High-Level Planner Synthesis for Whole-Body Locomotion in Unstructured Environments

Ye Zhao, Ufuk Topcu and Luis Sentis

***Abstract*— Contact-based decision and planning methods are increasingly being sought for task execution in humanoid robots. However, formal methods from the verification and synthesis communities have not been yet incorporated into the motion planning sequence for complex mobility behaviors in humanoid robots. This study takes a step toward formally synthesizing high-level reactive task planners for whole-body locomotion in unstructured environments. We formulate a two-player temporal logic game between the contact planner and its possibly adversarial environment. The resulting discrete planner satisfies the given task specifications expressed in a fragment of temporal logic. The resulting commands are executed by a low-level 3D phase-space motion planner algorithm. We devise various low-level locomotion modes based on centroidal momentum dynamics. Provable correctness of the low-level execution of the synthesized discrete task planner is guaranteed through the so-called simulation relations. Simulations of dynamic locomotion in unstructured environments support the effectiveness of the hierarchical planner protocol.**

I. INTRODUCTION

Many planning methods for mobility in humanoid robots are designed and implemented in ad-hoc manners. In contrast with existing techniques, we investigate formal-methods for high-level task planner synthesis of humanoid robot behaviors. Although widely used in the mobile robot motion planning community [1], [2], [3], formal verification and synthesis methods are still an underexploited area within the locomotion community. A possible reason is that legged robots are high dimensional and possess under-actuated dynamics. To circumvent this difficulty, we propose a planning strategy that leverages low-dimensional phase-space planning models for locomotion [4]. In particular, locomotion trajectories in the phase space are sequentially composed based on determining keyframes states. An example scenario is shown in Fig. 1. These type of dynamic behaviors need to satisfy given task specifications in a provably correct manner. We achieve this correctness by devising a hierarchical locomotion planner protocol to guarantee the correct low-level execution of the high-level reactive planner.

Our objective is to synthesize a correct high-level reactive task planner for the unified locomotion problem by using formal methods. The main contribution of this work is to devise a high-level planner switching strategy by solving a two-player game between the contact planner and its possibly adversarial unstructured environment. We employ linear temporal logic [5] to specify whole-body locomotion (WBL) behaviors. Different from abstraction-based methods applied to continuous dynamical systems [6], locomotion inherently exhibits hybrid dynamics prompting us to focus on discrete planning synthesis. We rely on a discretization of the phase space into keyframe states while we treat the choice of transitions as adversaries in a two-player game. We focus on the communication between the high level task planner and the low level motion planner via switching signals and the correctness of the interplay. We define various low-level locomotion modes from a centroidal-momentum model that specify the basic WBL behaviors. As an extension from rough terrain locomotion [7], [4], we focus not only on the whole-body locomotion but also on its response to various unforeseen environmental events such as stair crack and sudden appearance of a human in the scene. To the best of our knowledge, this study is the first attempt to use formal methods for WBL behaviors and task planner synthesis with guarantee of correctness.

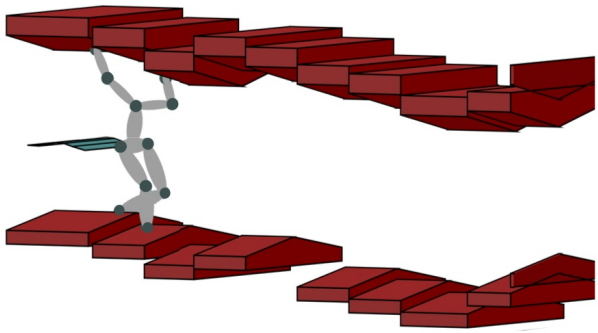

Fig. 1: A scenario of maneuvering in an unstructured environment via multi-contact whole-body locomotion.

II. RELATED WORK

Formal methods for motion and task planning have been widely investigated for mobile navigation [8], [9], [10]. The authors in [2] proposed an automated computational framework for decentralized communications and control of a team of mobile robots from global task specifications. This work suffers from high computational complexity and does not address reactive response to environmental changes. To alleviate the computation burden, the work in [1] proposed a receding-horizon-based hierarchical framework that reduced a complex synthesis problem to a set of significantly

Ye Zhao and Luis Sentis are with the Human Centered Robotics Laboratory, the Department of Mechanical Engineering, at The University of Texas at Austin, USA (yezhao@utexas.edu, lsentis@austin.utexas.edu)

Ufuk Topcu is with the Department of Aerospace Engineering and Engineering Mechanics, at The University of Texas at Austin, USA (utopcu@utexas.edu)

smaller problems of shorter horizon. The approach proposed in [11] allows mobile robots to react to the environment in real time. Deviating from the discretization approaches aforementioned, Signal Temporal Logic (STL) [9] allows to reason about dense-time, real-valued signals. Correspondingly, quantitative semantics are admitted to define the extent to which the specifications are satisfied or violated. This property makes STL especially suitable to quantify robustness measures [12]. Recently, the result of [8] extended the STL to a probabilistic framework such that machine learning methods can be used for safety control under uncertainty and predictions. However, these mobile robots have simple dynamics unlike humanoid robots.

More recently, formal methods have started to be used in humanoid robotics, such as in robotic manipulation by [13], [14]. However, formal methods are yet to be explored for legged locomotion, or for more complex WBL tasks. An abstraction based controller was proposed in [15] for bipedal hybrid systems, but this work lacks provable correctness guarantees, and the robot behavior is limited to level-ground bipedal walking. Recently, the work in [16] proposed an end-to-end approach to automatically implement a synthesized LTL-based planner on an Atlas humanoid robot. Reaction to low-level failures is formally incorporated by simply terminating the execution. Nevertheless, the robot behaviors are centered on manipulation and grasping tasks, and the locomotion problem is briefly discussed. Differently from the works above, our study focuses on the locomotion behaviors within highly unstructured terrains and emphasizes the need of using WBL responsive behaviors. Our interest focuses on the decision policies for effective WBL behaviors.

On controller synthesis, a recent focus has been on provable correctness of controllers [2], [11]. The work of [6] extended the controller synthesis with guaranteed-correctness to nonlinear switched systems and reacted to an adversarial environment at runtime. Our approach follows this work and designs suitable discrete abstractions by approximation techniques. Recently, The authors in [17] proposed a multi-layered synergistic framework such that the low-level sampling-based planner communicates with the high-level discrete planner through a middle synergetic layer. This hierarchy facilitates the interaction between high-level and low-level planners. However their work is not geared towards responding to sudden changes in the environment like we do.

Humanoid multi-contact planning and control are gaining increasing attention [18], [19], [20]. A seminal work [18] identified the rough terrain multi-contact locomotion as a hybrid control problem while the work in [19] phrased the multi-contact planning problem as a hierarchy that first reasons about contacts, and then interpolates these contacts by trajectories computed from a probabilistic planner. The recent study in [21] uses internal force feedback control in the framework of Whole-Body Operational Space Control to achieve balancing on a challenging disjointed terrain. However, all these works focused on either static or quasi-static mobility behaviors. Instead, our planning strategy is geared towards highly dynamic behaviors, i.e., non-periodic multi-contact dynamic locomotion over unstructured environments.

