full-body kinematic constraints. Our IMDP planner assumes that the robot CoM always reaches the desired $x - y$ region at each walking step. Furthermore, although high-level planning focuses on CoM positions, we also want to ensure that the robot avoids stepping in unsafe regions. We can enforce the CoM constraint with the appropriate selection of the cell length l of the $x-y$ grid regions, and we can obtain safety guarantees with respect to foot positions via graph pruning:

Theorem 2 (Safe Low-Level PIMDP Planning): Given CoM lateral apex deviation bounds $\Delta y_{1,\max}, \Delta y_{2,\max}$ as in Subsection VI.B, geometrically safe planning with respect to CoM apex positions is enforced if the grid region cell length satisfies $l > 2\Delta y_{1,\max}$. Additionally, safety with respect to foot positions (*i.e.*, no foot lands in a violating region) is enforced by pruning PIMDP states for which a violating $x - y$ grid region lies within a circle of radius $\Delta y_{1,\max} + \Delta y_{2,\max}$.

Proof:
First, we prove the CoM safety condition. Since the PIMDP controller (Section IV.C) produces discrete actions which always target the center of a 2D grid, the robot's CoM stays within the target grid region at each step under a CoM deviation of at most $\Delta y_{1,\max}$ if the distance from the center of each region to its boundary is at least $\Delta y_{1,\max}$ in all directions. For a hyperrectangular region, this is accomplished if each cell of the region has a side length greater than or equal to $2\Delta y_{1,\max}$. Note that since the target point under each step is always the center of a region, lateral deviations do not accumulate over multiple steps.
We now prove the condition of safety with respect to foot locations. We know that the maximum deviation of the foot position from the center of a targeted 2D grid region is

bounded by $\Delta y_{1,\max} + \Delta y_{2,\max}$. Therefore, the range of potential foot locations is bounded by a circle with radius $\Delta y_{1,\max} + \Delta y_{2,\max}$. If the circle for a PIMDP state intersects with a violating region, the PIMDP state is violating. Then, by including these new violating states in the graph pruning method detailed in Algorithm 1, we can again use Theorem 1 to guarantee that the robot will never step in a violating region. Figure 3 illustrates these safety conditions. ■

Theorem 2 can be applied to guarantee safety of the PIMDP planner in the presence of low-level full-body kinematic constraints and in particular to the special case of the backward step action in the IMDP-BL (i.e., action "S" in Equation 6). Physically, this motion is executed through a "turn-in-place" sequence wherein the robot performs a predefined series of steps in place to change its yaw angle 180° and then take a forward step to the desired location. Extending the arguments in Theorem 2, the safety of this "turn-in-place" sequence corresponds to the condition that there exists no violating region within radius $\Delta y_{1,\max} + \Delta y_{2,\max}$ of the region from which the turn will be initiated, since each step during the turn has a target waypoint corresponding to the center of the initial region with an increment only in the yaw angle.

An alternative approach for obtaining safety guarantees while considering low-level legged kinematic constraints would be to discretize the 2D grid environment in a coarser fashion, allowing for the foot positions to always remain in the same region as the robot CoM. This would also necessitate the use of IMDP-BL states with a longer sequence of actions corresponding to, *e.g.*, predefined multi-step sequences. However, we choose to focus on scenarios with fine grid-world discretizations in order to showcase the level of precision available with our proposed approach and to illustrate our algorithms on an IMDP-BL structure that explicitly accounts for legged kinematic constraints.

VII. Case Study

We demonstrate our methodologies utilizing the Digit bipedal robot from Agility Robotics [4]. The Digit robot weighs 45 kg and has 16 degrees of freedom: three joints in each arm (shoulder roll and pitch, elbow), and five in each leg (hip roll, pitch, and yaw, knee, and ankle).

