

Time-Varying Foot-Placement Control for Underactuated Humanoid Walking on Swaying Rigid Surfaces

Yuan Gao^{*,1}, Victor Paredes^{*,2}, Yukai Gong³, Zijian He⁴, Ayonga Hereid², Yan Gu⁴

Abstract—Locomotion on dynamic rigid surface (i.e., rigid surface accelerating in an inertial frame) presents complex challenges for controller design, which are essential for deploying humanoid robots in dynamic real-world environments such as moving trains, ships, and airplanes. This paper introduces a real-time, provably stabilizing control approach for underactuated humanoid walking on periodically swaying rigid surface. The first key contribution is the analytical extension of the classical angular momentum-based linear inverted pendulum model from static to swaying grounds. This extension results in a time-varying, nonhomogeneous robot model, which is fundamentally different from the existing pendulum models. We synthesize a discrete footstep control law for the model and derive a new set of sufficient stability conditions that verify the controller’s stabilizing effect. Another key contribution is the development of a hierarchical control framework that incorporates the proposed footstep control law as its higher-layer planner to ensure the stability of underactuated walking. The closed-loop stability of the complete hybrid, full-order robot dynamics under this control framework is provably analyzed based on nonlinear control theory. Finally, experiments conducted on a Digit humanoid robot, both in simulations and with hardware, demonstrate the framework’s effectiveness in addressing underactuated bipedal locomotion on swaying ground, even in the presence of uncertain surface motions and unknown external pushes.

Index Terms—Legged robot control, locomotion stability, reduced-order modeling, underactuation, dynamic ground.

I. INTRODUCTION

Locomotion is a fundamental capability of legged robots, enabling them to navigate diverse environments to perform a variety of real-world tasks, such as warehouse automation, home assistance, package delivery, and search and rescue. Although robust walking on complex yet static terrains such as slopes [1], stairs [2], and gravel [3] has been extensively studied, the focus has primarily been on overcoming significant challenges in controller design due to the inherently unstable, hybrid, nonlinear dynamics of robots, compounded by uncertainties such as terrain variations [4].

* Equal contribution.

¹Y. Gao is with the College of Engineering, University of Massachusetts Lowell, Lowell, MA 01854, USA. E-mail: yuan_gao@student.uml.edu.

²V. Paredes and A. Hereid are with the Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH 43210, USA. E-mails: paredescuana.1@buckeyemail.osu.edu, hereid.1@osu.edu.

³Y. Gong is with the Robotics Department, University of Michigan, Ann Arbor, MI 48105, USA. E-mail: ykgong@umich.edu.

⁴Z. He and Y. Gu are with the School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA. E-mails: {he348, yangu}@purdue.edu.

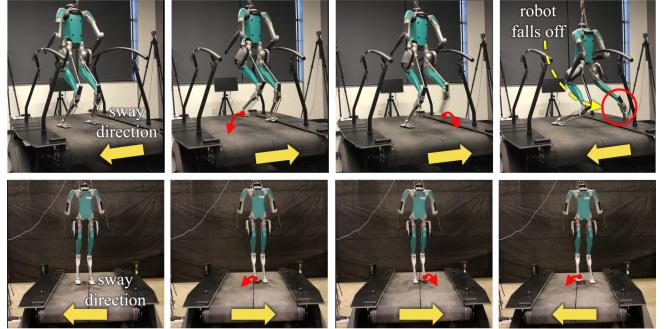


Fig. 1. Top: The proprietary stepping-in-place controller of the humanoid robot Digit fails to maintain stability during DRS sway at a frequency of 0.25 Hz and amplitude of 5 cm. Bottom: Our proposed framework reliably achieves stable walking under identical conditions.

Conversely, bipedal locomotion on dynamic rigid surface (DRS)—rigid surface that move within an inertial frame, such as those on buses, vessels, and airplanes—remains relatively underexplored. This is despite the critical importance of such capabilities in scenarios involving inspection, maintenance, and emergency response on moving transportation vehicles and oil platforms.

Current robust controllers designed for static terrains may only handle mild DRS motions and are prone to failure under more intense or prolonged DRS movements. For instance, as depicted in Fig. 1, the proprietary controller of the Digit humanoid robot fails to maintain stable locomotion under moderate DRS sway. The primary issue is that omitting explicit consideration of surface motion results in it acting as a temporally persistent disturbance, giving the robot insufficient recovery time. This may induce excessive control actions that exceed the robot’s actuator capabilities, leading to potential instability and safety risks.

Thus, this paper introduces a control framework that explicitly incorporates DRS motion into the dynamic modeling and controller design of the robot. Our approach not only aims to enable stable locomotion on DRS for underactuated legged robots but also seeks to offer performance guarantees through provable stabilization of the walking dynamics associated with DRS locomotion.

A. Related work

1) Stability criteria and control for fully actuated walking: Fully actuated walking has been long studied and can be used to facilitate complex tasks beyond walking, such as

simultaneous execution of locomotion and manipulation [5]. For fully actuated walking where a robot's number of degrees of freedom (DoFs) equals the number of actuators, an extensively studied walking control paradigm is based on the zero moment point (ZMP) condition [6], [7]. This condition constrains the robot's center of pressure to be strictly within the support polygon, thus restricting a support foot from rolling about its edge. By enforcing the ZMP condition and the other ground-contact constraints (e.g., friction cone and unilateral constraints) on the full-order robot model (i.e., a model describing the robot dynamics associated with all DoFs), existing motion generation strategies can produce reference motions that correspond to balanced walking. Despite the effectiveness of these methods for static terrain, their computational complexity often precludes real-time application due to the high dimensionality and nonlinear nature of legged robot dynamics.

To address this, reduced-order models such as the linear inverted pendulum (LIP) have been developed, simplifying the center of mass (CoM) dynamics to enable feasible motion planning under the ZMP constraints [8]–[10]. Methods including the capture point [11]–[14] and divergent component of motion (DCM) [15]–[17] also employ the LIP to enable computationally efficient gait generation.

Still, existing gait generation methods based on the ZMP, Capture Point, and DCM primarily focus on fully actuated robots, rather than underactuated walking where the number of DoFs is greater than the number of actuators. Further, as these approaches use robot models (e.g., the classical LIP model) that assume the ground is static, they may not satisfactorily ensure stability or robustness of walking when the ground accelerates in an inertial frame.

2) Stability criteria and control for underactuated walking: Underactuation in walking robots can be induced by a passive support foot (e.g., point foot). With deactivated foot joints, a robot with finite-size feet effectively become a robot with point feet, which simplifies the analysis of foot-ground contact [18]. Also, exploiting underactuation can facilitate dynamic, agile, and energy-efficient gait [19], [20]. Yet, underactuation poses additional challenges as unactuated dynamics cannot be directly altered by joint torques [21].

The hybrid zero dynamics (HZD) framework [21]–[24] is an extensively studied control approach for underactuated bipedal walking. This approach treats the legged robot's hybrid dynamics model, which describes both the continuous motion (e.g., foot swinging) and discrete events (e.g., foot landings), to ensure the provable stability of periodic orbits within the robot's state space. While effective for both fully actuated and underactuated robots, the complexity and computational demands of HZD can limit its application to pre-calculated gaits [25]–[30], which may be restrictive for real-world scenarios. Recent studies have attempted to mitigate these issues via offline generation of a family of different gaits [31], improving the robustness under disturbances [32] and enabling walking over discrete terrain [33].

To enable flexible and efficient real-time adjustment of gait characteristics such as step lengths, which are critical

for guaranteeing walking stability and enhancing robustness, Xiong et al. [34], [35] analytically extended the classical continuous-time LIP model with a discrete model of foot landings, creating a hybrid LIP (H-LIP) model. This development led to the establishment of provable stability conditions that guide the planning of stabilizing foot-landing positions. Similarly, Gong et al. [18] introduced a hybrid version of the angular-momentum-based LIP (ALIP) model, explicitly incorporating a robot's contact angular momentum as a state variable to improve robustness against motor torques and ensure the invariance of the contact angular momentum at foot-touchdown impacts. These reduced-order models have recently been extended to accommodate vertical CoM movements, facilitating navigation over rough terrain and stairs [36], [37], and have been integrated with adaptive regulators to enhance the robustness of locomotion [28].

While these approaches have achieved remarkable performance on physical robots traversing various complex real-world environments, they generally assume that the ground is stationary. Thus, significant ground motion introduces unmodeled uncertainties that can lead to walking instability under these previous control frameworks.

3) Robust walking control for static terrain: Beyond the existing work reviewed earlier, model predictive control schemes are well-suited for addressing perturbations such as uncertain rough terrain since they explicitly enforce the physical limits and constraints [38]–[41]. In addition to model-based approaches, reinforcement learning based control has realized highly robust and agile walking in a wide variety of complex real-world terrains [42]–[47]. However, these methods often assume static ground conditions and does not explicitly account for ground motion, which may not ensure locomotion stability on accelerating ground.

4) Stability conditions and control for accelerating ground: Producing stable gait on DRS presents a complex control challenge due to the inherently time-varying robot dynamics emerging from robot-DRS interactions (Fig. 1). Recently, there has been an increasing interest in studying the modeling, control design, and stability analysis of quadrupedal walking on vertically moving DRS. Iqbal et al. [48] introduced a provably stabilizing nonlinear control approach that explicitly deals with the hybrid, nonlinear, and time-varying, full-order robot dynamics associated with quadrupedal walking on DRS. This approach, however, relies heavily on offline motion generation to meet real-time implementation needs, limiting its practical application.

Furthermore, the classical LIP model has been modified to overcome its static terrain assumption by explicitly considering the vertical movement of DRS, allowing for real-time motion generation and control for quadrupedal walking [49]–[51]. Although these modifications provide stability conditions for the extended LIP models during vertical DRS movements, they cannot be directly applied to address horizontally swaying surfaces. This is because the robot dynamics differ fundamentally during horizontal versus vertical motions. This research aims to fill this gap by creating a new dynamics model and control framework

for locomotion on swaying DRS.

B. Contribution

This study aims to create a real-time, provably stabilizing control approach for underactuated humanoid walking during periodic and horizontal DRS motions, even in the presence of uncertainties such as sudden pushes and uncertain ground movements. To the best of our knowledge, this study represents one of the earliest efforts to explicitly address temporally persistently swaying ground and enable reliable underactuated bipedal locomotion on accelerating ground.

Our preliminary work [52] analytically extended the ALIP model [18] from static to swaying DRS, producing a new reduced-order model referred to as ALIP-DRS. This initial study also provably analyzed the model's stability and synthesized a hierarchical control framework based on the resulting stability condition. Yet, the condition assumed that the robot's gait period matched the DRS motion period exactly, which can be restrictive for practical applications. Moreover, the stability of the complete full-order closed-loop system under the proposed framework was not analyzed, and experimental validation on hardware was absent.

Building upon our preliminary work [52], the substantial, new contributions of this study are:

- Theoretical generalization of the stability condition for the ALIP-DRS model by allowing the ratio between the gait and the DRS motion periods to be any positive rational number. Previously, the preliminary condition [52] assumed a unity ratio. Also, supporting propositions of the main stability theorem are provided, whereas previously only the theorem was presented.
- Development of a three-layer control framework that achieves stable underactuated bipedal walking during DRS sway. The framework extends beyond the preliminary control framework in [52], which was limited to the special stability condition for the ALIP-DRS.
- Overall Lyapunov-based stability analysis for the complete closed-loop full-order system under the proposed control framework, which verifies the provable stability of the system and was not given in [52].
- Experimental validation assessing the effectiveness of the proposed control approach in ensuring walking stability and robustness under various DRS sway motions and uncertainties such as uncertain DRS motions in multiple directions and unknown sudden pushes.

This paper is structured as follows. Section II introduces the proposed ALIP-DRS model for locomotion during DRS sway. Section III presents the proposed discrete footstep control and stability analysis for the ALIP-DRS. Section IV describes a hierarchical control framework that utilizes the ALIP-DRS footstep controller to indirectly stabilize the unactuated dynamics of a full-dimensional underactuated robot. Section V provides the stability analysis of the complete full-order system under the proposed framework. Sections VI and VII report the simulation and hardware experiment results. Section VIII discusses the capabilities and limitations of this work. Section IX gives the concluding remarks.

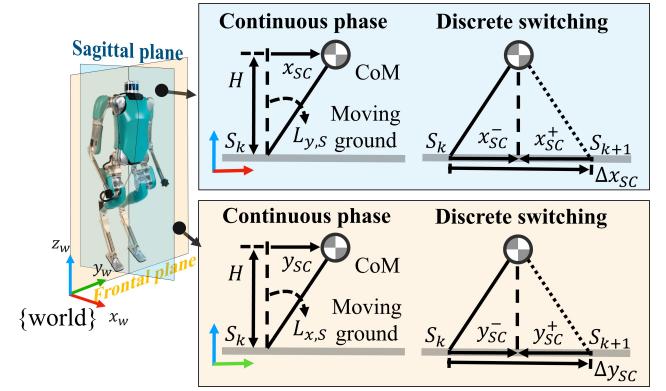


Fig. 2. Illustration of the proposed hybrid ALIP-DRS model, showing both continuous and discrete components in the sagittal and frontal planes.

II. ALIP-DRS MODEL

This section introduces the derivation of the proposed ALIP-DRS model for bipedal walking on swaying DRS.

The key to the derivation is the analytical extension of the classical ALIP model [18] from static to dynamic terrain, achieved by explicitly incorporating the DRS sway in the model. This incorporation leads to the distinct time-varying and nonhomogeneous nature of the ALIP-DRS. Also, we explicitly consider the discrete robot dynamics at foot-landing events, which allows us to exploit foot placement in the controller design to ensure ALIP-DRS's stability. This consideration results in the hybrid nature of the ALIP-DRS. Figure 2 illustrates the three-dimensional (3-D) ALIP-DRS.

A. Contact Angular Momentum during DRS Sway

To analytically extend the classical ALIP from static ground to swaying DRS, we first mathematically define the robot's contact angular momentum during DRS sway.

The contact angular momentum, denoted as $\mathbf{L}_S = [L_{x,S}, L_{y,S}, L_{z,S}]^\top$, refers to the robot's total angular momentum about the ground-contact point. We choose L_S as a state variable due to the benefits [18] reviewed in Sec. I.

We use S to denote the contact point on the DRS, which moves along with the DRS in the inertial world frame. Also, let m , \mathbf{v}_{CoM} , \mathbf{p}_{SC} , and \mathbf{L}_{CoM} denote the robot's total mass, absolute CoM velocity in the world frame, CoM position relative to the contact point S , and angular momentum about the CoM, respectively. Then, \mathbf{L}_S is given by:

$$\mathbf{L}_S = \mathbf{L}_{CoM} + \mathbf{p}_{SC} \times (m\mathbf{v}_{CoM}). \quad (1)$$

To derive the dynamics of \mathbf{L}_S , we introduce the point A , which is a static point in the world frame that aligns with S at the given time instant. Further, let \mathbf{p}_{AC} and \mathbf{p}_{SA} be the position vectors pointing from A to CoM and from S to A , respectively. Then, we introduce the angular momentum about the point A , denoted as $\mathbf{L}_A \in \mathbb{R}^3$, which is related to \mathbf{L}_{CoM} through $\mathbf{L}_A = \mathbf{L}_{CoM} + \mathbf{p}_{AC} \times (m\mathbf{v}_{CoM})$. Thus, we have:

$$\mathbf{L}_S = \mathbf{L}_A + \mathbf{p}_{SA} \times (m\mathbf{v}_{CoM}). \quad (2)$$

Note that \mathbf{p}_{SA} is a zero vector since A and S coincide at the given time, but its time derivative is not necessarily zero due to the nontrivial movement of S in the world frame.

B. Continuous-Phase Dynamics of ALIP-DRS

During the continuous phase of bipedal walking, one foot contacts the ground while the other swings in the air.

1) *Dynamics of contact angular momentum \mathbf{L}_S :* To model the dynamics of \mathbf{L}_S , we first derive that for \mathbf{L}_A . As $\dot{\mathbf{L}}_A$ equals the sum of the external moments about point A , we have:

$$\dot{\mathbf{L}}_A = \mathbf{p}_{AC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A = \mathbf{p}_{SC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A, \quad (3)$$

where $\mathbf{g} = [0, 0, -g]^\top$ is the gravitational acceleration with magnitude g and $\boldsymbol{\tau}_A = [\tau_{A,x}, \tau_{A,y}, \tau_{A,z}]^\top \in \mathbb{R}^3$ is the external torque that is applied to the contact point.