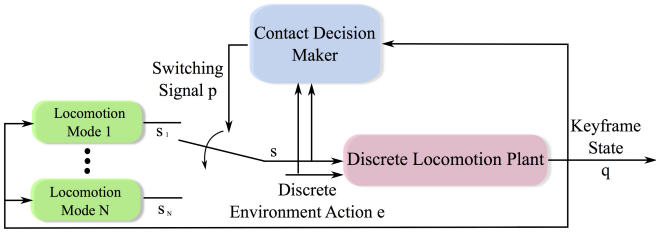

Fig. 2: Logic-based planner structure. Each mode, indexed by a switching signal p, corresponds to a locomotion model. Four modes are modeled for the maneuvering in unstructured dynamic environments. The stair height is represented by the variable e while the control action is represented by s describing the limb contact configuration. The discretized phase-space keyframe is $q = (p_{\text{contact}}, \dot{x}_{\text{apex}})$.

III. Preliminaries and Problem Formulation

We formulate the locomotion planning problem as a switched system, discuss temporal logic preliminaries and propose a discrete task planner synthesis problem.

A. Switched Systems and Phase-Space Planning

The dynamics of an WBL process can be defined as

$$\dot{\boldsymbol{\xi}}(\zeta) = f_p\big(\boldsymbol{\xi}(\zeta), \boldsymbol{u}(\zeta)\big),\ p \in \mathcal{P}, \tag{1}$$

where $\boldsymbol{\xi}(\zeta)$ denotes the system state vector at $\zeta \in \mathbb{R}_{\geq 0}$. The phase progression variable ζ, analogous to time, represents the current progression on a trajectory. $\boldsymbol{u}(\zeta)$ denotes the control input. The switching signal p indexes a specific locomotion mode [22] and belongs to the set $\mathcal{P}$. $f_p(\cdot)$ denotes a vector field associated with mode p. A logic-based switched system is shown in Fig. 2. Given a sequence of switching signals, the low-level motion planner evolves continuously and computes the contact transitions as defined below.

Definition 1 (Phase-Space Contact Switch): A phase-space contact switch, i.e., a contact transition, is defined by the intersection of two adjacent phase-space trajectories [4].

Our contact-based switching strategy is especially suitable for non-periodic locomotion. Non-periodic stability is defined as a progression map Φ between keyframe states, that is, driving the robot's center-of-mass from one desired keyframe to the next one via the control input $\boldsymbol{u}$, i.e. $(p_{\text{contact}_{k+1}}, \dot{x}_{\text{apex}_{k+1}}) = \Phi(p_{\text{contact}_k}, \dot{x}_{\text{apex}_k}, \boldsymbol{u})$, where p_{contact_k} and $\dot{x}_{\text{apex}_k}$ denote the k^{th}-step CoM sagittal position and velocity at the contact apex, respectively. To accomplish unified locomotion behaviors, we will compose a sequence of locomotion modes with planned keyframes. This can be achieved by synthesizing a high-level task planner protocol which makes proper contact decisions like the ones shown in Fig. 3 and determines the switching strategy of the low-level motion planner.

B. Preliminaries

We now define an open finite transition system and its execution, describe temporal logic preliminaries and introduce a specific linear temporal logic (LTL) formula.

Definition 2 (Open Finite Transition System): An open finite transition system (OFTS) $\mathcal{TS}$ is a tuple,

$$\mathcal{TS} := (\mathcal{Q}, \mathcal{P}, \mathcal{E}, \mathcal{S}, \mathcal{T}, \mathcal{I}, AP, \mathcal{L}), \tag{2}$$

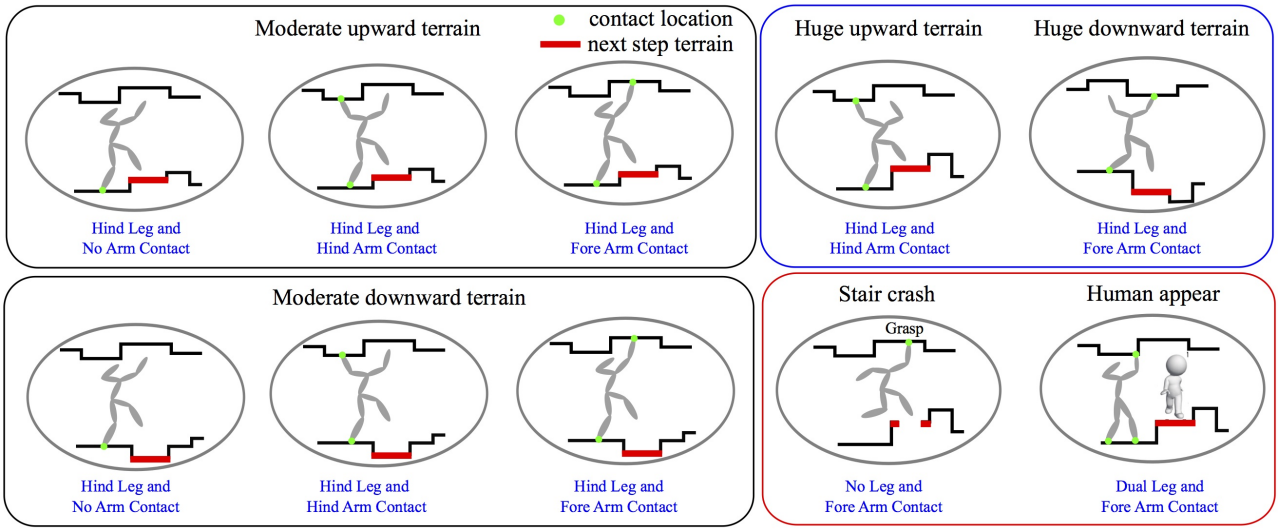

Fig. 3: Contact planning strategies according to rough terrains. We discretize the terrain height of the next walking step, and design it as the environment action. Given a moderate upward or downward terrain height variation, there can be multiple nondeterministic contact actions. Two unexpected events, i.e., stair crack and human appearance, are treated as emergent scenarios, and incorporated into the allowable environment. Detailed definitions of environment and system actions are provided in Section IV-A.

where $\mathcal{Q}$ is a finite set of states, $\mathcal{P}$ is a set of system modes defined in Eq. (1), $\mathcal{E}$ is a finite set of uncontrollable environmental actions, $\mathcal{S}$ is a finite set of controllable robot contact actions, $\mathcal{T} \subseteq (\mathcal{Q} \times \mathcal{P}) \times \mathcal{E} \times \mathcal{S} \times (\mathcal{Q} \times \mathcal{P})$ is a transition, $\mathcal{I} = \mathcal{Q}_0 \times \mathcal{P}_0 \subseteq \mathcal{Q} \times \mathcal{P}$ is a set of initial states, AP is a set of atomic propositions, $\mathcal{L} : (\mathcal{Q} \times \mathcal{P}) \to 2^{AP}$ is a labeling function mapping the state to an atomic proposition. $\mathcal{TS}$ is finite if $\mathcal{Q}, \mathcal{P}, \mathcal{E}, \mathcal{S}$ and AP are finite.