We consider an environment discretized as a 10x10 grid region as shown in Figure 4, with each square region having a side length of 0.2 meters. Each region is divided into 24 yaw angle bins which are 15° wide. For each grid region and yaw angle bin combination, there exists two IMDP-BL states: one for a left-foot contact state, and one for a right-foot contact state. At each IMDP-BL state, the robot has up to four potential actions. First, the robot can walk forward along its current yaw 0.4 meters (two regions forward) or turn around and walk backward 0.4 meters (two regions backward). Additionally, in left-foot states the robot can turn right 15° and step 0.4 meters (two regions forward and one region right), and in right-foot states the robot can turn left

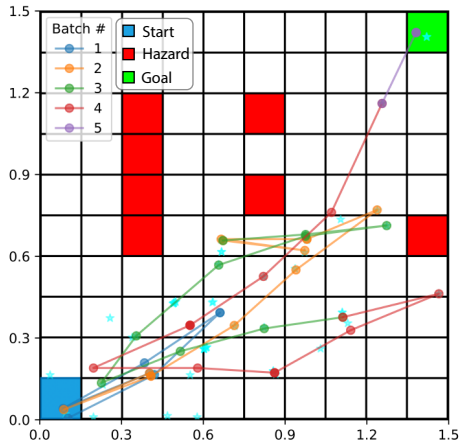


FIGURE 4. Illustration of the first case study results. The objective is to avoid the red hazard states and reach the green goal state. The robot starts at the bottom-left blue state and walks along the lower half of the environment, sampling the terrain elevation and model uncertainty until it is confident enough to traverse further up the environment and eventually reach the goal. The cyan stars correspond to footsteps, and each color of dots/trajectories corresponds to a sampling batch.

15° and step 0.4 m (two regions forward and one region left). In all cases, the discrete IMDP-BL action targets the center of the IMDP-BL state which is closest to the target point under these parameters. Finally, actions which would lead outside the boundaries of the environment are removed. Note that this formulation, which accounts for bipedal legged kinematic constraints, restricts the motion trajectories of the robot considerably as compared to, *e.g.*, a mobile robot which can target any adjacent state during grid cell transition.

We assume that the terrain elevation varies continuously within $[-0.1, 0.1]$m in the environment, and construct a yaw perturbation for each step which is a nonlinear function of the commanded action as well as the terrain elevation error between the robot's estimation and ground truth.

A. Reach-Avoid Case Study

First, we consider a reach-avoid specification wherein the robot must reach a goal state while avoiding hazard states, written as the LTL specification

$$\phi_1 = \neg \texttt{Haz}\ \mathcal{U}\ \texttt{Goal}. \tag{25}$$

To illustrate the effectiveness of our approach under highly-perturbed conditions, we assume *a priori* that the learnable motion perturbations can cause the robot to deviate up to 15° in yaw angle from its target during each step, along with up to an additional 10° of perturbation due to bounded stochastic noise, which we take as a zero mean, $\sigma_\nu = 2.5^\circ$ standard deviation Gaussian truncated at $\pm 10^\circ = 4\sigma_\nu$. This gives an initial probability of satisfaction of 0.18.

Figure 4 shows the results of the case study simulation. The robot initially cannot safely traverse through the obstacles so it traverses the lower half of the environment, training its GPs until it gains enough confidence to reach the upper half of the environment. From there, after collecting more samples, the probability of satisfaction $\check{P}_{\max}$ exceeds

the desired threshold $P_{\text{sat}} = 0.99$, and the robot is able to successfully reach the goal.

The simulation was run using Georgia Tech's PACE computing infrastructure [61]. Using a 24-core Intel Xeon Gold 6226 CPU and a RTX-6000 GPU with 32 GB of memory, the complete algorithm took 24 minutes 28 seconds to run, comprised of four iterations of the batch sampling. The initial sampling batch had four steps to allow the robot to gain a baseline understanding of the environment, and subsequent batches consisted of eight steps to gather more data.

