PH.D. DISSERTATION

TIME-VARYING CONTROL AND STATE ESTIMATION FOR BIPEDAL LOCOMOTION

A DISSERTATION PRESENTED

BY

YUAN GAO

SUBMITTED TO COLLEGE OF ENGINEERING
UNIVERSITY OF MASSACHUSETTS LOWELL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

February 2024

MECHANICAL ENGINEERING PROGRAM

PH.D. DISSERTATION
TIME-VARYING CONTROL AND STATE ESTIMATION FOR BIPEDAL LOCOMOTION

BY

YUAN GAO

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
FRANCIS COLLEGE OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
UNIVERSITY OF MASSACHUSETTS LOWELL

Signature of Author . . . . . .
Yuan Gao
February, 2024

Signature of Advisor . . . . . .
Christopher Niezrecki

Signature of Co-Advisor . . . . . .
Yan Gu

Signature of Other Thesis Committee Members

. . . . . .
Kshitij Jerath

. . . . . .
Kelilah Wolkowicz

. . . . . .
Xingye Da

PH.D. DISSERTATION
TIME-VARYING CONTROL AND STATE ESTIMATION FOR BIPEDAL LOCOMOTION

BY

YUAN GAO

ABSTRACT OF A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
FRANCIS COLLEGE OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
UNIVERSITY OF MASSACHUSETTS LOWELL
February 2024

Dissertation Supervisor: Christopher Niezrecki

Distinguished University Professor, Mechanical and Industrial Engineering

Abstract

This dissertation focuses on advancing bipedal robot control and state estimation within the context of dynamic environments. It encompasses several key stages, starting with the establishment of global-position tracking (GPT) control for multi-domain walking, where walking involves alterations in contact conditions or actuator configurations during motion. This initial effort, as described in Chapter 3, lays the foundation for precise trajectory tracking and stability, essential for navigating rapid changing environments. Developing a tracking controller for bipedal robots poses unique challenges due to the intricate, time-varying, and hybrid nature of robot dynamics, particularly in the context of multi-domain walking, encompassing various phases with distinct actuation characteristics including full actuation, over actuation, and underactuation. In response to this challenge, a continuous-phase GPT control law for multi-domain walking is introduced, which provides provable exponential convergence of the entire error state within full and over actuation domains, as well as the directly regulated error state within the underactuation domain. The simulation results confirm the accuracy and reasonable convergence rate of the proposed control approach.

Subsequently, the dissertation addresses the open problems of state estimation and control for bipedal locomotion on dynamic rigid surfaces (DRSes), which are rigid surfaces moving in the inertial frame. Real-world examples of DRSes include trains and ships. Towards solving the state estimation problem, the dissertation explores the expansion of existing state estimation techniques from static terrains to DRSes, as reported in Chapter 4. This extension aims to enhance the accuracy of state estimation with DRS motion involved, which is needed to inform controller designs. A new invariant extended Kalman filter is introduced for estimating the robot's pose and velocity during DRS locomotion using common sensors found on legged robots, such as inertial measurement units (IMU), joint encoders, and RGB-D cameras. The filter explicitly accounts for the nonstationary surface-foot contact point and the hybrid robot behaviors. It also exhibits attractive properties such as guaranteed convergence in the absence of IMU biases for the deterministic case. Furthermore, the observability of state variables is analyzed, revealing the impact of DRS movement on the observability. Experimental validations with a Digit humanoid robot walking on a pitching treadmill confirm the efficacy of the proposed filter under uncertainties (e.g., sensor noise and biases and estimation errors) and various DRS pitch movements.

The last direction of this dissertation research focuses on the design and evaluation of a hierarchical control approach that ensures stable underactuated bipedal walking on a DRS with a known, periodic, horizontal motion. Chapter 5 summarizes the findings along this direction. The framework comprises three layers: higher-layer foot-placement planning, middle-layer full-body reference generation, and lower-layer feedback control. The key novelty of the proposed framework lies in the footstep planning. By incorporating a new angular momentum-based linear inverted pendulum (ALIP) model and a DRS forcing input, the higher layer povably stabilizes the hybrid, linear, time-varying ALIP model. The middle-layer walking pattern generator produces smooth foot placement transitions using Bézier polynomials encoded by a time-based phase variable, while the lower layer adopts existing feedback controllers to drive the robot to its desired state. Validations using the Digit robot,

both in simulations and hardware experiments, support the framework's effectiveness in realizing stable bipedal walking on a horizontally moving surface.

Acknowledgments

I wish to extend my heartfelt gratitude to my advisors, Dr. Christopher Niezrecki and Dr. Yan Gu, for their unwavering support and invaluable guidance throughout my Ph.D. journey. Dr. Gu provided me with the remarkable opportunity to delve into the captivating realm of leg robotics. It is a field poised to make a profound impact on our daily lives, and for this, I am grateful. Dr. Niezrecki's support in handling administrative and technical matters following Dr. Gu's departure was significant in ensuring that the final stage of my Ph.D. journey was not marred by frustration. I am deeply appreciated of his contributions.

I would also like to express my sincere thanks to my esteemed Ph.D. committee members, both past and present, including Dr. Jerath Kshitij, Dr. Kelilah Wolkowicz, Dr. Zhu Mao, and Dr. Xingye Da. Their constructive comments and insightful feedback on my defense proposal were invaluable. A special note of gratitude goes to Dr. Da for his expert guidance on various optimization techniques that proved instrumental in advancing my research in leg locomotion.

I extend my appreciation to the New England Robotics Validation and Experimentation (NERVE) Center for providing the necessary space and equipment for conducting robotic experiments, many of which contributed to my published work. My thanks also go to Joyce Sidopoulos from MassRobotics, whose support and willingness to aid in advancing my career have been invaluable.

My gratitude extends to my collaborators on my last project, Dr. Ayonga Hereid, Dr. Yukai Gong, and Victor Paredes. Dr. Hereid's insights into control theory and implementation, along with his indispensable FROST toolbox, significantly contributed to my research. Dr. Gong's exceptional mathematical skills and profound understanding of the linear inverted pendulum (LIP) model and angular momentum provided me with a fresh perspective on leg locomotion. His hands-on experiences and unwavering support were truly appreciated. Victor's dedication to implementing the controller, coupled with his extensive hardware expertise, were indispensable. Without his assistance, I would not have achieved

the results we did. His willingness to travel great distances to assist with experiments is deeply valued.

I wish to acknowledge the members of my EX2 team, Dr. Murat Inalpolat and Stephanie Epstein, for their guidance and support in advancing the project's progress. Stephanie's dedication in assisting with the collection of experimental data, even when it meant carrying heavy EOD suits, is commendable.

My gratitude extends to my lab members at the Terrain Robotics Advanced Control and Experimentation (TRACE) Lab. Amir Iqbal, a longstanding friend and a true brother, brought joy and entertainment to our lab. Zenan Zhu's invaluable assistance during the 5G robotic challenges and his kind and generous nature were deeply appreciated. I am grateful to Zenan for lending me his PS4, which provided much-needed off-duty relaxation. Thanks to Ruochen Wang for his incredible cooking skill during my Purdue visit. To Aaron Soibelman, Caroline Bisio, Stephen Misenti, and Chengyue Li, your support during my experiments was incredibly helpful, and I am thankful for your contributions. Cyrus Ghorbani and Zijian He, thank you for helping me sharpen my mentorship skills, and Zijian, your efforts in facilitating my experiments at Purdue University were greatly appreciated.

I owe an immeasurable debt of gratitude to my parents, Zonghua Zhang and Fengge Gao, for their unconditional love and support. They are the inspiration behind my Ph.D. journey.

Last but certainly not least, I want to express my sincere gratitude to my wife, Xinyue (Sherry) Liu. Your unwavering love, boundless support, and unflagging belief in me have been the cornerstone of my journey. Your presence has illuminated every step of this arduous path, making it far less frustrating and infinitely more rewarding. The birth of our little girl, Joanna, during this journey was a significant milestone, and I am grateful for your incredible strength and love as we welcomed her into our lives.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Bipedal walking systems hold great potential for various applications, including delivery services and emergency response. Their remarkable capability to navigate through challenging terrains like uneven surfaces, stairs, narrow passages, and even moving platforms sets them apart from wheeled or tracked robots.

To ensure the success of bipedal robot walking, two critical components come into play: controllers and state estimators. Controllers play a central role in enabling reliable legged robot locomotion by directly altering the dynamic behavior of the robot. Meanwhile, state estimation is essential in supplying the controller with accurate estimates of the movement variables in real-time.

However, designing controllers and state estimators for bipedal robots presents substantial challenges. These machines exhibit complex hybrid dynamical behaviors that encompass both continuous dynamics, such as foot swinging, and discrete jumps, such as the role switching between the support and the swing feet at a foot-landing event. These discrete jumps are nonlinear, uncontrolled, and triggered by state-dependent conditions [2, 3]. Both the continuous and the discrete behaviors are subject to various ground contact constraints [4]. Also, modern bipedal robots are often equipped with on-board sensors that may not directly measure all the relevant state variables or can be influenced by hardware imperfections such as bias and noise [5]. Furthermore, in real-world applications, bipedal robots may encounter dynamic rigid surfaces (DRSes) [6], which are rigid surfaces moving in the inertial frame, such as ships, aircraft, trains, and elevators (see Fig. 1-1-a)). These surfaces introduce extra complexity due to the time-varying movement of the foot-ground contact region, in contrast to the static contact area on static terrain.

The robot hardware used in this dissertation research is shown in Fig. 1-1-b).

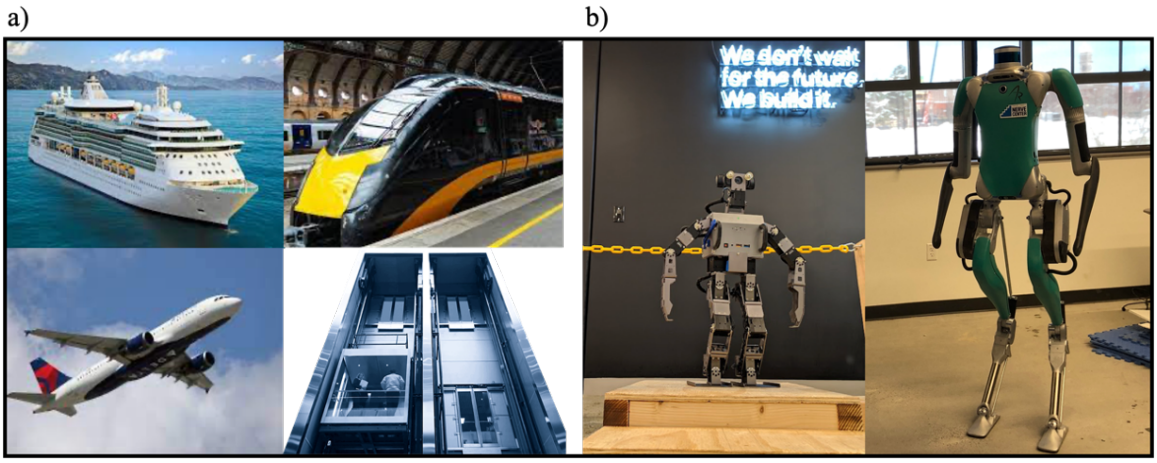

Figure 1-1: a) Illustration of real-world dynamic rigid surfaces (DRSes). b) Illustration of the robot hardware used in this dissertation. The left hand side is the OP3 robot (manufactured by ROBOTIS Co., Ltd), and the right hand side is the Digit robot (manufactured by Agility Robotics).

1.2 Related Work

To provide a context for this dissertation, several existing approaches of dynamic modelling, state estimation, and control strategies are reviewed in this section.

1.2.1 Dynamic Modeling for Bipedal Walking

Various model-based control approaches have achieved reliable walking performance on a wide range of bipedal robot platforms. These approaches rely on dynamic models, including reduced-order and full-order models, each with its own strengths and weaknesses in control design. Models are indispensable in bipedal robot controller design as they provide mathematical representations of robot dynamics and the environment. Reduced-order models offer computational efficiency, making them suitable when approximations are acceptable, whereas full-order models provide high-fidelity representations for precise control but require greater computational resources. The choice between these models depends on the need to balance accuracy with computational demands in different scenarios, highlighting the fundamental role of models in ensuring effective bipedal robot control.

1.2.1.1 Linear Inverted Pendulum Model

The most extensively studied reduced-order model of legged locomotion is the linear inverted pendulum (LIP) [7, 8] model. The LIP model approximates the legged robot as a mass point atop a massless leg, and captures the essential behaviors of legged locomotion (e.g., the transfer between the robot's potential and kinetic energy during walking). The major advantage of the LIP model stems from its simplicity (i.e., low dimensionality and linearity). As a legged robot typically has ten to forty joints, the full-order model of legged locomotion dynamics can be high-dimensional. The high dimensionality could induce an overly high computational load for the implementation of motion planners and controllers that prevent real-time implementation. In contrast, the LIP model has a significantly smaller number of state variables, and thus can be utilized to enable online planning and control. Besides simplicity, the LIP model also provides analytical tractability as well as physical intuitions into the complex dynamics of legged locomotion. Furthermore, Y. Gong et al. [9] have proposed an angular momentum-based linear inverted pendulum (ALIP) model. Compared with the classical LIP model, the ALIP model takes into account both the inertial and angular motions of the bipedal robot, resulting in a higher level of model fidelity. Yet, the major disadvantage of the LIP model, which is commonly shared by reduced-order models, is its inaccuracy in capturing the robot's complete dynamic behaviors as compared with a full-order model, especially for legged robots whose whole-body mass cannot be accurately assumed to be concentrated at a single point.

1.2.1.2 Spring-Loaded Inverted Pendulum Model

The Spring-Loaded Inverted Pendulum (SLIP) model is a widely used reduced-order model in the study of biomechanics and robotics, particularly in the context of understanding and simulating bipedal and quadrupedal locomotion [10]. This model simplifies the complex dynamics of legged locomotion into a single leg with a spring, representing the combined mass, inertia, and muscle action of an animal or a robot's leg. The essence of the SLIP model is its ability to capture the fundamental mechanics of running and hopping motions, where energy is stored in the spring during the stance phase and released during the take-off phase,

creating a cyclical movement pattern [11]. This model has been significant in uncovering fundamental locomotion principles in the natural world and has influenced the design and development of legged robotic systems [12].

1.2.1.3 Centroidal Dynamics

Another widely adopted reduced-order model is the centroidal dynamics [13, 14, 15, 16], which describes the dynamics of a robot's center of mass (CoM) and its whole-body angular momentum about the CoM in response to external forces (i.e., gravitational and ground reaction forces). Compared to the LIP model, the centroidal dynamics provides a more realistic approximation of the actual robot dynamics for several reasons. First, it does not rely on some of the relatively restrictive assumptions underlying the LIP model, such as the constant CoM height above the support point. Second, in contrast to the LIP model that only considers the effective ground reaction forces at the center of pressure (CoP), the centroidal dynamics considers the ground reaction forces at each support foot, which could be utilized to allow planners and controllers to explicitly optimize those forces at all support feet for ensuring secured foot-ground contact. However, the centroidal dynamics is not as analytically tractable as the LIP model. For instance, the whole-body angular momentum about the CoM cannot be integrated analytically based on the centroidal dynamics.

1.2.1.4 Hybrid Zero Dynamics (HZD)

The full-order model of a bipedal robot describes the complete dynamic behaviors of a biped's full degrees of freedom (DOF). The major advantage of using a full-order model in controller design lies in its higher accuracy than a reduced-order model. Yet, the full-order model of a bipedal robot can be highly complex due to its high dimensionality, nonlinearity, and hybrid nature. To simplify the full-order robot model without compromising its accuracy, the concept of hybrid zero dynamics (HZD) [2, 17, 18] has been proposed for underactuated walking robots whose independent joint motors are fewer than their DOFs. The HZD is the low-dimensional zero dynamics of the uncontrolled portion of the robot's full-order dynamics under the input-output linearizing control that is built upon the full-order model. The HZD enjoys both the accuracy of a general full-order model as well as the simplicity of a

standard reduced-order model. Different from the standard reduced-order models, the HZD also captures the hybrid behaviors of bipedal walking. Indeed, bipedal walking is inherently hybrid and possesses discrete-time dynamics, such as the role switching of the support and swing feet at the foot-landing events as well as the sudden jumps in the joint velocities caused by landing impacts. By capturing the hybrid behaviors, the HZD allows controllers to explicitly address them for ensuring locomotion stability in the presence of significant discrete behaviors. This is in sharp contrast to the majority of the existing reduced-order models, which only consider the continuous-time dynamics of bipedal walking, i.e., the swing-phase dynamics.

1.2.2 State Estimation for Legged Locomotion

State estimation plays a significant role in controller design and motion planning for robots by providing real-time estimates of the robot's movement state, such as the position, velocity, and orientation of the robot's base/trunk segment. The key performance measures of state estimation include: (a) accuracy, which is used to assess how closely the state estimates match the ground truth; (b) computational efficiency, which is a crucial metric for real-time implementation; (c) reliance on a minimal number of sensors, which is important for cost-effectiveness and simplicity in hardware setups; and (d) error convergence rate, which characterizes how fast the state estimate converges to its steady-state value.

State estimation methods include filtering and smoothing [19]. Filtering utilizes only the previous estimation results from the last time step to estimate the state variables at the current time step, whereas smoothing (e.g., factor graph [20, 21]) uses multiple previous estimation results, sometimes even all results up to the last time step, to provide state estimates at the current time step. Compared to smoothing, filtering is typically computationally more efficient due to its lighter computational load, and thus is more suitable for real-time state estimation. We concentrate on reviewing common filtering methods next as our work primarily addresses filtering instead of smoothing.

1.2.2.1 Kalman Filtering and Its Variantions

Kalman filtering (KF) is a commonly used real-time state estimation method for linear systems, involving propagation and measurement update steps. In the propagation step, it estimates the true state's probability density function using linear system dynamics. When sensor measurements arrive, state estimates are updated using linear measurement models [22]. In practical scenarios, process and measurement models are often nonlinear, leading to KF's performance limitations. The extended Kalman filter (EKF) extends KF to nonlinear models based on the linearization of the nonlinear model at the estimated state [23]. While computationally efficient, EKF may perform poorly in the presence of significant initial estimation errors and lacks provable guarantees [24]. In addition, the EKF has been reported to cause observability inconsistency issues [25] because they might treat the unobservable state as if they are observable. To overcome the inaccuracies of linearization in EKF, the unscented Kalman filter (UKF) provides an alternative by employing weighted statistical linear regression [26]. In general, the UKF produces equal or better results than the EKF [23].

1.2.2.2 Invariant Extended Kalman Filtering

Recently, the invariant extended Kalman filtering (InEKF) method has been introduced to resolve the inaccurate linearization of system models in order to achieve real-time, rapidly convergent estimation [24]. The InEKF formulates the filter design on the matrix Lie group for a class of continuous-time or discrete-time systems with group affine process models and invariant measurement models. For such systems, the linearization of the error dynamics on the Lie group is exact and independent of the state estimates for the deterministic portion of the systems. Due to this attractive feature, the InEKF provides provable guarantees on the estimation performance while demanding a low computational burden to allow for real-time implementation. The InEKF has achieved remarkable estimation performance for drones [27], under water vehicles [28], and legged robots [29, 30].

1.2.2.3 Real-Time State Estimation of Legged Locomotion

Legged robots are often equipped with sensors, such as IMUs (typically attached to the robot's trunk), joint encoders, contact sensors, cameras, and Lidars, to measure their pose and motion. However, these sensors are prone to hardware imperfections, uncertainties during real-world operations, and might not directly measure all variables needed to inform motion planners and controllers. To address this, various filters have been designed to provide real-time, accurate state estimates.

Early filter designs for legged robots primarily employ the robot's forward kinematics chain [31] but struggle with relatively large state estimation errors due to foot slippage on the ground, sensor noise, and kinematic model uncertainties. Improvements involve constructing a measurement model based on the forward kinematics from the ground contact point to the IMU, assuming static contact points [32]. This approach has been implemented on robots such as MIT's mini Cheetah quadruped [33], where the orientation of the robot's trunk, directly measured by the IMU, is utilized.

An EKF approach, employing unit quaternions to represent IMU rotations, has been introduced by Bloesch et al. for quadrupedal and bipedal robots [32]. This approach has enabled accurate and real-time estimation of movement variables such as the linear velocity and roll and pitch angles of the IMU/trunk frame. Yang et al. have further extendeded this method to a contact-centric UKF for legged locomotion on challenging terrains [34].

Hartley et al. [29] have applied the InEKF [32] to legged robot locomotion, reformulating theprevious EKF-based estimator [32] on matrix Lie groups. In contrast to the EKF approach, the InEKF ensures the asymptotic error convergence for the deterministic scenario in the absence of IMU biases, and significantly improves the error convergence rate. Lin et al. [35] have introduced a learning-based approach for detecting robot-terrain contacts without requiring modifications of existing hardware. This improved contact detection method enhances the effectiveness of the InEKF.