The $\mathcal{TS}$ defined above is "open" because it has uncontrollable actions $\mathcal{E}$. Without loss of generality, it is assumed that for every pair $(q, p) \in \mathcal{Q} \times \mathcal{P}$, there exists at least one pair (q', p') such that $(q, p) \xrightarrow{\mathcal{T}} (q', p')$. The OFTS considered in this study has non-deterministic transitions.

Definition 3 (Execution and word of an OFTS): An execution $\boldsymbol{\gamma}$ of an OFTS $\mathcal{TS}$ is an infinite path sequence $\boldsymbol{\gamma} = (q_0, p_0, e_0, s_0)(q_1, p_1, e_1, s_1)(q_2, p_2, e_2, s_2)\ldots$, with $\gamma_i = (q_i, p_i, e_i, s_i) \in \mathcal{Q} \times \mathcal{P} \times \mathcal{E} \times \mathcal{S}$ and $\gamma_i \xrightarrow{\mathcal{T}} \gamma_{i+1}$. The *word* generated from $\boldsymbol{\gamma}$ is $w_\gamma = w_\gamma(0)w_\gamma(1)w_\gamma(2)\ldots$, with $w_\gamma(i) = \mathcal{L}(\gamma_i),\ \forall i \geq 0$.

The word w_γ is said to satisfy a LTL formula φ, if and only if the execution $\boldsymbol{\gamma}$ satisfies φ. If all executions of $\mathcal{TS}$ satisfy φ, we say that $\mathcal{TS}$ satisfies φ, i.e., $\mathcal{TS} \models \varphi$.

Linear temporal logic is an extension of propositional logic that incorporate temporal operators. An LTL formula φ is composed of atomic propositions $\pi \in AP$. The generic form of a LTL formula has the following grammar,

$$\varphi ::= \pi \;\Big|\; \neg\varphi \;\Big|\; \varphi_1 \wedge \varphi_2 \;\Big|\; \varphi_1 \vee \varphi_2 \;\Big|\; \bigcirc \varphi \;\Big|\; \varphi_1 \mathcal{U} \varphi_2,$$

where the Boolean constants true and false are expressed by $\mathrm{false} = \neg\mathrm{true}$ and $\mathrm{true} = \varphi \vee \neg\varphi$, and we have the temporal operators $\bigcirc$ ("next"), $\mathcal{U}$ ("until"), $\neg$ ("negation") and $\wedge$ ("conjunction"). We can also define $\vee$ ("disjunction"), $\Rightarrow$ ("implication"), $\Leftrightarrow$ ("equivalence"). Another two key operators in LTL are $\Diamond$ ("Eventually") and $\Box$ ("Always"). We can interpret them $\Diamond\varphi := \mathrm{true}\,\mathcal{U}\varphi$ for "Eventually" and $\Box\varphi := \neg\Diamond\neg\varphi$ for "Always".

LTL formulae are interpreted over an infinite sequence of states $(\boldsymbol{q}, \boldsymbol{p})$. We define $(\boldsymbol{q}_i, \boldsymbol{p}_i) = (q_i, p_i)(q_{i+1}, p_{i+1})(q_{i+2}, p_{i+2})\ldots$ as a run from i^{th} position. It is said that a LTL formula φ holds at i^{th} position of $(\boldsymbol{q}, \boldsymbol{p})$, represented as $(q_i, p_i) \models \varphi$, if and only if φ holds for the remaining sequence of $(\boldsymbol{q}, \boldsymbol{p})$ starting at i^{th} position. As to more detailed LTL semantics, we refer readers to the previous works [1], [11] and a brief summary in the supplementary material[1].

C. Discrete Task Planner Synthesis Formulation

Given the preliminaries above, we formulate a discrete task planner synthesis problem and introduce a specific fragment of the temporal logic for the task specifications.

Discrete Contact Planner Switching Synthesis: Given a transition system $\mathcal{TS}$ and a LTL specification φ following the assume-guarantee form,

$$\varphi := \big((\varphi_q \wedge \varphi_e) \Rightarrow \varphi_s\big), \tag{3}$$

where φ_q represents the liveness assumption incorporating the transience property to enforce the continuous trajectory starting from a discretized region to eventually leave that region [6] (in our study, φ_q represents that the phase-space trajectory can not always use the same keyframe state); φ_e and φ_s are the propositions for admissible environment actions and correct overall system behaviors, respectively; we synthesize a contact planner switching strategy γ that generates only correct executions (q, p, e, s), i.e., $(q, p, e, s) \models \varphi$.

To make the computation tractable, we employ a fragment of LTL formulae with favorable polynomial complexity, named as the Generalized Reactivity (1) (GR (1)) formulae [23]. This class of formulae is expressed as, for $v \in \{e, s\}$,

$$\varphi_v = \varphi^v_{\text{init}} \bigwedge_{i \in I_{\text{safety}}} \Box\varphi^v_{\text{trans},i} \bigwedge_{i \in I_{\text{goal}}} \Box\Diamond\varphi^v_{\text{goal},i}, \tag{4}$$

[1] https://drive.google.com/open?id=0B_7VcYBOhr8uWlpGb3FqcGpMeWs

where φ^v_{init} are the propositional formulae defining initial conditions. $\varphi^v_{\text{trans},i}$ refer to the transitional propositional formulae (i.e., safety conditions) incorporating the state at next step. $\varphi^v_{\text{goal},i}$ are the propositional formulae describing the goals to be reached infinitely often (i.e., liveness conditions).

IV. Unified Locomotion Task Specifications and Planner Synthesis

We now introduce the model and the temporal logic specifications for the high-level planner tasks in response of the possibly adversarial environment actions.

A. Specifications for Environment and System Actions

This subsection defines an environment action set $\mathcal{E}$ and a system action set $\mathcal{S}$. Then we propose the corresponding specifications for these actions. First, the environment action is denoted by the set of the variation levels in the stair height

$$\mathcal{E} \coloneqq \{e_{md}, e_{hd}, e_{mu}, e_{hu}, e_{sc}, e_{ha}\}, \tag{5}$$

where the first four actions denote different compositions of downward and upward stairs with moderate and huge height variations, respectively. For instance, e_{md} denotes moderateDownward. The last two actions e_{sc} and e_{ha} represent unexpected events: stairCrack and humanAppear.