B. Intermediate Persistence Case Study

Next, we consider a more complex scenario to demonstrate the expressivity of the specifications considered. The specification is now for the robot to reach a goal state while avoiding hazards and also returning to a set of "checkpoint" regions at least once every three steps along its trajectories. Written as a LTL specification, this task becomes

$$\phi_2 = \neg \texttt{Haz}\ \mathcal{U}\ \texttt{Goal} \wedge \neg(\neg \texttt{Checkpoint} \wedge \bigcirc \neg \texttt{Checkpoint} \wedge \bigcirc \bigcirc \neg \texttt{Checkpoint}) \tag{26}$$

Note that this specification effectively quadruples the size of the PIMDP state space from 4800 states in the first case study to 19200 states. In this case study, the number of hazard states is reduced in order to maintain feasibility of the problem under the checkpoint constraint; otherwise, the parameters are the same as in the first case study.

Figure 5 shows the simulation results for this case study. In this case, the trajectory of the robot is noticeably altered from the first case study in order to favor "checkpoint" states although the general structure of the environment remains similar. As in the first case study, the robot traverses in the safer lower region (with an initial probability of satisfying the specification of 0.46) before collecting enough data to move upwards and eventually reach the goal.

Although the hazard set has been reduced and there exist many checkpoint states, the low-level legged kinematic constraints on the robot's motion make this problem challenging to solve. In particular, it is possible for the robot to enter a non-checkpoint region in a configuration such that it cannot reach a checkpoint region within the next two steps, even if there exist such regions adjacent to the robot's position. Because the PIMDP planner performs infinite-horizon control policy synthesis, states with these nonobvious unsafe configurations are pruned, enforcing safe robot trajectories. Thus, this case study illustrates both the planning capabilities of our abstraction-based approach as well as the unique challenges inherent to legged locomotion systems as compared to, *e.g.*, mobile robots. The complete simulation run took 10 hours 11 minutes to run, with four batches and 18 robot steps per batch. Thus, the curse of dimensionality with regards to PIMDP state space size is clearly illustrated, which will be addressed in future works.

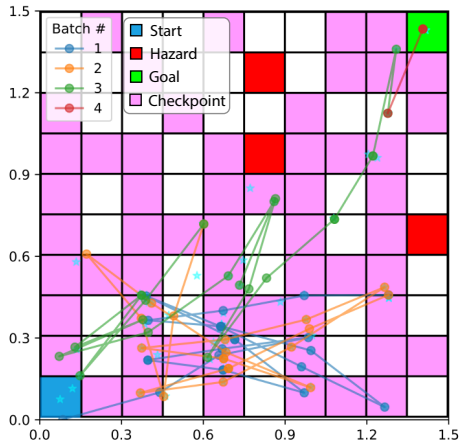


FIGURE 5. Illustration of the second case study results. The robot's objective is to avoid the red hazard states and reach the green goal state while touching pink states every three steps or less along its trajectory. In this case, the trajectory of the robot is altered from the reach-avoid case study so that the robot is able to return to pink states, but it is still able to train its GP models by traversing the safe lower half of the state space until it is able to gain confidence to reach the goal state.

VIII. Discussion, Conclusion, and Future Work

In this work, we introduced a temporal-logic-based planner for bipedal robot locomotion which allows for GP online learning of system and environmental uncertainties. We first formulated an IMDP model of bipedal locomotion which produces control policies compatible with low-level legged kinematic constraints. Additionally, we incorporated stacked GP learning of correlated multi-source yaw and terrain uncertainty. Furthermore, we developed a safe learning procedure which allows a robot to traverse its environment while respecting complex LTL specifications. Finally, we demonstrated the simulation case study results of our algorithms on reduced-order PIPM bipedal dynamics. Our approach intersects the fields of formal methods, learning, and locomotion in order to perform planning with respect to complex specifications on robots with nonlinear, high-DOF dynamics in highly uncertain environments.