To date, the majority of state estimators for legged robots primarily address cases where the ground remains static, and state estimation for moving surfaces remains underexplored.

1.2.3 Control Design

Controllers directly alter a robot's dynamic behavior and are essential in ensuring reliable legged locomotion performance. The primary challenges in designing controllers for legged robot locomotion arise from the complex robot dynamics, which are nonlinear, hybrid, and high-dimensional in nature. Furthermore, factors such as environmental uncertainties, sensor noise, and external disturbances contribute to the overall complexity of controller design. Common assessment measures of legged locomotion control include stability, agility, efficiency, versatility, and robustness. Stability primarily involves a robot's ability to maintain balance and prevent falls, making it the most important performance measure. Agility is the skill of effectively traversing different terrains, efficiency involves lowering energy usage, and versatility is the robot's ability to adjust its walking patterns to various environments. Robustness indicates the controller's capacity to manage model uncertainties and external disturbances. Different control approaches will be introduced next.

1.2.3.1 Zero Moment Point (ZMP) Based Control Design

The ZMP control design method is one of the most extensively studied methods for bipedal walking [7, 36]. This method relies on the satisfaction of the ZMP balance criterion to ensure walking stability. A ZMP is a reference point on the robot's foot about which the sum of all horizontal moments of ground reaction forces is equal to zero [37]. The ZMP balance criterion requires that the ZMP should be strictly within the support polygon for preventing a support foot from rolling over its edge. The ZMP controller design method typically utilizes the LIP model as its basis, and has been successfully implemented on a variety of biped robot platforms, such as ASIMO [38] and NAO [39].

The advantage of the ZMP control method lies in its high walking versatility, and has enabled stable walking with different gait types through accurate trajectory tracking. Despite high versatility, the classical ZMP-based controller is unable to address underactuated walking, which exists during foot slipping as well as the heel-off phase of human-like walking.

1.2.3.2 Capture Point Based Control Design

Besides the ZMP point, another extensively studied reference point in controller design is the capture point [40, 41]. The capture point is a point on the walking surface that a robot needs to step to in order to come to a complete stop. A controller designed based on the capture point can be used to guarantee walking robustness against external disturbances such as sudden pushes.

In general, the real-time computation of the capture point based on a full-order model is challenging due to the complex walking dynamics [42]. However, with a simplified model such as the LIP model, the analytical expression of a capture point can be obtained. Thus, implementing the capture point based control strategy online is possible due to the low computational cost of the LIP model [43]. A limitation of the LIP model is its assumption of constant CoM height, which proves inadequate when the robot navigates uneven terrain. To address this, S. Caron [44] has introduced the variable-height inverted pendulum (VHIP) model, which enables the robot to adapt to uneven terrain using capture inputs that are computed within a few microseconds.

Still, using a reduced-order model in controller design may not be effective for robot platforms (e.g., bipedal humanoid robots with substantial leg masses) whose dynamics cannot be sufficiently accurately captured by such models. Also, both the ZMP and the capture point based control methods only address the continuous walking dynamics without explicitly considering the discrete-time dynamics of bipedal walking, and thus cannot ensure walking stability when the discrete behaviors, such as sudden jumps in joint velocities upon foot landings, are significant.

1.2.3.3 Hybrid Zero Dynamics (HZD) Based Control Design

In contrast to the ZMP and capture point based methods, the HZD contol framework ensures the walking stability by provably stabilizing the desired motions of the hybrid, nonlinear, full-order walking dynamics [2, 17, 18].

The HZD framework has been implemented for fully actuated [45], underactuated [46, 47], and multi-domain walking [48, 49], as well as running [50] and robot-assisted human

walking [51]. Recently, velocity regulation [36] and learning-based gait library design [52] have been incorporated into the HZD framework to enhance walking versatility beyond periodic walking. In addition, the reinforcement leaning approach has also adapted into the HZD framework [53]. As the HZD framework considers the full-order robot model without relying on the restrictive model simplification assumptions as in the ZMP and capture point approaches, it has achieved much higher walking performance in terms of agility and energy efficiency. However, the HZD framework has been largely focused on stabilizing periodic walking motions based on orbital stabilization instead of provable trajectory tracking.

1.2.3.4 Learning Based Control Design

Thanks to the availability of abundant data, the data-driven methods, notably reinforcement learning (RL), offer new ways of control design for bipedal robots. There has been a growing body of work that employs RL in a model-free manner [54, 55]. These model-free methods generally require extensive training time and face challenges in transitioning from simulated environments to physical hardware. They also typically do not provide guarantees for system stability and performance and lack interpretability. In response to these limitations, hierarchical controller structures that merge RL with model-based approaches have been developed [56]. This methodology integrates various established models such as the LIP [57], SLIP [58], and full-order models [53]. By utilizing various models, the controller can capture the dynamics of bipedal walking systems more effectively [59]. This combination of model- and learning-based approaches represents a significant advancement in the field of robotic locomotion. It harnesses the predictive power of established models while simultaneously leveraging the generalizability of RL. The result is a more robust and versatile control system that can better navigate the complexities inherent in bipedal locomotion. Although the primary focus of this dissertation is on model-based methods instead of learning-based techniques, the potential and efficacy of these approaches are promising for further exploration.

1.3 Contributions

The research laid out in this dissertation has led to:

- Derivation of a nonlinear control approach that exploits Lyapunov-based controller design methodologies to realize accurate global-position tracking for multi-domain bipedal robot walking;
- Creation and experimental validation of an invariant filtering method that guarantees real-time, rapid error convergence for bipedal walking over both stationary and DRSes, by theoretically extending the existing InEKF method from systems without state-triggered jumps to hybrid dynamical systems.
- Formulation and experimental assessement of a new control approach that employs a new hybrid and time-varying reduced-order model to achieve stable underactuated bipedal walking on a horizontally oscillating DRS.

1.4 Overall Structure of this Dissertation

A brief summary of the following chapters are explained next.

Chapter 2 Background: In this chapter, we establish the fundamental mathematical concepts that readers will need so as to grasp the content presented in the rest of this dissertation. The covered topics include coordinate systems, the LIP model, hybrid systems, and an introduction to the basics of Lie groups. These concepts will reappear across the entirety of the dissertation. As we delve into coordinate systems, readers will gain insight into how the robot's movements are described. The derivation of the LIP model lays the groundwork for more advanced discussions in Section 5. The exploration of hybrid system dynamics guides us through the intricate behaviors of robots, providing a pivotal framework for comprehending multi-domain locomotion and full-order control strategies. Additionally, the section on the matrix Lie groups sets the stage for understanding the core principles of the InEKF, which is utilized for the robot's state estimation.

Chapter 3 Global-Position Tracking Control for 3-D Multi-Domain Bipedal Walking: This chapter introduces a new time-based nonlinear control method for accurate global-position tracking (GPT) in multi-domain bipedal walking. The complexity of the hybrid and nonlinear robot dynamics poses challenges in deriving tracking controllers, especially for multi-domain walking with full, over, and under actuation phases. Our approach employs a continuous-phase GPT control law that ensures exponential error state convergence across all domains. Sufficient multiple-Lyapunov stability conditions for the hybrid multi-domain tracking error system are established. Simulation results demonstrate the effectiveness of the proposed control method in both three-domain and two-domain walking scenarios, showcasing accurate tracking and convergence across various speeds and paths.

Chapter 4 Invariant Filtering for Legged Humanoid Locomotion: This chapter presents an InEKF that estimates the robot's trunk pose and velocity during DRS locomotion by using common sensors of legged robots (e.g., IMU, joint encoders, and RDB-D camera). A key feature of the filter lies in that it explicitly addresses the nonstationary surface-foot contact point and the hybrid robot behaviors. Another key feature is that, in the absence of IMU biases, the filter satisfies the attractive group affine and invariant observation conditions, and is thus provably convergent for the deterministic continuous phases. The observability analysis is performed to reveal the effects of DRS movement on the state observability, and the convergence property of the hybrid, deterministic filter system is examined for the observable state variables. Experiments of a Digit humanoid robot walking on a pitching treadmill validate the effectiveness of the proposed filter under sensor noise and biases as well as large initial estimation errors and DRS movement uncertainties.

Chapter 5 Foot-Placement Control for Underactuated Bipedal Walking on Dynamic Rigid Surfaces (DRSes): This chapter introduces a real-time control framework that realizes stable underactuated walking on DRSes with a known, periodic, horizontal motion. The framework consists of three layers: foot-placement planning, full-body reference generation, and feedback control. By incorporating the angular momentum-based linear inverted pendulum (ALIP) model and a DRS forcing input, the framework ensures stable walking by stabilizing the hybrid, linear, time-varying ALIP model. The walking pattern generator produces smooth foot placement transitions using Bèzier polynomials encoded

by a time-based phase variable, while the lower layer implements feedback controllers to handle holonomic constraints. Validations on the Digit robot, both in simulations and hardware, demonstrate the effectiveness of the framework in addressing DRSes.

Chapter 6 Conclusion and Future Work: This chapter presents an overview of the dissertation research, including our GPT control for multi-domain walking, InEKF-based state estimation for DRS locomotion, and ALIP-based control for underactuated walking on a horizontally moving DRS. Additionally, this chapter outlines promising directions for future research in the context of the dissertation.

Chapter 2 Background

This chapter introduces the essential mathematical concepts underlying the content in the rest of this dissertation. The covered topics encompass coordinate systems, hybrid systems, and an introduction of matrix Lie groups.

2.1 Coordinate Systems and Generalized Coordinates

This section explains the three coordinate systems used in the proposed controller and estimator designs. Figure 2-1 illustrates the three frames, with the x-, y-, and z-axes respectively highlighted in red, green, and blue.

2.1.1 Coordinate Systems

The world frame, also known as the inertial frame, is rigidly attached to the ground (see "{World}" in Fig. 2-1). The base frame, illustrated as "{Base}" in Fig. 2-1, is rigidly attached to the robot's trunk. The x-direction (red) points forward, and the z-direction (blue) points towards the robot's head. The origin of the vehicle frame (see "{Vehicle}" in Fig. 2-1) coincides with the base frame, and its z-axis remains parallel to that of the world frame. The vehicle frame rotates only about its z-axis by a certain heading (yaw) angle. The yaw angle of the vehicle frame with respect to (w.r.t.) the world frame equals that of the base frame w.r.t. the world frame, while the roll and pitch angles of the vehicle frame w.r.t. the world frame are 0.

2.1.2 Generalized coordinates

To use Lagrange's method to derive robot dynamics models, we need to first introduce the generalized coordinates to represent the base pose (i.e., position and orientation) and joint angles of the robot.

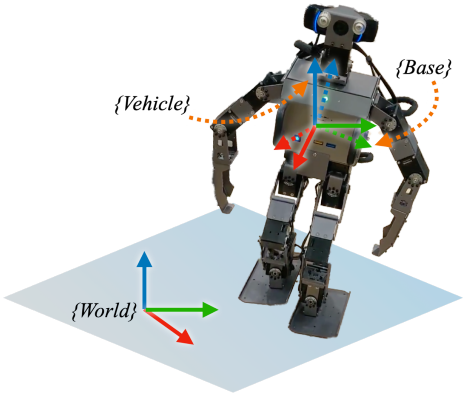

Figure 2-1: Illustration of the three coordinate systems used in the study: world frame, vehicle frame, and base frame.

We use $\mathbf{p}^b \in \mathbb{R}^3$ and $\boldsymbol{\gamma}^b \in SO(3)$ to respectively denote the absolute base position and orientation w.r.t. the world frame, and their coordinates are represented by (x_b, y_b, z_b) and $(\phi_b, \theta_b, \psi_b)$. Here ϕ_b, θ_b, ψ_b are the roll, pitch, and yaw angles, respectively. Then, the 6-D pose $\mathbf{q}_b$ of the base is given by: $\mathbf{q}_b := [(\mathbf{p}^b)^T, (\boldsymbol{\gamma}^b)^T]^T$.

Let the scalar real variables $q_1, \ ..., \ q_n$ represent n revolute joints of the robot. Then, the generalized coordinates of a 3-D robot, which has a floating base and total of n independent revolute joints, can be expressed as:

$$\mathbf{q} = \left[\mathbf{q}_b^T, \ q_1, \ ..., \ q_n\right]^T \in \mathcal{Q}, \tag{2.1}$$

where $\mathcal{Q} \subset \mathbb{R}^{n+6}$ is the configuration space. Note that the number of degrees of freedom (DOFs) of this robot without subjecting to any holonomic constraints is $n+6$.

2.2 Hybrid System

A hybrid control system $\mathcal{HC}$ is a tuple:

$$\mathcal{HC} = (\Gamma, D, U, S, \Delta, FG), \tag{2.2}$$

where

- The oriented graph $\Gamma = (V, E)$ comprises a set of vertices $V = \{v_1, v_2, ..., v_N\}$ and a set

of edges $E = \{e_1, e_2, ..., e_N\}$, where N is the total number of elements in each set. In this dissertation, each vertex v_i represents the i^{th} domain, while each edge e_i represents the transition from the source domain to the target domain, thereby indicating the ordered sequence of all domains.

- D is a set of domains of admissibility.
- U is the set of admissible control inputs.
- S is a set of switching surfaces determining the occurrence of switching between domains.
- Δ is a set of reset maps, which represents the impact dynamics between a robot's swing foot and the ground.
- FG is a set of vector fields on the state manifold.

The elements of these sets are explained next.

2.2.1 Continuous-phase dynamics

Within any domains in D, the robot only exhibits continuous movements, and its dynamics model is naturally continuous-time. Applying Lagrange's method, we obtain the second-order, nonlinear robot dynamics as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\mathbf{u} + \mathbf{J}^T\mathbf{F}_c, \tag{2.3}$$

where $\mathbf{M}(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^{(n+6)\times(n+6)}$ is the inertia matrix. The vector $\mathbf{c} : \mathcal{TQ} \to \mathbb{R}^{(n+6)}$ is the summation of the Coriolis, centrifugal, and gravitational terms, where $\mathcal{TQ}$ is the tangent bundle of $\mathcal{Q}$. The matrix $\mathbf{B} \in \mathbb{R}^{(n+6)\times n_a}$ is the input matrix. The vector $\mathbf{u} \in U \subset \mathbb{R}^{n_a}$ is the joint torque vector. The matrix $\mathbf{J}(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^{n_c\times(n+6)}$ represents the Jacobian matrix. The vector $\mathbf{F}_c \in \mathbb{R}^{n_c}$ is the constraint force that the ground applies to the foot-ground contact region of the robot. Note that during multi-domain walking, the dimensions of $\mathbf{J}$ and $\mathbf{F}_c$ vary among the different domains due to differences in the ground-contact conditions and other types of holonomic constraints.

The holonomic constraints associated with legged locomotion on stationary ground can be expressed as:

$$\mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{0}, \tag{2.4}$$

where $\mathbf{0}$ is a zero matrix with an appropriate dimension.

Combining Eqs. (2.3) and (2.4), we compactly express the continuous-phase dynamics model as [60]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}) = \bar{\mathbf{B}}(\mathbf{q})\mathbf{u}, \tag{2.5}$$

where the vector $\bar{\mathbf{c}}$ and matrix $\bar{\mathbf{B}}$ are defined as: $\bar{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}) := \mathbf{c} - \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\mathbf{J}\mathbf{M}^{-1}\mathbf{c} - \dot{\mathbf{J}}\dot{\mathbf{q}})$ and $\bar{\mathbf{B}}(\mathbf{q}) := \mathbf{B} - \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}\mathbf{B}$.

2.2.2 Switching surfaces

When a robot's state reaches a switching surface $S_i \in S$, where i is the index of the switching surface, it exits the source domain and enters the targeted domain. We use switching surfaces to describe the conditions under which a switching event occurs. The switching surface varies for different walking type, as we will look more into details in Chap. 3 and Chap. 5.

2.2.3 Discrete impact dynamics

When a rigit swing foot hits a rigid walking surface, an instantaneous rigid-body impact occurs, resulting in the robot's generalized velocity $\dot{\mathbf{q}}$ experiencing a sudden jump. Unlike velocity $\dot{\mathbf{q}}$, the configuration $\mathbf{q}$ remains continuous across an impact event as long as there is no coordinate swap of the two legs at any switching event. Let $\dot{\mathbf{q}}^-$ and $\dot{\mathbf{q}}^+$ represent the values of $\dot{\mathbf{q}}$ just before and after an impact, respectively. The impact dynamics can be described by the following nonlinear reset map [2]:

$$\dot{\mathbf{q}}^+ = \mathbf{\Delta}_{\dot{q}}(\mathbf{q})\dot{\mathbf{q}}^-, \tag{2.6}$$

where $\mathbf{\Delta}_{\dot{q}} : \mathcal{Q} \to \mathbb{R}^{(n+6)\times(n+6)}$ is an element in Δ and a nonlinear matrix-valued function relating the pre-impact generalized velocity $\dot{\mathbf{q}}^-$ to the post-impact value $\dot{\mathbf{q}}^+$. The derivation

of $\mathbf{\Delta}_{\dot{q}}$ relies on the specific impact scenario, which will be explored in detail in Chap. 3 where various foot-surface contact conditions will be discussed.

2.3 Bézier polynomial

The Bézier polynomial $\phi(s)$, which is used to parameterize the desired trajectory, is defined as follows:

$$\phi(s) := \sum_{m=0}^{M} \alpha_m \frac{M!}{m!(M-m)!} s^m (1-s)^{M-m}, \tag{2.7}$$

where M represents the order of the Bézier polynomial and α_m is the coefficient associated with the m^{th} term. Bézier polynomials have the convenient property that $\phi(0) = \alpha_0$ and $\phi(1) = \alpha_M$, which allows us to relate the coefficients to the values of the polynomial at the endpoints. By adjusting the values of the coefficients, we can modify the desired trajectory accordingly. Moreover, Bézier polynomials exhibit relatively small oscillations even with small variations in the parameters, making them a suitable choice for encoding desired trajectories for robots.

2.4 Fundamentals of Matrix Lie Groups

Robotic state estimation is vital for a robot to understand its environment and navigate effectively, involving the deduction of the robot's internal state from sensor data. The integration of matrix Lie groups into robotic state estimation can help effectively address the nonlinearities of system dynamics [29]. This section offers a brief introduction to matrix Lie groups.

2.4.1 Lie Groups

A Lie group $\mathcal{G}$ is a smooth manifold whose elements, such as $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, satisfies the following group axioms:

Closure: $\mathcal{X} \circ \mathcal{Y} \in \mathcal{G}$,

Identity: $\mathcal{X} \circ \mathcal{I} = \mathcal{I} \circ \mathcal{X} = \mathcal{X}$,

Inverse: $\mathcal{X} \circ \mathcal{X}^{-1} = \mathcal{X}^{-1} \circ \mathcal{X} = \mathcal{I}$,

Associativity: $(\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z})$,

where $\mathcal{I}$ is the group identity and $\circ$ is the group composition operation.

2.4.2 Matrix Lie Groups

A matrix Lie group $\mathcal{G}_M$ [61] is a specific type of Lie groups in which the elements of the group are represented by certain invertible square matrices, and the group operation is matrix multiplication, denoted as $\cdot$ here. Accordingly, to ensure that matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ belong to the matrix Lie group, they must satisfy the following conditions, which are derived from the group axioms:

Closure: $\mathbf{X} \cdot \mathbf{Y} \in \mathcal{G}_M$,

Identity: $\mathbf{X} \cdot \mathbf{I} = \mathbf{I} \cdot \mathbf{X} = \mathbf{X}$,

Inverse: $\mathbf{X} \cdot \mathbf{X}^{-1} = \mathbf{X}^{-1} \cdot \mathbf{X} = \mathbf{I}$,

Associativity: $(\mathbf{X} \cdot \mathbf{Y}) \cdot \mathbf{Z} = \mathbf{X} \cdot (\mathbf{Y} \cdot \mathbf{Z})$,

where $\mathbf{I}$ is the identity matrix with a proper dimension. Since this dissertation primarily uses matrix Lie group in the proposed state estimator design, the subsequent content focuses on introducing several basic concepts associated with matrix Lie groups, rather than the broader context of a general Lie group.

2.4.3 Lie Algebra

For any point (matrix) $\mathbf{X} \in \mathcal{G}_M$, there is an unique tangent space, denoted as $T_{\mathbf{X}}\mathcal{G}_M$. Such a tangent space at the group identity, denoted as $T_I\mathcal{G}_M$, is called the Lie algebra $\mathfrak{g}$. The Lie algebra is a set of $n \times n$ square matrices with dimension of dim$\mathfrak{g}$. Lie algebra $\mathfrak{g}$ can be mapped directly to the vector space using the linear operator $(\cdot)^\vee : \mathfrak{g} \to \mathbb{R}^{dim\mathfrak{g}}$ (Vee operator), whose inverse operator is $(\cdot)^\wedge : \mathbb{R}^{dim\mathfrak{g}} \to \mathfrak{g}$ (hat operator).