Given these environment actions, the system actions of the robot are denoted by a contact configuration set

$$\mathcal{S} \coloneqq \{a_{li\text{-}aj},\ \forall (i,j) \in \mathcal{A}_{\text{index}}\}, \tag{6}$$

where the indices 'l' and 'a' are short for leg and arm, respectively. (i,j) corresponds to the contact limb. We have $\mathcal{A}_{\text{index}} = \{(h,n),(h,h),(h,f),(n,f),(d,h),(d,f)\}$, where the letters '$h$', '$f$', '$d$' and '$n$' represent hind, fore, dual and no contacts, respectively. For instance, $a_{lh\text{-}af}$ denotes the legHindArmFore contact configuration.

We now design environment and system action specifications. The environment actions are assumed not to occur at the initial time, and therefore $\varphi^e_{\text{init}} = \neg e_{sc} \wedge \neg e_{ha}$. Accordingly, the robot can initially not take the actions designated for these unexpected events, i.e., $\varphi^s_{\text{init}} = \neg a_{ln\text{-}af} \wedge \neg(a_{ld\text{-}ah} \vee a_{ld\text{-}af})$. Transitional and goal specifications are defined next

- (S0) None of the huge terrain height variations (i.e., hugeDownward or hugeUpward), humanAppear and stairCrack occur in two consecutive steps,

$$\square(e_{hd} \Rightarrow \bigcirc\neg e_{hd}) \bigwedge \square(e_{hu} \Rightarrow \bigcirc\neg e_{hu})$$
$$\bigwedge \square(e_{ha} \Rightarrow \bigcirc\neg e_{ha}) \bigwedge \square(e_{sc} \Rightarrow \bigcirc\neg e_{sc}).$$

- (S1) In normal scenarios, the mappings between environment and system actions, as illustrated in Fig. 3, are specified by

$$\square(e_{md} \Rightarrow (a_{lh\text{-}an} \vee a_{lh\text{-}ah} \vee a_{lh\text{-}af}))$$
$$\bigwedge \square(e_{mu} \Rightarrow (a_{lh\text{-}an} \vee a_{lh\text{-}ah} \vee a_{lh\text{-}af}))$$
$$\bigwedge \square(e_{hu} \Rightarrow a_{lh\text{-}ah}) \bigwedge \square(e_{hd} \Rightarrow a_{lh\text{-}af}),$$

where moderate terrain variations allow more nondeterministic contact actions than those allowed by huge terrain variations. For instance, if $e = e_{hu}$, i.e., hugeUpward, the robot has only one action option, which makes its hind arm in contact such that its CoM gains a larger acceleration to step up as shown in Fig. 3.

- (S2) If a stair crack occurs, the robot will grab a handle on the ceiling wall (i.e., $a_{ln\text{-}af}$). On the contrary, when the stair is not cracked, we obviously have $a \neq a_{ln\text{-}af}$:

$$\square(e_{sc} \Rightarrow a_{ln\text{-}af}) \bigwedge \square(\neg e_{sc} \Rightarrow \neg a_{ln\text{-}af}).$$

- (S3) If a human appears in front of the robot, the robot comes to a stop with the action $a_{ld\text{-}af}$. Contrarily, when the human disappears, the robot starts to walk from where it stops:

$$\square\big(e_{ha} \Rightarrow (a_{ld\text{-}ah} \vee a_{ld\text{-}af})\big)$$
$$\bigwedge \square\big(\neg e_{ha} \Rightarrow \neg(a_{ld\text{-}ah} \vee a_{ld\text{-}af})\big).$$

- (S4) Neither unexpected events (i.e., humanAppear and stairCrack) occur infinitely often:

$$\square\Diamond\neg e_{ha} \bigwedge \square\Diamond\neg e_{sc}.$$

Among these specifications, we have (S0) representing φ^e_{trans}, (S1)-(S3) representing φ^s_{trans} and (S4) representing φ^e_{goal}, where $\varphi^e_{\text{trans}}, \varphi^s_{\text{trans}}$ and φ^e_{goal} are defined in Eq. (4). Currently, the contact action specifications above are designed in a heuristic manner. We leave the optimal LTL design of contact actions for future work.

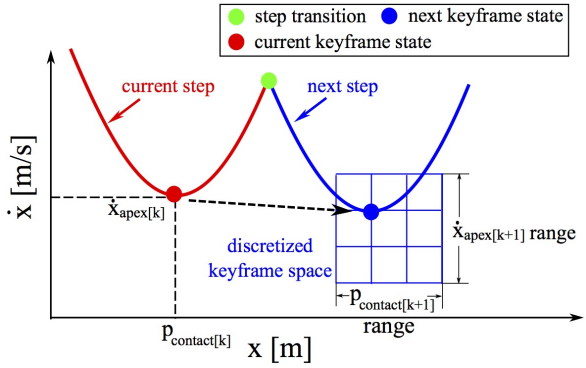

Fig. 4: Phase-space discretization for the next walking step. We choose a rational phase-space region based on robot kinematics, and uniformly discretize this region. The keyframe state $(p_{\text{contact}}, \dot{x}_{\text{apex}})$ is determined according to environment actions.

B. Specifications for Keyframe States

Our discrete phase-space motion planner has a keyframe state vector $q = \{p_{\text{contact}}, \dot{x}_{\text{apex}}\}$ as defined in Section III-A. Since we always focus on the keyframe state for next walking step, the discretization of a particularly selected phase-space region is shown in Fig. 4. Then the keyframe states are represented by

$$\mathcal{Q} \coloneqq \{q_{i\text{-}j},\ \forall (i,j) \in \mathcal{Q}_{\text{index}}\} \bigcup \{q_{\text{swing}}, q_{\text{stop}}\}, \tag{7}$$

where the apex velocity index i and the step length index j are assigned to three different "levels of degree" (LOD): s (Small), n (Normal) and l (Large). For instance, $q_{s\text{-}l}$ represents velSmallstepLarge, a keyframe with a small apex velocity and a large step length. The index set $\mathcal{Q}_{\text{index}}$ comprises

9 elements in total. Additionally, two more keyframe states q_{swing} (velSwingstepSwing) and q_{stop} (velStopstepStop) are designed as unexpected events. A future work is to propose a phase-space partition refinement for the keyframe state.

Remark 1: While we use a fixed discretization over the range of CoM apex velocity and step length, finer or varying discretization may enhance the flexibility of the method at the expense of increase computation cost.

Given environment actions, the propositions for the keyframe states are designed as follows.