Next, we take the time derivative of both sides of (2), yielding $\dot{\mathbf{L}}_S = \dot{\mathbf{L}}_A + \dot{\mathbf{p}}_{SA} \times (m\mathbf{v}_{CoM}) + \mathbf{p}_{SA} \times (m\dot{\mathbf{v}}_{CoM})$. Since $\mathbf{p}_{SA} = \mathbf{0}$ with $\mathbf{0}$ a zero vector with an appropriate dimension and $\dot{\mathbf{p}}_{SA} = -\dot{\mathbf{p}}_S$ with $\mathbf{p}_S = [x_S, y_S, z_S]^\top \in \mathbb{R}^3$ the absolute position of the point S , this equation becomes:

$$\dot{\mathbf{L}}_S = \dot{\mathbf{L}}_A - \dot{\mathbf{p}}_S \times (m\mathbf{v}_{CoM}). \quad (4)$$

Combining (3) and (4) gives the dynamics equation of \mathbf{L}_S :

$$\dot{\mathbf{L}}_S = \mathbf{p}_{SC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A - \dot{\mathbf{p}}_S \times (m\mathbf{v}_{CoM}). \quad (5)$$

Here we assume that the CoM velocity is parallel to the DRS velocity at point S . This helps prevent leg overstretch during surface motion, and simplifies the robot model by leading to $\dot{\mathbf{p}}_S \times (m\mathbf{v}_{CoM}) = \mathbf{0}$. Thus, (5) becomes:

$$\dot{\mathbf{L}}_S = \mathbf{p}_{SC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A. \quad (6)$$

As the toe joints of Digit's support foot are disabled to simplify the treatment of foot-ground contact as reviewed in Sec. I, we have $\tau_{A,x} = \tau_{A,y} = 0$. Then, the scalar form of (6) in x - and y -directions is given by:

$$\begin{bmatrix} \dot{L}_{x,S} \\ \dot{L}_{y,S} \end{bmatrix} = \begin{bmatrix} -mg y_{SC} \\ mg x_{SC} \end{bmatrix}. \quad (7)$$

2) *Dynamics of CoM position \mathbf{p}_{SC} :* By definition, the relative CoM velocity $\dot{\mathbf{p}}_{SC}$ is $\dot{\mathbf{p}}_{SC} = \mathbf{v}_{CoM} - \dot{\mathbf{p}}_S$. Under the assumption that the CoM velocity is parallel to the DRS velocity at S and given that the DRS only moves horizontally, we have $\dot{z}_S = 0$ and $\dot{z}_{SC} = 0$. By expressing \mathbf{v}_{CoM} using (1), the dynamics of the horizontal CoM position is:

$$\begin{bmatrix} \dot{x}_{SC} \\ \dot{y}_{SC} \end{bmatrix} = \begin{bmatrix} \frac{1}{mH} L_{y,S} \\ \frac{-1}{mH} L_{x,S} \end{bmatrix} - \begin{bmatrix} \dot{x}_S(t) \\ \dot{y}_S(t) \end{bmatrix} - \begin{bmatrix} \frac{1}{mH} L_{y,CoM} \\ \frac{-1}{mH} L_{x,CoM} \end{bmatrix}, \quad (8)$$

where $L_{x,CoM}$ and $L_{y,CoM}$ are L_{CoM} 's horizontal elements.

3) *Continuous-phase ALIP-DRS dynamics:* As robots do not typically demonstrate significant angular momentum about the CoM during walking, we consider $\mathbf{L}_{CoM} = \mathbf{0}$ [18]. Then, based on (7) and (8), the ALIP-DRS dynamics in the sagittal and frontal planes are respectively expressed as:

$$\begin{bmatrix} \dot{x}_{SC} \\ \dot{L}_{y,S} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \frac{1}{mH} \\ mg & 0 \end{bmatrix}}_{=: \mathbf{A}_x} \underbrace{\begin{bmatrix} x_{SC} \\ L_{y,S} \end{bmatrix}}_{=: \mathbf{x}} + \underbrace{\begin{bmatrix} -\dot{x}_S(t) \\ 0 \end{bmatrix}}_{=: \mathbf{f}_x(t)} \quad \text{and} \quad (9)$$

$$\begin{bmatrix} \dot{y}_{SC} \\ \dot{L}_{x,S} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -\frac{1}{mH} \\ -mg & 0 \end{bmatrix}}_{=: \mathbf{A}_y} \underbrace{\begin{bmatrix} y_{SC} \\ L_{x,S} \end{bmatrix}}_{=: \mathbf{y}} + \underbrace{\begin{bmatrix} -\dot{y}_S(t) \\ 0 \end{bmatrix}}_{=: \mathbf{f}_y(t)}. \quad (10)$$

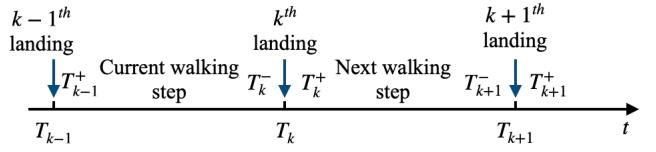


Fig. 3. Illustration of foot-landing time instants. T_{k-1}^+ and T_k^- are the start and end time instants of the current walking step, respectively. The next walking step begins at $t = T_k^+$ and ends at T_{k+1}^- .

Remark 1 (Periodic DRS sway): This study considers DRS motions $\dot{x}_S(t)$ and $\dot{y}_S(t)$ that are continuously differentiable and periodic with $\dot{x}_S(t) = \dot{x}_S(t + T_{x,DRS})$ and $\dot{y}_S(t) = \dot{y}_S(t + T_{y,DRS})$ for any $t > 0$. Here $T_{x,DRS}$ and $T_{y,DRS}$ are the least periods. To highlight the explicit time dependence of $\dot{x}_S(t)$ and $\dot{y}_S(t)$, their expressions keep the time argument.

Note that the continuous-phase ALIP-DRS model in (9) and (10) is explicitly time-varying and nonhomogeneous due to the presence of the time-varying forcing terms $\mathbf{f}_x(t)$ and $\mathbf{f}_y(t)$. These terms are induced by the horizontal DRS motion $\dot{x}_S(t)$ and $\dot{y}_S(t)$. This time-varying and nonhomogeneous property is fundamentally different from existing LIP models, which are further discussed later in Remark 2.

C. Discrete Foot-Switching Model of ALIP-DRS

To derive the time evolution of the CoM position (x_{SC} , y_{SC}) and contact angular momentum ($L_{x,S}$, $L_{y,S}$) across a foot-landing event, we consider the k^{th} landing event at the time instant T_k^- for any $k \in \mathbb{N}$, as illustrated in Fig. 3. We use $(\cdot)^-$ and $(\cdot)^+$ to denote the values of (\cdot) just before and after a foot-landing instant, respectively.

1) *Impact invariance of contact angular momentum \mathbf{L}_S :* Upon completing one walking step, the swing foot touches the walking surface while the old support foot begins to swing. Thus, there is a sudden change in the contact-point position, as illustrated in Fig. 2.

Let $\mathbf{L}_{S,k}$ denote the value of the angular momentum \mathbf{L}_S about the k^{th} contact point. This contact point is initiated at the $(k-1)^{th}$ landing at T_{k-1}^- . Let $\mathbf{L}_{S,k+1}$ represent the angular momentum about the new $(k+1)^{th}$ contact point initiated at T_k^- . At T_k^- , $\mathbf{L}_{S,k+1}^-$ and $\mathbf{L}_{S,k}^-$ are related through:

$$\mathbf{L}_{S,k+1}^- = \mathbf{L}_{S,k}^- + \mathbf{p}_{(k+1) \rightarrow k} \times (m\mathbf{v}_{CoM}^-), \quad (11)$$

where $\mathbf{p}_{(k+1) \rightarrow k}$ is the position vector pointing from the $(k+1)^{th}$ to the k^{th} contact point.

As the robot walks on a horizontal terrain with zero vertical CoM velocity (i.e., $\dot{z}_{SC} = 0$) [18], we have $\mathbf{p}_{(k+1) \rightarrow k} \times (m\mathbf{v}_{CoM}^-) = \mathbf{0}$, leading to $\mathbf{L}_{S,k+1}^- = \mathbf{L}_{S,k}^-$. Meanwhile, since the k^{th} landing impact results in zero impulse torque about the $(k+1)^{th}$ contact point, $\mathbf{L}_{S,k+1}$ remains unchanged across the k^{th} landing event; that is, $\mathbf{L}_{S,k+1}^+ = \mathbf{L}_{S,k+1}^-$. Thus, the robot's contact angular momentum \mathbf{L}_S is invariant to the landing impact, with $\mathbf{L}_{S,k+1}^+ = \mathbf{L}_{S,k}^-$. Then, the changes in the contact angular momenta across a foot-landing event satisfy $\Delta L_{y,S} := L_{y,S}^+ - L_{y,S}^- = 0$ and $\Delta L_{x,S} := L_{x,S}^+ - L_{x,S}^- = 0$.

2) *Jump in relative CoM position:* At a foot-landing event, the horizontal CoM position (x_{SC} , y_{SC}) jumps because

the horizontal contact-point position (x_S , y_S) jumps (Fig. 2). We use Δx_{SC} and Δy_{SC} to denote such changes in x_{SC} and y_{SC} , respectively; that is, $\Delta x_{SC} := x_{SC}^+ - x_{SC}^-$ and $\Delta y_{SC} := y_{SC}^+ - y_{SC}^-$.

Using u_x and u_y to denote the step lengths along the x - and y -directions, we obtain the discrete open-loop dynamics of x_{SC} and y_{SC} as: $\Delta x_{SC} = -u_x$ and $\Delta y_{SC} = -u_y$. Note that $\mathbf{p}_{(k+1) \rightarrow k} = [-u_x, -u_y, 0]^\top$ for a horizontal, swaying DRS.

Here we treat the footstep position (u_x, u_y) as the discrete control input to the proposed ALIP-DRS model, to be designed later. This treatment allows us to explicitly and systematically consider foot placement in the stability analysis and control of the ALIP-DRS model.

3) Discrete dynamics: In summary, the discrete portion of the open-loop ALIP-DRS dynamics is expressed as:

$$\begin{aligned} \begin{bmatrix} \Delta x_{SC} \\ \Delta L_{y,S} \end{bmatrix} &= \begin{bmatrix} -1 \\ 0 \end{bmatrix} u_x \quad \text{and} \quad \begin{bmatrix} \Delta y_{SC} \\ \Delta L_{x,S} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} u_y. \end{aligned} \quad (12)$$

$=:\Delta\mathbf{x}$ $=:\mathbf{b}_x$ $=:\Delta\mathbf{y}$ $=:\mathbf{b}_y$

D. Open-loop Hybrid ALIP-DRS Model

From the continuous-phase dynamics in (9) and (10) and the discrete one in (12), we know the sagittal and frontal subsystems of the ALIP-DRS share the same structure. For brevity and without loss of generality, we primarily focus on the sagittal-plane subsystem in the controller design and stability analysis for the ALIP-DRS model. Combining the sagittal components of those three equations gives the state-space form of the open-loop ALIP-DRS system as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}_x \mathbf{x} + \mathbf{f}_x(t), & \text{if } t \neq T_k^-; \\ \Delta \mathbf{x} = \mathbf{b}_x u_x, & \text{if } t = T_k^-. \end{cases} \quad (13)$$

Remark 2 (Time-varying and nonhomogeneous ALIP-DRS): Equation (13) reveals that the open-loop ALIP-DRS dynamics is linear, hybrid, time-varying, and nonhomogeneous. The ALIP-DRS is fundamentally different from the existing LIP [8], H-LIP [35], and ALIP [18], which are time-invariant and nonhomogeneous since they consider static terrains. With a stationary ground, $\mathbf{f}_x(t) = \mathbf{0}$ holds, and ALIP-DRS's continuous part reduces to the existing ALIP. Further, the ALIP-DRS is distinct from the existing time-varying LIP models for vertically moving surface [49], [50] and variable CoM height [14], which are all homogeneous.

III. ALIP-DRS FOOTSTEP CONTROL AND STABILITY ANALYSIS

This section introduces the discrete footstep control and stability analysis for the ALIP-DRS. The controller serves as the higher-layer planner of a hierarchical control framework, which indirectly stabilizes the unactuated dynamics of an underactuated walking robot, as detailed in Sec. IV.

The design of this controller builds upon the existing footstep controller [18], adapting it from the classical ALIP model to the proposed ALIP-DRS model, as discussed in subsection A. Given that the ALIP-DRS is time-varying and nonhomogeneous in contrast to the time-invariant and homogeneous nature of the classical ALIP, the previous

stability conditions are not suitable for ALIP-DRS. Thus, new conditions are introduced in subsections B and C.

A. Discrete-Time Foot Placement Control

The objective of the discrete-time footstep control is to find the expression of the foot-landing position u_x such that the hybrid ALIP-DRS system in (13) is provably stabilized.

The key idea of the controller is by adjusting the swing-foot landing location at the end of the current step (e.g., $t = T_k^-$), the contact angular momentum at the end of the next step (e.g., $t = T_{k+1}^-$) can be regulated. These time instants are illustrated in Fig. 3.

1) Solution to the continuous-phase ALIP-DRS dynamics: From the linear system theory, the solution of (9) is:

$$\begin{bmatrix} x_{SC}(t_2) \\ L_{y,S}(t_2) \end{bmatrix} = e^{\mathbf{A}_x(t_2-t_1)} \begin{bmatrix} x_{SC}(t_1) \\ L_{y,S}(t_1) \end{bmatrix} + \begin{bmatrix} V_{x,1} \\ V_{x,2} \end{bmatrix} \quad (14)$$

on $t \in [t_1, t_2]$, for any $t_1, t_2 > 0$. Here, $[V_{x,1}(t_1, t_2), V_{x,2}(t_1, t_2)]^\top := \mathbf{V}_x(t_1, t_2) := \int_{t_1}^{t_2} e^{\mathbf{A}_x(t_2-\tau)} \mathbf{f}_x(\tau) d\tau$.

This study considers known DRS motion, and thus $\mathbf{f}_x(t)$ is known for any $t > 0$. Then, since $\mathbf{f}_x(t)$ is known, the value of $\mathbf{V}_x(t_1, t_2)$ can be readily computed for any $t_1, t_2 > 0$.

2) Expressing pre-impact contact angular momentum at $t = T_{k+1}^-$: From the second row of (14), the contact angular momentum at the end of the next step, $L_{y,S}(T_{k+1}^-)$, is related to that at the beginning of the next step, $L_{y,S}(T_k^+)$, through:

$$\begin{aligned} L_{y,S}(T_{k+1}^-) &= mHl \sinh(lT_{step}) x_{SC}(T_k^+) \\ &\quad + \cosh(lT_{step}) L_{y,S}(T_k^+) + V_{x,2}(T_k^+, T_{k+1}^-), \end{aligned} \quad (15)$$

where $l := \sqrt{\frac{g}{H}}$ and $T_{step} := T_{k+1}^- - T_k^-$ is the step duration.

The role switching of the swing and support feet leads to:

$$x_{SC}(T_k^+) = x_{SwC}(T_k^-), \quad (16)$$

where x_{SwC} is the forward CoM position relative to the swing foot. Meanwhile, since contact angular momentum is impact invariant as discussed in Sec. II, we have:

$$L_{y,S}(T_k^+) = L_{y,S}(T_k^-). \quad (17)$$

Then, combining (15)-(17) gives:

$$\begin{aligned} L_{y,S}(T_{k+1}^-) &= mHl \sinh(lT_{step}) x_{SwC}(T_k^-) \\ &\quad + \cosh(lT_{step}) L_{y,S}(T_k^-) + V_{x,2}(T_k^+, T_{k+1}^-). \end{aligned} \quad (18)$$

3) Matching the predicted and desired contact angular momenta at T_{k+1}^- : Let $\bar{L}_{y,S}$ denote the desired contact angular momentum in the sagittal plane, which can be treated as a user input. Based on (8) and given the desired forward ALIP-DRS velocity v_x^{des} , $\bar{L}_{y,S}$ can be specified as: $\bar{L}_{y,S} = mHv_x^{des}$.

From (18), we know that to ensure $L_{y,S}(T_{k+1}^-) = \bar{L}_{y,S}(T_{k+1}^-)$, the swing-foot landing position at the end of the current step T_k^- should be:

$$x_{SwC}(T_k^-) = \frac{\bar{L}_{y,S}(T_{k+1}^-) - V_{x,2}(T_k^+, T_{k+1}^-) - \cosh(lT_{step}) L_{y,S}(T_k^-)}{mHl \sinh(lT_{step})}. \quad (19)$$

In (19), $V_{x,2}(T_k^+, T_{k+1}^-)$ can be directly computed as mentioned earlier. The terms $\cosh(lT_{step})$ and $mHl \sinh(lT_{step})$ are known since l , T_{step} , m , and H are known parameters

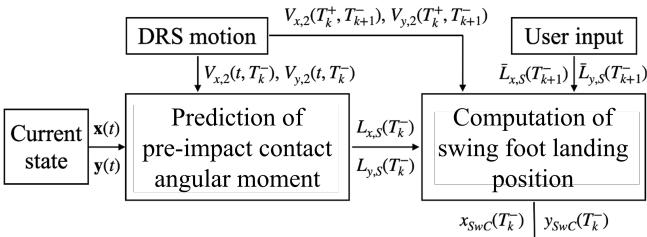


Fig. 4. Computation of the swing-foot landing location for the end of the current walking step at T_k^- . The ALIP-DRS state at the current time step t ($\mathbf{x}(t)$ and $\mathbf{y}(t)$) and the nominal DRS motion profile ($x_{\text{SwC}}(t)$, $y_{\text{SwC}}(t)$) are used to compute the estimated pre-impact contact angular momentum ($L_{x,S}(T_k^-)$, $L_{y,S}(T_k^-)$) based on (14) and (27). This prediction and the corresponding user-specified desired angular momentum ($\bar{L}_{x,S}(T_{k+1}^-)$, $\bar{L}_{y,S}(T_{k+1}^-)$) are used to compute the swing-foot placement ($x_{\text{SwC}}(T_k^-)$, $y_{\text{SwC}}(T_k^-)$). Here $V_{x,2}$, $\bar{L}_{y,S}$, and y_{SwC} are counterparts of $V_{x,2}$, $\bar{L}_{x,S}$, and x_{SwC} for the frontal plane.

of ALIP-DRS. Also, $L_{y,S}(T_k^-)$ can be computed based on the second row of (14) with $t_1 = T_{k-1}$ and $t_2 = T_k$. Thus, the forward foot-landing position at the end of the current step, $x_{\text{SwC}}(T_k^-)$, can be readily calculated. The procedure for computing $x_{\text{SwC}}(T_k^-)$ is illustrated in Fig. 4.