It is worth noting that the Lie algebra can be defined locally w.r.t. a specific point $\mathbf{X}$ on the matrix Lie group, which may not necessarily be the group identity. To precisely denote

this tangent space, a left superscript can be used, as seen in expressions such as ${}^{X}\mathbf{v} \in T_X\mathcal{G}_M$ and ${}^{I}\mathbf{v} \in T_I\mathcal{G}_M$, where $\mathbf{v} \in \mathfrak{g}$ represents an arbitrary element within the Lie algebra.

2.4.4 Exponential, Logarithmic, and Adjoint Operations

The exponential map, denoted as $\exp : \mathfrak{g} \to \mathcal{G}_M$, maps the element of the Lie algebra to the Lie group exactly. In the context of matrix Lie groups, the exp function corresponds to the matrix exponential and possesses the following properties:

- $\exp(t\mathbf{v} + s\mathbf{v}) = \exp(t\mathbf{v}) \cdot \exp(s\mathbf{v})$,
- $\exp(t\mathbf{v}) = (\exp(\mathbf{v}))^t$,
- $\exp(-\mathbf{v}) = (\exp(\mathbf{v}))^{-1}$,
- $\exp(\mathbf{X} \cdot \mathbf{v} \cdot \mathbf{X}^{-1}) = \mathbf{X} \cdot \exp(\mathbf{v}) \cdot \mathbf{X}^{-1}$.

where t and s are any real numbers. The inverse operation is the logarithmic map, denoted as $\log : \mathcal{G}_M \to \mathfrak{g}$.

The Lie algebra at the group identity and the Lie algebra at other points can be transformed to each other through the adjoint transformation, $\mathrm{Ad}_X : \mathfrak{g} \to \mathfrak{g}$, via the following expression:

$$ {}^{I}\mathbf{v} = \mathrm{Ad}_X({}^{X}\mathbf{v}) = \mathbf{X} \cdot \mathbf{v} \cdot \mathbf{X}^{-1}. \tag{2.8} $$

Chapter 3 Global-Position Tracking Control for Multi-Domain Bipedal Robot Walking

This chapter introduces a new time-based nonlinear control method that achieves accurate global-position tracking (GPT) for multi-domain bipedal walking on a static ground by explicitly treating the hybrid, nonlinear, full-order robot dynamics.

Multi-domain walking in legged locomotion is naturally hybrid, involving both continuous foot-swinging phases and discrete foot-landing behaviors within a gait cycle. This occurs due to changes in foot-ground contact conditions and actuation authority [62, 63], which is observed in human walking as well. These phases encompass full actuation, underactuation, and over actuation, each having different actuation characteristics. In contrast to fully actuated walking, multi-domain walking offers greater agility and efficiency thanks to underactuation. However, this underactuation introduces substantial control challenges owing to limited control authority.

Several control strategies have been proposed to achieve stable multi-domain walking in bipedal and quadrupedal robots. These strategies use hybrid models to capture multi-domain robot dynamics [63]. M. Dai et al. have introduced the multi-domain linear inverted pendulum (MLIP) model and employed step-to-step dynamics to achieve human-like multi-domain walking on a three-dimensional (3-D) full-order Cassie robot [64]. Meanwhile, M. Tucker et al. have demonstrated multi-domain walking on an exoskeleton platform [65]. While these approaches have demonstrated stability and reliable performance on physical legged robot robot platforms, they may not directly address general global-position tracking (GPT) control problems. In real-world scenarios, robots often require precise global position control and timing, which can be challenging to achieve with orbital stabilization mechanisms employed in previous methods[66, 67, 68, 69, 70]. This is because those previous methods rely on orbital stabilization to ensure locomotion stability, which cannot guarantee accurate and precise tracking of time-varying global-position trajectories [71, 72, 73, 74, 75].

To address this tracking control problem for multi-domain walking, we have developed

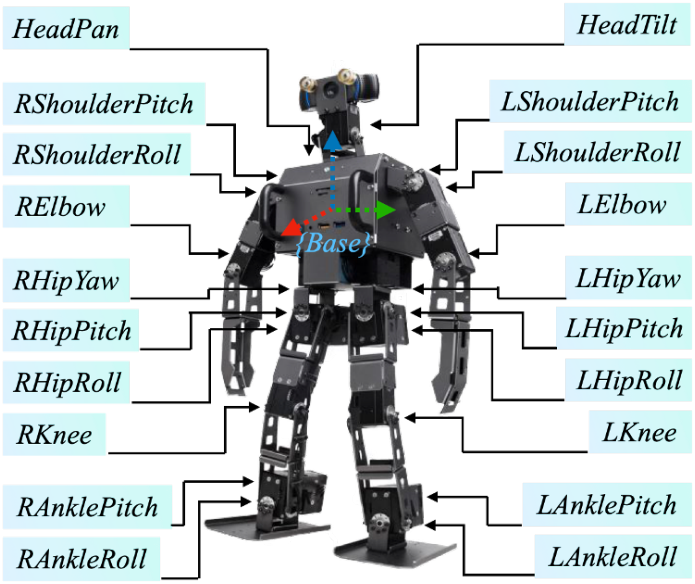

Figure 3-1: Illustration of the Darwin OP3 robot, which is used to validate the proposed global-position tracking control approach. Darwin OP3 is a bipedal humanoid robot with twenty revolute joints, designed and manufactured by ROBOTIS [1]. The reference frame of the robot's floating base, highlighted as "{Base}", is located at the center of the chest.

a GPT control method for 2-D fully actuated bipedal walking robots [71, 72, 76] and extendeded it to 3-D fully actuated robots [73, 77, 78, 75, 60]. Additionally, we have formulated the dynamics of fully actuated quadrupedal walking on a dynamic rigid surface (DRS) in the inertial frame as a hybrid time-varying system [79, 6, 80], enabling the development of a GPT control law for fully actuated quadrupeds. However, these methods are not directly applicable to the multi-domain control problem because they do not explicitly handle the underactuated robot dynamics associated with multi-domain walking.

3.1 Full-Order Dynamic Modeling of Three-Domain Walking

This section presents the hybrid model of bipedal robot dynamics associated with three-domain walking. The robot model used in this chapter is illustrated in Fig. 3-1.

3.1.1 Walking Domain Description

For simplicity and without loss of generality, we consider the following assumptions on the foot-ground contact conditions during 3-D walking:

A3.1 The toe and heel are the only parts of a support foot that can contact the ground [62].

A3.2 While contacting the ground, the toes and/or heels have line contact with the ground.

A3.3 There is no foot slipping on the ground.

Also, we consider the common assumption below about the robot's actuators:

A3.4 All the n revolute joints of the robot are independently actuated.

Let n_a denote the number of independent actuators, and $n_a = n$ holds under assumption (A3.4). At the switching instant, we consider the following assumptions:

A3.5 The landing impact between the robot's foot and the ground is a contact between rigid bodies.

A3.6 The impact occurs instantaneously and lasts for an infinitesimal period of time.

Figure 3-2 illustrates the complete gait cycle of human-like walking with a rolling support foot. As the figure displays, the complete walking cycle involves three continuous phases/domains and three discrete behaviors connecting the three domains. The three domains are:

(i) Full actuation (FA) domain, where n_a equals the number of DOFs;

(ii) Underactation (UA) domain, where the number of independent actuators (n_a) is less than that of the robot's degrees of freedom (DOFs); and

(iii) Over actuation (OA) domain, where n_a is greater than the number of DOFs.

The actuation types associated with the three domains are different because those domains have distinct foot-ground contact conditions, which are explained next under assumptions (A3.1)-(A3.4).

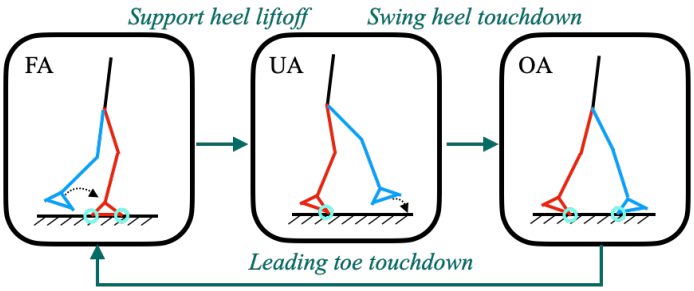

Figure 3-2: The directed cycle of 3-D three-domain walking. The green circles in the diagram highlight the portions of a foot that are in contact with the ground. The position trajectory of the swing foot is indicated by the dashed arrow. The red and blue legs respectively represent the support and swing legs. Note that when the robot exits the OA domain and enters the FA domain, the swing and support legs switch their roles, and accordingly the leading and trailing legs swap their colors.

3.1.1.1 FA domain

As illustrated in the "FA" portion of Fig. 3-2, only one foot is in support and it is static on the ground within the FA domain. Under assumption (A3.1), we know both the toe and heel of the support foot contact the ground. From assumptions (A3.2) and (A3.3), we can completely characterize the foot-ground contact condition with six independent scalar holonomic constraints. Using n_c to denote the number of holonomic constraints, we have $n_c = 6$ within an FA domain, and the number of DOFs becomes $\text{DOF} = n + 6 - n_c = n$. Meanwhile, $n_a = n$ holds under assumption (A3.4). Since $\text{DOF} = n_a$, all of the DOFs are directly actuated; that is, the robot is indeed fully actuated.

3.1.1.2 UA domain

The "UA" portion of Fig. 3-2 shows that the robot's support foot rolls about its toe within a UA domain. Under assumptions (A3.2) and (A3.3), the number of holonomic constraints is five, i.e., $n_c = 5$. This is because the support foot can only rotate about the line toe but its motion is fully restricted in terms of the 3-D translation and the roll and yaw rotation. Then, the number of DOFs is: $\text{DOF} = n + 6 - 5 = n + 1$. Since the number of independent actuators, n_a, equals n under assumption (A3.4) and is lower than the number of DOFs, $(n+1)$, the robot is underactuated with one degree of underactuation.

3.1.1.3 OA domain

Upon exiting the UA domain, the robot's swing-foot heel strikes the ground and enters the OA domain (Fig. 3-2). Within an OA domain, both the trailing toe and the leading heel of the robot contact the ground, which is described by ten scalar holonomic constraints (i.e., $n_c = 10$). Thus, the DOF becomes $\text{DOF} = n + 6 - n_c = n - 4$, which is less than the number of actuators under assumption (A3.4), meaning the robot is over actuated.

3.1.2 Hybrid Multi-Domain Dynamics

This subsection presents the full-order model of the robot dynamics that corresponds to multi-domain walking. Since multi-domain walking involves both continuous-time dynamics and discrete-time behaviors, a hybrid model is employed to describe the robot dynamics. Recall Sec. 2.2 for fundamentals of the hybrid system.

3.1.2.1 Continuous-phase dynamics

Within any of the three domains, the robot only exhibits continuous movements, and its dynamics model is naturally continuous-time as described by Eq. (2.3). Note that the dimensions of $\mathbf{J}$ and $\mathbf{F}_c$ in Eq. (2.3) vary among the three domains due to differences in the ground-contact conditions.

3.1.2.2 Switching surfaces

As displayed in Fig. 3-2, the human-like, three-domain walking involves three switching events, which are:

(i) Switching from FA to UA ("Support heel liftoff");

(ii) Switching from UA to OA ("Swing heel touchdown"); and

(iii) Switching from OA to FA ("Leading toe touchdown").

The occurrence of these switching events is completely determined by the position and velocity of the robot's swing foot in the world frame as well as the ground-reaction force

experienced by the support foot. We use switching surfaces to describe the conditions under which a switching event occurs.

When the heel of the support foot takes off at the end of the FA phase, the robot enters the UA domain (Fig. 3-2). This support heel liftoff condition can be described using the vertical ground-reaction force applied at the support heel, denoted as $F_{c,z} : \mathcal{TQ} \times U \to \mathbb{R}$. We use $S_{F\to U}$ to denote the switching surface connecting an FA domain and its subsequent UA domain, and express it as:

$$S_{F\to U} := \{(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) \in \mathcal{TQ} \times U : F_{c,z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) = 0\}.$$

The UA-to-OA switching occurs when the swing foot's heel lands on the ground (Fig. 3-2). Accordingly, we express the switching surface that connects a UA domain and its subsequent OA domain, denoted as $S_{U\to O}$, as:

$$S_{U\to O}(\mathbf{q}, \dot{\mathbf{q}}) := \{(\mathbf{q}, \dot{\mathbf{q}}) \in \mathcal{TQ} : z_{swh}(\mathbf{q}) = 0,\ \dot{z}_{swh}(\mathbf{q}, \dot{\mathbf{q}}) < 0\},$$

where $z_{swh} : \mathcal{Q} \to \mathbb{R}$ represents the height of the lowest point within the swing-foot heel above the ground.

As the leading toe touches the ground at the end of an OA phase, a new FA phase is activated (Fig. 3-2). In this study, we assume that the leading toe landing and the trailing foot takeoff occur simultaneously at the end of an OA phase, which is reasonable because the trailing foot typically remains contact with the ground only for a brief period (e.g., approximately 3% of a complete human gait cycle [62]) after the touchdown of the leading foot's toe. The switching surface, $S_{O\to F}$, that connects an OA domain and its subsequent FA domain is then expressed as:

$$S_{O\to F}(\mathbf{q}, \dot{\mathbf{q}}) := \{(\mathbf{q}, \dot{\mathbf{q}}) \in \mathcal{TQ} : z_{swt}(\mathbf{q}) = 0,\ \dot{z}_{swt}(\mathbf{q}, \dot{\mathbf{q}}) < 0\},$$

where $z_{swt} : \mathcal{Q} \to \mathbb{R}$ represents the height of the swing-foot toe above the walking surface.

3.1.2.3 Discrete impact dynamics

The complete walking cycle involves two foot-landing instantaneous impacts; one impact occurs at the landing of the swing-foot heel (i.e., transition from UA to OA), and the other at the touchdown of the leading-foot toe between the OA and FA phases. Note that the switching from FA to UA, characterized by the support heel liftoff, is a continuous process that does not induce any impacts.

Recall Eq. (2.6) for the expression of the impact dynamics. Note that the dimension of $\boldsymbol{\Delta}_{\dot{q}}$ is invariant across the three domains since it characterizes the jumps of all floating-base generalized coordinates.

3.2 Controller Design for Three-Domain Walking

This section introduces the proposed GPT controller design based on the hybrid model of multi-domain bipedal robotic walking introduced in Section 3.1. The resulting controller provably ensures the exponential error convergence for the directly regulated DOFs within each domain. The sufficient conditions under which the proposed controller guarantees the stability for the overall hybrid system are provided in Section 3.3.

3.2.1 Desired Trajectory Encoding

As the primary control objective is to provably drive the global-position tracking error to zero, one set of desired trajectories that the proposed controller aims to reliably track is the robot's desired global-position trajectories. Since a bipedal humanoid robot typically has many more DOFs and actuators than the desired global-position trajectories, the controller could regulate additional variables of interest (e.g., swing-foot pose).

We use both time-based and state-based phase variables to encode these two sets of desired trajectories, as explained next.

3.2.1.1 Time-based encoding variable

We choose to use the global time variable t to encode the desired global-position trajectories so that a robot's actual horizontal position trajectories in the world (i.e., x_b and y_b) can be accurately controlled with precise timing, which is crucial for real-world tasks such as dynamic obstacle avoidance.

We use $x_d(t) : \mathbb{R}^+ \to \mathbb{R}$ and $y_d(t) : \mathbb{R}^+ \to \mathbb{R}$ to denote the desired global-position trajectories along the x- and y-axis of the world frame, respectively, and $\psi_d(t) : \mathbb{R}^+ \to \mathbb{R}$ is the desired heading/yaw angle. We assume that the desired horizontal global-position trajectories $x_d(t)$ and $y_d(t)$ are supplied by a higher-layer planner, and the design of this planner is not the focus of this chapter. Given $x_d(t)$ and $y_d(t)$, the desired heading direction $\psi_d(t)$ can be designed as a function of $x_d(t)$ and $y_d(t)$, which is $\psi_d(t) := \tan^{-1}(y_d/x_d)$. Such a definition ensures that the robot is facing forward during walking.

We consider the following assumption on the regularity condition of $x_d(t)$ and $y_d(t)$:

A3.7 The desired global-position trajectories $x_d(t)$ and $y_d(t)$ are planned as continuously differentiable on $t \in \mathbb{R}^+$ with the norm of $\dot{x}_d(t)$ and $\dot{y}_d(t)$ bounded above by a constant number; that is, there exists a positive constant L_d such that

$$\|\dot{x}_d(t)\|,\ \|\dot{y}_d(t)\| \leq L_d \tag{3.1}$$

for any $t \in \mathbb{R}^+$.

Under assumption (A3.7), the time functions $x_d(t)$ and $y_d(t)$ are Lipschitz continuous on $t \in \mathbb{R}^+$ [81], which we utilize in the proposed stability analysis.

3.2.1.2 State-based encoding variable

As robotic walking inherently exhibits a cyclic movement pattern in the robot's configuration space, it is natural to encode the desired motion trajectories of the robot with a phase variable that represents the walking progress within a cycle.

To encode the desired trajectories other than the desired global-position trajectories, we choose to use a state-based phase variable, denoted $\theta(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}$, that represents the

total horizontal distance traveled within a walking step. Accordingly, the phase variable $\theta(\mathbf{q})$ increases monotonically within each walking step during straight-line or curved-path walking, which ensures a unique mapping from $\theta(\mathbf{q})$ to the encoded desired trajectories. In contrast, in our previous work [73, 75], the phase variable is chosen as the walking distance projected along a single direction on the ground, which may not ensure such a unique mapping during curved-path walking.

Since the phase variable $\theta(\mathbf{q})$ is essentially the length of a 2-D curve that represents the horizontal projection of the 3-D walking path on the ground, we can use the actual horizontal velocities ($\dot{x}_b$ and $\dot{y}_b$) of the robot's base to express $\theta(\mathbf{q})$ as:

$$\theta(\mathbf{q}(t)) = \int_{t_0}^{t} \sqrt{\dot{x}_b^2(t) + \dot{y}_b^2(t)}dt, \tag{3.2}$$

where $t_0 \in \mathbb{R}^+$ represents the actual initial time instant of the given walking step and t is the current time.

The normalized phase variable, which represents the percentage completion of a walking step, is given by:

$$s(\theta) := \frac{\theta}{\theta_{max}}, \tag{3.3}$$

where the real scalar parameter θ_{max} represents the maximum value of the phase variable (i.e., the planned total distance to be traveled within a walking step). At the beginning of each step, the normalized phase variable takes a value of 0, while at the end of the step, it equals 1.

3.2.2 Output Function Design

An output function is a function that represents the difference between a control variable and its desired trajectory, which is essentially the trajectory tracking error. The proposed controller aims to drive the output function to zero for the overall hybrid walking process.

Due to the distinct robot dynamics among different domains, we design different output functions (including the control variables and desired trajectories) for different domains.

3.2.2.1 FA domain

We use $\mathbf{h}_c^F(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^n$ to denote the vector of n control variables that are directly commanded within the FA domain. Without loss of generality, we use the OP3 robot shown in Fig. 3-1 as an example to explain a common choice of control variables within the FA domain.

The OP3 robot has twenty directly actuated joints (i.e., $n = n_a = 20$) including eight upper body joints. Also, using n_{up} to denote the number of upper body joints, we have $n_{up} = 8$.

We choose the twenty control variables as follows:

(i) The robot's global position and orientation represented by the 6-D absolute base pose (i.e., position $\mathbf{p}_b$ and orientation $\boldsymbol{\gamma}_b$) w.r.t. the world frame;

(ii) The position and orientation of the swing foot w.r.t. the vehicle frame, respectively denoted as $\mathbf{p}_{sw}(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^3$ and $\boldsymbol{\gamma}_{sw}(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^3$; and

(iii) The angles of the n_{up} upper body joints $\mathbf{q}_{up} \in \mathbb{R}^{n_{up}}$.

We choose to directly control the global position of the robot to ensure that the robot's base follows the desired global-position trajectory. The base orientation is also directly commanded to guarantee a steady trunk (e.g., for mounting cameras) and the desired heading direction. The swing foot pose is regulated to ensure an appropriate foot posture at the landing event, and the upper body joints are controlled to avoid any unexpected arm motions that may affect the overall walking performance.