- (S5) If $\bigcirc e = \bigcirc e_{md}$ (i.e., moderateDownward), the LOD for the next keyframe state $\bigcirc q$ remains constant or increases by one either from step length or apex velocity:

$$\begin{aligned}
&\square\big((q_{s\text{-}s} \wedge \bigcirc e_{md}) \Rightarrow \bigcirc(q_{s\text{-}s} \vee q_{s\text{-}n} \vee q_{n\text{-}s})\big) \\
&\bigwedge \square\big((q_{s\text{-}n} \wedge \bigcirc e_{md}) \Rightarrow \bigcirc(q_{s\text{-}n} \vee q_{s\text{-}l} \vee q_{n\text{-}n})\big) \\
&\quad \dots \\
&\bigwedge \square\big((q_{l\text{-}n} \wedge \bigcirc e_{md}) \Rightarrow \bigcirc(q_{l\text{-}n} \vee q_{l\text{-}l})\big) \bigwedge \square\big((q_{n\text{-}l} \\
&\wedge \bigcirc e_{md}) \Rightarrow \bigcirc(q_{n\text{-}l} \vee q_{l\text{-}l})\big) \bigwedge \square\big((q_{l\text{-}l} \wedge \bigcirc e_{md}) \\
&\Rightarrow \bigcirc q_{l\text{-}l}\big) \bigwedge \square\Big(\big((q_{\text{swing}} \vee q_{\text{stop}}) \wedge \bigcirc e_{md}\big) \\
&\Rightarrow \bigcirc(q_{s\text{-}n} \vee q_{n\text{-}n} \vee q_{l\text{-}n})\Big),
\end{aligned}$$

  where, if $q = q_{s\text{-}s}$, $\bigcirc q$ can be $q_{s\text{-}s}$ (remaining constant), $q_{s\text{-}n}$ (step length increases one degree) or $q_{n\text{-}s}$ (apex velocity increases one degree). All the other keyframes in normal scenarios follow the same pattern. There are three special cases: (i) when $q = q_{l\text{-}n}$, there are only two choices for $\bigcirc q$, i.e., $q_{l\text{-}n}$ and $q_{l\text{-}l}$; (ii) the same situation applies to $q_{n\text{-}l}$; (iii) when $q = q_{l\text{-}l}$, the only choice is $\bigcirc q = \bigcirc q_{l\text{-}l}$. In unexpected cases, we assign $\bigcirc q$ as one of $q_{s\text{-}n}$, $q_{n\text{-}n}$ and $q_{l\text{-}n}$.
- (S6) If $e = e_{sc}$ (i.e., stairCrack), then $q = q_{\text{swing}}$ regardless of the previous q:

$$\square(e_{sc} \Rightarrow q_{\text{swing}}).$$

  Note that q_{swing} is excluded from the previous state q since we specify that a stair crack can not appear two steps consecutively as shown in (S0).
- (S7) If $e = e_{ha}$ (i.e., humanAppear), then $q = q_{\text{stop}}$ regardless of the previous q:

$$\square(e_{sc} \Rightarrow q_{\text{stop}}).$$

The remaining eight normal scenarios involving different environment and system action compositions are defined in a similar pattern and they are omitted here for brevity. The aforementioned supplementary material illustrates one scenario when $e = e_{hd}$ (i.e., hugeDownward). All the specifications in (S5)-(S7) represent φ^e_{trans}. The goal specification φ^s_{goal} in our scenario is trivial, that is, repeatedly selecting contact configurations among $\mathcal{S}$. Now all the specifications have been proposed, and $\varphi = \big((\varphi_q \wedge \varphi_e) \Rightarrow \varphi_s\big)$ holds.

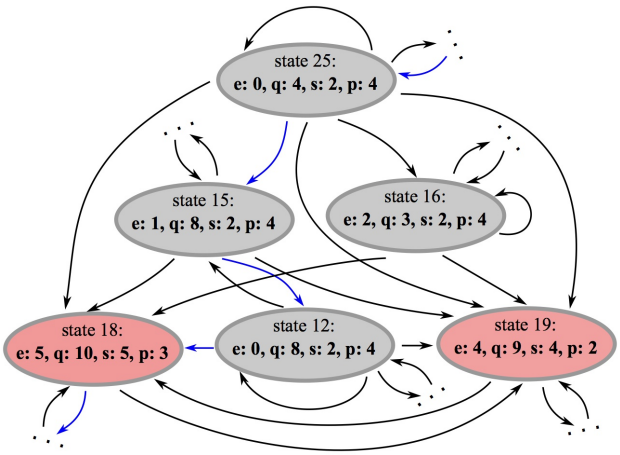

Fig. 5: A fragment of the synthesized automaton for the WBL contact planner. Nondeterministic transitions are encoded in this automaton. The blue transitions represent a specific execution. For illustration convenience, we index both the environment action $\mathcal{E}$ in Eq. (5) and the system action $\mathcal{S}$ in Eq. (6) as $\{0, \dots, 5\}$ in order, respectively. The system keyframe state $\mathcal{Q}$ is indexed as $\{0, \dots, 10\}$ in order. For instance, when the automaton state is at number 25, we have environment state $e = 0$ and keyframe $q = 4$. The winning strategy assigns system action $s = 2$ and system switching mode $p = 4$.

C. High-Level Reactive Task Planner Synthesis

Given the task specifications, we now synthesize a reactive planner by formulating the high-level WBL planning problem as a game between the robot and its environment.

Definition 4 (Game of the WBL Task Planner): A game for the whole-body locomotion task planner is a tuple

$$\mathcal{G} := \langle \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_i, \theta_o, \rho_i, \rho_o, \phi_{\text{win}} \rangle,$$

where $\mathcal{X} := \mathcal{E} \times \mathcal{Q}$ is a set of input variables for player 1; $\mathcal{Y} := \mathcal{S} \times \mathcal{P}$ is a set of output variables for player 2; $\mathcal{V} = \mathcal{X} \times \mathcal{Y}$ is a finite set of proposition state variables over finite domains in the game; θ_i and θ_o are atomic propositions characterizing initial states of the input and output variables, respectively; $\rho_i(\mathcal{V}, \mathcal{X}')$ and $\rho_o(\mathcal{V}, \mathcal{X}', \mathcal{Y}')$ are the transition relation for the input and output variables at next step, respectively; ϕ_{win} is the winning condition given by an LTL formula.

A winning strategy of the switched system for the pair $(\mathcal{TS}, \varphi)$ is defined as a partial function $(\gamma_0 \gamma_1 \cdots \gamma_{i-1}, (q_i, e_i)) \mapsto (s_i, p_i)$, where a switching mode and a contact configuration are chosen according to the state sequence history and the current keyframe state and environment action in order to satisfy Eq. (3). All the specifications are satisfied whatever admissible yet uncontrollable environment actions are.

Proposition 1 (Existence of A Winning WBL Strategy): A winning WBL strategy exists for the game $\mathcal{G}$ in Definition 4 if and only if $(\mathcal{TS}, \varphi)$ is *realizable*.

Fig. 5 shows an automaton fragment of the WBL contact planner. Self-transition exists in moderateUpward states (e.g., state 16) and moderateDownward states (e.g., state 12 and 25) while hugeDownward states (e.g., state 15) do not have a self-transition according to proposition (S1). There is no transition between states 12 and 16 due to infeasible keyframe state transition. States 18 and 19 in red nodes represent humanAppear and stairCrack scenarios, respectively.