4) *Expressing discrete-time footstep control law u_x :* Recall $u_x(T_k^-) = x_{\text{SC}}(T_k^-) - x_{\text{SwC}}(T_k^-)$ (see Fig. 2). Then, the discrete footstep control law at the k^{th} landing is:

$$u_x(T_k^-) = x_{\text{SC}}(T_k^-) - \frac{\bar{L}_{y,S}(T_{k+1}^-) - V_{x,2}(T_k^+, T_{k+1}^-) - \cosh(lT_{\text{step}})L_{y,S}(T_k^-)}{mHl \sinh(lT_{\text{step}})}. \quad (20)$$

B. Closed-loop ALIP-DRS model

Combining (9), (12), and (20) yields the following closed-loop ALIP-DRS system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}_x \mathbf{x} + \mathbf{f}_x(t), & t \neq T_k^-; \\ \Delta \mathbf{x} = \mathbf{B}_x \mathbf{x}^- + \mathbf{g}_x(t), & t = T_k^-, \end{cases} \quad (21)$$

where

$$\mathbf{B}_x = \begin{bmatrix} -1 & -\frac{\cosh(lT_{\text{step}})}{mHl \sinh(lT_{\text{step}})} \\ 0 & 0 \end{bmatrix} \text{ and } \mathbf{g}_x(T_k^-) = \begin{bmatrix} \frac{\bar{L}_{y,S}(T_{k+1}^-) - V_{x,2}(T_k^+, T_{k+1}^-)}{mHl \sinh(lT_{\text{step}})} \\ 0 \end{bmatrix}.$$

1) *Periodic solution to the closed-loop ALIP-DRS system:* As the DRS of interest to this study sways periodically (Remark 1), we focus on periodic solutions of the closed-loop ALIP-DRS system in (21), which corresponds to cyclic walking motions. We use $\psi_x(t)$ to denote the periodic solution with a least period of T_{sys} , i.e., $\psi_x(t) = \psi_x(t + T_{\text{sys}})$ for any $t > 0$. To analyze the stability of $\psi_x(t)$ under the proposed footstep control law, we introduce the homogeneous system associated with the ALIP-DRS in (21).

2) *Homogeneous system associated with ALIP-DRS:* To analyze the stability of linear time-varying nonhomogeneous hybrid systems, which include the closed-loop ALIP-DRS system in (21), we need to consider their corresponding homogeneous systems [53]. The homogeneous system associated with (21) is given as:

$$\begin{cases} \dot{\mathbf{z}} = \mathbf{A}_x \mathbf{z}^-, & t \neq T_k^-; \\ \Delta \mathbf{z} = \mathbf{B}_x \mathbf{z}^-, & t = T_k^-, \end{cases} \quad (22)$$

where \mathbf{z} is the state vector of the homogeneous system.

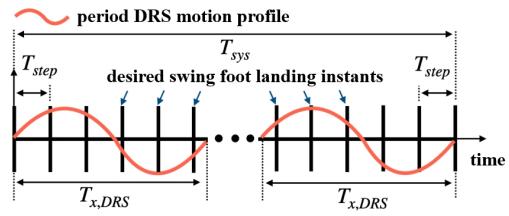


Fig. 5. Illustration of the periods of the walking step (T_{step}), forward DRS motion (T_{DRS}), and solution of the closed-loop ALIP-DRS system (T_{sys}). This study considers both the special case where $T_{\text{sys}} = T_{\text{step}} = T_{x,\text{DRS}}$ and the general case where $T_{\text{sys}} = N_1 T_{\text{step}} = N_2 T_{x,\text{DRS}}$ for any $N_1, N_2 \in \mathbb{N}$.

3) *Monodromy matrix under the special case where $T_{x,\text{DRS}} = T_{\text{step}} = T_{\text{sys}}$:* When the least periods of the DRS motion and the periodic solution are equal to the duration of one walking step (i.e., $T_{x,\text{DRS}} = T_{\text{sys}} = T_{\text{step}}$), the monodromy matrix [53] of the homogeneous system, denoted as \mathbf{M}_x , is expressed as:

$$\mathbf{M}_x = (\mathbf{I} + \mathbf{B}_x) e^{\mathbf{A}_x T_{\text{step}}} = \begin{bmatrix} -\cosh(lT_{\text{step}}) & -\frac{\cosh^2(lT_{\text{step}})}{mHl \sinh(lT_{\text{step}})} \\ mHl \sinh(lT_{\text{step}}) & \cosh(lT_{\text{step}}) \end{bmatrix}, \quad (23)$$

where \mathbf{I} is an identity matrix with an appropriate dimension.

The eigenvalues of \mathbf{M}_x , denoted as $\lambda_{x,1}$ and $\lambda_{x,2}$, can be calculated as: $\lambda_{x,1} = \lambda_{x,2} = 0$.

Remark 3 (Interpretation of the monodromy matrix): Based on the ALIP-DRS equation in (21) and the definition of the monodromy matrix \mathbf{M}_x in (23), \mathbf{M}_x relates the post-impact state values $\mathbf{x}(T_{k-1}^+)$ and $\mathbf{x}(T_k^+)$ as:

$$\mathbf{x}(T_k^+) = \mathbf{M}_x \mathbf{x}(T_{k-1}^+) + \tilde{\mathbf{V}}_x(T_{k-1}^+, T_k^-), \quad (24)$$

where $\tilde{\mathbf{V}}_x := (\mathbf{I} + \mathbf{B}_x) \mathbf{V}_x(T_{k-1}^+, T_k^-)$. Equation (24) is the step-to-step dynamics [35] of the ALIP-DRS. When the ground is static, $\tilde{\mathbf{V}}_x(T_{k-1}^+, T_k^-)$ becomes zero, and (24) reduces to the step-to-step dynamics of the ALIP for static terrain.

4) *Monodromy matrix under the general case where $N_1 T_{\text{step}} = N_2 T_{x,\text{DRS}} = T_{\text{sys}}$:* The rest of this paper considers the general case where $N_1 T_{\text{step}} = N_2 T_{x,\text{DRS}} = T_{\text{sys}}$ for any $N_1, N_2 \in \mathbb{N}$. Under this general case, the ratio between the DRS motion period and one walking-step duration can be any real positive number, instead of one as in the special case. Figure 5 illustrates T_{step} , $T_{x,\text{DRS}}$, and T_{sys} .

Using (24) recursively, we obtain the following step-to-step dynamics of the closed-loop ALIP-DRS system:

$$\begin{aligned} \mathbf{x}(T_{k+N_1}^+) &= \mathbf{M}_x \mathbf{x}(T_{k+N_1-1}^+) + \tilde{\mathbf{V}}_x(T_{k+N_1-1}^+, T_{k+N_1}^-) \\ &= \mathbf{M}_x [\mathbf{M}_x \mathbf{x}(T_{k+N_1-2}^+) + \tilde{\mathbf{V}}_x(T_{k+N_1-2}^+, T_{k+N_1-1}^-)] \\ &\quad + \tilde{\mathbf{V}}_x(T_{k+N_1-1}^+, T_{k+N_1}^-) \\ &= \mathbf{M}_x^2 \mathbf{x}(T_{k+N_1-2}^+) + \mathbf{M}_x \tilde{\mathbf{V}}_x(T_{k+N_1-2}^+, T_{k+N_1-1}^-) \\ &\quad + \tilde{\mathbf{V}}_x(T_{k+N_1-1}^+, T_{k+N_1}^-) \\ &= \dots = \underbrace{\mathbf{M}_x^{N_1} \mathbf{x}(T_k^+)}_{\bar{\mathbf{M}}_x} + \sum_{i=1}^{N_1} \mathbf{M}_x^{N_1-i} \tilde{\mathbf{V}}_x(T_{k+i-1}^+, T_{k+i}^-). \end{aligned} \quad (25)$$

Equation (25) shows that the monodromy matrix $\bar{\mathbf{M}}_x$ of the homogeneous system for this general scenario is related to \mathbf{M}_x through: $\bar{\mathbf{M}}_x = \mathbf{M}_x^{N_1}$. Thus, the two eigenvalues of $\bar{\mathbf{M}}_x$,

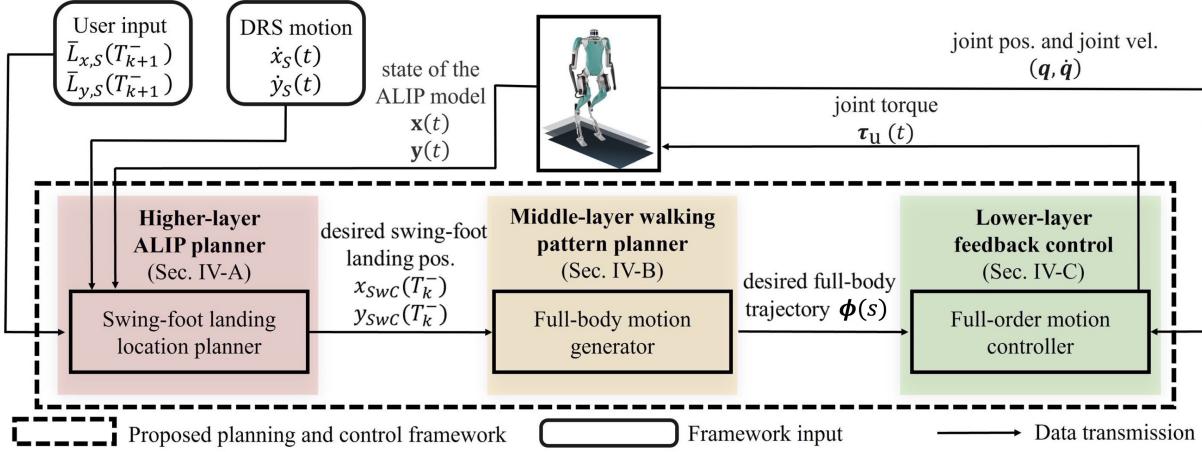


Fig. 6. Overview of the proposed hierarchical control approach. The higher-layer planner generates the desired swing-foot landing locations (x_{SWC} , y_{SWC}) based on the ALIP-DRS model to help stabilize the unactuated robot dynamics. Subsequently, the middle-layer walking pattern generator plans the desired trajectories $\boldsymbol{\phi}(s)$ for the actively actuated joints, ensuring they are compatible with both the ALIP-DRS model and the desired swing-foot landing positions. Lastly, the lower-layer controller outputs the joint torques τ_u to reliably track the full-body trajectories generated by the middle layer.

denoted as $\bar{\lambda}_{x,1}$ and $\bar{\lambda}_{x,2}$, can be evaluated as:

$$\bar{\lambda}_{x,1} = \lambda_{x,1}^{N_1} = 0 \text{ and } \bar{\lambda}_{x,2} = \lambda_{x,2}^{N_1} = 0. \quad (26)$$

5) *Stability of homogeneous system:* With the properties of the monodromy matrix analyzed, we are now ready to introduce the stability condition for the homogeneous system under the general case where $N_1 T_{step} = N_2 T_{x,DRS} = T_{sys}$.

Proposition 1 (Stability of homogeneous system under discrete footstep control): Under the general case where $N_1 T_{step} = N_2 T_{x,DRS} = T_{sys}$ with any $N_1, N_2 \in \mathbb{N}$, the homogenous system in (22) is exponentially stable under the proposed discrete footstep control law in (20). \diamond

Proof: From $T_{sys} = N_1 T_{step}$ and the definition of T_{step} (i.e., $T_{step} = T_{k+1} - T_k$), we know T_{sys} satisfies $T_{sys} = T_{k+N_1} - T_k$. Also, by definition, the matrix \mathbf{A}_x is constant. Thanks to these properties, we can assess the stability of the homogeneous system based on the stability theorem of general linear time-invariant impulsive systems given in Theorem 3.5 of [53]. By that theorem, because the eigenvalues $\bar{\lambda}_{x,1}$ and $\bar{\lambda}_{x,2}$ are strictly less than one in modulus under the proposed discrete control law in (20), the homogeneous system is exponentially stable under the control law. \square

With the stability of the homogeneous system confirmed in Proposition 1, the exponential stability of the periodic solution $\boldsymbol{\psi}_x(t)$ for the nonhomogeneous closed-loop ALIP-DRS system in (21) is discussed next.

C. Closed-Loop Stability Analysis for ALIP-DRS

The closed-loop stability analysis is conducted to verify the stability of the periodic solution $\boldsymbol{\psi}_x(t)$ of the closed-loop ALIP-DRS system, as described in (21), under the proposed footstep controller. The stability of $\boldsymbol{\psi}_x(t)$ is assessed by applying the stability condition for general linear time-varying nonhomogeneous hybrid systems, which include the closed-loop ALIP-DRS system in (21).

Theorem 1 (Exponential stability of ALIP-DRS under footstep control): Consider the general case where $N_1 T_{step} = N_2 T_{x,DRS} = T_{sys}$ with any $N_1, N_2 \in \mathbb{N}$. Under the proposed

discrete footstep control law in (20), the periodic solution $\boldsymbol{\psi}_x(t)$ of the nonhomogeneous hybrid ALIP-DRS system in (21) is exponentially stable. \diamond

Proof: As confirmed in Proposition 1, the homogeneous system in (22) is exponentially stable under the proposed discrete footstep control law in (20) for $N_1 T_{step} = N_2 T_{x,DRS} = T_{sys}$. Then, by the stability theory of linear nonhomogeneous impulsive systems (Theorem 4.2 in [53]), the footstep controller ensures the exponential stability of $\boldsymbol{\psi}_x(t)$ for the nonhomogeneous ALIP-DRS system in (21). \square

From (26), we know that the algebraic multiplicity of the eigenvalue 0 is two while the geometric multiplicity is one. Thus, any state \mathbf{x} does not necessarily converge to the periodic solution $\boldsymbol{\psi}_x(t)$ within one period of T_{sys} but is guaranteed to converge exactly to $\boldsymbol{\psi}_x(t)$ within $2T_{sys}$.

IV. HIERARCHICAL CONTROL FRAMEWORK

This section presents the proposed hierarchical control framework designed to achieve stable underactuated walking during DRS sway. The key novelty of the framework lies in its utilization of the proposed ALIP-DRS footstep control law as a higher-layer footstep planner.

As shown in Fig. 6, the framework is structured into three layers to ensure the computational efficiency for real-time motion generation while guaranteeing stability for the complex full-order dynamics of underactuated humanoid walking on DRS. Its higher-layer footstep planner uses the proposed ALIP-DRS footstep controller in (20) to generate the desired footstep locations in real-time.

The middle layer produces the task-space reference trajectories for the full-order robot model that agree with the desired footstep positions supplied by the higher layer. These trajectories conform to the simplifying assumptions of the ALIP-DRS model, such as parallel velocities for the CoM and the support point, which minimizes discrepancies between the reduced-order ALIP-DRS and the full-order robot model. This alignment ensures that the desired footstep positions provided by the higher layer are physically feasible

for the actual robot to follow, indirectly supporting the stabilization of the full-order underactuated robot model, as detailed in the next section.

The lower-layer controller ensures that the actual robot faithfully executes the planned full-body trajectories.

The overall stability of the complete full-order system under this hierarchical control framework is analyzed and verified in Sec. V.

A. Higher-Layer ALIP-DRS Footstep Planner

This subsection introduces the higher-level planner, which generates the desired CoM and foot-landing positions based on the ALIP-DRS model.

The desired CoM position trajectories along the x - and y -directions are set as the solutions of the ALIP-DRS model in the sagittal and frontal planes, respectively. The desired CoM position in the x -direction is given in (14). From the frontal ALIP-DRS model in (10), its solution between $[t_1, t_2]$ for any $t_1, t_2 > 0$ is:

$$\begin{bmatrix} y_{SC}(t_2) \\ L_{x,S}(t_2) \end{bmatrix} = e^{\mathbf{A}_y(t_2-t_1)} \begin{bmatrix} y_{SC}(t_1) \\ L_{x,S}(t_1) \end{bmatrix} + \begin{bmatrix} V_{y,1}(t_1, t_2) \\ V_{y,2}(t_1, t_2) \end{bmatrix}, \quad (27)$$

where $[V_{y,1}(t_1, t_2), V_{y,2}(t_1, t_2)]^\top := \mathbf{V}_y(t_1, t_2) := \int_{t_1}^{t_2} e^{\mathbf{A}_y(t_2-\tau)} \mathbf{f}_y(\tau) d\tau$.

The discrete footstep control law of the ALIP-DRS model is used to determine the desired foot-landing locations for the actual full-dimensional robot. This controller directly stabilizes the reduced-order ALIP-DRS model, thus contributing to the stabilization of the actual dynamics of the unactuated variables \mathbf{x} and \mathbf{y} as explained in Sec. V.

The desired forward landing position is given by (20). As the ALIP-DRS dynamics in the sagittal and frontal planes share the same structure, the desired lateral landing location is planned in a way similar to the sagittal-plane footstep controller design introduced in Sec. III. Analogous to (19), the lateral CoM location relative to the swing foot at the end of the current step is expressed as:

$$y_{SWC}(T_k^-) = \frac{-L_{x,S}(T_{k+1}^-) + V_{y,2}(T_k^+, T_{k+1}^-) + \cosh(iT_{step}) L_{x,S}(T_k^-)}{mHl \sinh(iT_{step})}, \quad (28)$$

where $\bar{L}_{x,S}$ is the desired contact angular momentum.