The stack of control variables $\mathbf{h}_c^F(\mathbf{q})$ are expressed as:

$$\mathbf{h}_c^F(\mathbf{q}) = \begin{bmatrix} x_b \\ y_b \\ \psi_b \\ z_b \\ \phi_b \\ \theta_b \\ \mathbf{p}_{sw} \\ \boldsymbol{\gamma}_{sw} \\ \mathbf{q}_{up} \end{bmatrix}. \tag{3.4}$$

We use $\mathbf{h}_d^F(t,s) : \mathbb{R}^+ \times [0,1] \to \mathbb{R}^n$ to denote the desired trajectories for the control variables $\mathbf{h}_c^F(\mathbf{q})$ within the FA domain. These trajectories are encoded by the global time t and the normalized state-based phase variable $s(\theta)$ as follows: (i) the desired trajectories of the base position variables x_b and y_b and the base yaw angle ψ_b are encoded by the global time t, while (ii) those of the other $(n-3)$ control variables, including the base height z_b, base roll angle ϕ_b, base pitch angle θ_b, swing-foot pose $\mathbf{p}_{sw}$ and $\boldsymbol{\gamma}_{sw}$, and upper joint angle $\mathbf{q}_{up}$, are encoded by the normalized phase variable $s(\theta)$.

The desired trajectory $\mathbf{h}_d^F(t,s)$ is expressed as:

$$\mathbf{h}_d^F(t,s) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ \psi_d(t) \\ \boldsymbol{\phi}^F(s) \end{bmatrix}, \tag{3.5}$$

where $x_d(t)$, $y_d(t)$, and $\psi_d(t)$ are defined in Section 3.2.1.1, and the function $\boldsymbol{\phi}^F(s) : [0,1] \to \mathbb{R}^{n-3}$ represents the desired trajectories of the control variables z_b, ϕ_b, θ_b, ψ_b, $\mathbf{p}_{sw}$, $\boldsymbol{\gamma}_{sw}$, and $\mathbf{q}_{up}$.

The desired function $\boldsymbol{\phi}^F(s)$ is parameterized by the Bézier polynomials (recall Sec. 2.3). The output function during an FA phase is defined as:

$$\mathbf{h}^F(t,\mathbf{q}) := \mathbf{h}_c^F(\mathbf{q}) - \mathbf{h}_d^F(t,s). \tag{3.6}$$

3.2.2.2 UA domain

As explained in Section 3.1.1, a robot has $(n+1)$ DOF within the UA domain but only n_a actuators. Thus, only n_a (i.e, n) variables can be directly commanded within the UA domain.

We opt to control individual joint angles within the UA domain to mimic human-like walking. By "locking" the joint angles, the robot can perform a controlled falling about the support toe, similar to human walking.

Thus, the control variable $\mathbf{h}_c^U(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^n$ is:

$$\mathbf{h}_c^U(\mathbf{q}) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dots \\ q_n \end{bmatrix}. \tag{3.7}$$

Let $\mathbf{h}_d^U(s) : [0,1] \to \mathbb{R}^n$ denote the desired joint position trajectories within the UA domain. These desired trajectories $\mathbf{h}_d^U(s)$ are parameterized using Bézier polynomials $\boldsymbol{\phi}^U(s) : [0,1] \to \mathbb{R}^n$; that is, $\mathbf{h}_d^U = \boldsymbol{\phi}^U(s)$. The function $\boldsymbol{\phi}^U(s)$ can be expressed similarly to $\boldsymbol{\phi}^F(s)$.

The associated output function is then given by:

$$\mathbf{h}^U(\mathbf{q}) := \mathbf{h}_c^U(\mathbf{q}) - \mathbf{h}_d^U(s). \tag{3.8}$$

3.2.2.3 OA domain

Let $\mathbf{h}_c^O(\mathbf{q}) : \mathcal{Q} \to \mathbb{R}^{n-4}$ denote the control variables within the OA domain. Recall that the robot has n_a actuators and $(n-4)$ DOFs within the OA domain.

We choose the $(n-4)$ control variables as:

(i) The robot's 6-D base pose w.r.t. the world frame;

(ii) The angles of the n_{up} upper body joints, $\mathbf{q}_{up}$; and

(iii) The pitch angles of the trailing and leading feet, denoted as $\theta_t(\mathbf{q})$ and $\theta_l(\mathbf{q})$, respectively.

Similar to the FA domain, we choose to directly command the robot's 6-D base pose within the OA domain to ensure satisfactory global-position tracking performance, as well as the upper body joints to avoid unexpected arm movements that could compromise the robot's balance. Also, regulating the pitch angle of the leading foot helps ensure a flat-foot posture upon switching into the subsequent FA domain where the support foot remains flat on the ground. Meanwhile, controlling the pitch angle of the trailing foot can prevent overly early or late foot-ground contact events.

Thus, the control variable $\mathbf{h}_c^O(\mathbf{q})$ is:

$$\mathbf{h}_c^O(\mathbf{q}) = \begin{bmatrix} x_b \\ y_b \\ \psi_b \\ z_b \\ \phi_b \\ \theta_b \\ \theta_t \\ \theta_l \\ \mathbf{q}_{up} \end{bmatrix}. \tag{3.9}$$

The desired trajectory $\mathbf{h}_d^O(t,s) : \mathbb{R}^+ \times [0,1] \to \mathbb{R}^{n-4}$ within the OA domain is expressed as:

$$\mathbf{h}_d^O(t,s) := \begin{bmatrix} x_d(t) \\ y_d(t) \\ \psi_d(t) \\ \boldsymbol{\phi}^O(s) \end{bmatrix}, \tag{3.10}$$

where $\boldsymbol{\phi}^O(s) : [0,1] \to \mathbb{R}^{n-7}$ represents the desired trajectories of z_b, ϕ_b, θ_b, θ_t, θ_l, and $\mathbf{q}_{up}$,

which, similar to $\boldsymbol{\phi}^F(s)$ and $\boldsymbol{\phi}^U(s)$, can be chosen as Bézier curves.

The tracking error $\mathbf{h}^O(t,\mathbf{q})$ is expressed as:

$$\mathbf{h}^O(t,\mathbf{q}) := \mathbf{h}_c^O(\mathbf{q}) - \mathbf{h}_d^O(t,s). \tag{3.11}$$

3.2.3 Input-Output Linearizing Control

The output functions representing the trajectory tracking errors can be compactly expressed as:

$$\mathbf{y}_i = \mathbf{h}^i(t,\mathbf{q}), \tag{3.12}$$

where the subscript $i \in \{F,U,O\}$ indicates the domain.

Due to the nonlinearity of the robot dynamics and the time-varying nature of the desired trajectories, the dynamics of the output functions are nonlinear and time-varying. To reduce the complexity of controller design, we use input-output linearization to convert the nonlinear, time-varying error dynamics into a linear time-invariant one.

Let $\mathbf{u}_i$ ($i \in \{F,U,O\}$) denote the joint torque vector within the given domain. We exploit the input-output linearizing control law [81]

$$\mathbf{u}_i = (\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}}\mathbf{M}^{-1}\bar{\mathbf{B}})^{-1}[(\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}})\mathbf{M}^{-1}\bar{\mathbf{c}} + \mathbf{v}_i - \frac{\partial^2 \mathbf{h}^i}{\partial t^2} - \frac{\partial}{\partial \mathbf{q}}(\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}}\dot{\mathbf{q}})\dot{\mathbf{q}}] \tag{3.13}$$

to linearize the continuous-phase output function dynamics (i.e., Eq. (2.5)) into $\ddot{\mathbf{y}}_i = \mathbf{v}_i$, where $\mathbf{v}_i$ is the control law of the linearized system. Here, the matrix $\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}}\mathbf{M}^{-1}\bar{\mathbf{B}}$ is invertible on $\mathcal{Q}$ because (i) $\mathbf{M}$ is invertible on $\mathcal{Q}$, (ii) $\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}}$ is full row rank on $\mathcal{Q}$ by design, and (iii) $\bar{\mathbf{B}}$ is full column rank on $\mathcal{Q}$.

It should be noted that $\mathbf{u}_i$ has different expressions in different domains, due to the variations in the control variables and desired trajectories. For instance, as the output function is time-independent within the UA domain, the function $\frac{\partial^2 \mathbf{h}^U}{\partial t^2}$ in Eq. (3.13) is always a zero vector because the output function $\mathbf{h}^U$ is explicitly time-independent.

We design $\mathbf{v}_i$ as a proportional-derivative (PD) controller

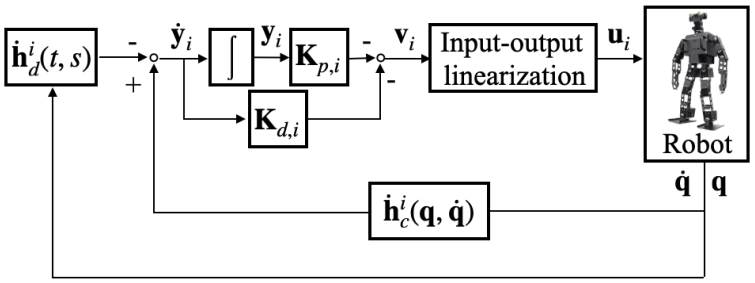

Figure 3-3: Block diagram of the proposed global-position tracking control law within each domain. Here $i \in \{F,U,O\}$ indicates the domain type.

$$\mathbf{v}_i = -\mathbf{K}_{p,i}\mathbf{y}_i - \mathbf{K}_{d,i}\dot{\mathbf{y}}_i, \tag{3.14}$$

where $\mathbf{K}_{p,i}$ and $\mathbf{K}_{d,i}$ are positive-definite diagonal matrices containing the proportional and derivative control gains, respectively. It is important to note that the dimension of the gains $\mathbf{K}_{p,i}$ and $\mathbf{K}_{d,i}$ depends on that of the output function in each domain; their dimension is $n \times n$ in FA and UA domains, and $(n-4) \times (n-4)$ in the OA domain.

We call the GPT control law in Eqs. (3.13) and (3.14) the "IO-PD" controller in the rest of this chapter, and the block diagram of the controller is shown in Fig. 3-3.

Under the IO-PD control laws, the closed-loop output function dynamics within domain i becomes linear:

$$\ddot{\mathbf{y}}_i = -\mathbf{K}_{d,i}\dot{\mathbf{y}}_i - \mathbf{K}_{p,i}\mathbf{y}_i.$$

Drawing upon the well-studied linear systems theory, we can ensure the exponential convergence of $\mathbf{y}_i$ to zero within each domain by properly choosing the values of the PD gain matrices ($\mathbf{K}_{p,i}$ and $\mathbf{K}_{d,i}$) [81].

3.3 Closed-Loop Stability Analysis for Three-Domain Walking

This section explains the proposed stability analysis of the closed-loop hybrid control system under the continuous IO-PD control law.

The continuous GPT law introduced in Section 3.2 with properly chosen PD gains achieves exponential stabilization of the output function state within each domain. Nevertheless, the stability of the overall hybrid dynamical system is not automatically ensured for two main reasons. First, within the UA domain, the utilization of the input-output linearization technique and the absence of actuators to directly control all the DOFs induce internal dynamics, which the control law cannot directly regulate [3, 82]. Second, the impact dynamics in Eq. (2.6) is uncontrolled due to the infinitesimal duration of an impact between rigid bodies (i.e., ground and swing foot). As both internal dynamics and reset maps are highly nonlinear and time-varying, analyzing their effects on the overall system stability is not straightforward.

To ensure satisfactory tracking error convergence for the overall hybrid closed-loop system, we analyze the closed-loop stability via the construction of multiple Lyapunov functions [83]. The resulting sufficient stability conditions can be used to guide the parameter tuning of the proposed IO-PD law for ensuring system stability and satisfactory tracking.

3.3.1 Hybrid Closed-Loop Dynamics

This subsection introduces the hybrid closed-loop dynamics under the proposed IO-PD control law in Eqs. (3.13) and (3.14), which serves as the basis of the proposed stability analysis.

3.3.1.1 State variables within different domains

The state variables of the hybrid closed-loop system include the output function state $(\mathbf{y}_i, \dot{\mathbf{y}}_i)$ $(i \in \{F, O, U\})$. This choice of state variables allows our stability analysis to exploit the linear dynamics of the output function state within each domain, thus greatly reducing the complexity of the stability analysis for the hybrid, time-varying, nonlinear closed-loop system.

We use $\mathbf{x}_F \in \mathbb{R}^{2n}$ and $\mathbf{x}_O \in \mathbb{R}^{2n-8}$ to respectively denote the state within the FA and OA domains, which are exactly the output function state:

$$\mathbf{x}_F := \begin{bmatrix} \mathbf{y}_F \\ \dot{\mathbf{y}}_F \end{bmatrix} \quad \text{and} \quad \mathbf{x}_O := \begin{bmatrix} \mathbf{y}_O \\ \dot{\mathbf{y}}_O \end{bmatrix}.$$

Within the UA domain, the output function state, denoted as $\mathbf{x}_\xi \in \mathbb{R}^{2n-2}$, is expressed as:

$$\mathbf{x}_\xi := \begin{bmatrix} \mathbf{y}_U \\ \dot{\mathbf{y}}_U \end{bmatrix}.$$

Besides $\mathbf{x}_\xi$, the complete state $\mathbf{x}_U$ within the UA domain also include the uncontrolled state, denoted as $\mathbf{x}_\eta \in \mathbb{R}^2$. Since the stance-foot pitch angle $\theta_{st}(\mathbf{q})$ is not directly controlled within the UA domain, we define $\mathbf{x}_\eta$ as:

$$\mathbf{x}_\eta := \begin{bmatrix} \theta_{st} \\ \dot{\theta}_{st} \end{bmatrix}.$$

Thus, the complete state within the UA domain is:

$$\mathbf{x}_U := \begin{bmatrix} \mathbf{x}_\xi \\ \mathbf{x}_\eta \end{bmatrix}. \tag{3.15}$$

3.3.1.2 Closed-loop error dynamics

The hybrid closed-loop error dynamics associated with the FA and OA domains share the following similar form:

$$\begin{cases} \dot{\mathbf{x}}_F = \mathbf{A}_F \mathbf{x}_F & \text{if } (t, \mathbf{x}_F^-) \notin S_{F\to U} \\ \mathbf{x}_U^+ = \boldsymbol{\Delta}_{F\to U}(t, \mathbf{x}_F^-) & \text{if } (t, \mathbf{x}_F^-) \in S_{F\to U} \end{cases}$$
$$\begin{cases} \dot{\mathbf{x}}_O = \mathbf{A}_O \mathbf{x}_O & \text{if } (t, \mathbf{x}_O^-) \notin S_{O\to F} \\ \mathbf{x}_F^+ = \boldsymbol{\Delta}_{O\to F}(t, \mathbf{x}_O^-) & \text{if } (t, \mathbf{x}_O^-) \in S_{O\to F} \end{cases} \tag{3.16}$$

with

$$\mathbf{A}_F := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_{p,F} & -\mathbf{K}_{d,F} \end{bmatrix} \text{ and } \mathbf{A}_O := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_{p,O} & -\mathbf{K}_{d,O} \end{bmatrix}, \tag{3.17}$$

where $\mathbf{I}$ is an identity matrix with an appropriate dimension, and $\boldsymbol{\Delta}_{F\to U} : \mathbb{R}^+ \times \mathbb{R}^{2n} \to \mathbb{R}^{2n+2}$ and $\boldsymbol{\Delta}_{O\to F} : \mathbb{R}^+ \times \mathbb{R}^{2n-8} \to \mathbb{R}^{2n}$ are respectively the reset maps of the state vectors $\mathbf{x}_F$ and $\mathbf{x}_O$. The expressions of $\boldsymbol{\Delta}_{F\to U}$ and $\boldsymbol{\Delta}_{O\to F}$ are omitted for space consideration and can be directly obtained by combining the expressions of the reset map $\boldsymbol{\Delta}_{\dot{q}}$ of the generalized coordinates in Eq. (2.6) and the output functions $\mathbf{y}_F$, $\mathbf{y}_O$, and $\mathbf{y}_U$.

The closed-loop error dynamics associated with the continuous UA phase and the subsequent UA→OA impact map can be expressed as:

$$\begin{cases} \begin{cases} \dot{\mathbf{x}}_\xi = \mathbf{A}_\xi \mathbf{x}_\xi \\ \dot{\mathbf{x}}_\eta = \mathbf{f}_\eta(t, \mathbf{x}_\eta, \mathbf{x}_\xi) \end{cases} & \text{if } (t, \mathbf{x}_U^-) \notin S_{U\to O} \\ \mathbf{x}_O^+ = \boldsymbol{\Delta}_{U\to O}(t, \mathbf{x}_\xi^-, \mathbf{x}_\eta^-) & \text{if } (t, \mathbf{x}_U^-) \in S_{U\to O} \end{cases} \tag{3.18}$$

where

$$\mathbf{A}_\xi := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_{p,U} & -\mathbf{K}_{d,U} \end{bmatrix}. \tag{3.19}$$

The expression of $\mathbf{f}_\eta$ in Eq. (3.18) can be directly derived using the continuous-phase dynamics equation of the generalized coordinates and the expression of the output function $\mathbf{y}_U$. Similar to $\boldsymbol{\Delta}_{F\to U}$ and $\boldsymbol{\Delta}_{O\to F}$, we can readily obtain the expression of the reset map $\boldsymbol{\Delta}_{U\to O} : \mathbb{R}^+ \times \mathbb{R}^{2n+2} \to \mathbb{R}^{2n-8}$ based on the reset map in Eq. (2.6) and the expressions of $\mathbf{y}_U$ and $\mathbf{y}_O$.

3.3.2 Multiple Lyapunov-Like Functions

The proposed stability analysis via the construction of multiple Lyapunov functions begins with the design of the Lyapunov-like functions. We use $V_F(\mathbf{x}_F)$, $V_U(\mathbf{x}_U)$, and $V_O(\mathbf{x}_O)$ to respectively denote the Lyapunov-like functions within the FA, UA, and OA domains, and introduce their mathematical expressions next.

3.3.2.1 FA and OA domains

As the closed-loop error dynamics within the continuous FA and OA phases are linear and time-invariant, we can construct the Lyapunov-like functions $V_F(\mathbf{x}_F)$ and $V_O(\mathbf{x}_O)$ as [84]:

$$V_F(\mathbf{x}_F) = \mathbf{x}_F^T \mathbf{P}_F \mathbf{x}_F \text{ and } V_O(\mathbf{x}_O) = \mathbf{x}_O^T \mathbf{P}_O \mathbf{x}_O$$

with $\mathbf{P}_i$ ($i \in \{F, O\}$) the solution to the Lyapunov equation

$$\mathbf{P}_i \mathbf{A}_i + \mathbf{A}_i^T \mathbf{P}_i = -\mathbf{Q}_i,$$

where $\mathbf{Q}_i$ is any symmetric positive-definite matrix with a proper dimension.

3.3.2.2 UA domain

As the input-output linearization technique is utilized and not all DOFs within the UA domain can be directly controlled, internal dynamics exist that cannot be directly controlled [72]. We design the Lyapunov-like function V_U for the UA domain as:

$$V_U = V_\xi(\mathbf{x}_\xi) + \beta \|\mathbf{x}_\eta\|^2, \tag{3.20}$$

where $V_\xi(\mathbf{x}_\xi)$ is a positive-definite function and β is a positive constant to be designed.

As the dynamics of the output function state $\mathbf{x}_\xi$ are linear and time-invariant, the construction of $V_\xi(\mathbf{x}_\xi)$ is similar to that of V_F and V_O:

$$V_\xi(\mathbf{x}_\xi) = {\mathbf{x}_\xi}^T \mathbf{P}_\xi \mathbf{x}_\xi,$$

where $\mathbf{P}_\xi$ is the solution to the Lyapunov equation

$$\mathbf{P}_\xi \mathbf{A}_\xi + \mathbf{A}_\xi^T \mathbf{P}_\xi = -\mathbf{Q}_\xi$$

with $\mathbf{Q}_\xi$ any symmetric positive-definite matrix with an appropriate dimension.

3.3.3 Definition of Switching Instants

In the following stability analysis, the three domains of the k^{th} ($k \in \{1,2,...\}$) walking step are, without loss of generality, ordered as:

$$FA \rightarrow UA \rightarrow OA.$$

For the k^{th} walking step, we respectively denote the *actual* values of the initial time instants of the FA phase, the $FA \rightarrow UA$ switching instant, the $UA \rightarrow OA$ switching instant, and the final time instant of the OA phase as:

$$T_{3k-3},\ T_{3k-2},\ T_{3k-1},\ \text{and}\ T_{3k}.$$

The corresponding *desired* switching instants are denoted as:

$$\tau_{3k-3},\ \tau_{3k-2},\ \tau_{3k-1},\ \text{and}\ \tau_{3k}.$$

Using these notations, the k^{th} actual complete gait cycle on $t \in (T_{3k-3}, T_{3k})$ comprises:

(i) Continuous FA phase on $t \in (T_{3k-3}, T_{3k-2})$;

(ii) FA$\rightarrow$UA switching at $t = T_{3k-2}^{-}$;

(iii) Continuous UA phase on $t \in (T_{3k-2}, T_{3k-1})$;

(iv) UA$\rightarrow$OA switching at $t = T_{3k-1}^{-}$;

(v) Continuous OA phase on $t \in (T_{3k-1}, T_{3k})$; and

(vi) OA$\rightarrow$FA switching at $t = T_{3k}^{-}$.