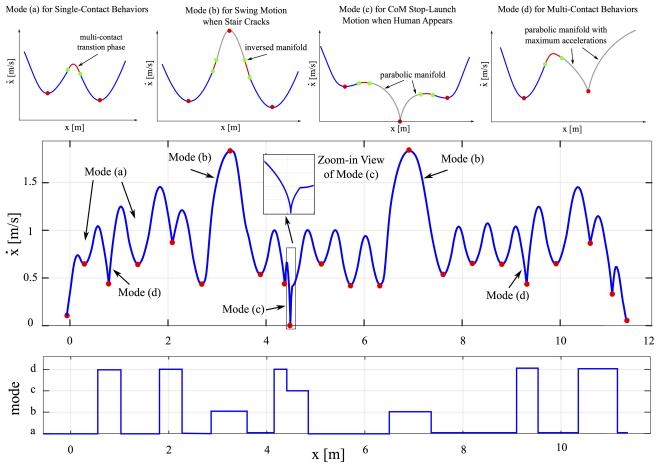

Fig. 6: Sagittal CoM phase-space trajectories and mode switchings for a 20-step WBL maneuver. The top four figures illustrate phase-space manifolds of the four locomotion modes. The mode switching is governed by the synthesized high-level contact planner. Among these steps, two stair crack and one human appearance are taken into account.

Remark 2: The synthesized automaton incorporates non-deterministic transitions, caused by the following reasons: (i) Environment actions are non-deterministic. (ii) Given an environment action, several non-deterministic keyframe states can be chosen. (iii) Even when both an environment action and a keyframe state are given, non-deterministic system contact actions exist in certain transitions.

V. Low-level Locomotion Modes

Given the high-level discrete task planner, let us design a three-dimensional phase-space motion planner that consists of a set of locomotion modes and composes them sequentially to guarantee the correctness of overall implementation. To begin with, we introduce centroidal momentum dynamics in a general sense. Dynamics of mechanical systems can be represented by their rate of linear and angular momenta, which are affected by external wrenches (force/torque) exerted on the system. We characterize this class of dynamical systems via the balance of moments around the system's centroid.

$$\dot{\boldsymbol{l}} = m\ddot{\boldsymbol{p}}_{\text{com}} = \sum_{i}^{N_c} \boldsymbol{f}_i + m\boldsymbol{g}, \tag{8}$$

$$\dot{\boldsymbol{k}} = \sum_{i}^{N_c} (\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times \boldsymbol{f}_i + \boldsymbol{\tau}_i, \tag{9}$$

where $\boldsymbol{l} \in \mathbb{R}^3$ and $\boldsymbol{k} \in \mathbb{R}^3$ represent the centroidal linear and angular momenta, respectively. $\boldsymbol{f}_i \in \mathbb{R}^3$ is the i^{th} ground reaction force, m is the total mass of the robot, $\boldsymbol{g} = (0, 0, -g)^T$ corresponds to the gravity field, $\boldsymbol{f}_{\text{com}} = m\ddot{\boldsymbol{p}}_{\text{com}} = m(\ddot{x}, \ddot{y}, \ddot{z})^T$ is the vector of center-of-mass inertial forces. Eq. (8) represents the rate of spatial linear momentum is equal to the total linear external forces. $\boldsymbol{p}_i = (p_{i,x}, p_{i,y}, p_{i,z})^T$ is the position of the i^{th} limb contact position. $\boldsymbol{\tau}_i \in \mathbb{R}^3$ is the i^{th} contact torque. Eq. (9) reveals that the rate of angular momentum is equal to the sum of the torques generated by contact wrenches at the CoM. Given this general model, certain assumptions are commonly imposed to make the problem tractable [24]. In our case, four specific locomotion modes, under different mild assumptions, are proposed according to specific WBL contact configurations.

Mode (a): Prismatic Inverted Pendulum Model. For single foot contact, Eq. (9) is simplified to $(\boldsymbol{p}_{\text{com}} - \boldsymbol{p}_{\text{foot}}) \times (\boldsymbol{f}_{\text{com}} + m\,\boldsymbol{g}) = -\boldsymbol{\tau}_{\text{com}}$. Given a piece-wise linear CoM path surface to follow, the system dynamics are expressed as

$$\begin{pmatrix} \dot{l}_x \\ \dot{l}_y \end{pmatrix} = m \cdot \omega_{\text{PIPM}}^2 \begin{pmatrix} x - x_{\text{foot}} - \frac{\tau_y}{mg} \\ y - y_{\text{foot}} - \frac{\tau_x}{mg} \end{pmatrix},$$

where l_x and l_y are linear momenta aligned with sagittal and lateral directions as defined in Eq. (8). The PIPM phase-space asymptotic slope [4] is defined as $\omega_{\text{PIPM}} = \sqrt{g/z_{\text{PIPM}}^{\text{apex}}}$, $z_{\text{PIPM}}^{\text{apex}} = (a \cdot x_{\text{foot}} + b - z_{\text{foot}})$, where a and b are the slope and constant scalars for the piecewise linear CoM path surface $\psi_{\text{CoM}}(x, y, z) = z - ax - b = 0$. Thus, dynamics in vertical direction are represented by $\dot{l}_z = a\dot{l}_x$ and not explicitly shown here. The control input is $\boldsymbol{u} = (x_{\text{foot}}, y_{\text{foot}}, \omega_{\text{PIPM}}, \tau_x, \tau_y)^T$. For more details, please refer to the result in [4].

Mode (b): Pendulum Model. When the robot grasps the hook on the ceiling wall to swing over an unsafe region, the system dynamics can be approximated as a pendulum model (PM). For a single arm contact, we have

$$\begin{pmatrix} \dot{l}_x \\ \dot{l}_y \end{pmatrix} = -m \cdot \omega_{\text{PM}}^2 \begin{pmatrix} x - x_{\text{arm}} - \frac{\tau_y}{mg} \\ y - y_{\text{arm}} - \frac{\tau_x}{mg} \end{pmatrix},$$

where similarly we can define $\omega_{\text{PM}} = \sqrt{g/z_{\text{PM}}^{\text{apex}}}$, $z_{\text{PM}}^{\text{apex}} = (z_{\text{arm}} - a \cdot x_{\text{arm}} - b)$, with the same parameter definitions and CoM path surface in Mode (a). A difference between Modes (a) and (b) lies in that PM dynamics are stable since the CoM is always attracted to move towards the apex position while the PIPM dynamics are not.

Mode (c): Stop-Launch Model. When a human appears, the robot has to come to a stop and wait until human disappears. The task in this mode consists on decelerating the CoM motion to zero and accelerating it from zero again. We name this model as a stop-launch model (SLM).

$$\dot{l}_x = ma_x,\ \dot{l}_y = ma_y,\ \dot{l}_z = ma_z,$$

where the control input $\boldsymbol{u} = (a_x, a_y, a_z)^T$ are CoM accelerations. For simplicity, constant accelerations are used and the resulting phase-space trajectory is a parabolic manifold. More elegant acceleration profiles can be designed as needed.