Accordingly, the discrete footstep control law for the frontal plane, at the k^{th} landing event, is given by:

$$u_y(T_k^-) = y_{SC}(T_k^-) - \frac{-\bar{L}_{x,S}(T_{k+1}^-) + V_{y,2}(T_k^+, T_{k+1}^-) + \cosh(iT_{step}) L_{x,S}(T_k^-)}{mHl \sinh(iT_{step})}. \quad (29)$$

The value of $\bar{L}_{x,S}(T_{k+1}^-)$ can be determined based on the previous approach used for walking on static terrain [54], which produces a zero average lateral speed. By setting $\bar{L}_{x,S}(T_{k+1}^-)$ based on (10) to allow a periodic lateral motion with a desired step width W , $\bar{L}_{x,S}$ is given by:

$$\bar{L}_{x,S} = \begin{cases} \frac{1}{2} mHW \frac{l \sinh(iT_{step})}{1 + \cosh(iT_{step})} & (\text{right-foot-in-support}); \\ -\frac{1}{2} mHW \frac{l \sinh(iT_{step})}{1 + \cosh(iT_{step})} & (\text{left-foot-in-support}). \end{cases} \quad (30)$$

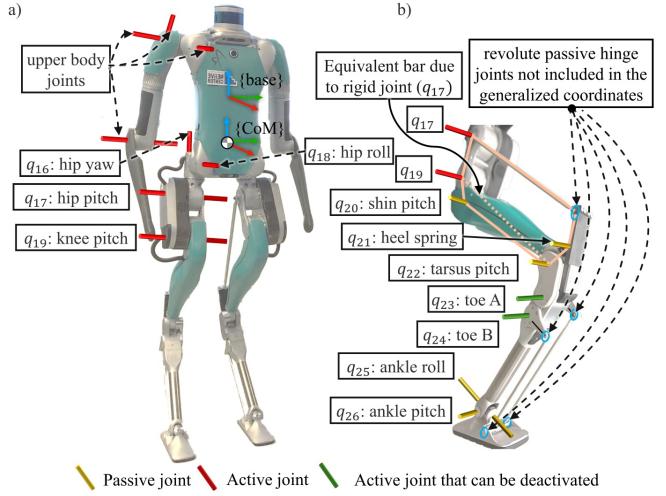


Fig. 7. Illustration of Digit's leg joints from a) the front view and b) the side view. The joints highlighted in green, specifically the toe joints, are active during the swinging phase of the corresponding leg but are deactivated upon ground contact. The side view additionally depicts the three closed-loop linkages, which are governed by holonomic constraints.

B. Middle-Layer Walking Pattern Planner

This subsection explains the middle-layer planner, with the Digit robot as an illustrating example. This layer aims to generate reference trajectories for the full-order robot model that are compatible with the ALIP-DRS model and align with the desired footstep locations determined by the higher-layer planner. These trajectories then serve as inputs to the lower-layer controller, as described in Subsection C.

1) *Degree of underactuation:* To introduce the design of the desired full-dimensional reference trajectories, we first determine the number of DoFs that can be directly commanded (i.e., conversely, the degree of underactuation).

At a floating-base configuration where the robot is detached from the ground, the Digit robot has 36 DoFs, which includes the 6 DoFs from the floating-base coordinates. An illustration of Digit's revolute joints is given in Fig. 7.

One set of six holonomic constraints are formed at the foot-ground contact region when the contact is full and static. Another set of ten holonomic constraints are induced by the kinematic structure of the Digit robot, including: (i) two passive shin joints, with joint angles q_5 (left leg) and q_{20} (right leg), which are both treated as rigid joints due to their high stiffness; (ii) two heel springs (q_{21} and q_6) on both legs, which are considered as rigid links given their high stiffness; and (iii) three closed-loop linkages on each leg, which are used to actuate the passive joints and inherently possess three holonomic constraints per leg.

Under these holonomic constraints, Digit's number of DoFs during walking is: DoF = 36 – 6 – 10 = 20.

The robot has 20 independent joint actuators. To enable point contact for aligning with the ALIP-DRS model, the Toe-A and Toe-B joints (see Fig. 7) of the support foot are deactivated. Thus, the robot has 18 active actuators.

Given that the number of active actuators, denoted as n_a , is 18, we know $n_a = 18 < \text{DoF}$. Thus, the Digit walking is underactuated, with two degrees of underactuation. The re-

sulting four-dimensional unactuated subsystem corresponds to the ALIP-DRS model in the sagittal and frontal planes.

2) *Full-dimensional control variable selection*: With 18 active actuators, 18 control variables can be directly commanded. Leaving the ALIP-DRS state \mathbf{x} and \mathbf{y} uncontrolled, the control variables $\mathbf{h}_c \in \mathbb{R}^{18}$ are chosen as:

$$\mathbf{h}_c(\mathbf{q}) = \begin{bmatrix} z_{SC}(\mathbf{q}) \\ \boldsymbol{\gamma}_{Tr} \\ x_{Sw}(\mathbf{q}) \\ y_{Sw}(\mathbf{q}) \\ z_{Sw}(\mathbf{q}) \\ \boldsymbol{\gamma}_{Sw}(\mathbf{q}) \\ \mathbf{q}_{upper} \end{bmatrix}. \quad (31)$$

Here, $\mathbf{q} \in Q$ is the generalized coordinates, with $Q \subset \mathbb{R}^{36}$ the configuration space. $\boldsymbol{\gamma}_{Tr} \in \mathbb{R}^3$ is the trunk orientation. x_{Sw} and y_{Sw} are the swing-foot positions relative to the CoM in the x - and y -directions, respectively, while z_{Sw} is the absolute swing-foot height. $\boldsymbol{\gamma}_{Sw}(\mathbf{q}) \in \mathbb{R}^3$ is the swing-foot orientation. $\mathbf{q}_{upper} \in \mathbb{R}^8$ is the upper-body joint angles.

We choose to directly regulate the CoM height z_{SC} to be constant, so as to align the actual CoM height with the ALIP-DRS model. Also, we control the trunk orientation $\boldsymbol{\gamma}_{Tr}$ to maintain an upright posture, and keep the upper-body joint angles \mathbf{q}_{upper} fixed for simplicity and for avoiding unexpected arm movements. Finally, controlling the swing-foot position (x_{Sw} , y_{Sw} , and z_{Sw}) enables the actual robot to accurately execute the desired footstep locations provided by the higher-layer planner.

3) *Full-body trajectory generation*: Let $\boldsymbol{\phi}$ denote the desired trajectories of \mathbf{h}_c , with $\boldsymbol{\phi} := [\phi_1, \phi_2, \dots, \phi_{18}]^\top \in \mathbb{R}^{18}$. We utilize Bézier polynomials [21] encoded by a time-based phase variable s [30] to design $\boldsymbol{\phi}$. The encoding variable s represents how long a walking step has progress within a walking step, and is defined as $s := \frac{t - T_{k-1}}{T_{step}}$ for $t \in [T_{k-1}, T_k]$. The Bézier polynomial ϕ_i is defined as $\phi_i(s) := \sum_{j=0}^M \alpha_{i,j} \frac{M!}{j!(M-j)!} s^j (1-s)^{M-j}$, where M represents the user-specified order of the polynomial and $\alpha_{i,j}$ is the coefficient of the j^{th} term. An example of $\boldsymbol{\phi}$ is given in Sec. VII-A.

Based on the desired footstep locations generated by the higher-layer planner as well as the robot's current actual states, the middle layer continuously updates the Bézier coefficients of $\phi_5(s)$ and $\phi_6(s)$, which are the desired swing-foot trajectories for x_{Sw} and y_{Sw} , respectively. The update procedure at time step t is outlined in Algorithm 1, where the value of a variable (\cdot) at t is denoted by $(\cdot)_t$.

C. Lower-Layer Feedback Control

This subsection describes the feedback controller designed to track the desired trajectories planned by the middle layer.

Various existing control techniques can be utilized to construct the lower-level controller to ensure reliable trajectory tracking. In this study, we employ the input-output linearizing control approach, which is inspired by the Hybrid Zero Dynamics (HZD) framework [21], as explained next.

1) *Input-output linearizing control*: The input-output linearizing control law has been previously implemented in

Algorithm 1 Pseudocode for updating the Bézier polynomial coefficients at time step t based on ALIP-DRS planner

```

while True do
    At time step  $t$ , obtain the current generalized coordinates
     $\mathbf{q}_t$  of the full-order robot.
    if  $s=0$  then
        Assign the current swing-foot position in the  $x$ - and
         $y$ -directions to  $\alpha_{5,0}$  and  $\alpha_{6,0}$ :
         $\alpha_{5,0} \leftarrow x_{Sw}(\mathbf{q}_t)$ ,  $\alpha_{6,0} \leftarrow y_{Sw}(\mathbf{q}_t)$ .
    end
    if  $0 \leq s \leq 1$  then
        Convert  $\mathbf{q}_t$  into the current ALIP-DRS state:
         $\mathbf{x}_t \leftarrow \mathbf{x}(\mathbf{q}_t)$ ,  $\mathbf{y}_t \leftarrow \mathbf{y}(\mathbf{q}_t)$ .
        Compute the pre-impact angular momentum
         $L_{y,S}(T_k^-)$  and  $L_{x,S}(T_k^-)$  using (14) and (27), based
        on  $\mathbf{x}_t$ ,  $\mathbf{y}_t$ , and known DRS motion.
        Compute  $(x_{SwC}(T_k^-), y_{SwC}(T_k^-))$  using (19) and (28).
        Assign the planned swing-foot landing locations
         $(x_{SwC}(T_k^-), y_{SwC}(T_k^-))$  to the appropriate Bézier co-
        efficients of  $\phi_5$  and  $\phi_6$ :
         $\alpha_{5,M} \leftarrow -x_{SwC}(T_k^-)$ ,  $\alpha_{6,M} \leftarrow -y_{SwC}(T_k^-)$ .
        Assign  $\alpha_{5,1}, \dots, \alpha_{5,M-1}$  as the linear interpolation be-
        tween  $\alpha_{5,0}$  and  $\alpha_{5,M}$ .
        Assign  $\alpha_{6,1}, \dots, \alpha_{6,M-1}$  as the linear interpolation be-
        tween  $\alpha_{6,0}$  and  $\alpha_{6,M}$ .
    else
        No update on  $\alpha_{5,0}, \dots, \alpha_{5,M}$  and  $\alpha_{6,0}, \dots, \alpha_{6,M}$ .
    end
end

```

simulations for humanoid walking on static surface [30] and quadrupedal robot walking during vertical DRS motion [51].

This controller uses the full-order continuous-phase robot model to exactly linearize the dynamics of the output function, and then shapes the output function dynamics based on the well-studied linear system theory. To ensure accurate trajectory tracking, we choose the output function as the tracking error \mathbf{h} defined as $\mathbf{h} := \mathbf{h}_c - \boldsymbol{\phi}$.

Using Lagrange's method, we can obtain the continuous-phase full-order model as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\boldsymbol{\tau}_u + \mathbf{J}_c^\top \mathbf{f}_c, \quad (32)$$

where \mathbf{M} is the inertia matrix, \mathbf{c} is the sum of the gravitational, Coriolis, and centrifugal terms, \mathbf{B} is the input-selection matrix, and $\boldsymbol{\tau}_u$ is a joint-torque vector. \mathbf{f}_c is the force vector that enforces the 16 holonomic constraints mentioned earlier, which includes the ground-reaction force. $\mathbf{J}_c(\mathbf{q})$ is a Jacobian matrix associated with those holonomic constraints, whose expression is omitted for brevity.

The robot's 16 holonomic constraints are expressed as:

$$\mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \ddot{\mathbf{p}}_c(t). \quad (33)$$

Here the time-varying vector $\mathbf{p}_c(t)$ is defined as $\mathbf{p}_c(t) := [\mathbf{p}_S^\top(t), \boldsymbol{\gamma}_S^\top(t), \mathbf{0}]^\top$, where $\mathbf{p}_S(t)$ and $\boldsymbol{\gamma}_S(t)$ are the position and orientation of the foot-surface contact region that reside in the DRS, respectively, and the dimension of $\mathbf{0}$ is 1×10 . The zero vector corresponds to the ten holonomic constraints in

Digit's kinematic structure, as explained in subsection B.

Integrating (32) and (33), we obtain:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{c}}(t, \mathbf{q}, \dot{\mathbf{q}}) = \bar{\mathbf{B}}\boldsymbol{\tau}_u, \quad (34)$$

where $\bar{\mathbf{c}} := \mathbf{c} - \mathbf{J}_c^\top (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^\top)^{-1} (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{c} - \dot{\mathbf{J}}_c \dot{\mathbf{q}} + \ddot{\mathbf{p}}_c(t))$ and $\bar{\mathbf{B}} := \mathbf{B} - \mathbf{J}_c^\top (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^\top)^{-1} (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{B})$ [27].

Due to the nonlinearity of this full-order dynamics model, the output-function dynamics is also nonlinear. To linearize the output-function dynamics, the input-output linearizing control law is expressed as:

$$\boldsymbol{\tau}_u = \left(\frac{\partial \mathbf{h}_c}{\partial \mathbf{q}} \mathbf{M}^{-1} \bar{\mathbf{B}} \right)^{-1} \left[\left(\frac{\partial \mathbf{h}_c}{\partial \mathbf{q}} \right) \mathbf{M}^{-1} \bar{\mathbf{c}} + \mathbf{v} - \frac{\partial}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{h}_c}{\partial \mathbf{q}} \dot{\mathbf{q}} \right) \dot{\mathbf{q}} + \frac{1}{T_{step}^2} \frac{\partial \boldsymbol{\phi}}{\partial s^2} \right], \quad (35)$$

where the function \mathbf{v} is designed later. Note that \mathbf{h}_c can be chosen such that there exists an open subset of the robot's configuration space Q on which $\frac{\partial \mathbf{h}_c}{\partial \mathbf{q}} \mathbf{M}^{-1} \bar{\mathbf{B}}$ is invertible.

By applying the control law in (35), we obtain the linear closed-loop dynamics of the output function \mathbf{h} during a continuous phase: $\dot{\mathbf{h}} = \mathbf{v}$. Without loss of generality, \mathbf{v} can be chosen as a stabilizing proportional derivative (PD) controller. That is, $\mathbf{v} = -\mathbf{K}_p \mathbf{h} - \mathbf{K}_d \dot{\mathbf{h}}$, with the PD gains \mathbf{K}_p and \mathbf{K}_d tuned to ensure the closed-loop stability of the output-function dynamics given by $\dot{\mathbf{h}} + \mathbf{K}_d \dot{\mathbf{h}} + \mathbf{K}_p \mathbf{h} = \mathbf{0}$.

Although the input-output linearizing control approach explicitly addresses the nonlinear full-order model in (34), its effectiveness relies on the model's accuracy. Due to the presence of springs in the Digit robot (Fig. 7), there is a notable mismatch between the model and the actual robot dynamics. Thus, this study also considers an inverse kinematics approach [55] as a candidate low-level controller.

2) *Inverse kinematics approach:* The closed-form inverse kinematics approach is independent of a robot's dynamics model, and thus can be more robust than the input-output linearizing controller in (35), even in the presence of Digit's springs. As the original inverse kinematics approach [55] is designed for fully actuated robots, we adapt it to underactuated walking robots as follows.

Let $\mathbf{q}_a \in \mathbb{R}^{18}$ denote the angles of all actively actuated joints. Then $\mathbf{q}_a = \mathbf{B}_a \mathbf{q}$ with \mathbf{B}_a a constant selection matrix.

The inverse kinematics controller treats the velocities of the actuated joints, $\dot{\mathbf{q}}_a$, as control inputs, which can be directly set to any arbitrary values within the joint limits to command \mathbf{q}_a . The objective of the controller is to make the closed-loop output-function system stable and first-order, with a linear form of $\dot{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\kappa}\mathbf{h}(\mathbf{q}) = \mathbf{0}$, where $\boldsymbol{\kappa} \in \mathbb{R}^{18 \times 18}$ is a positive-definite control gain matrix.

By the definition of the output function \mathbf{h} , we know $\dot{\mathbf{h}} = \dot{\mathbf{h}}_c - \dot{\boldsymbol{\phi}}$, and this first-order system can be re-written as:

$$\mathbf{J}_{h_c}(\mathbf{q})\dot{\mathbf{q}} = \dot{\boldsymbol{\phi}} - \boldsymbol{\kappa}\mathbf{h}(\mathbf{q}), \quad (36)$$

where $\mathbf{J}_{h_c}(\mathbf{q}) := \frac{\partial \mathbf{h}_c(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{18 \times 36}$. Since \mathbf{J}_{h_c} is not invertible, we cannot directly use this equation to solve for the reference velocities $\dot{\mathbf{q}}$ (which contains $\dot{\mathbf{q}}_a$).

The non-invertibility of \mathbf{J}_{h_c} is essentially due to the underactuation induced by Digit's passive support-toe joints as well as the presence of the 16 holonomic constraints in Digit's kinematic structure. Thus, to solve for $\dot{\mathbf{q}}$, we map $\dot{\mathbf{q}}$ into the passive support-toe velocities $\dot{\mathbf{q}}_{Toe} \in \mathbb{R}^2$ while

explicitly expressing the holonomic constraints using $\dot{\mathbf{q}}$.

Mapping $\dot{\mathbf{q}}$ into $\dot{\mathbf{q}}_{Toe}$, we obtain $\mathbf{S}_{Toe}\dot{\mathbf{q}} = \dot{\mathbf{q}}_{Toe}$, where $\mathbf{S}_{Toe} \in \mathbb{R}^{2 \times 36}$ is a known constant selection matrix. Meanwhile, to respect the 16 holonomic constraints, $\dot{\mathbf{q}}$ needs to satisfy $\mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{p}}_c(t)$, which corresponds to (33). Combining these two equations with (36) gives:

$$\mathbf{J}_{IK}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{v}_{IK}(t, \mathbf{q}, \dot{\mathbf{q}}) := \begin{bmatrix} \dot{\boldsymbol{\phi}} - \boldsymbol{\kappa}\mathbf{h}(\mathbf{q}) \\ \dot{\mathbf{p}}_c(t) \\ \dot{\mathbf{q}}_{Toe} \end{bmatrix}, \quad (37)$$

where $\mathbf{J}_{IK}(\mathbf{q}) := [\mathbf{J}_{h_c}^\top(\mathbf{q}), \mathbf{J}_c^\top(\mathbf{q}), \mathbf{S}_{Toe}^\top]^\top \in \mathbb{R}^{36 \times 36}$. Note that we can select the control variables \mathbf{h}_c such that $\mathbf{J}_{IK}(\mathbf{q})$ is invertible on Q . Thus, the velocity command $\dot{\mathbf{q}}_a$ is given by:

$$\dot{\mathbf{q}}_a = \mathbf{B}_a \mathbf{J}_{IK}^{-1}(\mathbf{q}) \mathbf{v}_{IK}(t, \mathbf{q}, \dot{\mathbf{q}}). \quad (38)$$

Since the actual robot is directly actuated by joint torques instead of joint velocities, we construct a PD joint-torque controller as inspired by [27], which treats the joint velocity command in (38) as the desired velocities denoted as $\mathbf{q}_{a,d}$. Then, from (38), we have $\mathbf{q}_{a,d} = \mathbf{B}_a \mathbf{J}_{IK}^{-1}(\mathbf{q}) \mathbf{v}_{IK}(t, \mathbf{q}, \dot{\mathbf{q}})$.