For brevity in notation in the following analysis, the values of any (scalar or vector) variable $\star$ at $t = T_{3k-j}^{-}$ and $t = T_{3k-j}^{+}$, i.e.,

$$\star(T_{3k-j}^{-}) \text{ and } \star(T_{3k-j}^{+}),$$

are respectively denoted as:

$$\star|_{3k-j}^{-} \text{ and } \star|_{3k-j}^{+}$$

for any $k \in \{1,2,...\}$ and $j \in \{0,1,2,3\}$.

3.3.4 Continuous-Phase Convergence and Boundedness of Lyapunov-Like Functions

As the output function state $\mathbf{x}_i$ ($i \in \{F, O, \xi\}$) is directly controlled, we can readily analyze the convergence of the output functions (and their associated Lyapunov-like functions, V_F, V_O, and V_ξ) within each domain based on the well-studied linear systems theory [81].

Proposition 1. (**Continuous-phase output function convergence within each domain**) *Consider the IO-PD control law in Eq.* (3.13)*, assumptions (A3.1)-(A3.7), and the following condition:*

(B1) The PD gains are selected such that $\mathbf{A}_F$*,* $\mathbf{A}_O$*, and* $\mathbf{A}_\xi$ *are Hurwitz.*

Then, there exist positive constants r_i*,* c_{1i}*,* c_{2i}*, and* c_{3i} *(*$i \in \{F, O, \xi\}$*) such that the Lyapunov-like functions* V_F*,* V_O*, and* V_ξ *satisfy the following inequalities*

$$c_{1i}\|\mathbf{x}_i\|^2 \leq V_i(\mathbf{x}_i) \leq c_{2i}\|\mathbf{x}_i\|^2 \quad \textit{and} \quad \dot{V}_i \leq -c_{3i}V_i \tag{3.21}$$

within their respective domains for any

$$\mathbf{x}_i \in B_{r_i}(\mathbf{0}) := \{\mathbf{x}_i : \|\mathbf{x}_i\| \leq r_i\},$$

where $\mathbf{0}$ *is a zero vector with an appropriate dimension.*

Moreover, Eq. (3.21) *yields*

$$V_F|_{3k-2}^{-} \leq e^{-c_{3F}(T_{3k-2}-T_{3k-3})}V_F|_{3k-3}^{+}, \tag{3.22}$$

$$V_O|_{3k}^{-} \leq e^{-c_{3O}(T_{3k}-T_{3k-1})}V_O|_{3k-1}^{+}, \tag{3.23}$$

and

$$V_\xi|_{3k-1}^{-} \leq e^{-c_{3\xi}(T_{3k-1}-T_{3k-2})} V_\xi|_{3k-2}^{+}, \tag{3.24}$$

which describe the exponential continuous-phase convergence of V_F, V_O, and V_ξ within their respective domains.

The proof of Proposition 1 is omitted as Proposition 1 is a direct adaptation of the Lyapunov stability theorems from [81]. Note that the explicit relationship between the PD gains and the continuous-phase convergence rates c_{3F}, c_{3O}, and $c_{3\xi}$ can be readily obtained based on Remark 6 of our previous work [75].

Due to the existence of the uncontrolled internal state, the Lyapunov-like function V_U does not necessarily converge within the UA domain despite the exponential continuous-phase convergence of V_ξ guaranteed by the proposed IO-PD control law that satisfies condition (B1). Still, we can prove that within the UA domain of any k^{th} walking step, the value of the Lyapunov-like function V_U just before switching out of the domain, i.e., $V_U|_{3k-1}^{-}$, is bounded above by a positive-definite function of the "switching-in" value of V_U, i.e., $V_U|_{3k-2}^{+}$, as summarized in Proposition 2.

Proposition 2. (**Boundedness of Lyapunov-like function within UA domain**) *Consider the IO-PD control law in Eq.* (3.13) *and all conditions in Proposition 1. There exists a positive real number r_{U1} and a positive-definite function $w_u(\cdot)$ such that*

$$V_U|_{3k-1}^{-} \leq w_u(V_U|_{3k-2}^{+})$$

holds for any $k \in \{1,2,...\}$ and $\mathbf{x}_U \in B_{r_{U1}}(\mathbf{0})$.

Rationale of proof: The proof of Proposition 2 is given in Appendix A.1.1. The boundedness of the Lyapunov-like function $V_U(\mathbf{x}_U)$ at $t = T_{3k-1}^{-}$ is proven based on the definition of $V_U(\mathbf{x}_U)$ given in Eq. (3.20) and the boundedness of $\left\|\mathbf{x}_U|_{3k-1}^{-}\right\|$. Recall $\mathbf{x}_U := \begin{bmatrix}\mathbf{x}_\xi^T & \mathbf{x}_\eta^T\end{bmatrix}^T$. We establish the needed bound on $\left\|\mathbf{x}_U|_{3k-1}^{-}\right\|$ through the bounds on $\left\|\mathbf{x}_\xi|_{3k-1}^{-}\right\|$ and $\left\|\mathbf{x}_\eta|_{3k-1}^{-}\right\|$, which are respectively obtained based on the bounds of their continuous-phase dynamics of $\mathbf{x}_\xi$ and $\mathbf{x}_\eta$ and the integration of those bounds within the given continuous UA phase. ■

3.3.5 Boundedness of Lyapunov-Like Functions across Jumps

Proposition 3. (**Boundedness across jumps**) *Consider the IO-PD control law in Eq.* (3.13)*, all conditions in Proposition 1, and the following two additional conditions:*

(B2) The desired trajectories $\mathbf{h}_d^i$ *(*$i \in \{F,U,O\}$*) are planned to respect the impact dynamics with a small, constant offset* γ_Δ*; that is,*

$$\|\boldsymbol{\Delta}_{F\to U}(\tau_{3k-2},\mathbf{0})\| \leq \gamma_\Delta, \tag{3.25}$$

$$\|\boldsymbol{\Delta}_{U\to O}(\tau_{3k-1},\mathbf{0})\| \leq \gamma_\Delta, \text{ and} \tag{3.26}$$

$$\|\boldsymbol{\Delta}_{O\to F}(\tau_{3k},\mathbf{0})\| \leq \gamma_\Delta. \tag{3.27}$$

(B3) The PD gains are chosen to ensure a sufficiently high convergence rate (i.e., c_{3F}*,* c_{3O}*, and* $c_{3\xi}$ *in Eqs.* (3.22)-(3.24)*) of* V_F*,* V_O*, and* V_ξ*.*

Then, there exists a positive real number r *such that for any* $k \in \{1,2,...\}$*,* $\mathbf{x}_i \in B_r(\mathbf{0})$*, and* $i \in \{F,U,O\}$*, the following inequalities*

$$\begin{aligned} &... \leq V_F|_{3k}^+ \leq V_F|_{3k-3}^+ \leq ... \leq V_F|_3^+ \leq V_F|_0^+, \\ &... \leq V_U|_{3k+1}^+ \leq V_U|_{3k-2}^+ \leq ... \leq V_U|_4^+ \leq V_U|_1^+, \\ &\text{and} \\ &... \leq V_O|_{3k+2}^+ \leq V_O|_{3k-1}^+ \leq ... \leq V_O|_5^+ \leq V_O|_2^+ \end{aligned} \tag{3.28}$$

hold; that is, the values of each Lyapunov-like function at their associated "switching-in" instants form a nonincreasing sequence.

Rationale of proof: The proof of Proposition 3 is given in Appendix A.1.2. The proof shows the derivation details for the first inequality in Eq. (3.28) (i.e., $V_F|_{3k}^+ \leq V_F|_{3k-3}^+$ for any $k \in \{1,2,...\}$), which can be readily extended to prove the other two inequalities.

The proposed proof begins the analysis of the time evolution of the three Lyapunov-like functions within a complete gait cycle from $t = T_{3k-1}^+$ to $t = T_{3k}^+$, which comprises three continuous phases and three switching events as listed in Section 3.3.3.

Based on the time evolution, the bounds on the Lyapunov-like functions V_F, V_O, and V_U at the end of their respective continuous phases are given in Proposition 1 and 2, while their bounds at the beginning of those continuous phases are established through the analysis of the reset maps $\mathbf{\Delta}_{F\to U}$, $\mathbf{\Delta}_{U\to O}$, and $\mathbf{\Delta}_{O\to F}$. Finally, we combine these bounds to prove $V_F|_{3k}^{+} \leq V_F|_{3k-3}^{+}$. ■

The offset γ_Δ is introduced in condition (B2) for two primary reasons. Firstly, since the system's actual state trajectories inherently possess the impact dynamics, the desired trajectories need to respect the impact dynamics sufficiently closely (i.e., γ_Δ is small enough) in order to avoid overly large errors after an impact [85, 86]. If the desired trajectories do not agree with the impact dynamics sufficiently closely, the tracking errors at the beginning of a continuous phase could be overly large even when the errors at the end of the previous continuous phase are small. Such error expansion could induce aggressive control efforts at the beginning of a continuous phase, which could reduce energy efficiency and might even cause torque saturation. Secondly, while it is necessary to enforce the desired trajectories to respect the impact dynamics (e.g., through motion planning), requiring the exact agreement with the highly nonlinear impact dynamics (i.e., $\gamma_\Delta = 0$) could significantly increase the computationally burden of planning, which could be mitigated by allowing a small offset.

3.3.6 Main Stability Theorem

We derive the stability conditions for the hybrid error system in Eqs. (3.16) and (3.18) based on Propositions 1-3 and the general stability theory via the construction of multiple Lyapunov functions [83].

Theorem 1. *(***Closed-loop stability conditions***) Consider the IO-PD control law in Eq.* (3.13). *If all conditions in Proposition 3 are met, the origin of the hybrid closed-loop error system in Eqs.* (3.16) *and* (3.18) *is locally stable in the sense of Lyapunov.*

Rationale of proof: The full proof of Theorem 1 is given in Appendix A.1.3. The key idea of the proof is to show that the closed-loop control system satisfies the general multiple-Lyapunov stability conditions given in [83] if all conditions in Proposition 3 are met.

■

3.4 Extension from Three-Domain Walking with Full Motor Activation to Two-Domain Walking with Inactive Ankle Motors

This section explains the design of a GPT control law for a two-domain walking gait to further illustrate the proposed controller design method. The controller is a direct extension of the proposed controller design for three-domain walking (with full motor activation). For brevity, this section focuses on describing the distinct aspects of the two-domain case compared to the three-domain case explained earlier.

We consider the case of two-domain walking where underactuation is caused due to intentional ankle motor deactivation instead of loss of full contact with the ground as in the case of three-domain walking. Bipedal gait is sometimes intentionally designed as underactuated through motor deactivation at the support ankle [87], which could simplify the controller design. Specifically, by switching off the support ankle motors, the controller can treat the support foot as part of the ground and only handle a point foot-ground contact instead of a finite support polygon.

Figure 3-4 illustrates a complete cycle of a two-domain walking gait, which comprises an FA and a UA domain, with the UA phase induced by intentional motor deactivation. The FA and UA phases share the same foot-ground contact conditions; that is, the toe and heel of the support foot are in a static contact with the ground. Yet, within the UA domain, the ankle-roll and ankle-pitch joints of the support foot are disabled, leading to $\text{DOF} = n_a + 2 > n_a$ (i.e., underactuation).

To differentiate from the case of three-domain walking, we add a "†" superscript to the left of mathematical symbols when introducing the two-domain case.

Hybrid robot dynamics: The continuous-time robot dynamics within the FA domain of two-domain walking have exactly the same expression as those of the three-domain dynamics in Eq. (2.3). The robot dynamics within the UA domain are also the same as Eq. (2.3) except for the input matrix $\mathbf{B}$ (due to the ankle motor deactivation).

The complete gait cycle contains one foot-landing impact event, which occurs as the

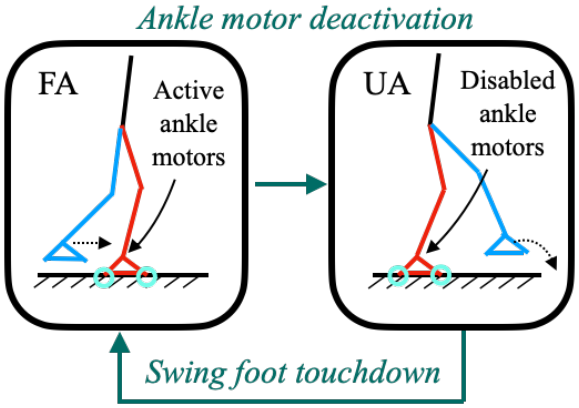

Figure 3-4: Illustration of a complete two-domain walking cycle. The green circles show the portions of the feet that touch the ground. The leg in red represents the support leg, while the leg in blue the swing leg. The movement of the swing foot is shown by the dashed arrow.

robot's state leaves the UA domain and enters the FA domain. The form of the associated impact map is similar to the impact map in Eq. (2.6) of the three-domain case. For brevity, we omit the expression and derivation details of the impact map.

There are two switching events, F→U and U→F, within a complete gait cycle, which are respectively denoted as ${}^{\dagger}S_{F\to U}$ and ${}^{\dagger}S_{U\to F}$ and given by:

$$
\begin{aligned}
&{}^{\dagger}S_{F\to U} := \{\mathbf{q} \in \mathcal{Q} : \theta(\mathbf{q}) > l_s\} \text{ and} \\
&{}^{\dagger}S_{U\to F} := \{(\mathbf{q}, \dot{\mathbf{q}}) \in \mathcal{TQ} : z_{sw}(\mathbf{q}) = 0, \dot{z}_{sw}(\mathbf{q}, \dot{\mathbf{q}}) < 0\},
\end{aligned}
$$

where $\theta(\mathbf{q})$ is defined as in Eq. (3.2) and the scalar positive variable l_s represents the desired traveling distance of the robot's base within the FA phase.

Local time-based phase variable: To allow the convenient adjustment of the intended period of motor deactivation, we introduce a new phase variable ${}^{\dagger}\theta(t)$ for the UA phase representing the elapsed time within this phase: ${}^{\dagger}\theta(t) = t - T_{Uk}$, where T_{Uk} is the initial time instant of the k^{th} UA phase.

The normalized phase variable is defined as: ${}^{\dagger}s({}^{\dagger}\theta) := \frac{{}^{\dagger}\theta}{\delta_{\tau_U}}$, where δ_{τ_U} is the expected duration of the UA. δ_{τ_U} can be assigned as a gait parameter that a motion planner adjusts for ensuring a reasonable duration of motor deactivation.

Output functions: The output function design within the FA domain is the same as the three-domain case.

The control variables within FA, denoted as ${}^{\dagger}\mathbf{h}_c^F(\mathbf{q})$, are chosen the same as the three-domain walking case in Eq. (3.4). Then, we have ${}^{\dagger}\mathbf{h}_c^F(\mathbf{q}) = \mathbf{h}_c^F(\mathbf{q})$. Accordingly, the desired trajectories ${}^{\dagger}\mathbf{h}_d^F(t,s)$ can be chosen the same as $\mathbf{h}_d^F(t,s)$, leading to the output function expressed as: ${}^{\dagger}\mathbf{h}^F(t,s) = {}^{\dagger}\mathbf{h}_c^F(\mathbf{q}) - {}^{\dagger}\mathbf{h}_d^F(t,s)$.

With two ankle (roll and pitch) motors disabled during the UA phase, the number of variables that can be directly controlled is reduced by two compared to the FA domain. Without loss of generality, We choose the control variables within the UA domain to be the same as the FA domain except that the base roll angle ϕ_b and base pitch angle θ_b are no longer controlled.

The control variables ${}^{\dagger}\mathbf{h}_c^U$ within the UA domain are then expressed as:

$$
{}^{\dagger}\mathbf{h}_c^U(\mathbf{q}) := \begin{bmatrix} x_b \\ y_b \\ \psi_b \\ z_b \\ \mathbf{p}_{sw}(\mathbf{q}) \\ \boldsymbol{\gamma}_{sw}(\mathbf{q}) \end{bmatrix}. \tag{3.29}
$$

The desired trajectories ${}^{\dagger}\mathbf{h}_d^U$ are given by:

$$
{}^{\dagger}\mathbf{h}_d^U(t,{}^{\dagger}s) := \begin{bmatrix} x_d(t) \\ y_d(t) \\ \psi_d(t) \\ {}^{\dagger}\boldsymbol{\phi}^U({}^{\dagger}s) \end{bmatrix}, \tag{3.30}
$$

where ${}^{\dagger}\boldsymbol{\phi}^U({}^{\dagger}s) : [0,1] \to \mathbb{R}^{n_a-5}$ represents the desired trajectories of z_b, $\mathbf{p}_{sw}$, and $\boldsymbol{\gamma}_{sw}$.

Then, we obtain the output function ${}^{\dagger}\mathbf{h}^U(t,\mathbf{q})$ as:

$$
{}^{\dagger}\mathbf{h}^U(t,\mathbf{q}) := {}^{\dagger}\mathbf{h}_c^U(\mathbf{q}) - {}^{\dagger}\mathbf{h}_d^U(t,{}^{\dagger}s). \tag{3.31}
$$

With the output function ${}^{\dagger}\mathbf{h}^i$ ($i \in \{F,U\}$) designed, we can use the same form of the

IO-PD control law in Eqs. (3.13) and (3.33) and the stability conditions in Theorem 1 to design the needed GPT controller for two-domain walking.

3.5 SIMULATION

This section reports the simulation results to demonstrate the satisfactory global-position tracking performance of the proposed controller design.

3.5.1 Comparative Controller: Input-Output Linearizing Control with Quadratic Programming

This subsection introduces the formulation of the proposed IO-PD controller as a quadratic program (QP) that handles the limited joint-torque capacities of real-world robots while ensuring a relatively accurate global-position tracking performance. We refer to the resulting controller as the "IO-QP" controller in this chapter. Besides enforcing the actuator limits and providing tracking performance guarantees, another benefit of the QP formulation lies in its computational efficiency for real-time implementation.

3.5.1.1 Constraints

We incorporate the IO-PD controller in Eq. (3.13) as an equality constraint in the proposed IO-QP control law. The proposed IO-QP also includes the torque limits as inequality constraints. We use $\mathbf{u}_{max,i}$ and $\mathbf{u}_{min,i}$ ($i \in \{F,U,O\}$) to denote the upper and lower limits of the torque command $\mathbf{u}_i$ given in Eq. (3.13). Then, the linear inequality constraint that the control signal $\mathbf{u}_i$ should respect can be expressed as: $\mathbf{u}_{min,i} \leq \mathbf{u}_i \leq \mathbf{u}_{max,i}$.

To ensure the control command $\mathbf{u}_i$ respects the actuator limits, we incorporate a slack variable $\boldsymbol{\delta}_{QP} \in \mathbb{R}^{n_a}$ in the equality constraint representing the IO-PD control law:

$$\mathbf{u}_i = \mathbf{N}(\mathbf{q},\dot{\mathbf{q}}) + \boldsymbol{\delta}_{QP}, \tag{3.32}$$

where $\mathbf{N} = (\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}}\mathbf{M}^{-1}\bar{\mathbf{B}})^{-1}[(\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}})\mathbf{M}^{-1}\bar{\mathbf{c}} + \mathbf{v}_i - \frac{\partial^2 \mathbf{h}^i}{\partial t^2} - \frac{\partial}{\partial \mathbf{q}}(\frac{\partial \mathbf{h}^i}{\partial \mathbf{q}}\dot{\mathbf{q}})\dot{\mathbf{q}}]$. To avoid overly large deviation from the original control law in Eq. (3.13), we include the slack variable in the cost

Table 3.1: Mass distribution of the OP3 robot.

Body component	Mass (kg)	Length (cm)
trunk	1.34	63
left/right thigh	0.31	11
left/right shank	0.22	11
left/right foot	0.07	12
left/right upper arm	0.19	12
left/right lower arm	0.04	12
head	0.15	N/A

function to minimize its norm as explained next.

3.5.1.2 Cost function

The proposed cost function is the sum of two components. One term is $\mathbf{u}_i^T \mathbf{u}_i$ and indicates the magnitude of the control command $\mathbf{u}_i$. Minimizing this term helps guarantee the satisfaction of the torque limit and the energy efficiency of walking.