Our work lays the theoretical and algorithmic foundations for the application of temporal-logic-based planning techniques onto legged locomotion systems. In particular, we demonstrate an approach which synthesizes high-level control policies that are compatible with low-level bipedal locomotion kinematic constraints. Our future work will focus on physical hardware implementation. We note that there remain several technical challenges in order to translate the simulation results in this work onto physical hardware experiments. Among these are the need to develop a robust low-level controller to track the desired reduced-order PIPM trajectory and the need for experimental characterizations of the motion perturbations experienced by the Digit robot, which are the subject of ongoing research [51]. Additionally, future works will address the computational complexity associated with our algorithms, in particular finding techniques to reduce the effect of the curse of dimensionality associated with abstraction-based techniques.

REFERENCES

[1] M. Hutter, C. Gehring, D. Jud, A. Lauber, D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. A. Höpflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, 2016.

[2] C. Vaudel, "Unitree b1 robotic equipment for research." [Online]. Available: https://www.robotlab.com/higher-ed-robots/store/unitree-b1

[3] E. Ackerman and E. Guizzo, "The next generation of boston dynamics' atlas robot is quiet, robust, and tether free," Jan 2023. [Online]. Available: https://spectrum.ieee.org/next-generation-of-boston-dynamics-atlas-robot

[4] J. Hurst, "Building robots that can go where we go," Jan 2023. [Online]. Available: https://spectrum.ieee.org/building-robots-that-can-go-where-we-go

[5] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[6] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 1207–1213.

[7] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 175–191, 2011.

[8] M. Cai, H. Peng, Z. Li, and Z. Kan, "Learning-based probabilistic ltl motion planning with environment and motion uncertainties," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2386–2392, 2020.

[9] E. Plaku and S. Karaman, "Motion planning with temporal-logic specifications: Progress and challenges," *AI communications*, vol. 29, no. 1, pp. 151–162, 2016.

[10] B. Lacerda, D. Parker, and N. Hawes, "Optimal and dynamic planning for markov decision processes with co-safe ltl specifications," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1511–1516.

[11] Y. Zhao, Y. Li, L. Sentis, U. Topcu, and J. Liu, "Reactive task and motion planning for robust whole-body dynamic locomotion in constrained environments," *The International Journal of Robotics Research*, vol. 41, no. 8, pp. 812–847, 2022.

[12] S. Kulgod, W. Chen, J. Huang, Y. Zhao, and N. Atanasov, "Temporal logic guided locomotion planning and control in cluttered environments," in *American Control Conference*. IEEE, 2020, pp. 5425–5432.

[13] P. Jagtap, S. Soudjani, and M. Zamani, "Formal synthesis of stochastic systems via control barrier certificates," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3097–3110, 2021.

[14] M. Cai, S. Xiao, Z. Li, and Z. Kan, "Optimal probabilistic motion planning with potential infeasible ltl constraints," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 301–316, 2023.

[15] S. Bharadwaj, R. Dimitrova, and U. Topcu, "Synthesis of surveillance strategies via belief abstraction," in *IEEE Conference on Decision and Control*. IEEE, 2018, pp. 4159–4166.

[16] S. Sarid, B. Xu, and H. Kress-Gazit, "Guaranteeing high-level behaviors while exploring partially known maps," 07 2012.

[17] M. R. Maly, M. Lahijanian, L. E. Kavraki, H. Kress-Gazit, and M. Y. Vardi, "Iterative temporal motion planning for hybrid systems in partially unknown environments," in *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC. Association for Computing Machinery, 2013, p. 353–362.

[18] M. Enqi Cao, J. Warnke, Y. Han, X. Ni, Y. Zhao, and S. Coogan, "Leveraging heterogeneous capabilities in multi-agent systems for environmental conflict resolution," *IEEE International Conference on Automation Science and Engineering*, pp. arXiv–2206, 2022.

[19] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *ACM International Conference on Hybrid Systems: Computation and Control*, 2010.

[20] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2908–2927, 2022.

[21] T. Li, R. Calandra, D. Pathak, Y. Tian, F. Meier, and A. Rai, "Planning in learned latent action spaces for generalizable legged locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2682–2689, 2021.

[22] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9805–9811.

[23] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[24] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control using divergent component of motion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2600–2607.