Thus, the PD joint-torque controller is designed as:

$$\boldsymbol{\tau}_u = -\bar{\mathbf{K}}_p(\mathbf{q}_a - \mathbf{q}_{a,d}) - \bar{\mathbf{K}}_d(\dot{\mathbf{q}}_a - \dot{\mathbf{q}}_{a,d}), \quad (39)$$

where $\bar{\mathbf{K}}_p, \bar{\mathbf{K}}_d \in \mathbb{R}^{18 \times 18}$ are PD matrices to be appropriately tuned. This controller is experimentally implemented on the physical Digit robot, as described in Sec. VII.

V. STABILITY ANALYSIS FOR FULL-ORDER MODEL

This section presents the stability analysis for the hybrid closed-loop full-order system under the proposed framework.

The stability of the closed-loop full-order system needs to be analyzed, as it is not automatically guaranteed by the proposed control framework. This necessity arises because the framework does not explicitly treat the inevitable inaccuracy in the ALIP-DRS model. While the higher-layer planner of the framework stabilizes the ALIP-DRS model, which approximates the unactuated dynamics, it does not directly stabilize the complete nonlinear model of the unactuated subsystem. Meanwhile, the lower-layer controller regulates the fully actuated output-function dynamics but does not directly influence the unactuated subsystem.

Given the nonlinear and hybrid nature of the full-order robot model, the proposed stability analysis employs Lyapunov functions [56]. Without loss of generality, this analysis assumes that the framework's lower layer is the input-output linearizing controller detailed in (35). The analysis considers the general case where $N_1 T_{step} = N_2 T_{x,DRS} = T_{sys}$.

A. Closed-loop Error Dynamics

1) *State definition:* Let \mathbf{X} denote the state of the full-order control system, comprising state variables of the unactuated dynamics, which is approximated by the ALIP-DRS model, and the directly controlled output-function dynamics.

We compactly denote the output function state as $\mathbf{x}_h \in \mathbb{R}^{2n_a}$, and define it as $\mathbf{x}_h := [\mathbf{h}^\top(t, \mathbf{q}), \dot{\mathbf{h}}^\top(t, \mathbf{q}, \dot{\mathbf{q}})]^\top$. Recall $n_a = 18$ for the underactuated Digit, as explained earlier.

Correspondingly, we introduce a new state $\mathbf{x}_\eta \in \mathbb{R}^4$ to denote the tracking error state associated with the unactuated ALIP-DRS state: $\mathbf{x}_\eta := [\mathbf{x}^\top(\mathbf{q}), \mathbf{y}^\top(\mathbf{q})]^\top - \boldsymbol{\psi}(t)$, where $\boldsymbol{\psi}(t)$ is the periodic solution to the closed-loop ALIP-DRS model.

Then, the complete state is defined as: $\mathbf{X} := [\mathbf{x}_h^\top, \mathbf{x}_\eta^\top]^\top$.

2) Hybrid, nonlinear full-order closed-loop dynamics:

Based on the proposed low-level control law, the hybrid closed-loop ALIP-DRS model, and the output-function dynamics model, the closed-loop full-order dynamics is:

$$\begin{cases} \dot{\mathbf{x}}_h = \mathbf{A}_h \mathbf{x}_h & \text{if } t \neq T_k^-; \\ \dot{\mathbf{x}}_\eta = \mathbf{A}_\eta \mathbf{x}_\eta + \mathbf{d}_A & \\ \mathbf{x}_h^+ = \Delta_h(t, \mathbf{x}_h^-, \mathbf{x}_\eta^-) & \text{if } t = T_k^-, \\ \mathbf{x}_\eta^+ = \Delta_\eta \mathbf{x}_\eta^- + \mathbf{d}_\Delta & \end{cases} \quad (40)$$

where $k \in \mathbb{N}$, $\mathbf{A}_h := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_p & -\mathbf{K}_d \end{bmatrix}$, and $\mathbf{A}_\eta := \begin{bmatrix} \mathbf{A}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_y \end{bmatrix}$.

The vector-valued nonlinear function Δ_h is the reset map of \mathbf{x}_h , which can be readily obtained based on the definition of \mathbf{x}_h and the reset map of the robot's generalized coordinates and velocities given in [21]. The constant matrix Δ_η is defined as $\Delta_\eta := \begin{bmatrix} \mathbf{B}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_y \end{bmatrix}$ with \mathbf{B}_y the frontal-plane counterpart of \mathbf{B}_x . \mathbf{d}_A and \mathbf{d}_Δ represent the discrepancies between the reduced-order ALIP-DRS and the full-order model.

Here, $\|\mathbf{d}_A\|$ and $\|\mathbf{d}_\Delta\|$ are assumed to be bounded above by positive, real numbers L_A and L_Δ : $\|\mathbf{d}_A\| \leq L_A$ and $\|\mathbf{d}_\Delta\| \leq L_\Delta$ for any bounded initial state satisfying $\mathbf{X}(0) \in B_{r_d}(\mathbf{0}) := \{\mathbf{X} : \|\mathbf{X}\| \leq r_d\}$, where r_d is a real positive constant.

Without loss of generality, the stability analysis considers the k^{th} gait cycle on $t \in (T_k, T_{k+1})$ with $k \in \mathbb{N}$, which comprises a continuous phase on $t \in (T_k, T_{k+1})$ and a switching event at $t = T_{k+1}^-$. For brevity, the values of a variable \star at $t = T_k^+$ and $t = T_k^-$ are denoted as $\star|_k^+$ and $\star|_k^-$, respectively.

B. Lyapunov Function Candidate

As the full-order robot model comprises unactuated and fully actuated subsystems, we design the Lyapunov function candidate $V(\mathbf{X})$ as $V(\mathbf{X}) = \beta_h V_h(\mathbf{x}_h) + \beta_\eta V_\eta(\mathbf{x}_\eta)$, where $V_h(\mathbf{x}_h)$ and $V_\eta(\mathbf{x}_\eta)$ are positive-definite functions and β_h and β_η are positive constants to be designed.

1) *Closed-form construction of $V_h(\mathbf{x}_h)$:* We construct the component $V_h(\mathbf{x}_h)$ as the Lyapunov function for the closed-loop, continuous-phase output-function dynamics (i.e., $\dot{\mathbf{h}} + \mathbf{K}_d \dot{\mathbf{h}} + \mathbf{K}_p \mathbf{h} = \mathbf{0}$). Since the lower-layer controller can stabilize this output-function dynamics with appropriately chosen PD gains \mathbf{K}_p and \mathbf{K}_d , the Converse Lyapunov Theorem [56] implies that the Lyapunov function $V_h(\mathbf{x}_h)$ exists for this dynamics. Further, since this dynamics is linear and time-invariant, the closed-form expression of $V_h(\mathbf{x}_h)$ can be constructed via a Lyapunov equation.

2) *Closed-form construction of $V_\eta(\mathbf{x}_\eta)$:* We define $V_\eta(\mathbf{x}_\eta)$ as the Lyapunov function for the closed-loop, step-to-step ALIP-DRS dynamics. This dynamics is analytically attractive and is exponentially stabilized by the proposed footstep controller, as established in Theorem 1. The Lyapunov equation for general linear discrete-time systems, which include this dynamics, is employed to derive the

closed-form expression of $V_\eta(\mathbf{x}_\eta)$. Note that the sagittal-plane component of this dynamics is given in (25), and its frontal-plane counterpart can be derived similarly.

3) *Convergence and Boundedness of V_h and V_η :* To establish the stability condition for the closed-loop system in (40), we first analyze the convergence and boundedness of V_h and V_η , as summarized in the following propositions.

Proposition 2 (Continuous-phase convergence of V_h): Consider the proposed lower-layer control law in (35) and the closed-loop error dynamics in (40). Also, consider the condition that the PD gains of the lower-layer control law are selected such that \mathbf{A}_h is Hurwitz. Then, there exist positive constants r_h , c_{h1} , c_{h2} , and c_{h3} such that the function V_h satisfies $c_{h1} \|\mathbf{x}_h\|^2 \leq V_h(\mathbf{x}_h) \leq c_{h2} \|\mathbf{x}_h\|^2$ and $\dot{V}_h \leq -c_{h3} V_h$ on $t \in (T_k, T_{k+1})$. Accordingly, V_h exponentially converges as: $V_h|_{k+1}^- \leq e^{-c_3(T_{k+1}-T_k)} V_h|_k^+$ during continuous phases. \diamond

The proof of Proposition 2 is a direct adaptation of the Lyapunov stability theory from [56] and is thus omitted.

Proposition 3 (Step-to-step convergence of V_η): Consider the proposed control framework and the closed-loop system in (40). There exist positive real constants r_η , $c_{\eta1}$, $c_{\eta2}$, and $c_{\eta4}$ and non-negative real number $c_{\eta3} < 1$ such that V_η satisfies $c_{\eta1} \|\mathbf{x}_\eta\|^2 \leq V_\eta \leq c_{\eta2} \|\mathbf{x}_\eta\|^2$ and $V_\eta|_{k+1}^- \leq c_{\eta3} V_\eta|_k^+ + c_{\eta4}$ for any bounded initial state satisfying $\mathbf{X}(0) \in B_{r_\eta}(\mathbf{0})$. \diamond

Proposition 3 holds because by Theorem 2, the proposed discrete footstep control law is exponentially stabilizing for the ALIP-DRS model and the inaccuracy of the ALIP-DRS model (i.e., \mathbf{d}_A and \mathbf{d}_Δ) are bounded as mentioned earlier.

C. Main Stability Theorem

Based on Propositions 1-3, the stability conditions for the hybrid error dynamics in (40) are introduced next.

Theorem 2 (Closed-loop stability conditions for the full-order model): Let all conditions in Propositions 1-3 hold. If the continuous-phase convergence rates $\frac{1}{c_{\eta3}}$ and c_{h3} are sufficiently high, then the origin of the hybrid closed-loop error system in (40) is locally stable in the sense of Lyapunov under the proposed control framework. \diamond

Rationale of proof: By the theory of multiple Lyapunov functions [57], the origin of the hybrid closed-loop error system in (40) is locally stable if: (S1) the “switch-out” value of the Lyapunov function candidate $V(\mathbf{X})$ is bounded above by a positive-definite function of the “switch-in” value of $V(\mathbf{X})$ within each continuous phase and (S2) the values of $V(\mathbf{X})$ immediately after each switching event satisfy $V|_{k+1}^+ \leq V|_k^+$. To prove Theorem 2, we show that $V(\mathbf{X})$ satisfies these two conditions if all conditions in Theorem 2 are met.

Proof: We begin the proof by first analyzing the evolution of the Lyapunov function candidate $V(\mathbf{X})$ for the k^{th} complete gait cycle on $t \in (T_k, T_{k+1}]$, which comprises one continuous phase and one switching event.

The continuous-phase boundedness of V_h is given in Proposition 2. Meanwhile, since the continuous-phase uncertainty \mathbf{d}_A in the unactuated dynamics is locally bounded and the matrix \mathbf{A}_η is constant and finite, we know that there exist positive real constants L_1 , L_2 , and r_1 such that

$$V_\eta|_{k+1}^- \leq L_1 V_\eta|_k^+ + L_2 \quad (41)$$

holds for any initial state satisfying $\mathbf{X}(0) \in B_{r_1}$.

From Proposition 2 and (41), we know the following inequality holds for any $\mathbf{X}(0) \in B_{r_1}(\mathbf{0})$:

$$\begin{aligned} V|_{k+1}^- &= \beta_h V_h|_{k+1}^- + \beta_\eta V_\eta|_{k+1}^- \\ &\leq \beta_h e^{-c_3(T_{k+1}-T_k)} V_h|_k^+ + \beta_\eta L_1 V_\eta|_k^+ + \beta_\eta L_2 \\ &\leq L_v V|_k^+ + \beta_\eta L_2, \end{aligned} \quad (42)$$

where $L_v := \max(e^{-c_3(T_{k+1}-T_k)}, L_1)$. This inequality verifies that the stability condition (S1) is met.

Since the nonlinear reset map Δ_h is continuously differentiable in t and \mathbf{X} [48], we know Δ_h is locally Lipschitz in t and \mathbf{X} [56]. Then, by applying the triangle inequality to the reset map Δ_h [58], we know there exists positive real scalars L_Δ , ε_Δ , and r_Δ such that

$$\left\| \mathbf{x}_h|_{k+1}^+ \right\|^2 \leq L_h \left\| \mathbf{x}_h|_{k+1}^- \right\|^2 + L_\eta \left\| \mathbf{x}_\eta|_{k+1}^- \right\|^2 + \varepsilon_\Delta \quad (43)$$

holds for a bounded initial state $\mathbf{X}(0) \in B_{r_\Delta}(\mathbf{0})$.

From Propositions 2 and 3 and (41)-(43), we have:

$$V_h|_{k+1}^+ \leq \frac{L_h c_{h2}}{c_{h1}} e^{-c_3(T_{k+1}-T_k)} V_h|_k^+ + \frac{L_\eta L_1 c_{h2}}{c_{\eta 1}} V_\eta|_k^+ + \varepsilon_h, \quad (44)$$

where $\varepsilon_h := \varepsilon_\Delta c_{h2} + \frac{L_\eta L_2 c_{h2}}{c_{\eta 1}}$.

Then, from the definition of V , Proposition 3, and (44), the step-to-step convergence of V is given by:

$$V|_{k+1}^+ \leq c_{v1} V|_k^+ + c_{v2}, \quad (45)$$

where $c_{v1} := \max(\frac{L_h c_{h2}}{c_{h1}} e^{-c_{h3}(T_{k+1}-T_k)}, c_{\eta 3} + \frac{L_1 L_\eta \beta_h c_{h2}}{\beta_\eta c_{\eta 1}})$ and $c_{v2} := \beta_h \varepsilon_h + \beta_\eta c_{\eta 4}$.

Since β_h and β_η can be chosen arbitrarily, we know c_{v2} can be set arbitrarily small, along with $\frac{L_1 L_\eta \beta_h c_{h2}}{\beta_\eta c_{\eta 1}}$. Meanwhile, we know $c_{\eta 3} < 1$. Thus, we can choose β_h and β_η such that $c_{\eta 3} + \frac{L_1 L_\eta \beta_h c_{h2}}{\beta_\eta c_{\eta 1}}$ is less than one. Meanwhile, by carefully choosing the PD gains in (35), the continuous-phase convergence rate c_{h3} can be sufficiently large such that $\frac{L_h c_{h2}}{c_{h1}} e^{-c_{h3}(T_{k+1}-T_k)}$ can be any positive number less than one. Thus, $c_{v1} < 1$ is ensured.

Since both c_{v1} and c_{v2} can be any arbitrarily small positive number with $c_{v1} < 1$, $V|_{k+1}^+ \leq V|_k^+$ is guaranteed for any $k \in \mathbb{N}$ and $\mathbf{X}(0) \in B_{r_v}(\mathbf{0})$ with $r_v = \min(r_d, r_\eta, r_\Delta, r_1)$. Thus, condition (S2) is met, which completes the proof. \square

VI. SIMULATION VALIDATION

This section presents simulations performed on a Digit robot in MATLAB, validating the proposed approach under various surface motions and uncertainties.

Simulation validation is necessary as it enables us to evaluate the proposed framework beyond the physical limitations of hardware experiments, such as the frequency and magnitude constraints of the DRS movement. The simulation video is available at: <https://youtu.be/NtAT0DFtMCY>.

A. Simulation Setup

1) *Setup of higher-layer planner:* Based on Digit's inertial and geometric properties, the parameters of the ALIP-DRS are set as: $H = 0.9$ m, $m = 46.1$ kg, $T_{step} = 0.4$ s,

TABLE I
SIMULATION CASES

Cases	DRS motion (m)	$\bar{L}_{y,S}$ (kg · m ² /s)
A	$x_S(t) = 0.04 \cos(\frac{2\pi}{0.4}t)$	4.1
B	$x_S(t) = 0.14 \cos(\frac{2\pi}{6}t)$	12.5
C	$y_S(t) = 0.06 \cos(\frac{2\pi}{0.72}t)$	0
D	$x_S(t) = 0.04 \cos(\frac{2\pi}{0.4}t)$ $y_S(t) = 0.1 \cos(\frac{2\pi}{6}t)$	6.27

and $W = 0.2$ m. As the ALIP-DRS model in (9) indicates, setting the desired angular momentum $\bar{L}_{y,S}$ is equivalent to setting the desired forward CoM velocity. For comprehensive assessment across various desired forward CoM velocities, different values of $\bar{L}_{y,S}$ are used, as specified in Table I.

2) *Setup of middle and lower layers:* In MATLAB simulations, the Bézier polynomials defining the desired full-dimensional trajectories ϕ are chosen as follows:

- ϕ_1 : Set to H so as to enforce the CoM height of the ALIP-DRS model on the full-order Digit model.
- ϕ_2 , ϕ_3 , and ϕ_4 : Set to 0 for a constant trunk orientation.
- ϕ_5 and ϕ_6 : Set according to Algorithm 1.
- ϕ_7 : Set with Bézier coefficients $[0, 0.02, 0.07, 0.15, 0.07, 0.02, 0]^\top$ to minimize the desired swing-foot height and reduce swing-foot motion.
- ϕ_8 , ϕ_9 , and ϕ_{10} : Set to 0 to ensure a proper swing-foot orientation at touchdowns.
- ϕ_{11} to ϕ_{18} : Set to 0 to keep the arms static.