The other term indicates the weighted norm of the slack variable $\boldsymbol{\delta}_{QP}$, i.e., $p\boldsymbol{\delta}_{QP}^T \boldsymbol{\delta}_{QP}$, with the real positive scalar constant p the slack penalty weight. By including the slack penalty term in the cost function, the deviation of the control signal from the original IO-PD form, which is caused by the relaxation, can be minimized.

3.5.1.3 QP formulation

Summarizing the constraints and cost function introduced earlier, we arrive at a QP given by:

$$\begin{aligned} \min_{\mathbf{u}_i, \boldsymbol{\delta}_{QP}} \quad & \mathbf{u}_i^T \mathbf{u}_i + p\ \boldsymbol{\delta}_{QP}^T \boldsymbol{\delta}_{QP} \\ \text{s.t.} \quad & \mathbf{u}_i = \mathbf{N} + \boldsymbol{\delta}_{QP} \\ & \mathbf{u}_i \geq \mathbf{u}_{min,i} \\ & \mathbf{u}_i \leq \mathbf{u}_{max,i} \end{aligned} \tag{3.33}$$

We present validation results for both IO-PD and IO-QP in the following section to demonstrate their effectiveness and performance comparison.

Table 3.2: Desired global-position trajectories.

Traj. index	$x_d(t)$ (cm)	$y_d(t)$ (cm)	Time interval (s)
(GP1)	$8t$	0	$[0,+\infty)$
(GP2)	$19.1t$	$5.9t$	$[0,+\infty)$
(GP3)	$25t$	0	$[0,3.13)$
	$3000\sin(\frac{t-3.13}{80})+78.2$	$3000\cos(\frac{t-3.13}{80})-3000$	$[3.13,4.25)$
	$24(t-4.25)+120$	$-7(t-4.25)-0.3$	$[4.25,+\infty)$

3.5.2 Simulation Setup

3.5.2.1 Robot model

The robot used to validate the proposed control approach is an OP3 bipedal humanoid robot developed by ROBOTIS, Inc. (see Fig.3-1). The OP3 robot is 50 cm tall and weighs approximately 3.2 kg. It is equipped with 20 active joints, as shown in Fig. 3-1. The mass distribution and geometric specifications of the robot are listed in Table 3.1. To validate the proposed controller, we use the MATLAB ODE solver `ODE45` to simulate the dynamics models of the OP3 robot for both three-domain walking and two-domain walking. The default tolerance settings of the `ODE45` solver are used.

3.5.2.2 Desired global-position trajectories and walking patterns

As mentioned earlier, this study assumes that the desired global-position trajectories are provided by a higher-layer planner. To assess the effectiveness of the proposed controller, three different desired global-position (GP) trajectories are tested, including single-direction and varying-direction trajectories. These trajectories are specified in Table. 3.2.

The GPs include two straight-line global-position trajectories with distinct heading directions, labeled as (GP1) and (GP2). We set the velocities of (GP1) and (GP2) to be different to evaluate the performance of the controller under different walking speeds. To assess the effectiveness of the proposed control law in tracking the desired global-position trajectories along a path with different walking directions, we also consider a walking trajectory (GP3) consisting of two straight-line segments connected via an arc.

The desired functions $\boldsymbol{\phi}^F$, $\boldsymbol{\phi}^U$, $\boldsymbol{\phi}^O$, and ${}^{\dagger}\boldsymbol{\phi}^U$ are designed as Bézier curves (Section 3.2). To respect the impact dynamics as prescribed by condition (B2), their parameters could be

Table 3.3: Initial tracking error norms for three cases.

Tracking error norm	Case A	Case B	Case C
swing foot position (% of step length)	27.5	27.5	40
base orientation (deg.)	0	17	12
base position (% of step length)	15	15	8

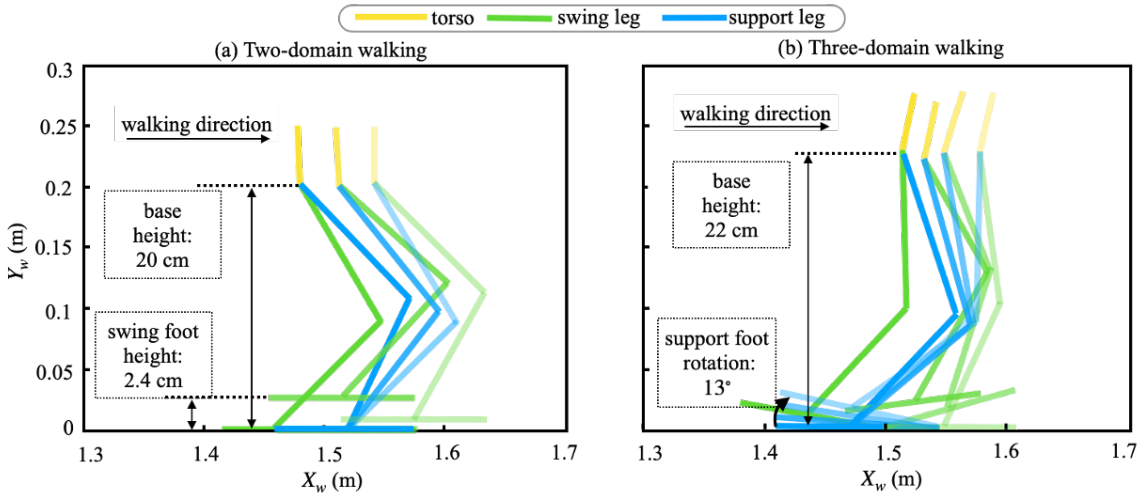

Figure 3-5: Desired walking patterns for (a) two-domain walking (Cases A and B) and (b) three-domain walking (Case C) in the sagittal plane. The labels X_w and Y_w represent the x- and y-axes of the world frame, respectively.

designed using the methods introduced in [67]. The desired walking patterns corresponding to the desired functions $\boldsymbol{\phi}^F$, $\boldsymbol{\phi}^U$, $\boldsymbol{\phi}^O$, and ${}^{\dagger}\boldsymbol{\phi}^U$ used in this study are illustrated in Fig. 3-5. In three-domain walking (3-5 (a)), the FA, UA, and OA phases take up approximately 33%, 8%, and 59% of one walking step, respectively, while the FA and UA phases of the two-domain walking gait (Fig. 3-5 (b)) last 81% and 19% of a step, respectively. For both walking patterns, the step length and maximum swing foot height are 7.1 cm and 2.4 cm, respectively.

3.5.2.3 Simulation cases

To validate the proposed controller under different desired global-position trajectories, walking patterns, and initial errors, we simulate the following three cases:

(Case A): Combination of desired trajectory (GP1) and two-domain walking pattern (Fig. 3-

5, left).

(Case B): Combination of desired trajectory (GP2) and two-domain walking pattern (Fig. 3-5, left).

(Case C): Combination of desired trajectory (GP3) and three-domain walking pattern (Fig. 3-5, right).

Table 3.3 summarizes the initial tracking error norms for all cases. Note that the initial swing-foot position tracking error is roughly 30-40% of the nominal step length.

3.5.2.4 Controller setting

For the IO-PD and IO-QP controllers, the PD controller gains are set as $\mathbf{K}_{p,i} = 225 \cdot \mathbf{I}$ and $\mathbf{K}_{d,i} = 50 \cdot \mathbf{I}$ to ensure the matrix $\mathbf{A}_i$ ($i \in \{F,U,O\}$) is Hurwitz matrix. For the IO-QP controller, the slack penalty weight p (Eq. (3.33)) is set as $p = 10^7$. On a desktop with an i7 CPU and 32GB RAM running MATLAB, it takes approximately 1 ms to solve the QP problem in Eq. (3.33).

To verify the stability of the multi-domain walking system, we construct the three Lyapunov-like functions V_f, V_u, and V_O as introduced in Section 3.3. In all domains, the matrix $\mathbf{P}_i$ (where $i \in \{F,U,O\}$) is obtained by solving the Lyapunov equation using the gain matrices $\mathbf{K}_{p,i}$ and $\mathbf{K}_{d,i}$ and the matrix $\mathbf{Q}_i$. Here without loss of generality, we choose $\mathbf{Q}_i$ as an identity matrix. For the UA phase, the value of β in the definition of V_U in Eq. (3.20) is set as 0.001.

3.5.3 Simulation Results

This subsection presents the tracking results of our proposed IO-PD and IO-QP controller for Cases A through C.

3.5.3.1 Global-position tracking performance

Figures 3-6 and 3-7 show the tracking performance of the proposed IO-PD and IO-QP controllers under Cases A and B, respectively. As explained earlier, Cases A and B share the

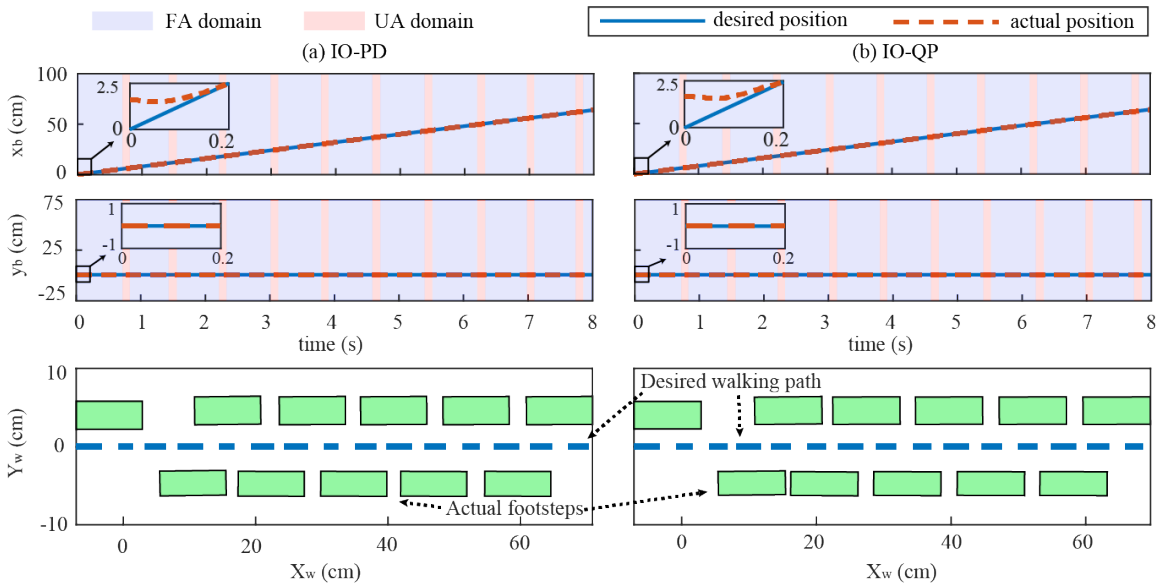

Figure 3-6: Satisfactory global-position tracking performance under Case A. The top row shows the global-position tracking results, and the bottom row displays the straight-line desired walking path and the actual footstep locations. The initial errors are listed in the Table 3.3.

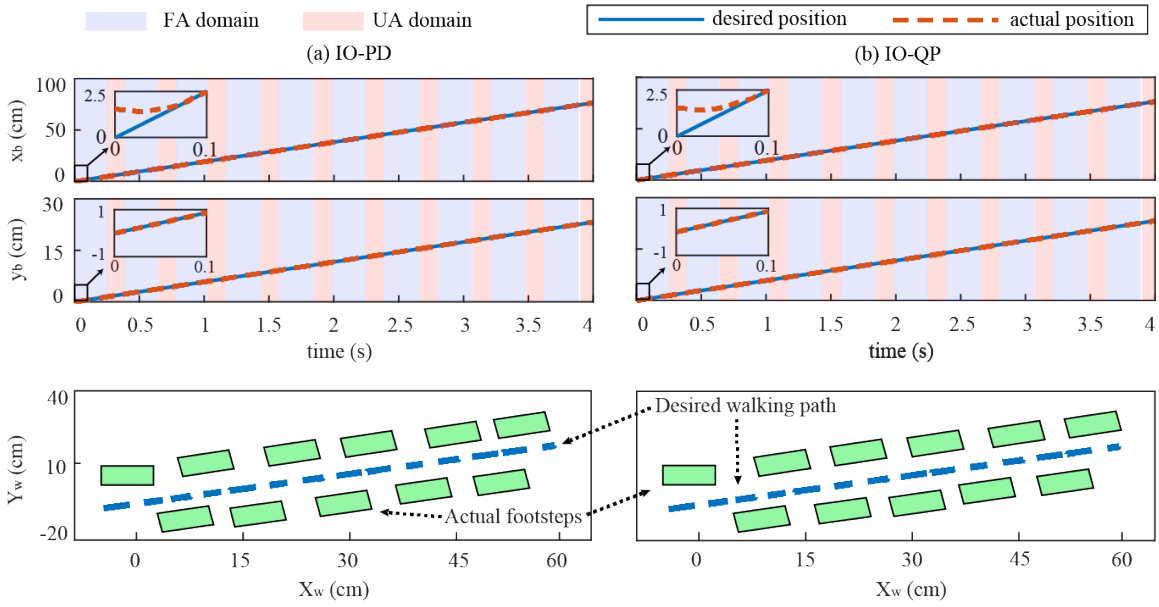

Figure 3-7: Satisfactory global-position tracking performance under Case B. The top row shows the global-position tracking results, and the bottom row displays the desired straight-line walking path and the actual footstep locations. The initial errors are listed in the Table 3.3.

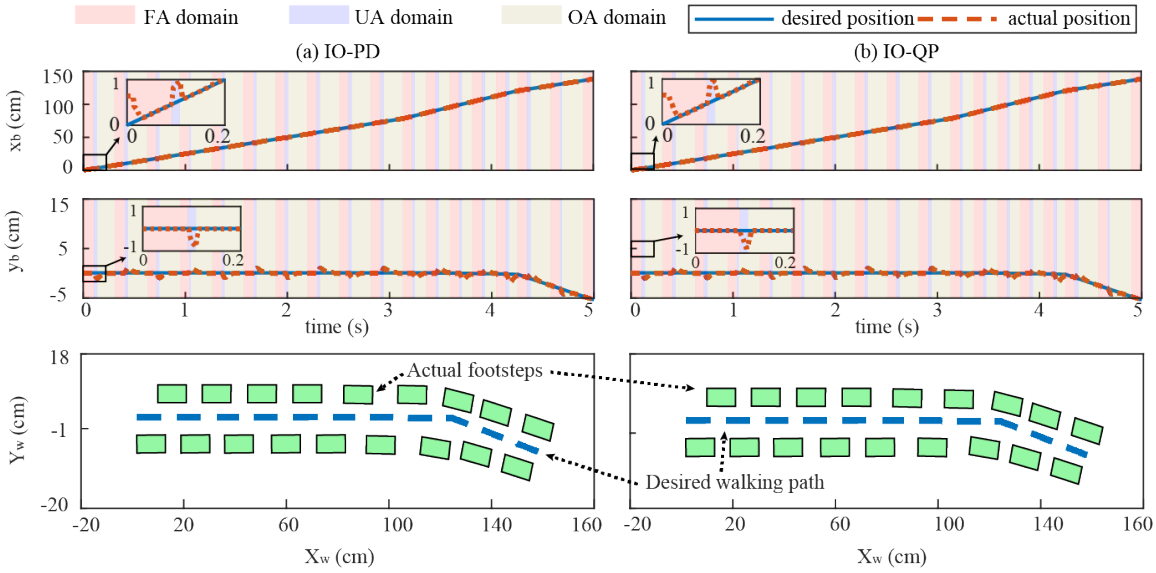

Figure 3-8: Satisfactory global-position tracking performance under Case C. The top row shows the global-position tracking results, and the bottom row displays the desired walking path and the actual footstep locations. The desired walking path consists of two straight lines connected by an arc. The initial errors are listed in the Table 3.3.

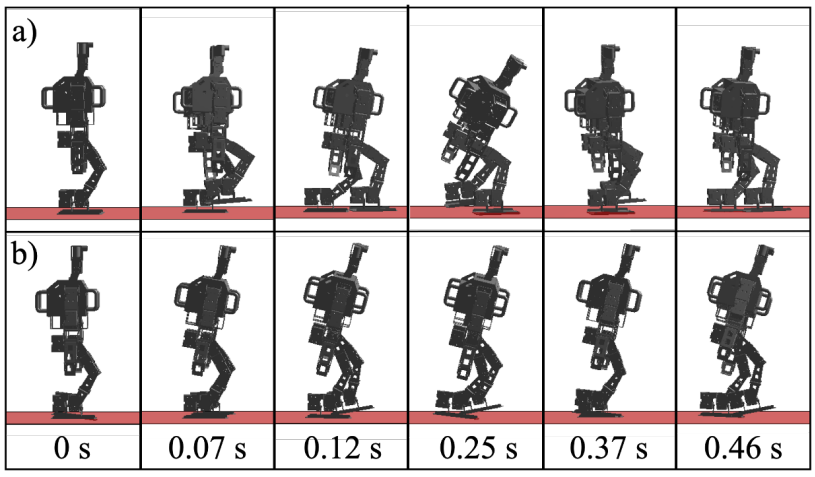

Figure 3-9: Time-elapsed illustration of three-domain walking.

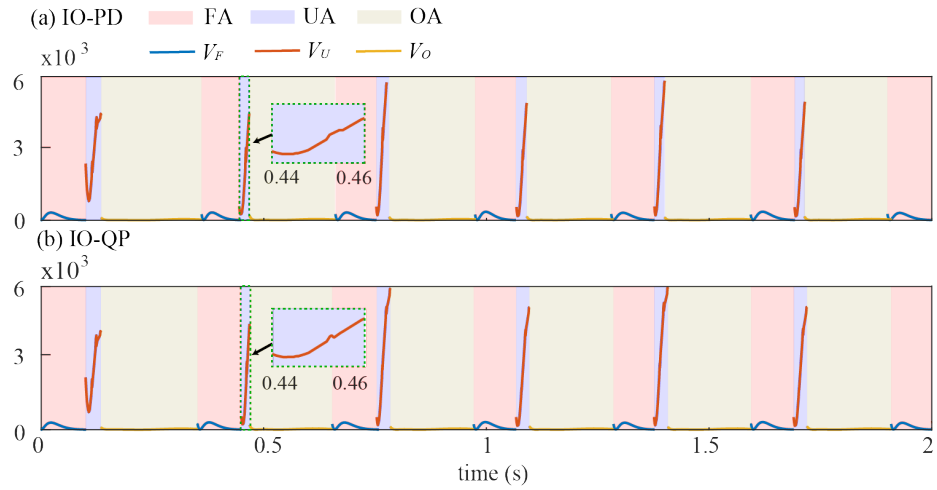

Figure 3-10: Time evolutions of multiple Lyapunov-like functions under Case C. The closed-loop stability is confirmed by the behaviors of the multiple Lyapunov functions, which complies with conditions (C1)-(C3) stated in the proof of Theorem 1 for both (a) IO-PD and (b) IO-QP control laws.

same desired walking pattern of two-domain walking (Fig. 3-9 a)), but they have different desired global-position trajectories and initial errors. For both cases, the IO-PD and IO-QP controllers satisfactorily drive the robot's actual horizontal global position (x_b, y_b) to the desired trajectories $(x_d(t), y_d(t))$, as shown in the top four plots in each figure. Also, from the footstep locations displayed at the bottom of each figure, the robot is able to walk along the desired walking path over the ground. In particular, the footstep trajectories in Fig. 3-7 demonstrate that even with a notable initial error (approx. 17°) of the robot's heading direction, the robot is able to quickly converge to the desired walking path.

Figure 3-8 displays the global-position tracking results of three-domain walking for Case C (Fig. 3-9 b)). The top two plots, i.e., the time profiles of the forward and lateral base position (x_b and y_b), show that the actual horizontal global position diverges from the reference within the UA phase during which the global position is not directly controlled. Despite the error divergence within the UA phase, the actual global position still converges to close to zero over the entire walking process thanks to convergence within the FA and OA domains, confirming the validity of Theorem 1.