[25] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *The Journal of Experimental Biology*, vol. 202, no. Pt 23, pp. 3325–3332, Dec. 1999.

[26] Y.-M. Chen and M. Posa, "Optimal reduced-order modeling of bipedal locomotion," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8753–8760.

[27] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot cassie." in *Robotics: Science and Systems*, vol. 101. Pittsburgh, Pennsylvania, USA, 2018, p. 14.

[28] K. Sreenath, C. R. Hill, and V. Kumar, "A partially observable hybrid system model for bipedal locomotion for adapting to terrain variations," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*, ser. HSCC '13. New York, NY, USA: Association for Computing Machinery, Apr. 2013, pp. 137–142. [Online]. Available: https://doi.org/10.1145/2461328.2461352

[29] K. Byl and R. Tedrake, "Metastable walking machines," *I. J. Robotic Res.*, vol. 28, pp. 1040–1064, 07 2009.

[30] 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.

[31] M. Dutreix, J. Huh, and S. Coogan, "Abstraction-based synthesis for stochastic systems with omega-regular objectives," *Nonlinear Analysis: Hybrid Systems*, vol. 45, p. 101204, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1751570X22000322

[32] Y. Meng and J. Liu, "Robustly complete finite-state abstractions for control synthesis of stochastic systems," 2023.

[33] R. Majumdar, K. Mallik, A.-K. Schmuck, and S. Soudjani, "Symbolic qualitative control for stochastic systems via finite parity games." *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 127–132, 2021, 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896321012611

[34] I. Gracia, D. Boskos, L. Laurenti, and M. M. J. au2, "Distributionally robust strategy synthesis for switched stochastic systems," 2022.

[35] A. Lavaei, S. Soudjani, A. Abate, and M. Zamani, "Automated verification and synthesis of stochastic hybrid systems: A survey," *Automatica*, vol. 146, p. 110617, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109822004794

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

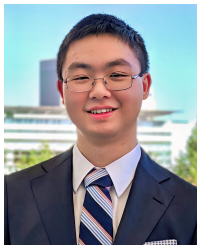
[37] G. Delimpaltadakis, L. Laurenti, and M. Mazo, "Abstracting the sampling behaviour of stochastic linear periodic event-triggered control systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 1287–1294.

[38] G. Delimpaltadakis, M. Lahijanian, M. M. Jr., and L. Laurenti, "Interval markov decision processes with continuous action-spaces," 2022.

[39] M. Dutreix and S. Coogan, "Specification-Guided Verification and Abstraction Refinement of Mixed Monotone Stochastic Systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 2975–2990, Jul. 2021.

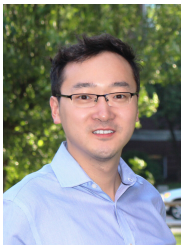
[40] B. Zhong, M. Zamani, and M. Caccamo, "Formal synthesis of controllers for uncertain linear systems against ω-regular properties: A set-based approach," 2022.