In MATLAB, the model-based input-output linearizing controller is implemented using (35). The full-order dynamic matrices \mathbf{M} and \mathbf{c} are obtained using FROST [26]. For all cases, the control gains are set to $\mathbf{K}_p = 2500 \cdot \mathbf{I}$ and $\mathbf{K}_d = 100 \cdot \mathbf{I}$. To meet the stability condition in Theorem 2, these gains are tuned to ensure a sufficiently high continuous-phase convergence rate for the output function state \mathbf{x}_h .

3) *Simulation cases:* To assess the control framework under both a unity ratio and a real-number ratio between the DRS motion period and one walking-step duration, as well as under different DRS motion directions, four sets of DRS motions are simulated, as summarized in Table I. The detailed descriptions of the surface motions are:

(Case A): DRS sways in the sagittal plane with the same period as walking (i.e., $T_{step} = T_{x,DRS} = 0.4$ s).

(Case B): DRS sways in the sagittal plane with its least period 15 times that of walking (i.e., $15T_{step} = T_{x,DRS}$).

(Case C): DRS sways in the frontal plane with its least period related to the walking period as $9T_{step} = 5T_{y,DRS}$.

(Case D): DRS sways with different amplitudes and periods in the forward and lateral directions. The DRS-walking period ratios are $T_{x,DRS}/T_{step} = 1$ and $T_{y,DRS}/T_{step} = 15$.

In all cases, the robot faces towards the x -axis of the world frame. The initial movement status of Digit is quite standing.

B. Comparative Simulations

To highlight the effectiveness of the proposed control framework (denoted as the “DRS framework”), we compare

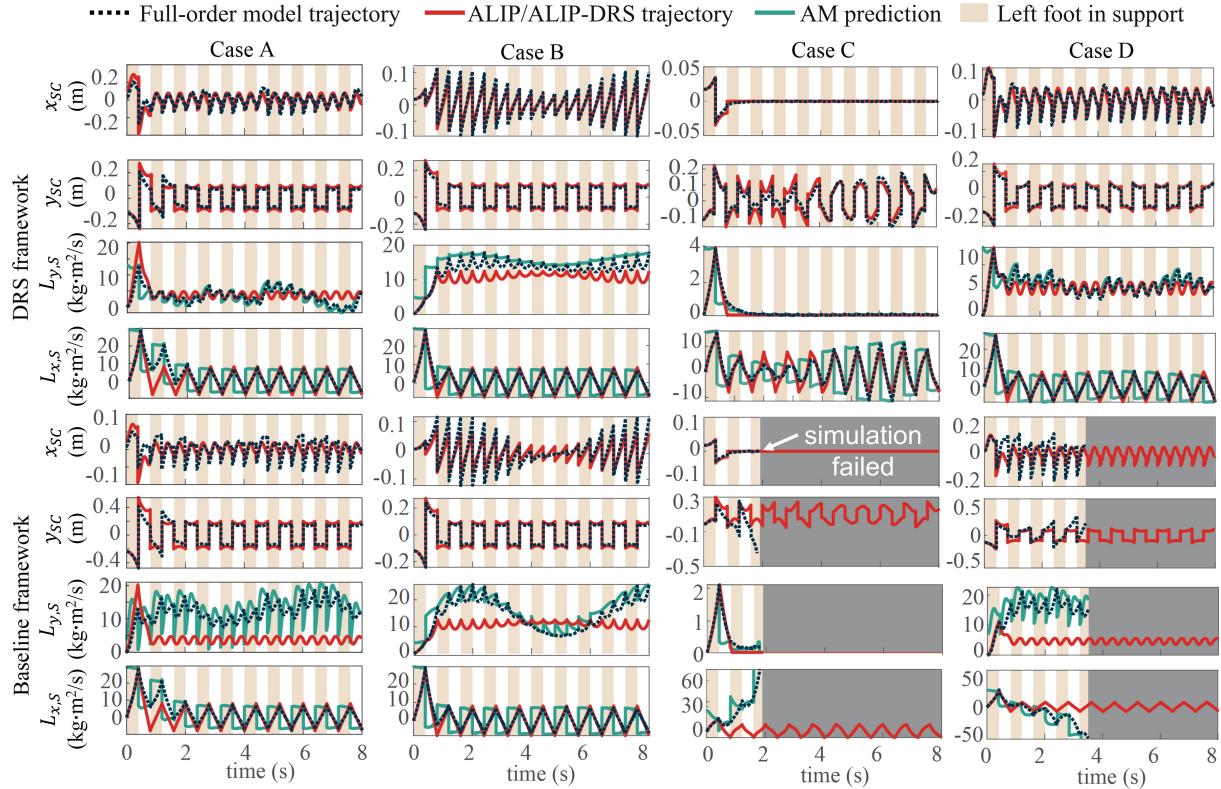


Fig. 8. Performance comparison between the proposed DRS framework and the baseline framework under simulation Cases A-D. The beige background highlights the left-foot-in-support phase, while the white background denotes the right-foot-in-support phase. Under the proposed framework, the robot's actual trajectories (labeled “Full-order model trajectory”) align closely with the ALIP-DRS trajectories, and the prediction of angular momentum (AM) at the end of the current step (green curve) is relatively accurate. In contrast, the robot operating under the baseline framework exhibits instability in Cases C and D. Additionally, significant discrepancies are observed between the actual trajectory of the contact angular momentum $L_{y,S}$ and the ALIP trajectory, and the prediction of $L_{y,S}$ at the end of the current step lacks the accuracy seen with the proposed framework.

it with a previous ALIP-based framework [18] designed for static terrain (denoted as the “baseline framework”). Figure 8 displays the simulation results under Cases A-D.

1) Baseline framework setup: The two frameworks have the same setup of middle and lower layers, including control variable selection, full-body trajectory parameterization, and the use of an input-output linearizing controller as the lower layer. The controller’s PD gains are tuned to guarantee accurate full-body trajectory tracking. The primary difference in the setup lies in the higher-layer planner: the DRS framework explicitly accounts for the DRS motion whereas the baseline framework assumes a static terrain.

2) Comparison on walking stability: As Fig. 8 indicates, the proposed DRS framework drives the full-dimensional trajectories (marked by black, dashed lines) significantly more closely to the desired trajectories (marked by red, solid lines), as compared with the baseline framework. The DRS framework also ensures stable walking across all four cases. Yet, the baseline framework fails to sustain stable walking in Cases C and D, causing the robot to fall over laterally.

3) Comparison on angular momentum prediction: The green curves in Fig. 8 show the predicted angular momentum at the end of the current walking step. As explained in Sec. III, at each time step within the current walking step, the higher-layer planner uses (14) and (27) and the robot’s current state to compute the predicted angular momentum at the end of the current walking step. The baseline framework

TABLE II
FORWARD VELOCITY REGULATION PERFORMANCE COMPARISON

Cases	Desired forward velocity (m/s)	DRS framework error (m/s)	Baseline framework error (m/s)
A	0.1	0.005	0.32
B	0.3	0.13	0.32
C	0	0.002	not stable
D	0.15	0.01	not stable

uses the same equations but with zero ground velocities.

Under the DRS framework, the predicted value of the contact angular momentum $L_{y,S}$ at the end of the current walking step (highlighted in green) is relatively consistent within each walking step and across different steps. In contrast, the baseline framework’s prediction varies significantly. This improved angular momentum prediction of the proposed framework leads to less variation in the planned foot placement location during a walking step. Thus, the desired swing-foot trajectories tracked by the lower-layer controller are smoother, causing the better tracking performance of the proposed framework compared to the baseline framework.

4) Comparison on velocity regulation accuracy: Table II shows the tracking errors for the forward CoM velocity under the proposed and baseline frameworks across Cases A-D. It also provides the desired forward CoM velocity for each case, corresponding to the desired contact angular momentum $L_{y,S}$ outlined in Table I. As indicated in the table,

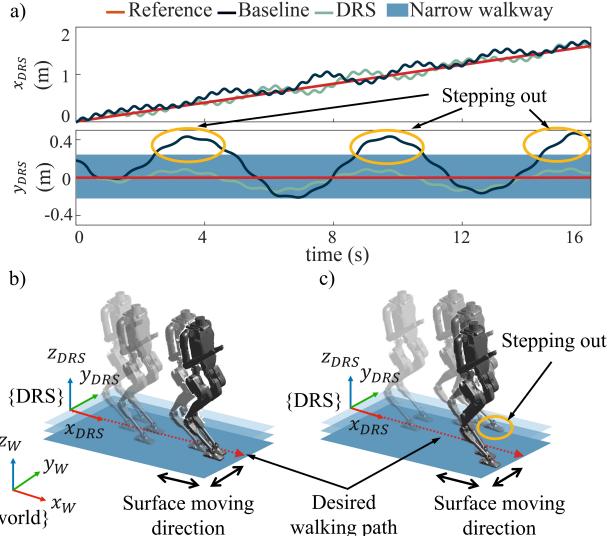


Fig. 9. Comparison of base-position tracking accuracy between the baseline and DRS frameworks under Case D. a) The DRS framework demonstrates higher tracking accuracy in the x_{DRS} -direction compared with the baseline framework, allowing the robot to remain within the confines of the narrow walkway. b) Snapshots depict stable walking by Digit using the proposed DRS framework. c) Under the baseline framework, Digit steps outside the designated walkway, illustrating reduced tracking accuracy.

the proposed framework achieves velocity tracking errors ranging from 0.005 m/s to 0.13 m/s across the four cases. In contrast, the baseline framework shows larger errors, reaching 0.32 m/s in Cases A and B. For Cases C and D, the baseline framework's errors are not reported due to its inability to maintain stable walking in these scenarios.

5) Comparison on path tracking: By adjusting the desired angular momenta $\bar{L}_{x,S}$ and $\bar{L}_{y,S}$, our framework enables the robot to track the desired position trajectory of its base (i.e., trunk). Accurate base-position tracking is critical for collision avoidance in real-world DRS environments, such as ships, which narrow walkways are common.

Using the mapping from the horizontal CoM velocities to the contact angular momenta given in (9) and (10), and by including a position-based feedback term, the desired contact angular momenta $\bar{L}_{x,S}$ and $\bar{L}_{y,S}$ are adjusted as follows:

$$\begin{aligned}\bar{L}_{x,S} &= K_y(y_b - y_{b,d}) + mH\dot{y}_{b,d} \text{ and} \\ \bar{L}_{y,S} &= K_x(x_b - x_{b,d}) + mH\dot{x}_{b,d},\end{aligned}$$

where x_b and y_b represent the actual base positions along the x - and y -axes of the DRS frame, respectively, $x_{b,d}$ and $y_{b,d}$ denotes the corresponding desired positions, and K_x and K_y are control gains assigned by the user.

Figure 9 depicts the base-tracking performance of the Digit robot navigating through a narrow pathway under Case D. The DRS framework allows Digit to closely track both the desired walking path and the desired forward velocity within the DRS frame. However, the robot fails to stay close to the desired path under the baseline framework.

C. Robustness Assessment

The robustness of the proposed framework is assessed under various uncertainties.

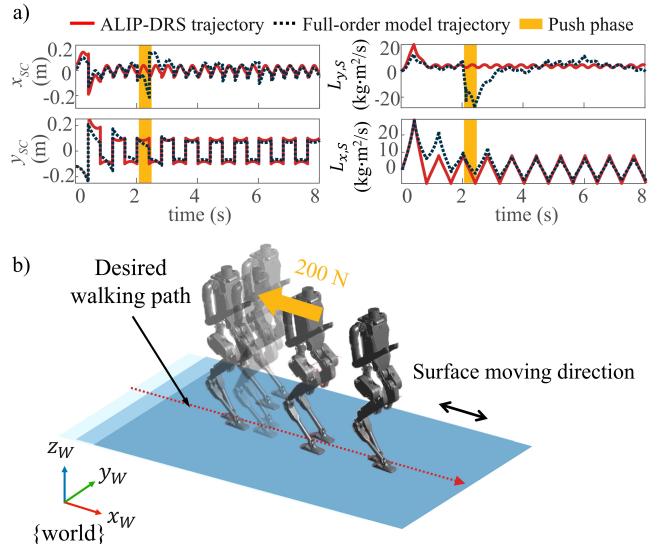


Fig. 10. Robustness evaluation results of the proposed framework under an unknown sudden push lasting 0.1 seconds, with DRS motion and desired robot velocity consistent with Case A, as detailed in Tables I and III. a) Time evolution of the CoM position and contact angular momentum trajectories, demonstrating fast convergence following the push. b) Illustration of stable robot walking before, during, and after push.

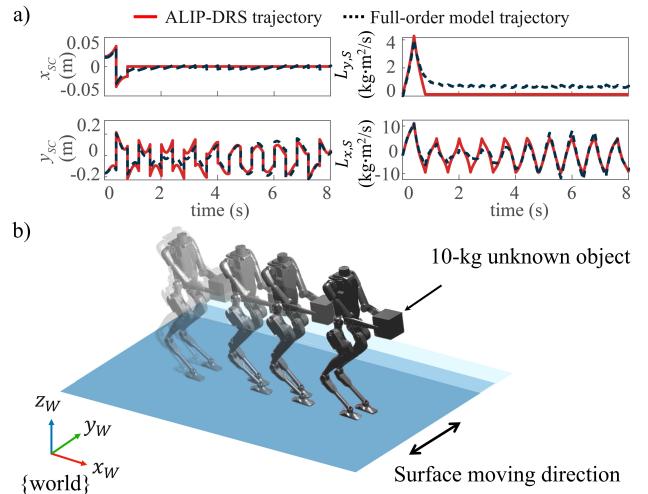


Fig. 11. Robustness evaluation results under an unknown 10-kg load, with DRS motion and desired robot velocity matching Case C, as outlined in Tables I and III. Subplot a) displays the time evolution of the CoM position (x_{SC} , y_{SC}) and contact angular momenta ($L_{x,S}$, $L_{y,S}$) for both the full-order robot model and the ALIP-DRS model, highlighting their close alignment and ALIP-DRS's accuracy even with an unknown external load. Subplot b) illustrates that the robot maintains stable walking despite forward position drift induced by the unknown load.

1) Sudden push: An unknown sudden force of 200 N is horizontally applied to Digit's chest under Case A. As shown in Fig. 10, the Digit robot is initially heavily affected by this unexpected push. However, after taking four steps, Digit is able to converge back to a close neighborhood of the desired trajectories, demonstrating the robustness of the proposed framework against unknown pushes.

2) Unknown load: Figure 11 presents the results for Case C, where the robot is carrying a box with an unknown weight of 10 kg. The desired behavior for the robot is to walk in place on the DRS. Due to the substantial unknown load,

TABLE III
ACTUAL WALKING SPEED (M/S) UNDER VARIOUS DRS MOTION UNCERTAINTIES δ_A AND δ_t

δ_t (s)	0	0.13	0.26	0.4
δ_A (m)	0	0.1005	0.5560	0.6338
0	0.1005	0.5560	0.6338	0.2837
0.013	-0.0007	0.5856	0.7521	-0.0150
0.026	-0.1024	0.5259	0.7013	-0.1703
0.04	-0.1145	0.6302	0.7408	-0.2001

the actual contact angular momentum $L_{y,S}$ shows a constant error of $0.7 \text{ kg}\cdot\text{m}^2/\text{s}$, as depicted in subplot a. This results in a forward position drift, as illustrated in subplot b. Still, the robot maintains stable walking throughout the simulation, despite the significant unknown load.

3) *DRS motion uncertainty*: To evaluate the robustness of the proposed framework under inaccurately known surface motions, the framework is fed with incorrect DRS motion under Case A. The tested inaccurate DRS motion $\bar{x}_s(t)$ is:

$$\bar{x}_s(t) = (0.04 + \delta_A) \cos(\frac{2\pi}{0.4}(t + \delta_t)), \quad (46)$$

where δ_A and δ_t are unknown offsets.

Table III summarizes the speed regulation performance under different levels of DRS motion uncertainty in amplitude δ_A and phase δ_t . Note that the maximum values of the uncertainties are relatively substantial, with $\delta_A = 0.04 \text{ m}$, which is the same as the nominal DRS motion amplitude, and $\delta_t = 0.4 \text{ s}$, which equals one walking-step duration. The shaded cells contain the actual walking speed corresponding to different uncertainty levels. In Case A, the desired forward speed is 0.1 m/s . Without any uncertainties ($\delta_A = \delta_t = 0$), the actual forward speed is 0.1005 m/s , demonstrating the close alignment with the desired value. Although the speed regulation performance degrades as the DRS motion uncertainty increases, Digit maintains stable walking in all cases.

VII. HARDWARE EXPERIMENT VALIDATION

This section presents the hardware experiment results that demonstrate the stability, tracking performance, and robustness of underactuated walking under the proposed DRS framework. A video of the experiment is available at: <https://youtu.be/NtAT0DFtMCY>.

Due to hardware constraints, such as the limited surface area and motion capabilities of the physical DRS, the locomotion tasks and DRS motions implemented in the experiments differ from those in the simulation cases. Still, the framework is tested under different DRS motion frequencies and directions, and is also evaluated in the presence of various uncertainties, including sudden pushes and uncertain surface motions.