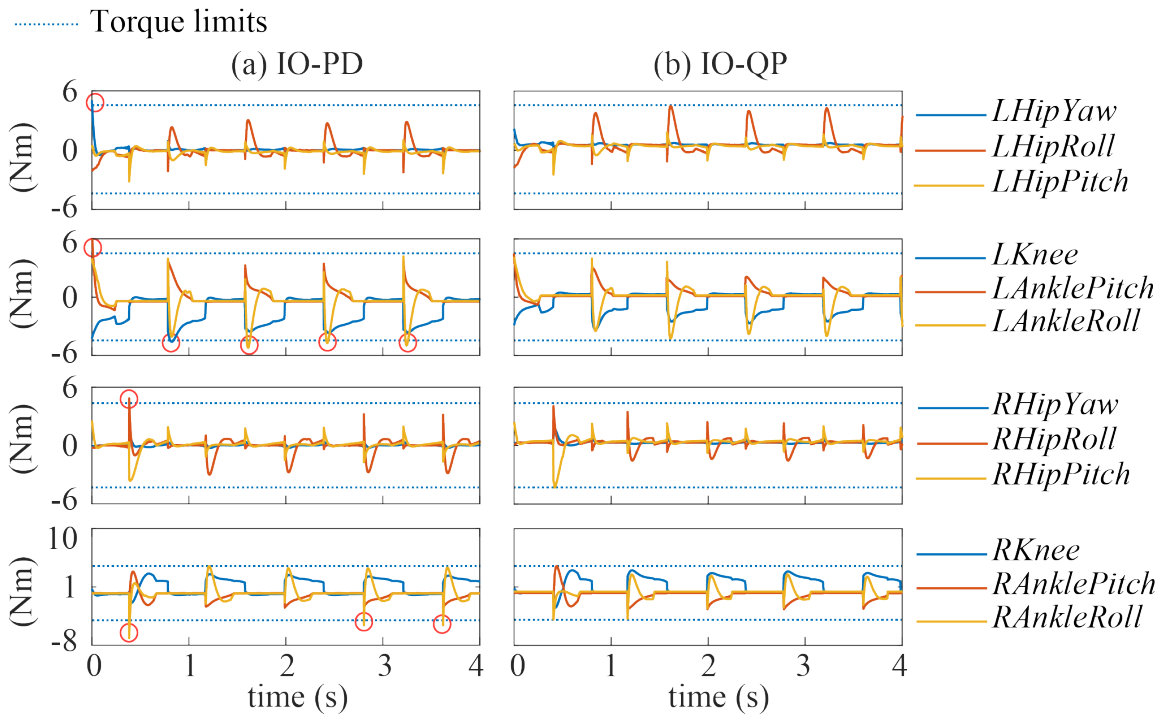

Figure 3-11: Torque profiles of each leg motor under the proposed (a) IO-PD and (b) IO-QP controllers for Case B. "L" and "R" stand for left and right, respectively. The red circles highlight the occurrence of torque limit violations. The jumps are more significant under the IO-PD controller than the IO-QP controller because the latter explicitly meets the torque limits. The blue dotted line represents the torque limits. It is evident that the torque profile of the IO-QP controller adheres to the torque limits, whereas the torque profile of the IO-PD controller may exceed the torque limits.

3.5.3.2 Convergence of Lyapunov-like functions

The multiple Lyapunov-like functions for case C, implemented with IO-PD and IO-QP control laws, is illustrated in Figure 3-10. Both control laws ensure the continuous-phase convergence of V_F and V_O satisfies condition (B1). Although V_U diverges during the UA phase, it remains bounded, thereby satisfying condition (B3). Moreover, we know the desired trajectories parameterized as Bézier curves are planned to satisfy (B2). Therefore, the multiple Lyapunov-like functions behave as predicted by conditions (C1)-(C3) in the proof of Theorem 1, indicating closed-loop stability.

3.5.3.3 Satisfaction of torque limits

Figure 3-11 illustrates the joint torque profiles of each leg motor under the IO-PD and IO-QP control methods for Case B. The torque limits u_{max} and u_{min} are set as 4.1 N and -4.1 N, respectively. It is observed that the torque experiences sudden spikes due to the foot-landing impact at the switching from the UA to the FA phases. Due to the notable initial tracking errors, there are also multiple spikes in the joint torques at the beginning of the entire walking process. These spikes tend to be more significant with the IO-PD controller than with the IO-QP controller. In fact, all of the torque peaks under IO-QP are within the torque limits whereas some of those peaks under IO-PD exceed the limits, which is primarily due to the fact that the IO-QP controller explicitly enforces the torque limits but IO-PD does not. This comparison highlights the advantage of using IO-QP over IO-PD in ensuring satisfaction of actuation constraints.

3.6 Discussion

This study has introduced a nonlinear GPT control approach for 3-D multi-domain bipedal robotic walking based on hybrid full-order dynamics modeling and multiple Lyapunov stability analysis. Similar to the HZD-based approaches [48, 88, 89] for multi-domain walking, our controller only acts within continuous phases, leaving the discrete impact dynamics uncontrolled. Another key similarity lies in that we also build the controller based on the hybrid, nonlinear, full-order dynamics model of multi-domain walking that faithfully captures the true robot dynamics and we exploit the input-output linearization technique to exactly linearize the complex continuous-phase robot dynamics.

Despite these similarities, our control law focuses on accurately tracking the desired global-position trajectories with the precise timing, whereas the HZD-based approach may not be directly extended to achieve such global-position tracking performance. This is essentially caused by the different stability types that the two approaches impose. The stability conditions proposed in this study enforce the stability of the desired global-position trajectory, which is a time function encoded by the global time. In contrast, the stability conditions underlying the HZD framework ensure the stability of the desired periodic orbit,

which is a curve in the state space on which infinitely many global-position trajectories reside.

Our previous GPT controller design [71] for the multi-domain walking of a 2-D robot is only capable of tracking straight-line paths. By explicitly modeling the robot dynamics associated with 3-D walking and considering the robot's 3-D movement in the design of the desired trajectories, the proposed approach is capable of ensuring satisfactory global-position tracking performance for 3-D walking.

One limitation of the proposed approach is that it may be non-feasible to meet the proposed stability conditions in practice if the duration of the underactuation phase, δ_{τ_U}, is overly large. From Eq. (A.20) in the proof of Proposition 3, we know that as δ_{τ_U} increases, α_2 will also increase, leading to a larger value of $\bar{N}$. If $\bar{N}$ is overly large, Eq. (3.28) will no longer hold, and the stability conditions will be invalid. To resolve this potential issue, the nominal duration of the UA domain cannot be set overly long. Indeed, the percentage of the UA phase within a complete gait cycle is respectively 19% and 8% of the simulated two-domain and three-domain walking, which is comparable to that of human walking (i.e., 18% [88]).

Another limitation of our control laws lies in that the robot dynamics model needs to be sufficiently accurate for the controller to be effective, due to the utilization of the input-output linearization technique. Yet, model parametric errors, external disturbances, and hardware imperfections (e.g., sensor noise) are prevalent in real-world robot operations [90]. To enhance the robustness of the proposed controller for real-world applications, we can incorporate robust control [91, 92, 93, 74, 94] into the GPT control law to address uncertainties. Furthermore, we can exploit online footstep planning [95, 96, 97, 98, 99, 100] to adjust the robot's desired behaviors in real-time to better reject modeling errors and external disturbances.

3.7 Summary

- A continuous tracking control law is introduced that achieves provably accurate global-position tracking for the hybrid model of multi-domain bipedal robotic walking

involving different actuation types.

- Both a three-domain and a two-domain walking gait are investigated to illustrate the effectiveness of the proposed approach, and the input-output linearizing controller was cast into a quadratic program (QP) to handle the actuator torque saturation.
- Finally, the performance of the input-output linearizing control law with and without the QP formulation is compared to highlight the effectiveness of the former in mitigating torque saturation while ensuring the closed-loop stability and trajectory tracking accuracy.
- We have previously published research on global-position tracking control for fully actuated robot walking [60, 75] and planar multi-domain walking [101]. Our most recent work on this topic has been submitted [102].

Chapter 4 Invariant Extended Kalman Filtering for Legged Robot Locomotion

In this chapter, we delve into locomotion on dynamic rigid surfaces (DRSes) and address state estimation challenges. The subsequent chapter will explore motion generation and control within a similar context. State estimation is crucial for providing a robot with accurate movement state information, including trunk pose and velocity, that are essential for planning and control. While extensive research has tackled state estimation on static and unstable surfaces, DRSes such as ships and aircraft remains underexplored. The challenge lies in the nonstationary nature of surface-foot contact points and the hybrid robot dynamics involving both continuous behaviors and discrete foot-landing events. The main objective of this chapter is to introduce the design of a state estimator that takes into explicit account the motion DRSes and achieves accurate, rapidly convergent, and real-time estimation.

The extended Kalman filter (EKF) [103, 104, 105] has been a standard method for real-time state estimation in legged locomotion on static surfaces using common on-board sensors. Recently, EKF-based estimators have been adapted for estimating a robot's trunk/base pose and velocity [32, 106, 107]. However, they suffer limitations in handling large estimation errors. To address these issues and ensure rapid convergence under substantial estimation errors, the previous EKF-based design [32] has be expanded into an invariant extended Kalman filter (InEKF) [24] for legged locomotion on static surfaces [29, 108]. Subsequently, the InEKF has been extended to incorporate smoothing techniques in works by Chauchat et al.[109] and by Yoon et al.[110]. The InEKF methodology has also found application in various other domains, including wheeled vehicles [111], underwater vehicles [112], aircraft [113], and human movement [114, 115]. However, its effectiveness in DRS locomotion, especially under significant surface motion, remains unclear due to its underlying assumption of stationary surface-foot contact.

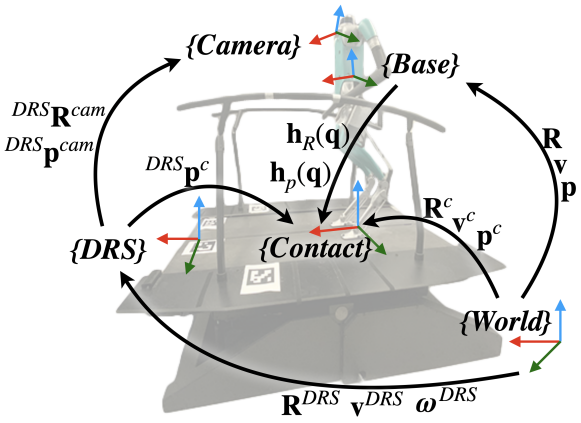

Figure 4-1: Illustration of coordinate frames and key variables. The treadmill is a DRS that rotates in the world frame.

4.1 Problem Formulation

In the proposed filter design, we focus on key state variables essential for locomotion planning and control. These include the robot's linear velocity ($\mathbf{v} \in \mathbb{R}^3$) and orientation ($\mathbf{R} \in SO(3)$) in the world frame, as well as the base position ($\mathbf{p}^b \in \mathbb{R}^3$) and contact-point position ($\mathbf{p}^c \in \mathbb{R}^3$). These selections allow us to incorporate forward kinematics between the base and contact/foot frames in the filter design [32, 29].

The considered DRS shares characteristics with real-world DRSes such as aircraft and vessels. Firstly, when operating on such surfaces, a robot only has access to surface-attached landmarks, rather than landmarks fixed in the inertial frame, due to limited visibility of the environment. Secondly, we assume a relatively accurate knowledge of the DRS's orientation ($\mathbf{R}^{DRS} \in SO(3)$) and linear and angular velocities ($\mathbf{v}^{DRS}, \boldsymbol{\omega}^{DRS} \in \mathbb{R}^3$) (see Fig.4-1). This assumption is reasonable as real-world DRSs typically incorporate high-precision motion monitoring systems [116, 117]. The proposed filter design explicitly accounts for the inevitable inaccuracies in the surface pose and motion knowledge, as discussed in Sec. 4.2.

The sensors utilized in this study consist of standard on-board equipment, including an inertial measurement unit (IMU) mounted on the robot's base/trunk, joint encoders, an RGB-D camera, and a contact indicator. The RGB-D camera tracks landmarks fixed to the DRS. This tracking allows the calculation of the camera's pose w.r.t. the DRS frame. The contact indicator serves to detect foot landing events. The encoders measure the joint angles

$\mathbf{q} \in \mathbb{R}^m$ with m the number of joints. Corrupted by white zero-mean Gaussian noise $\mathbf{w}^q$, the raw encoder data $\tilde{\mathbf{q}}$ is expressed as: $\tilde{\mathbf{q}} = \mathbf{q} + \mathbf{w}^q$. The IMU includes a gyroscope and an accelerator that respectively measure the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and linear acceleration $\mathbf{a} \in \mathbb{R}^3$ of the IMU in the IMU/base frame. Corrupted by white Gaussian zero-mean noise $\mathbf{w}^a, \mathbf{w}^\omega \in \mathbb{R}^3$, as well as biases $\mathbf{b}^a, \mathbf{b}^\omega \in \mathbb{R}^3$, the IMU readings $\tilde{\mathbf{a}}$ and $\tilde{\boldsymbol{\omega}}$ are expressed as: $\tilde{\mathbf{a}} = \mathbf{a} + \mathbf{b}^a + \mathbf{w}^a$ and $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \mathbf{b}^\omega + \mathbf{w}^\omega$.

4.1.1 Continuous-Phase IMU Motion and Bias Dynamics

To form the process model, we choose to adopt the IMU motion dynamics due to its accuracy and simplicity [32]. At time t, the IMU motion dynamics is given by:

$$\begin{aligned} \dot{\mathbf{R}}_t &= \mathbf{R}_t(\tilde{\boldsymbol{\omega}}_t - \mathbf{b}_t^\omega - \mathbf{w}_t^\omega)_\times, \\ \dot{\mathbf{v}}_t &= \mathbf{R}_t(\tilde{\mathbf{a}}_t - \mathbf{b}_t^a - \mathbf{w}_t^a) + \mathbf{g}, \text{ and } \dot{\mathbf{p}}_t = \mathbf{v}_t, \end{aligned} \tag{4.1}$$

where $(\cdot)_\times$ is a skew-symmetric matrix and $\mathbf{g}$ is the gravitational acceleration vector. The IMU bias dynamics is modeled as Brownian motion [29]:

$$\dot{\mathbf{b}}_t^a = \mathbf{w}_t^{ba} \text{ and } \dot{\mathbf{b}}_t^\omega = \mathbf{w}_t^{b\omega}, \tag{4.2}$$

where $\mathbf{w}_t^{ba}$ and $\mathbf{w}_t^{b\omega}$ are white Gaussian noise with zero mean.

4.1.2 Continuous-Phase Contact-Point Motion Dynamics

During DRS locomotion, the support foot moves w.r.t. the world frame due to the surface's motion. Consequently, the deterministic motion model differs from prior work on static surface locomotion [32, 29], where the contact point's motion was assumed to be $\dot{\mathbf{p}}_t^c = \mathbf{0}$. Instead, we explicitly consider the contact point velocity $\mathbf{v}_t^c$ in the model:

$$\dot{\mathbf{p}}_t^c = \mathbf{v}_t^c. \tag{4.3}$$

In this study, we inform the model in Eq.(4.3) by directly measuring the contact point

velocity. This measurement is based on the known surface pose and motion, along with the measured contact position in the DRS frame. The kinematics for this velocity measurement are as follows (see Fig.4-1):

$$\mathbf{v}_t^c = \mathbf{v}_t^{DRS} + \boldsymbol{\omega}_t^{DRS} \times (\mathbf{R}_t^{DRS\ DRS}\mathbf{p}_t^c). \tag{4.4}$$

Here, ${}^{DRS}\mathbf{p}_t^c$ represents the contact point position relative to the DRS frame, expressed in the DRS frame. This value can be computed using the robot's camera and encoder data. It is worth noting that we assume the surface orientation $\mathbf{R}_t^{DRS}$ and motion $\boldsymbol{\omega}^{DRS}, \mathbf{v}_t^{DRS}$ are known, as explained earlier. For an example of how to compute $\mathbf{v}_t^c$, refer to Sec. 4.4.

To account for velocity measurement inaccuracies, we consider:

$$\tilde{\mathbf{v}}_t^c = \mathbf{v}_t^c + \mathbf{R}_t \mathbf{w}_t^c, \tag{4.5}$$

Here, $\tilde{\mathbf{v}}_t^c \in \mathbb{R}^3$ represents the measured contact point velocity, and $\mathbf{w}_t^c$ models the inaccuracy as white Gaussian zero-mean noise expressed in the base frame.

4.1.3 Discrete Jump Dynamics at a Foot Landing

During a foot landing, there is a role switch between the swing and support legs, which results in a discrete jump in the contact point position $\mathbf{p}_t^c$. To ensure an appropriate propagation of the estimate and covariance during foot landings, the proposed filter explicitly considers this jump.

The jump map for the contact point position $\mathbf{p}_t^c$ is defined as:

$$\mathbf{p}_{t^+}^c = \mathbf{p}_t^c + \mathbf{R}_t \mathbf{h}_c(\mathbf{q}_t) \tag{4.6}$$

Here, the subscript t^+ denotes the timing just after the foot landing at time t. The function $\mathbf{h}_c$ represents the forward kinematics from the previous support-foot position to the new one, expressed in the base frame. Importantly, all other state variables remain continuous across foot switching, except for $\mathbf{p}_t^c$.

To approximate the nonlinear term in the jump dynamics (Eq. (4.6)) using a first-order

Taylor expansion, we have: $\mathbf{R}_t \mathbf{h}_c(\mathbf{q}_t) \approx \mathbf{R}_t \mathbf{h}_c(\tilde{\mathbf{q}}_t) - \mathbf{R}_t \frac{\partial \mathbf{h}_c}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_t)\mathbf{w}_t^q$.

4.1.4 Position based Forward Kinematics Measurement

To establish a connection between the contact and base frames, we employ the leg odometry measurement as described in [32, 29] (see Fig. 4-2-b):

$$\mathbf{R}_t^T(\mathbf{p}_t^c - \mathbf{p}_t^b) = \mathbf{h}_p(\mathbf{q}_t), \tag{4.7}$$

Here, the function $\mathbf{h}_p$ represents the support foot position relative to the base, expressed in the base frame. Given the inherent inaccuracy in the encoder reading $\tilde{\mathbf{q}}_t = \mathbf{q}_t + \mathbf{w}^q$, and utilizing a first-order Taylor expansion, we can rewrite the model in Eq. (4.7) as: $\mathbf{h}_p(\mathbf{q}_t) \approx \mathbf{h}_p(\tilde{\mathbf{q}}_t) - \frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_t)\mathbf{w}_t^q$.

4.1.5 Contact Orientation based Measurement

When the support foot and the surface have a full area contact, their normal vectors are parallel, whether the surface is stationary or moving (see Fig. 4-2-a)). In this study, we utilize this rotational kinematic relationship to form a measurement model. Suppose that the z-axes of the contact and surface frames are aligned and normal to the DRS. Then,

$$\mathbf{R}_t^{DRS}\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T = \mathbf{R}_t^c\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T = \mathbf{R}_t \mathbf{h}_R(\mathbf{q}_t)\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T \tag{4.8}$$

holds, where $\mathbf{R}_t^c \in SO(3)$ is the contact frame orientation and the forward kinematics matrix function $\mathbf{h}_R$ is the support foot orientation w.r.t. the base frame (see Fig.4-1). To address the inaccuracy of the known surface orientation $\tilde{\mathbf{R}}_t^{DRS}$, we assume the true orientation is corrupted by white Gaussian zero-mean uncertainty $\mathbf{w}_t^{DRS}$ as:

$$\mathbf{R}_t^{DRS} = \exp(-\mathbf{w}_t^{DRS})\tilde{\mathbf{R}}_t^{DRS} \approx (\mathbf{I}_3 - (\mathbf{w}_t^{DRS})_\times)\tilde{\mathbf{R}}_t^{DRS}, \tag{4.9}$$

where $\mathbf{I}_n$ is an $n \times n$ identity matrix.

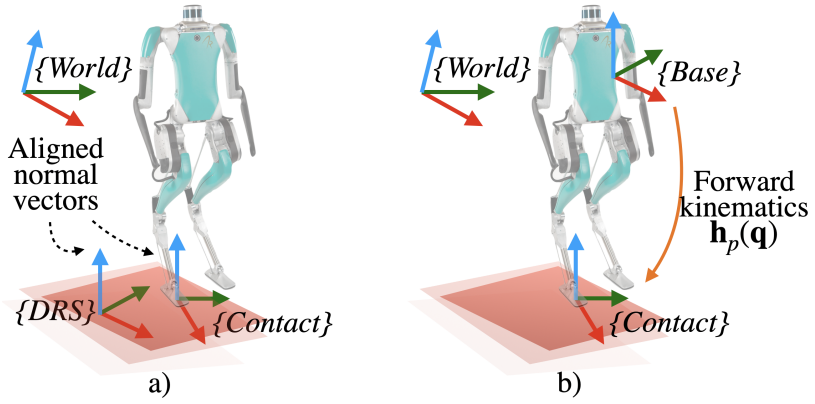

Figure 4-2: Illustrations of the observations: a) normal vector alignment of the contact and DRS frames and b) contact point position in the base frame.