[41] T. Banerjee, R. Majumdar, K. Mallik, A.-K. Schmuck, and S. Soudjani, "Fast symbolic algorithms for omega-regular games under strong transition fairness," *TheoretiCS*, vol. Volume 2, feb 2023. [Online]. Available: https://doi.org/10.46298%2Ftheoretics.23.4
[42] 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.
[43] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Strategy Synthesis for Partially-known Switched Stochastic Systems," *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pp. 1–11, May 2021, arXiv: 2104.02172. [Online]. Available: http://arxiv.org/abs/2104.02172
[44] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "Learning predictive terrain models for legged robot locomotion," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3545–3552.
[45] J. Morimoto, C. G. Atkeson, G. Endo, and G. Cheng, "Improving humanoid locomotive performance with learnt approximated dynamics via gaussian processes for regression," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 4234–4240.
[46] M. Budd, B. Lacerda, P. Duckworth, A. West, B. Lennox, and N. Hawes, "Markov decision processes with unknown state feature values for safe exploration using gaussian processes," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7344–7350.
[47] T. Seyde, J. Carius, R. Grandia, F. Farshidian, and M. Hutter, "Locomotion planning through a hybrid bayesian trajectory optimization," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5544–5550.
[48] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration in finite markov decision processes with gaussian processes," 2016.
[49] J. Jiang, Y. Zhao, and S. Coogan, "Safe learning for uncertainty-aware planning via interval MDP abstraction," *IEEE Control Systems Letters*, vol. 6, pp. 2641–2646, 2022.
[50] Y. Zhao, B. R. Fernandez, and L. Sentis, "Robust optimal planning and control of non-periodic bipedal locomotion with a centroidal momentum model," *The International Journal of Robotics Research*, vol. 36, no. 11, pp. 1211–1242, 2017.
[51] A. Shamsah, J. Warnke, Z. Gu, and Y. Zhao, "Integrated Task and Motion Planning for Safe Legged Navigation in Partially Observable Environments," 2021, _eprint: 2110.12097.
[52] 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. [Online]. Available: https://www.springer.com/gp/book/9783319507620
[53] F. Leibfried, V. Dutordoir, S. John, and N. Durrande, "A tutorial on sparse gaussian processes and variational inference," 2021, arXiv: 2012.13962 [cs.LG].
[54] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.
[55] M. Hartley, "Contact-aided state estimation on lie groups for legged robot mapping and control," Ph.D. dissertation, 2019.
[56] M. Bloesch, "State estimation for legged robots-kinematics, inertial sensing, and computer vision," Ph.D. dissertation, ETH Zurich, 2017.
[57] T. Hirabayashi, B. Ugurlu, A. Kawamura, and C. Zhu, "Yaw moment compensation of biped fast walking using 3d inverted pendulum," in *2008 10th IEEE International Workshop on Advanced Motion Control*, 2008, pp. 296–300.
[58] M. Neumann, K. Kersting, Z. Xu, and D. Schulz, "Stacked gaussian process learning," in *2009 Ninth IEEE International Conference on Data Mining*, 2009, pp. 387–396.
[59] 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, conference Name: IEEE Transactions on Software Engineering.
[60] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008, google-Books-ID: nDQiAQAAIAAJ.
[61] PACE, *Partnership for an Advanced Computing Environment (PACE)*, 2017. [Online]. Available: http://www.pace.gatech.edu



Jesse Jiang (Student Member, IEEE) received the B.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA in 2021. He is currently a NSF Graduate Research Fellow and Ph.D. student in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, GA, USA. His research interests lie in applying tools from formal methods and learning to enable novel behavior for legged locomotion systems.

Samuel Coogan (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA in 2010 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, Berkeley, CA, USA in 2012 and 2015. He is currently an associate professor and the Demetrius T. Paris Junior Professor at the Georgia Institute of Technology, Atlanta, GA, USA in the School of Electrical and Computer Engineering and the School of Civil and Environmental Engineering. Prior to joining Georgia Tech in 2017, he was an assistant professor at the University of California, Los Angeles from 2015 to 2017. Prof. Coogan received a CAREER Award from the National Science Foundation in 2018, a Young Investigator Award from the Air Force Office of Scientific Research in 2019, and the Donald P Eckman Award from the American Automatic Control Council in 2020. He is a member of SIAM and a senior member of IEEE.



Ye Zhao (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from The University of Texas at Austin, Austin, TX, USA, in 2016. He was a Post-Doctoral Fellow with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. He is currently an Assistant Professor with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include robust task and motion planning, contact-rich trajectory optimization, formal methods for legged locomotion and navigation. Dr. Zhao serves as an Associate Editor of IEEE Transactions on Robotics (TRO), IEEE Robotics and Automation Letters (RA-L) and IEEE Control Systems Letters (L-CSS). He is a Co-Chair of the IEEE Robotics and Automation Society (RAS) Technical Committee on Whole-Body Control and was a Co-Chair of the IEEE RAS Student Activities Committee.