A. Experiment Setup

1) *DRS motions*: A Motek M-Gait treadmill is used as the DRS (Fig. 12). Due to the movement limits of this treadmill, the experimentally implemented DRS motions are less aggressive than those in the simulations. This treadmill's walking area is $1 \text{ m} \times 2 \text{ m}$ (width \times length), and has a maximum sway amplitude of 5 cm . As the treadmill is not

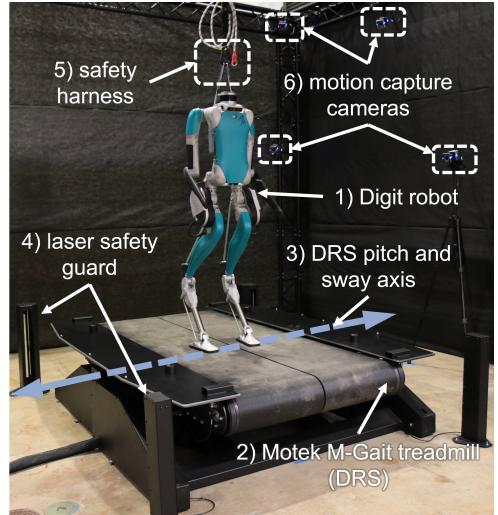


Fig. 12. The experiment setup includes: 1) Digit robot, 2) Motek M-Gait treadmill (serving as DRS), equipped with 3) a pitch and sway axis, 4) a laser safety guard, 5) safety harness, and 6) motion capture cameras.

TABLE IV
EXPERIMENT CASES

Cases	DRS expressions (m)	$\bar{L}_{y,S}$ ($\text{kg}\cdot\text{m}^2/\text{s}$)
A	$y_S(t) = 0.04 \cos(\frac{2\pi}{6.8}t)$	0
B	$y_S(t) = 0.04 \cos(\frac{2\pi}{5.6}t)$	0
C	$x_S(t) = 0.04 \cos(\frac{2\pi}{6.8}t)$	0
D	$x_S(t) = 0.04 \cos(\frac{2\pi}{5.6}t)$	0

capable of moving with a period as short as the robot gait cycle, which is approximately 0.4 s , we focus on testing DRS motions with a period multiple times that of the gait. Also, the treadmill can execute linear motions with a changing direction only along a single axis, which is the “sway axis” in Fig. 12. Thus, to generate DRS motions in the frontal (or sagittal) planes of the Digit robot, the robot is placed to face perpendicular (or parallel) to the treadmill’s sway axis.

2) *Experiment cases*: To evaluate the robot performance under different surface motions, four experiment cases are considered, as listed in Table IV. In all cases, the desired contact angular momentum in the sagittal plane, $\bar{L}_{y,S}$, is set to zero to enable walking-in-place for ensuring that the robot moves within the limited walking region of the treadmill.

3) *Implementation of the proposed DRS control framework*: The proposed DRS framework is implemented on the Digit robot using C++ within the ROS platform. To inform the control framework with the robot’s trunk pose and linear velocity, which are not directly measured by Digit’s onboard sensors, we utilize Digit’s proprietary state estimator considering its reasonable accuracy under the tested DRS motions.

For the lower-layer controller, instead of employing input-output linearizing control as in the simulations, the proposed inverse kinematics control is implemented. To ensure a sufficient accuracy in tracking the desired full-body trajectories ϕ , the control gain κ is set as a diagonal matrix with $[10, 30, 30, 5, 18, 45, 20, 0.01, 7, 10, \mathbf{1}_{1 \times 8}]$ as its diagonal. Here, $\star_{m \times n}$ denotes an $m \times n$ matrix whose elements all take the value of \star . Furthermore, the proportional

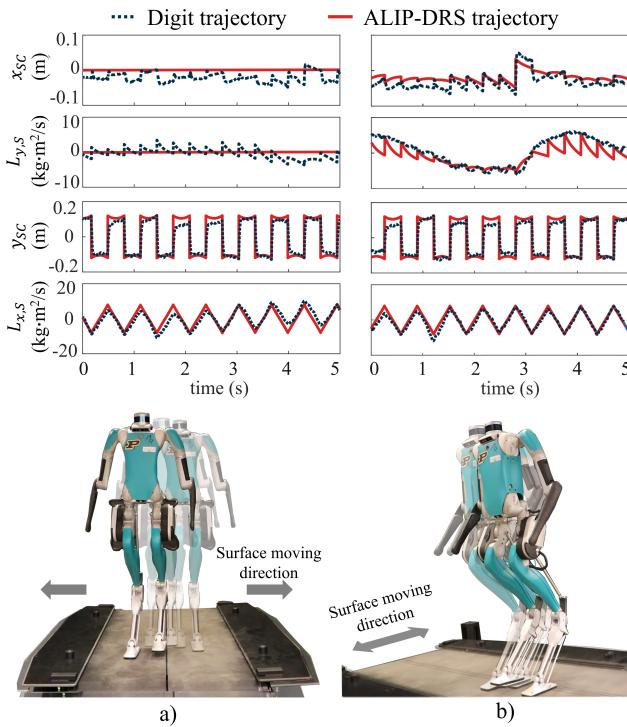


Fig. 13. Experimental trajectory tracking results under two different DRS motions: a) Case B and b) Case D. In Case B, where the desired robot speed is zero (stepping in place), the desired ALIP-DRS trajectories for x_{SC} and y_{SC} are set to zero. Digit's actual trajectories closely match the ALIP-DRS trajectories, demonstrating stable walking and ALIP-DRS's accuracy.

gain $\bar{\mathbf{K}}_p$ in (39) is chosen as a diagonal matrix whose diagonal values depend on the support foot. This diagonal is $100 \cdot [25_{1 \times 2}, 35, 25, 5_{1 \times 4}, 25_{1 \times 2}, 35, 25_{1 \times 3}, 5_{1 \times 4}]$ and $100 \cdot [25_{1 \times 2}, 35, 25_{1 \times 3}, 5_{1 \times 4}, 25_{1 \times 2}, 35, 25, 5_{1 \times 4}]$ for the left- and right-leg-in-support phases, respectively. Similar to the tuning of κ , $\bar{\mathbf{K}}_p$ is set to achieve accurate tracking. The derivative gain $\bar{\mathbf{K}}_d$ is set to zero because Digit's internal joint-space impedance controller, which is provided by the manufacturer, acts as a derivative control term.

The same set of control gains are used across all four cases to demonstrate the effectiveness of our proposed framework on hardware without extensive parameter tuning.

B. Experiment Results

This subsection presents the experimental results about walking stability, trajectory tracking, and robustness.

1) Walking stability and trajectory tracking of unactuated variables: Figure 13 illustrates the trajectory tracking results for the CoM position (x_{SC} , y_{SC}) and contact angular momenta ($L_{x,S}$, $L_{y,S}$) in Cases B and D, where the DRS sways in the robot's lateral and sagittal planes, respectively. As described in Secs. IV and V, these variables are not directly commanded by the lower-layer controller and correspond to the unactuated state \mathbf{x}_η . The figure shows that these unactuated variables remain close to the ALIP-DRS's solution, which serves as their desired trajectory as explained in Sec. V. This confirms that the proposed framework ensures stable underactuated walking during the various DRS motions in Cases A-D, and validates the accurate tracking of the actual unactuated variables.

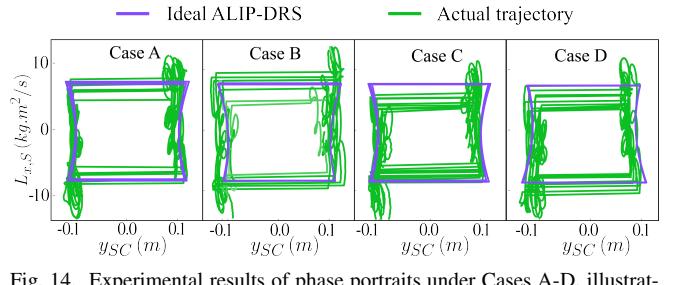


Fig. 14. Experimental results of phase portraits under Cases A-D, illustrating the close alignment between the actual Digit motion and the ideal ALIP-DRS trajectories. This alignment further confirms the reasonable accuracy of ALIP-DRS and validates that the proposed framework enables reliable trajectory tracking for the unactuated state \mathbf{x}_η across various DRS motions.

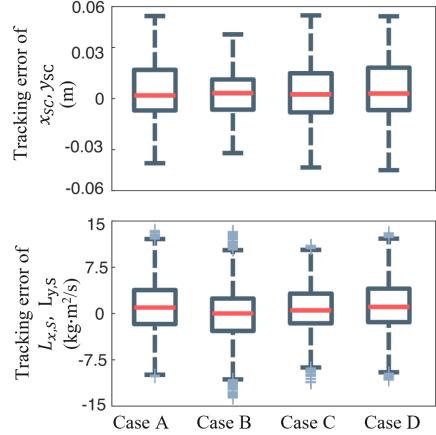


Fig. 15. Experimental results showing trajectory discrepancies between the actual robot and the ALIP-DRS model from Cases A to D. The top and bottom graphs display the combined errors in the CoM positions (x_{SC} , y_{SC}) and angular momenta ($L_{y,S}$, $L_{x,S}$), respectively.

Additionally, Fig. 14 shows the sagittal-plane phase portrait of both the ideal ALIP-DRS model and the actual robot under Cases A-D. Note that the ideal ALIP-DRS phase portrait exhibits multiple loops, as the solution period T_{sys} of the ALIP-DRS is several times the walking-step duration T_{step} . The close alignment between the two sets of phase portraits further demonstrates the walking stability and indicates the relative accuracy of the ALIP-DRS model.

The trajectory tracking errors of the unactuated variables under Cases A-D are presented as box-and-whisker plots in Fig. 15. In each plot, the red central mark represents the mean value, while the bottom and top edges of the box correspond to the 25th and 75th percentiles, respectively. Outliers are indicated by light blue “+” symbols. The mean values of the CoM position and contact angular momentum errors are close to zero across all four cases, with relatively small upper and lower quartiles that show consistent values across the cases. These relatively small tracking errors align with the results shown in Figs. 13 and 14, demonstrating the high tracking accuracy achieved by the proposed framework.

2) Trajectory tracking for directly commanded variables: The full-body trajectory tracking results for the directed commanded control variables \mathbf{h}_c are provided in Fig. 16. The CoM height z_{SC} shows slight oscillations around the desired height of 92 cm, with an average tracking error of 1 cm. Note that the desired CoM height trajectory in red is the

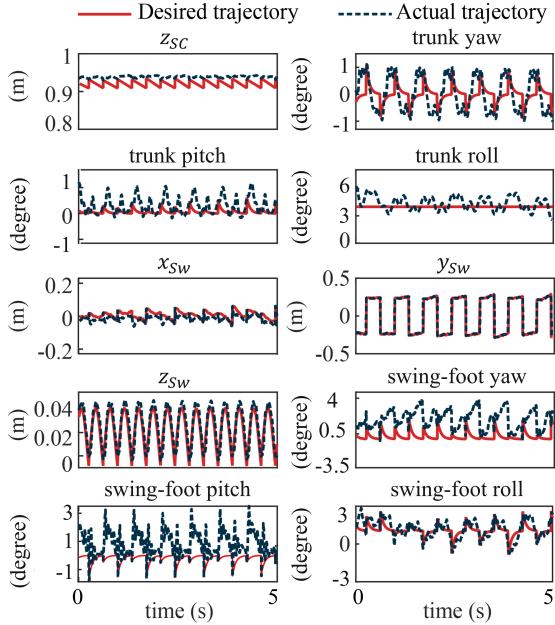


Fig. 16. Experiment results of task-space tracking for the directly commanded state \mathbf{x}_h under Case B. The CoM height z_{SC} closely matches the desired height, indicating that the actual robot effectively adheres to the constant CoM height assumption of the ALIP-DRS model. Also, the relatively small tracking error in the desired swing-foot pose confirms accurate foot placement, demonstrating faithful execution of the discrete footstep control designed based on the ALIP-DRS model.

modified desired trajectory. Its value at the beginning of a walking step is matched with the actual CoM height, which helps reduce the jumps in the tracking error immediately after the foot-landing events and thus avoids excessively high joint torques. The swing-foot position tracking is accurate in all directions, with a peak tracking error of 5.5 cm and an average tracking error of 1.2 cm. While the tracking of the swing-foot orientation is not highly accurate, the average error in each direction is within 2° , which is sufficient to ensure a proper foot posture before swing-foot touchdowns.

3) Robustness to pushes and unknown surface motion: The robustness of the proposed DRS framework is assessed against three types of external disturbances, which are: (i) sudden pushes applied by a long stick; (ii) uncertain forward and lateral DRS motions induced by orienting Digit away from its nominal facing direction by 45° ; and (iii) unknown pitching DRS motion with an amplitude of 1° and frequency of 10 s. These uncertainties are tested during a treadmill sway motion with an amplitude of 0.05 m and a period of 4 s, which is the same as the treadmill motion during which Digit's proprietary controller failed to sustain stable walking as shown in the top row of Fig. 1.

The phase portraits of the lateral CoM position y_{SC} and contact angular momentum $L_{x,S}$ under these three types of uncertainties are given in Fig. 17. Figure 18 shows the corresponding snapshots of Digit walking, highlighting that the proposed control framework maintains stable walking despite the uncertainties. The phase portraits in Fig. 17 indicate that in the presence of the uncertainties, the robot's actual trajectories exhibit a larger deviation from the ideal phase portraits of the ALIP-DRS, compared to the scenarios

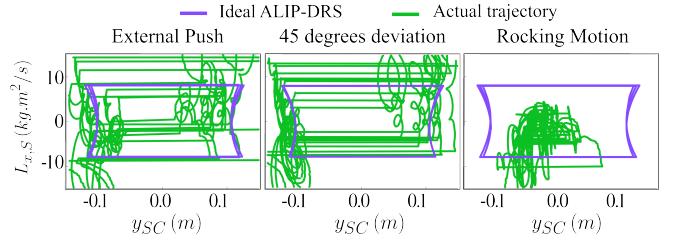


Fig. 17. Experiment results of phase portraits under varying uncertainties, including external pushes (left), unknown forward motion and uncertain lateral motion of the DRS (middle), and unknown DRS pitch motion (right).

without these uncertainties shown in Fig. 14. This is primarily due to the higher model inaccuracies induced by the unknown surface motions and external force disturbances.

In particular, under the unknown treadmill rocking motion, the robot's actual phase portraits notably deviate from the ALIP-DRS's, as shown in Fig. 17. This deviation may be due to the large modeling errors caused by the unmodeled vertical ground motions. The relative CoM height above the support point is assumed to be constant in the ALIP-DRS model, but a rocking motion can introduce a noticeable vertical CoM velocity. Moreover, the treadmill's rocking motion can introduce non-negligible centroidal momentum \mathbf{L}_{CoM} , which is assumed to be zero in the ALIP-DRS model. Still, despite the degraded tracking performance compared to the nominal cases, the robot successfully maintains stable walking and steps in place near the treadmill's center under all tested uncertainties. Potential solutions that address complex surface motions based on the current work are discussed in the next section.

VIII. DISCUSSION

This section discusses the capabilities and limitations of the proposed control approach, along with potential directions for future research.

To control underactuated bipedal walking on swaying DRS, this study introduces a new reduced-order robot model, ALIP-DRS, to approximate the associated hybrid, nonlinear, time-varying, nonhomogeneous unactuated robot dynamics. The key novelty of this model lies in its explicit consideration of the DRS sway motion. Due to this sway, the ALIP-DRS is nonhomogeneous, fundamentally distinguishing it from classical LIP models and their variations, such as the time-invariant H-LIP [35] and ALIP [18] and the time-varying HT-LIP [49] and variable-height LIP [14], which are all homogeneous. The close alignment between Digit's unactuated movement variables and the ALIP-DRS's trajectories is confirmed through both simulations (Figs. 8, 10, and 11) and hardware experiments (Figs. 13, 14, 15, and 17), highlighting the reasonable accuracy of the ALIP-DRS. This model's generality allows for extensions to other legged robot platforms beyond humanoids.

In addition, the stability for the nonhomogeneous closed-loop ALIP-DRS model under a discrete footstep control law is provably verified based on the stability of its homogeneous portion. The previously developed stability conditions

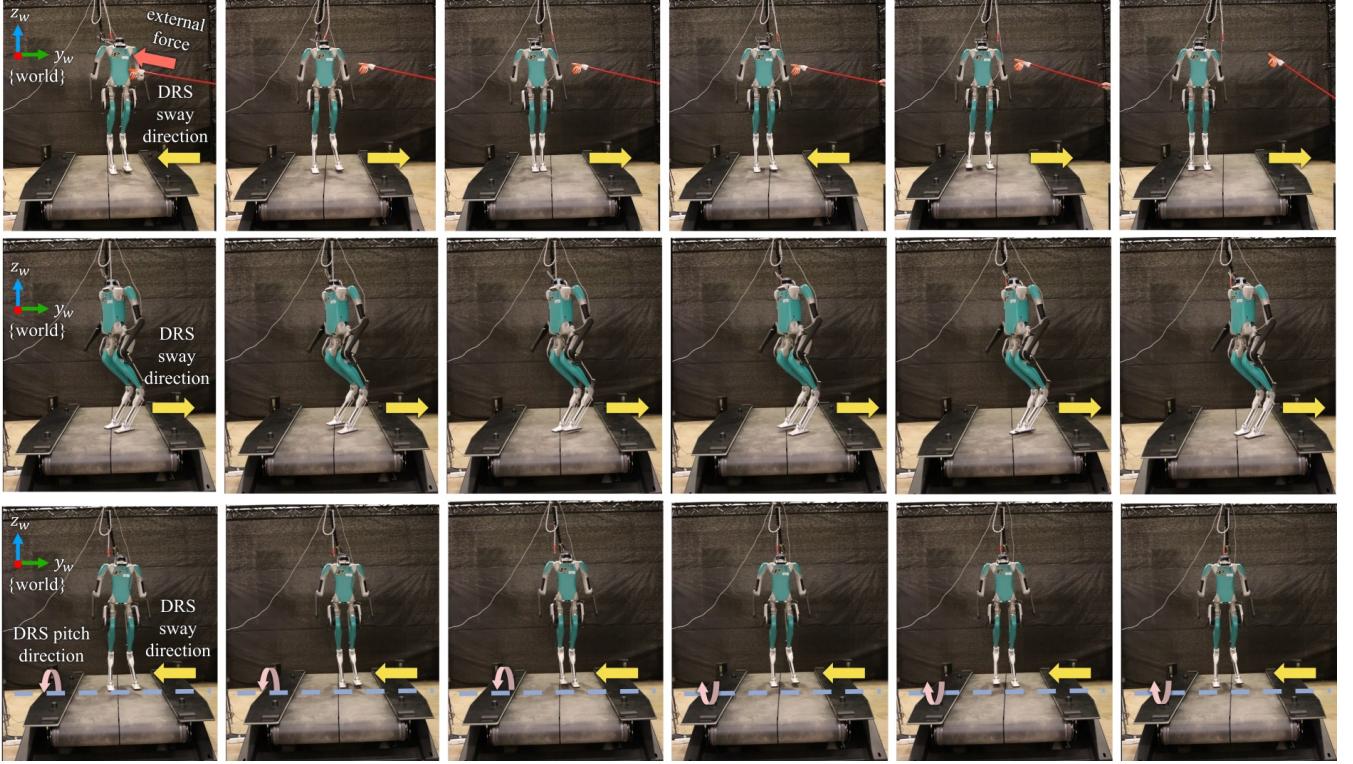


Fig. 18. Illustrations of the Digit and DRS motions under different uncertainty tests. Top: Digit steps in place on swaying DRS while experiencing unknown sudden pushes. Middle: Due to Digit's heading direction at 45° to the DRS sway direction, it faces uncertain DRS motions in both its sagittal and frontal planes. Bottom: Digit steps in place experiencing both sway and unknown pitch motions of the DRS. Note that although the treadmill sways left and right in all experiments, the snapshots in the middle and bottom rows capture moments when the treadmill's movement points in the same direction.

for existing LIP models, which are homogeneous, are not directly applicable to the nonhomogeneous ALIP-DRS. The proposed stability analysis, based on the general stability theory of linear impulsive nonhomogeneous systems, can be generalized to continuous-phase control laws for pendulum models as well as alternative state variable choices beyond those employed in this study.