Mode (d): Modified Multi-Contact Model. In this mode, a new multi-contact model is proposed built upon the centroidal momentum, and we name it the modified multi-contact model (MMCM). To make the dynamics tractable, we assume a known constant vertical acceleration a_z in each step and neglection of the angular momentum k_z around z-axis [24]. Therefore, we have a constant resultant vertical external force, i.e., $\sum_i^{N_c} f_{i,z} = m(a_z - g)$. Since our robot has point contacts, $\boldsymbol{\tau}_i = 0, \forall i \leq N_c$, and the dynamics are

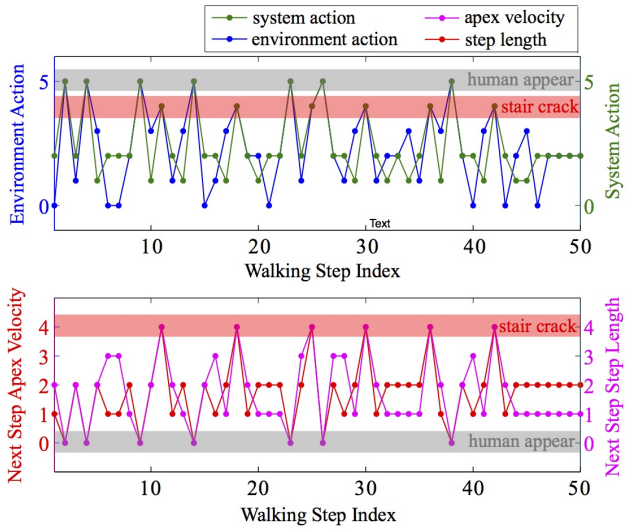

Fig. 7: Environment actions, system actions and keyframe states of 50 walking steps according to the synthesized automaton. Actions and states are indexed by numbers. Unexpected events, i.e, human appearance and stair crack, are highlighted in the shaded regions. In the second subfigure, the numbers 0-4 on the vertical axis correspond to $\{0, 0.4, 0.6, 0.8, 1.7\}$ m/s for next step apex velocity and $\{0.15, 0.5, 0.6, 0.7, 0.6\}$ m for next step step length.

derived as

$$\begin{pmatrix} \dot{l}_x \\ \dot{l}_y \\ \dot{k}_x \\ \dot{k}_y \end{pmatrix} = \begin{pmatrix} \sum_i^{N_c} f_{i,x} \\ \sum_i^{N_c} f_{i,y} \\ -m(a_z-g)y + z\sum_i^{N_c} f_{i,y} - \sum_i^{N_c} p_{i,z} f_{i,x} \\ + \sum_i^{N_c} p_{i,z} f_{i,z} \\ m(a_z-g)x - z\sum_i^{N_c} f_{i,x} + \sum_i^{N_c} p_{i,z} f_{i,x} \\ - \sum_i^{N_c} p_{i,y} f_{i,y} \end{pmatrix},$$

where k_x and k_y are angular momenta aligned with x and y directions as defined in Eq. (9). State vector $\boldsymbol{\xi} = (x, y, l_x, l_y, k_x, k_y)^T$ and control input $\boldsymbol{u} = (f_{1,x}, f_{1,y}, f_{1,z} \dots f_{N_c,x}, f_{N_c,y}, f_{N_c,z})^T$. The control inputs in this mode are the contact forces. These contact forces need to satisfy friction cone constraints while maximizing CoM accelerations. For the four modes above, time synchronization between sagittal and lateral dynamics is guaranteed by a Newton-Raphson foot placement searching algorithm [4]. Via these modes, the switching mode set is defined as

Definition 5 (Switching Mode Set): A phase-space planner switching mode set $\mathcal{P}$ is defined as

$$\mathcal{P} \coloneqq \{\text{PIPM}, \text{PM}, \text{SLM}, \text{MMCM}\}.$$

Given a switching signal commanded from the high-level task planner, the low-level phase-space motion planner generates the continuous CoM trajectory and computes the mode switching event, i.e., walking step transition.

VI. Correctness of The WBL Task Planner

In this section, we prove the correctness of the low-level motion planner implementations in Section V for the discrete contact planner in Section IV-C. To begin with, we define the low-level trajectory as $\delta = (\boldsymbol{\xi}, \rho, \eta, \sigma)$. The high-level execution is defined as $\gamma = (q, e, s, p)$ in Definition 3.

Definition 6: The low-level trajectory δ is a continuous implementation of the high-level execution γ, if there exists a sequence of non-overlapping phase intervals $\mathcal{H} = H_1 \cup H_2 \cup H_3 \dots$ and $\cup_{i=1}^{\infty} H_i = \mathbb{R}^+$ such that $\forall \zeta \in H_k, \forall k \geq 1$, the following mappings hold

$$\boldsymbol{\xi}(\zeta) \in T^{-1}(q_k), \rho(\zeta) = e_k, \eta(\zeta) = s_k, \sigma(\zeta) = p_k,$$

where T represents an abstraction mapping, which maps a certain continuous state $\boldsymbol{\xi}$ region into a discrete state q [6]. By this definition and stutter-equivalence [5], we can conclude $\delta \models \varphi$ if and only if $\gamma \models \varphi$. Additionally, if the left boundary point of H_k approaches to infinity as $k \to \infty$, the continuous implementation guarantees the Zeno behavior to be ruled out. For detailed explanations, reader can refer to [6] and the reference therein. Given these preliminaries, we have the following theorem:

Theorem 1 (Correctness of The WBL Task Planner): Given an over-approximation model and assuming Zeno phenomenon already ruled out, a winning WBL strategy synthesized from the two-player game is guaranteed to be correctly implemented by the underlying low-level phase-space motion planner.