The study also introduces a hierarchical control framework that incorporates the ALIP-DRS's foot-placement controller as its higher-layer footstep planner. By stabilizing the ALIP-DRS, the discrete control law indirectly stabilizes the actual unactuated dynamics of the robot during walking. The stability conditions for the complete full-order system under the control framework are also provided. Due to their general applicability, this stability analysis can be extended to prior hierarchical control methods based on LIP models [18], [35]. Experimental results confirm that the proposed framework effectively achieves stable underactuated walking under various DRS sway motions and uncertainties.

Building on these theoretical findings, future work will extend the stability conditions of the ALIP-DRS model to accommodate DRS motions in both vertical and horizontal directions. In the presence of vertical and horizontal surface motions, the hybrid ALIP-DRS model (13) will be updated with a time-varying matrix A_x , whose explicit time dependence is induced by vertical surface motion [50]. The stability conditions for this generalized ALIP-DRS model will be developed by integrating the stability analysis from

this study with that for locomotion control on vertical surface motions. This future research will address the observed deviation between the actual robot state and the ideal ALIP-DRS model during unknown DRS pitching motions (Fig. 17). Moreover, future efforts will explore real-time surface motion estimation using onboard robot sensors, so as to enable the proposed hierarchical control framework to adapt more effectively to unknown surface motions and further enhance locomotion robustness.

IX. CONCLUSION

This paper has introduced a real-time, provably stabilizing control approach that employs a new reduced-order robot model to enable underactuated bipedal walking on a swaying dynamic rigid surface (DRS). The classical angular-momentum-based linear inverted pendulum (ALIP) model was analytically extended to explicitly consider the time-varying DRS motion, resulting in a time-varying, nonhomogeneous system, which was referred to as the ALIP-DRS model. A discrete-time footstep control strategy was constructed, which was proven to be exponentially stabilizing for the ALIP-DRS model. With the ALIP-DRS footstep controller serving as a motion planner that generates the desired swing-foot landing positions in real-time, a hierarchical control framework was developed for the full-order robot model of an underactuated walking robot. The closed-loop stability conditions for the hybrid full-order control system were constructed by analyzing both the unactuated and fully

actuated subsystems under the proposed control framework. Both numerical simulations and hardware experiments on a Digit humanoid robot confirmed that the proposed ALIP-DRS model is reasonably accurate in describing the unactuated dynamics of the actual robot. Further, the proposed framework achieved stable walking under various DRS sway motions, and demonstrated high robustness under different uncertainties including sudden pushes and uncertain DRS motions in different directions.

ACKNOWLEDGMENT

The authors would like to thank I-Chia Chang for supporting the hardware experiments and providing constructive comments on this paper.

REFERENCES

- [1] J.-Y. Kim, I.-W. Park, and J.-H. Oh, “Walking control algorithm of biped humanoid robot on uneven and inclined floor,” *J. Intel. Rob. Syst.*, vol. 48, pp. 457–484, 2007.
- [2] S. Caron, A. Kheddar, and O. Tempier, “Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control,” in *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 277–283, 2019.
- [3] A. Pajon, S. Caron, G. De Magistri, S. Miossec, and A. Kheddar, “Walking on gravel with soft soles using linear inverted pendulum tracking and reaction force distribution,” in *Proc. IEEE-RAS Int. Conf. Humanoid Rob.*, pp. 432–437, 2017.
- [4] C. E. Bauby and A. D. Kuo, “Active control of lateral balance in human walking,” *J. Biomech.*, vol. 33, no. 11, pp. 1433–1440, 2000.
- [5] D. Kim, J. Lee, J. Ahn, O. Campbell, H. Hwang, and L. Sentis, “Computationally-robust and efficient prioritized whole-body controller with contact constraints,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 1–8, 2018.
- [6] M. Vukobratović, B. Borovac, and V. Potkonjak, “Zmp: A review of some basic misunderstandings,” *Int. J. Humanoid Rob.*, vol. 3, no. 02, pp. 153–175, 2006.
- [7] F. Firmani and E. J. Park, “Theoretical analysis of the state of balance in bipedal walking,” *J. Biomechan. Engr.*, vol. 135, no. 4, p. 041003, 2013.
- [8] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, vol. 1, pp. 239–246, 2001.
- [9] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, “A closed-form solution for real-time zmp gait generation and feedback stabilization,” in *Proc. IEEE-RAS Int. Conf. Humanoid Rob.*, pp. 936–940, 2015.
- [10] T. Sugihara, K. Imanishi, T. Yamamoto, and S. Caron, “3d biped locomotion control including seamless transition between walking and running via 3d zmp manipulation,” in *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 6258–6263, 2021.
- [11] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *Proc. IEEE-RAS Int. Conf. Humanoid Rob.*, pp. 200–207, 2006.
- [12] A. L. Hof, M. Gazendam, and W. Sinke, “The condition for dynamic stability,” *J. Biomech.*, vol. 38, no. 1, pp. 1–8, 2005.
- [13] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, “Bipedal walking control based on capture point dynamics,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 4420–4427, 2011.
- [14] S. Caron, A. Escande, L. Lanari, and B. Mallein, “Capturability-based pattern generation for walking with variable height,” *IEEE Trans. Rob.*, vol. 36, no. 2, pp. 517–536, 2019.
- [15] T. Takenaka, T. Matsumoto, and T. Yoshiike, “Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 1084–1091, 2009.
- [16] J. Englsberger, C. Ott, and A. Albu-Schäffer, “Three-dimensional bipedal walking control using divergent component of motion,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 2600–2607, 2013.
- [17] J. Englsberger, C. Ott, and A. Albu-Schäffer, “Three-dimensional bipedal walking control based on divergent component of motion,” *IEEE Trans. Rob.*, vol. 31, no. 2, pp. 355–368, 2015.
- [18] Y. Gong and J. W. Grizzle, “Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion,” *ASME J. Dyn. Sys., Meas., Contr.*, vol. 144, no. 12, p. 121006, 2022.
- [19] S. H. Collins and A. Ruina, “A bipedal walking robot with efficient and human-like gait,” in *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 1983–1988, 2005.
- [20] J. Reher and A. D. Ames, “Dynamic walking: Toward agile and efficient bipedal robots,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 535–572, 2021.
- [21] E. R. Westervelt, C. Chevallereau, J. H. Choi, B. Morris, and J. W. Grizzle, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.
- [22] I. Poulakakis and J. W. Grizzle, “The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper,” *IEEE Trans. Autom. Contr.*, vol. 54, no. 8, pp. 1779–1793, 2009.
- [23] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, “A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel,” *Int. J. Rob. Res.*, vol. 30, no. 9, pp. 1170–1193, 2011.
- [24] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, “Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel,” *Int. J. Rob. Res.*, vol. 32, no. 3, pp. 324–345, 2013.
- [25] A. Ramezani, J. W. Hurst, K. Akbari Hamed, and J. W. Grizzle, “Performance analysis and feedback control of atrias, a three-dimensional bipedal robot,” *ASME J. Dyn. Syst. Meas. Contr.*, vol. 136, no. 2, p. 021012, 2014.
- [26] A. Hereid and A. D. Ames, “FROST: Fast robot optimization and simulation toolkit,” in *Proc. Int. Conf. Intel. Rob. Syst.*, 2017.
- [27] Y. Gao and Y. Gu, “Global-position tracking control of a fully actuated NAO bipedal walking robot,” in *Proc. Amer. Contr. Conf.*, pp. 4596–4601, 2019.
- [28] V. Paredes and A. Hereid, “Dynamic locomotion of a lower-limb exoskeleton through virtual constraints based zmp regulation,” in *Proc. ASME Dyn. Syst. Contr. Conf.*, vol. 84270, p. V001T14A001, American Society of Mechanical Engineers, 2020.
- [29] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, “Optimization-based control for dynamic legged robots,” *IEEE Trans. Rob.*, 2023.
- [30] Y. Gao and Y. Gu, “Global-position tracking control of multi-domain planar bipedal robotic walking,” in *Proc. ASME Dyn. Syst. Contr. Conf.*, vol. 59148, p. V001T03A009, 2019.
- [31] M. S. Motahar, S. Veer, and I. Poulakakis, “Composing limit cycles for motion planning of 3d bipedal walkers,” in *Proc. IEEE Conf. Dec. Contr.*, pp. 6368–6374, 2016.
- [32] R. Hartley, X. Da, and J. W. Grizzle, “Stabilization of 3d underactuated biped robots: Using posture adjustment and gait libraries to reject velocity disturbances,” in *Proc. IEEE Conf. Control Tech. Appl.*, pp. 1262–1269, 2017.
- [33] Q. Nguyen, A. Agrawal, X. Da, W. C. Martin, H. Geyer, J. W. Grizzle, and K. Sreenath, “Dynamic walking on randomly-varying discrete terrain with one-step preview,” in *Rob Sc. Syst.*, vol. 2, pp. 384–399, 2017.
- [34] X. Xiong and A. D. Ames, “Orbit characterization, stabilization and composition on 3d underactuated bipedal walking via hybrid passive linear inverted pendulum model,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 4644–4651, 2019.
- [35] X. Xiong and A. Ames, “3-d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization,” *IEEE Trans. Rob.*, vol. 38, no. 4, pp. 2405–2425, 2022.
- [36] X. Xiong and A. Ames, “Slip walking over rough terrain via h-lip stepping and backstepping-barrier function inspired quadratic program,” *IEEE Rob. Autom. L.*, vol. 6, no. 2, pp. 2122–2129, 2021.
- [37] M. Dai, X. Xiong, and A. Ames, “Bipedal walking on constrained footholds: Momentum regulation via vertical com control,” in *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 10435–10441, 2022.
- [38] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, “Mpc for humanoid gait generation: Stability and feasibility,” *IEEE/ASME Trans. Mechatron.*, vol. 36, no. 4, pp. 1171–1188, 2020.
- [39] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, “Terrain-adaptive, alip-based bipedal locomotion controller via model predictive control and virtual constraints,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 6724–6731, 2022.

- [40] B. Acosta and M. Posa, "Bipedal walking on constrained footholds with mpc footstep control," in *Proc. IEEE-RAS Int. Conf. Humanoid Rob.*, pp. 1–8, 2023.
- [41] Z. Gu, Y. Zhao, Y. Chen, R. Guo, J. K. Leestma, G. S. Sawicki, and Y. Zhao, "Robust-locomotion-by-logic: Perturbation-resilient bipedal locomotion via signal temporal logic guided model predictive control," *arXiv preprint arXiv:2403.15993*, 2024.
- [42] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. Van de Panne, "Feedback control for cassie with deep reinforcement learning," in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 1241–1246, 2018.
- [43] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Robust feedback motion policy design using reinforcement learning on a 3d digit bipedal robot," in *Proc. IEEE Int. Conf. Intel. Rob. Syst.*, pp. 5136–5143, 2021.
- [44] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 2811–2817, 2021.
- [45] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," *arXiv preprint arXiv:2105.08328*, 2021.
- [46] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Reinforcement learning-based cascade motion policy design for robust 3d bipedal locomotion," *IEEE Acc.*, vol. 10, pp. 20135–20148, 2022.
- [47] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through reinforcement learning," *arXiv preprint arXiv:2302.09450*, 2023.
- [48] A. Iqbal, Y. Gao, and Y. Gu, "Provably stabilizing controllers for quadrupedal robot locomotion on dynamic rigid platforms," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 4, pp. 2035–2044, 2020.
- [49] A. Iqbal, S. Veer, and Y. Gu, "Asymptotic stabilization of aperiodic trajectories of a hybrid-linear inverted pendulum walking on a vertically moving surface," in *Proc. Amer. Contr. Conf.*, pp. 3030–3035, 2023.
- [50] A. Iqbal, S. Veer, and Y. Gu, "Analytical solution to a time-varying lip model for quadrupedal walking on a vertically oscillating surface," *Mechatron.*, vol. 96, p. 103073, 2023.
- [51] A. Iqbal, S. Veer, and Y. Gu, "DRS-LIP: Linear inverted pendulum model for legged locomotion on dynamic rigid surfaces," *arXiv preprint arXiv:2202.00151*, 2022.
- [52] Y. Gao, Y. Gong, V. Paredes, A. Hereid, and Y. Gu, "Time-varying alip model and robust foot-placement control for underactuated bipedal robotic walking on a swaying rigid surface," in *Proc. Amer. Contr. Conf.*, pp. 3282–3287, 2023.
- [53] D. Bainov and P. Simeonov, *Impulsive differential equations: periodic solutions and applications*, vol. 66. CRC Press, 1993.
- [54] Y. Gong and J. Grizzle, "One-step ahead prediction of angular momentum about the contact point for control of bipedal locomotion: Validation in a lip-inspired controller," in *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 2832–2838, 2021.
- [55] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *Int. J. Rob. Res.*, vol. 27, no. 6, pp. 737–757, 2008.
- [56] H. K. Khalil, *Nonlinear systems*. No. 5, Prentice Hall, 1996.
- [57] M. S. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Autom. Contr.*, vol. 43, no. 4, pp. 475–482, 1998.
- [58] Y. Gao, K. Barhydt, C. Nieuzeck, and Y. Gu, "Global-Position Tracking Control for Multi-Domain Bipedal Walking with Underactuation," *ASME J. Dyn. Syst. Meas. Contr.*, pp. 1–27, 04 2024.



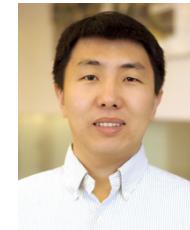
Victor Paredes received his BSc in Mechatronics from the National University of Engineering in Peru in 2012, and his MSc in Mechanical Engineering from Texas A&M University in 2016. He is currently a PhD student at The Ohio State University, specializing in Humanoid Robotics. His research focuses on optimization, planning, and control to develop stable walking gaits, aiming to advance the field of humanoid robotics by enhancing the efficiency and stability of robotic locomotion.



Yukai Gong received his B.E. degree in Naval Architecture and Ocean Engineering from Huazhong University of Science and Technology, Wuhan, China in 2015, his M.S.E degree in Mechanical Engineering and PhD. degree in Robotics from University of Michigan in 2022.



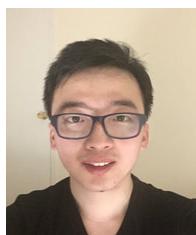
Zijian He received his B.Sc. and M.Sc. in Mechanical Engineering from Purdue University. He is currently a Ph.D. student at Purdue University. His research focuses on state estimation and control of bipedal humanoid robots.



Ayonga Hereid is an Assistant Professor in the Department of Mechanical and Aerospace Engineering at the Ohio State University. He received his B.S. and M.S. degrees in Mechanical Engineering from Zhejiang University in 2007 and 2010, respectively, and received his Ph.D. degree in Mechanical Engineering from the Georgia Institute of Technology in 2016. Prior to joining the Ohio State University, he was a postdoctoral research fellow in the Department of Electrical Engineering and Computer Science at the University of Michigan in Ann Arbor. Dr. Hereid was the recipient of the NSF CAREER Award in 2022. His work was recognized for the Best Student Paper Award in 2014 from the ACM International Conference on Hybrid System: Computation and Control and was nominated as the Best Conference Paper Award Finalist in 2016 at the IEEE International Conference on Robotics and Automation.



Yan Gu received the B.S. degree in Mechanical Engineering from Zhejiang University, China, in June 2011 and the Ph.D. degree in Mechanical Engineering from Purdue University, West Lafayette, IN, USA, in August 2017. She joined the faculty of the School of Mechanical Engineering at Purdue University in July 2022. Prior to joining Purdue, she was an Assistant Professor with the Department of Mechanical Engineering at the University of Massachusetts Lowell. Her research interests include nonlinear control, hybrid systems, and legged robotics. She was the recipient of the NSF CAREER Award in 2021 and the ONR YIP Award in 2024.



Yuan Gao received his B.S. degree in Mechanical Engineering from China Jiliang University, Hangzhou, China in 2014, the M.S. degree in Mechanical Engineering from Arizona State University in 2016, and the Ph.D. degree in Mechanical Engineering from University of Massachusetts Lowell in 2023. He is currently a control engineer at Brooks Automation.