Proof: The theorem above can be proved as follows. From the previous proposition in Section 1, a winning WBL strategy synthesized from the WBL task planner game solves the discrete locomotion planning problem. To study the correctness, we start by elaborating an over-approximated model of our locomotion motion planner. Initially, all the possible transitions in a given keyframe discretization (as shown in Fig. 4) are modeled in the finite transition system. According to environment actions and the LOD principle defined in proposition (S5) of Section IV-B, we remove unnecessary transitions and obtain a more accurate over-approximation model. Given this model, a winning switching strategy, represented as an automaton $\mathcal{A}_{\text{WBL}}$, is synthesized by solving the two-player game. According to $\mathcal{A}_{\text{WBL}}$, a sequence of system actions s and switching modes p are derived from a certain sequence of environment actions e and discretized keyframe states q. To verify the correct implementation of a certain execution γ, we use the switching strategy semantics: given an initial state $\boldsymbol{\xi}(0)$ and an initial environment action $\rho(0) = e_0$, we assign $\eta(0) = s_0$ and $\sigma(0) = p_0$ according to $\mathcal{A}_{\text{WBL}}$. Then the continuous dynamics $\boldsymbol{\xi}(\zeta)$ evolve by following a specific mode $\dot{\boldsymbol{\xi}} = f_{p_0}(\boldsymbol{\xi}, \boldsymbol{u})$. Once a new environment action e' is detected and this action satisfies all the environment assumptions, the switching mode p is updated immediately based on $\mathcal{A}_{\text{WBL}}$. Given this new switching mode, the same procedure is repeated as above. Therefore, it is proved that the low-level trajectory correctly implements one discrete execution of the synthesized automaton. ■

VII. Implementation Results

In this section, we demonstrate our WBL results by combining the task planner and the low-level motion planner via switched modes. Keyframe state $\mathcal{Q}$ is decomposed into two states: apex velocity and step length. For either state, Small, Normal and Large labels are assigned to $\{1, 2, 3\}$ in

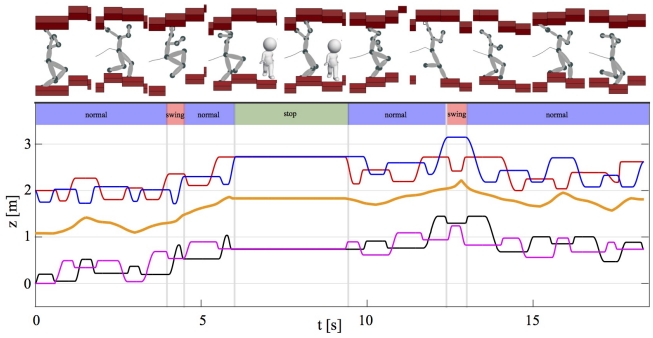

Fig. 8: Snapshots of WBL motions and continuous CoM vertical kinematic trajectories. The snapshots show a sequence of primary behaviors including swinging motion over the cracked stair and stopping motion when a human appears unexpectedly. The figure at the bottom shows the CoM vertical position trajectory (orange thick line), hand and feet trajectories (thin interlaced lines).

order while Stop and Swing labels are assigned to $\{0, 4\}$. The Temporal Logic Planning (TuLiP) toolbox, a python-based embedded control software [1], is used to synthesize the discrete contact planner. *Realizability* of this planner is checked by an off-the-shelf function. If the specifications are *realizable*, the synthesized planner is guaranteed by construction to satisfy all the proposed specifications. Our resulting discrete task planner is represented by a finite state automaton with 27 states and 148 transitions. Fig. 7 illustrates discrete environment and system actions, and their corresponding keyframe states. For the low-level implementation, four modes are alternated according to the high-level switching protocol. Fig. 6 shows the synthesized CoM sagittal phase-space trajectory of a 20-step walking. Fig. 8 illustrates dynamic motion snapshots and continuous kinematic trajectories of the vertical CoM, feet and hands. An accompanying video about the WBL behaviors is available at `https://youtu.be/urp7xu8vx3s`. Currently, we are leveraging the proposed planner synthesis to more generalized locomotion behaviors, such as locomotion with steering directions and maneuvering in cluttered environments.

VIII. Conclusions

This work employs temporal-logic-based formal methods to synthesize high-level reactive task planners for the complex locomotion behaviors in unstructured environments. Contact decisions are determined according to the synthesized switching protocol. A particular focus has been given to the correctness of the overall implementation from the high-level task planner to the low-level motion planner. We proposed new locomotion models at the low-level to deal with complex dynamic motions. The proposed hierarchical planning framework is validated through simulations of WBL maneuvers in an unstructured environment. We expect that this line of work acts as an entry point for the humanoid robotics community to employ formal methods to verify and synthesize planners and controllers.

References

[1] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2817–2830, 2012.

[2] M. Kloetzer and C. Belta, "Automatic deployment of distributed teams of robots from temporal logic motion specifications," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 48–61, 2010.

[3] J. Fu and U. Topcu, "Synthesis of joint control and active sensing strategies under temporal logic constraints," *IEEE Transactions on Automatic Control*, 2016.

[4] Y. Zhao, B. Fernandez, and L. Sentis, "Robust phase-space planning for agile legged locomotion over various terrain topologies," *Robotics: Science and System*, 2016.

[5] C. Baier, J.-P. Katoen *et al.*, *Principles of model checking*. MIT press Cambridge, 2008.

[6] J. Liu, N. Ozay, U. Topcu, and R. M. Murray, "Synthesis of reactive switching protocols from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 58, no. 7, pp. 1771–1785, 2013.

[7] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

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

[9] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 2015, pp. 239–248.

[10] J. A. DeCastro, J. Alonso-Mora, V. Raman, D. Rus, and H. Kress-Gazit, "Collision-free reactive mission and motion planning for multi-robot systems," in *International Symposium on Robotics Research*, September 2015.

[11] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[12] S. S. Farahani, V. Raman, and R. M. Murray, "Robust model predictive control for signal temporal logic synthesis," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 323–328, 2015.

[13] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *IEEE-RAS International Conference on Robotics and Automation*, 2015, pp. 346–352.

[14] S. Chinchali, S. C. Livingston, U. Topcu, J. W. Burdick, and R. M. Murray, "Towards formal synthesis of reactive controllers for dexterous robotic manipulation," in *IEEE-RAS International Conference on Robotics and Automation*, 2012, pp. 5183–5189.

[15] A. D. Ames, P. Tabuada, B. Schürmann, W.-L. Ma, S. Kolathaya, M. Rungger, and J. W. Grizzle, "First steps toward formal controller synthesis for bipedal robots," in *International Conference on Hybrid Systems: Computation and Control*, 2015, pp. 209–218.

[16] S. Maniatopoulos, P. Schillinger, V. Pong, D. C. Conner, and H. Kress-Gazit, "Reactive high-level behavior synthesis for an atlas humanoid robot," in *IEEE-RAS International Conference on Robotics and Automation*, 2016, pp. 4192–4199.

[17] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Motion planning with hybrid dynamics and temporal goals," in *IEEE Conference on Decision and Control*, 2010, pp. 1108–1115.

[18] T. Bretl, "Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 317–342, 2006.

[19] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 7–12.

[20] G. C. Thomas and L. Sentis, "Towards computationally efficient planning of dynamic multi-contact locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[21] D. Kim, Y. Zhao, G. Thomas, B. Fernandez, and L. Sentis, "Stabilizing series-elastic point-foot bipeds using whole-body operational space control," *IEEE Transactions on Robotics, In Press*, 2016.

[22] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2012.

[23] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Saar, "Synthesis of reactive (1) designs," *Journal of Computer and System Sciences*, vol. 78, no. 3, pp. 911–938, 2012.

[24] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4030–4035.