To handle the inaccuracy of the encoder reading $\tilde{\mathbf{q}}_t$, the support foot orientation $\mathbf{R}_t\mathbf{h}_R(\mathbf{q}_t)$ is approximated as: $\mathbf{R}_t\mathbf{h}_R(\mathbf{q}_t) \approx \mathbf{R}_t\mathbf{h}_R(\tilde{\mathbf{q}}_t) - \mathbf{R}_t\mathbf{J}_{h_R}(\tilde{\mathbf{q}}_t, \mathbf{w}_t^q)$. Here the matrix $\mathbf{J}_{h_R}(\tilde{\mathbf{q}}_t, \mathbf{w}_t^q)$ is obtained based on the Jacobian of each column of $\mathbf{h}_R \triangleq [\mathbf{h}_{R,1},\ \mathbf{h}_{R,2},\ \mathbf{h}_{R,3}]$ as:

$$\mathbf{J}_{h_R} \triangleq \left[\frac{\partial \mathbf{h}_{R,1}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n})\mathbf{w}_t^q,\ \frac{\partial \mathbf{h}_{R,2}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_t)\mathbf{w}_t^q,\ \frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_t)\mathbf{w}_t^q\right]. \tag{4.10}$$

Combining these equations yields:

$$\begin{aligned}&\mathbf{R}_t^T\tilde{\mathbf{R}}_t^{DRS}\begin{bmatrix}0\ 0\ 1\end{bmatrix}^T + \mathbf{R}_t^T(-\mathbf{w}_t^{DRS})_\times\tilde{\mathbf{R}}_t^{DRS}\begin{bmatrix}0\ 0\ 1\end{bmatrix}^T \\ \approx &\mathbf{h}_R(\tilde{\mathbf{q}}_t)\begin{bmatrix}0\ 0\ 1\end{bmatrix}^T - \frac{\partial \mathbf{h}_{R,3}}{\partial \tilde{\mathbf{q}}_t}(\tilde{\mathbf{q}}_t)\mathbf{w}_t^q.\end{aligned} \tag{4.11}$$

4.2 Filter Design

This section introduces the proposed InEKF design based on the models formulated in Sec. 4.1.

The proposed filter derivation begins with proper state representation. We adopt the representation in [29] since our filters estimate the same state. First, the state variables $\mathbf{R}_t$, $\mathbf{v}_t$, $\mathbf{p}_t^b$, and $\mathbf{p}_t^c$ are expressed on the matrix Lie group $\mathcal{G}$ as:

$$\mathbf{X}_t \triangleq \begin{bmatrix}\mathbf{R}_t & [\mathbf{v}_t,\ \mathbf{p}_t^b,\ \mathbf{p}_t^c] \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3\end{bmatrix} \in \mathcal{G}, \tag{4.12}$$

where $\mathbf{0}_{m\times n}$ is an $m \times n$ zero matrix. The Lie group $\mathcal{G}$ is $SE_3(3)$, an extension of the special Euclidean group $SE(3)$.

To explicitly handle IMU biases, they are also chosen as state variables. These biases are typically expressed on the vector space instead of $\mathcal{G}$ [118]; that is, $\boldsymbol{\theta}_t \triangleq [(\mathbf{b}_t^{\omega})^T, (\mathbf{b}_t^{a})^T]^T$.

Let $\bar{(\cdot)}$ denote the estimate of the variable $(\cdot)$. Based on the InEKF framework [24], we use the right-invariant error $\boldsymbol{\eta}_t$ to represent the estimation error of $\mathbf{X}_t$ on $\mathcal{G}$:

$$\boldsymbol{\eta}_t \triangleq \bar{\mathbf{X}}_t \mathbf{X}_t^{-1} \in \mathcal{G}. \tag{4.13}$$

The log of the invariant error, denoted as $\boldsymbol{\xi}_t$, is a vector on $\mathbb{R}^{\dim \mathfrak{g}}$ defined via $\boldsymbol{\eta}_t \triangleq \exp(\boldsymbol{\xi}_t^{\wedge})$.

The individual elements of $\boldsymbol{\eta}_t$, $\boldsymbol{\xi}_t$, and $\boldsymbol{\xi}_t^{\wedge}$ are given in Sec .B.2 of the Appendix (see Eq. (B.3)).

The IMU bias estimation error $\boldsymbol{\zeta}_t$ is defined as: $\boldsymbol{\zeta}_t \triangleq \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}$.

4.2.1 Continuous-Phase Process Model and Propagation Step

This subsection introduces the process model and propagation step of the proposed filter for the continuous phases.

4.2.1.1 Process model

Based on the IMU motion and bias dynamics and the contact point motion in Eqs. (4.1)-(4.3), the process model is expressed as:

$$\begin{aligned}\dot{\mathbf{X}}_t &= \begin{bmatrix} \mathbf{R}_t(\tilde{\boldsymbol{\omega}}_t - \mathbf{b}_t^{\omega})_{\times} & [\mathbf{R}_t(\tilde{\mathbf{a}}_t - \mathbf{b}_t^{a}) + \mathbf{g},\ \mathbf{v}_t,\ \tilde{\mathbf{v}}_t^{c}] \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix} - \mathbf{X}_t(\mathbf{w}_t^X)^{\wedge} \\ &\triangleq \mathbf{f}_{u_t}(\mathbf{X}_t, \boldsymbol{\theta}_t) - \mathbf{X}_t(\mathbf{w}_t^X)^{\wedge},\end{aligned} \tag{4.14}$$

with the noise vector $\mathbf{w}_t^X \triangleq \left[(\mathbf{w}_t^{\omega})^T, (\mathbf{w}_t^{a})^T, \mathbf{0}_{1\times 3}, (\mathbf{w}_t^{c})^T\right]^T$. Here we define the input $\mathbf{u}_t$ to consist of the IMU and encoder readings and the measured contact point velocity, i.e., $\mathbf{u}_t = \left[\tilde{\boldsymbol{\omega}}_t^T,\ \tilde{\mathbf{a}}_t^T,\ (\tilde{\mathbf{v}}_t^{c})^T,\ \tilde{\mathbf{q}}_t\right]^T$. Note that the encoder reading $\tilde{\mathbf{q}}_t$ is not an input to the continuous-phase process model in Eq. (4.14) but is used later in the jump process model.

4.2.1.2 Linearized error model

By using the first-order Taylor expansion $\boldsymbol{\eta}_t = \text{expm}(\boldsymbol{\xi}_t^{\wedge}) \approx \mathbf{I} + \boldsymbol{\xi}_t^{\wedge}$ and by applying the chain rule to express $\dot{\boldsymbol{\eta}}_t$, we obtain the linearized error equation:

$$\begin{bmatrix} \dot{\boldsymbol{\xi}}_t \\ \dot{\boldsymbol{\zeta}}_t \end{bmatrix} = \mathbf{A}_t \begin{bmatrix} \boldsymbol{\xi}_t \\ \boldsymbol{\zeta}_t \end{bmatrix} + \begin{bmatrix} \mathbf{Ad}_{\bar{\mathbf{X}}_t} & \mathbf{0}_{12\times 6} \\ \mathbf{0}_{6\times 12} & \mathbf{I}_6 \end{bmatrix} \mathbf{w}_t. \tag{4.15}$$

Here, the noise term $\mathbf{w}_t$ is defined as $\mathbf{w}_t \triangleq [(\mathbf{w}_t^X)^T, (\mathbf{w}_t^{b\omega})^T, (\mathbf{w}_t^{ba})^T]^T$, the adjoint matrix $\mathbf{Ad}_{\bar{\mathbf{X}}_t}$ is given in Section B.2 of the Appendix (see Eq. (B.5)), and the matrix $\mathbf{A}_t$ is:

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\bar{\mathbf{R}}_t & \mathbf{0}_{3\times 3} \\ (\mathbf{g})_\times & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -(\bar{\mathbf{v}}_t)_\times \bar{\mathbf{R}}_t & -\bar{\mathbf{R}}_t \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -(\bar{\mathbf{p}}_t)_\times \bar{\mathbf{R}}_t & \mathbf{0}_{3\times 3} \\ (\tilde{\mathbf{v}}_t^c)_\times & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -(\bar{\mathbf{p}}_t^c)_\times \bar{\mathbf{R}}_t & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} \end{bmatrix}. \tag{4.16}$$

Note that $\mathbf{A}_t$ contains the contact point velocity $\tilde{\mathbf{v}}_t^c$ because the process model explicitly considers it. Derivation of Eqs. (4.15) and (4.16) is given in B.2 of the Appendix.

4.2.1.3 Propagation

Let t_n ($n \in \{1, 2, ...\}$) denote the time when sensors return data for the filter to correct the estimates. Then, during the propagation step on $t \in [t_{n-1}, t_n)$, the estimates $\bar{\mathbf{X}}_t$ and $\bar{\boldsymbol{\theta}}_t$ are obtained via $\dot{\bar{\mathbf{X}}}_t = \mathbf{f}_{u_t}(\bar{\mathbf{X}}_t, \bar{\boldsymbol{\theta}}_t)$ and $\dot{\bar{\boldsymbol{\theta}}}_t = \mathbf{0}$ based on the process models in Eqs. (4.14) and (4.2).

By the InEKF methodology, the covariance matrix $\mathbf{P}_t$ is propagated via the Riccati equation associated with the linearized error model in Eq. (4.15): $\dot{\mathbf{P}}_t = \mathbf{A}_t\mathbf{P}_t + \mathbf{P}\mathbf{A}_t^T + \bar{\mathbf{Q}}_t$, where $\bar{\mathbf{Q}}_t \triangleq \begin{bmatrix} \mathbf{Ad}_{\bar{\mathbf{X}}_t} & \mathbf{0}_{12\times 6} \\ \mathbf{0}_{6\times 12} & \mathbf{I}_6 \end{bmatrix} \text{Cov}(\mathbf{w}_t) \begin{bmatrix} \mathbf{Ad}_{\bar{\mathbf{X}}_t} & \mathbf{0}_{12\times 6} \\ \mathbf{0}_{6\times 12} & \mathbf{I}_6 \end{bmatrix}^T$.

Remark 1 **(Group affine property):** Without IMU biases, the continuous process model in Eq. (4.14) is group affine as defined in [24]. Thus, without biases and in the deterministic case, the linear error dynamics in Eq. (4.15) is exact and independent of the true state, and

the covariance propagation is exact. Such features are different from the standard EKF whose linearization accuracy relies on the estimation error.

4.2.2 Continuous-Phase Measurement Models and Update Step

This subsection formulates the two measurements in Eqs. (4.7) and (4.8) into the right-invariant observation form defined in [24] and introduces the update step of the proposed InEKF at time t_n. These treatments result in an error update equation that is independent of the true state.

4.2.2.1 Right-invariant orientation based measurement

The orientation based measurement in Eq. (4.11) can be transformed into the following right-invariant observation form:

$$\underbrace{\begin{bmatrix} \mathbf{h}_R(\tilde{\mathbf{q}})_{t_n} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \mathbf{0}_{3\times 1} \end{bmatrix}}_{\mathbf{Y}_{1,t_n}} = \mathbf{X}_{t_n}^{-1} \underbrace{\begin{bmatrix} \tilde{\mathbf{R}}_{t_n}^{DRS} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \mathbf{0}_{3\times 1} \end{bmatrix}}_{\mathbf{d}_{1,t_n}} + \begin{bmatrix} \mathbf{V}_{1,t_n} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.17}$$

with $\mathbf{V}_{1,t_n} = \mathbf{R}_{t_n}^T(\tilde{\mathbf{R}}_{t_n}^{DRS} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T)_\times \mathbf{w}_{t_n}^{DRS} + \frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n})\mathbf{w}_{t_n}^q$.

4.2.2.2 Right-invariant position measurement

The position measurement in Eq. (4.7) has the right-invariant form [29]:

$$\underbrace{\begin{bmatrix} \mathbf{h}_p(\tilde{\mathbf{q}}_{t_n}) \\ 0 \\ 1 \\ -1 \end{bmatrix}}_{\mathbf{Y}_{2,t_n}} = \mathbf{X}_{t_n}^{-1} \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 1} \\ 0 \\ 1 \\ -1 \end{bmatrix}}_{\mathbf{d}_{2,t_n}} + \begin{bmatrix} \frac{\partial \mathbf{h}_p}{\partial \tilde{\mathbf{q}}}(\tilde{\mathbf{q}}_{t_n})\mathbf{w}_{t_n}^q \\ \mathbf{0}_{3\times 1} \end{bmatrix}. \tag{4.18}$$

4.2.2.3 Update

At time t_n, the updated estimates and covariance, denoted as $(\bar{\mathbf{X}}^{\dagger}_{t_n}, \bar{\boldsymbol{\theta}}^{\dagger}_{t_n})$ and $\mathbf{P}^{\dagger}_{t_n}$, are given by [24]:

$$\bar{\mathbf{X}}^{\dagger}_{t_n} = \exp\left(\mathbf{L}^{\xi}_{t_n}\mathbf{z}_{t_n}\right)\bar{\mathbf{X}}_{t_n},\ \bar{\boldsymbol{\theta}}^{\dagger}_{t_n} = \bar{\boldsymbol{\theta}}_{t_n} + \mathbf{L}^{\zeta}_{t_n}\mathbf{z}_{t_n},\ \mathbf{P}^{\dagger}_{t_n} = (\mathbf{I} - \mathbf{L}_{t_n}\mathbf{H}_{t_n})\mathbf{P}_{t_n}, \tag{4.19}$$

where $\mathbf{L}_{t_n} \triangleq \left[(\mathbf{L}^{\xi}_{t_n})^T, (\mathbf{L}^{\zeta}_{t_n})^T\right]^T$ is filter gain, $\mathbf{H}_{t_n}$ is the observation matrix, and

$$\mathbf{z}_{t_n} \triangleq \left[(\bar{\mathbf{X}}_{t_n}\mathbf{Y}_{1,t_n} - \mathbf{d}_{1,t_n})^T,\ (\bar{\mathbf{X}}_{t_n}\mathbf{Y}_{2,t_n} - \mathbf{d}_{2,t_n})^T\right]^T.$$

To derive the observation matrix $\mathbf{H}_{t_n}$, we first decompose it into $\mathbf{H}_{t_n} = \left[\mathbf{H}^T_{1,t_n},\ \mathbf{H}^T_{2,t_n}\right]^T$, where $\mathbf{H}_{1,t_n} \in \mathbb{R}^{6\times 12}$ and $\mathbf{H}_{2,t_n} \in \mathbb{R}^{6\times 12}$ are respectively associated with the measurement models in (4.17) and (4.18). Since the measurement models are not explicitly dependent on biases, the matrix $\mathbf{H}_{i,t_n}$ $(i = 1,2)$ can be further decomposed as $\mathbf{H}_{i,t_n} \triangleq \left[\tilde{\mathbf{H}}_{i,t_n},\ \mathbf{0}_{3\times 6}\ ;\mathbf{0}_{3\times 12},\ \mathbf{0}_{3\times 6}\right]$, where the element $\mathbf{0}_{3\times 6}$ correspond to the bias terms and the element $\mathbf{0}_{3\times 12}$ could be removed if a reduced-dimensional filter gain is instead used as in [29]. Based on the right-InEKF methodology [24], we obtain the submatrix $\tilde{\mathbf{H}}_{i,t_n}$ via $\tilde{\mathbf{H}}_{i,t_n}\boldsymbol{\xi}_{t_n} = -(\boldsymbol{\xi}_{t_n})^{\wedge}\mathbf{d}_{i,t_n}$: $\tilde{\mathbf{H}}_{1,t_n} \triangleq [(\mathbf{R}^{DRS}_{t_n}\,[0,0,1]^T)_{\times},\ \mathbf{0}_{3\times 9}]$ and $\tilde{\mathbf{H}}_{2,t_n} \triangleq [\mathbf{0}_{3\times 6},\ -\mathbf{I}_3,\ \mathbf{I}_3]$.

To compute $\mathbf{L}_{t_n}$, the linearized error update equation is obtained based on the update equation (Eq. (4.19)) as:

$$\begin{bmatrix}\boldsymbol{\xi}^{\dagger}_{t_n}\\ \boldsymbol{\zeta}^{\dagger}_{t_n}\end{bmatrix} = (\mathbf{I} - \mathbf{L}_{t_n}\mathbf{H}_{t_n})\begin{bmatrix}\boldsymbol{\xi}_{t_n}\\ \boldsymbol{\zeta}_{t_n}\end{bmatrix} + \mathbf{L}_{t_n}\begin{bmatrix}\bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n})\\ \mathbf{0}_{3\times 1}\\ \bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{p}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n})\\ \mathbf{0}_{3\times 1}\end{bmatrix}\mathbf{w}^q_{t_n}, \tag{4.20}$$

with derivation details given in B.3 of the Appendix. Then, applying the standard Kalman filtering methodology to this linear error update equation, we obtain the filter gain: $\mathbf{L}_{t_n} = \mathbf{P}_{t_n}\mathbf{H}^T_{t_n}\mathbf{S}^{-1}_{t_n}$, where $\mathbf{S}_{t_n} = \mathbf{H}_{t_n}\mathbf{P}_{t_n}\mathbf{H}^T_{t_n} + \mathbf{N}_{t_n}$, $\bar{\mathbf{N}}_{t_n} \triangleq \mathrm{diag}(\bar{\mathbf{N}}_{1,t_n}, \bar{\mathbf{N}}_{2,t_n})$, $\bar{\mathbf{N}}_{1,t_n} \triangleq \bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n})\mathrm{Cov}(\mathbf{w}^q_{t_n})(\frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n}))^T\bar{\mathbf{R}}^T_{t_n}$, and $\bar{\mathbf{N}}_{2,t_n} \triangleq \bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{p}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n})\mathrm{Cov}(\mathbf{w}^q_{t_n})(\frac{\partial \mathbf{h}_{p}}{\partial \mathbf{q}_t}(\tilde{\mathbf{q}}_{t_n}))^T\bar{\mathbf{R}}^T_{t_n}$.

Remark 2 **(Independence of true state):** The linearized error update equation (Eq. (4.20)) is independent of the true state $\mathbf{X}_t$ and $\boldsymbol{\theta}_t$ in the deterministic case. This is because both measurement models satisfy the right-invariant observation form w.r.t. $\mathbf{X}_t$ and are independent of $\boldsymbol{\theta}_t$ and because the update equation of $\bar{\mathbf{X}}_t$ is in the exponential form as prescribed by the InEKF methodology [24].

4.2.3 Discrete Process Model and Propagation Step

Without loss of generality and for simplicity, suppose that the foot-landing events and the updates do not coincide. Thus, the proposed filtering for the state jump focuses on estimate and covariance propagation without update. Except for the true contact point position $\mathbf{p}_t^c$, the rest of the true state is continuous across a foot landing, as explained in Sec. 4.1.

4.2.3.1 Process model

From the proposed jump dynamics in Sec. 4.1, the stochastic jump dynamics of $\mathbf{X}_t$ can be approximately expressed as:

$$\begin{aligned}\mathbf{X}_{t^+} &= \mathbf{X}_t \begin{bmatrix} \mathbf{I}_3 & [\mathbf{0}_{3\times1},\ \mathbf{0}_{3\times1},\ \mathbf{h}_c(\tilde{\mathbf{q}}_t)] \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} - \mathbf{X}_t \begin{bmatrix} \frac{\partial \mathbf{h}_c}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_t)\mathbf{w}_t^q \\ \mathbf{0}_{3\times1} \end{bmatrix} \\ &\triangleq \boldsymbol{\Delta}_{u_t}(\mathbf{X}_t) - \mathbf{X}_t \mathbf{w}_t^{\Delta},\end{aligned} \tag{4.21}$$

where the encoder data $\tilde{\mathbf{q}}_t$ serves as the input. As the biases are continuous under a jump event, $\boldsymbol{\theta}_{t^+} = \boldsymbol{\theta}_t$ holds.

Remark 3 **(Group affine property):** The jump map $\boldsymbol{\Delta}_{u_t}$ of the state $\mathbf{X}_t$ possesses the discrete-time group affine property defined in [119], and is independent of IMU biases $\boldsymbol{\theta}_t$. Thus, the jump dynamics of the error $\boldsymbol{\xi}_t$ is independent of the true state and is exactly linear. Moreover, from the expression of $\boldsymbol{\Delta}_{u_t}$ in Eq. (4.21), we can see that $\boldsymbol{\Delta}_{u_t}$ is a group action on $SE_3(3)$, under which the error $\boldsymbol{\xi}_t$ naturally does not change.

4.2.3.2 Error equation

From Eq. (4.21), we obtain the dynamics of the logarithmic error $\boldsymbol{\xi}_t$ as: $\boldsymbol{\xi}_{t^+} = \boldsymbol{\xi}_t - \mathbf{Ad}_{\bar{\mathbf{X}}}\mathbf{w}_t^{\Delta}$. Indeed, as analyzed in Remark 3, the error does not jump under $\boldsymbol{\Delta}_{u_t}$. Also, $\boldsymbol{\zeta}_{t^+} = \boldsymbol{\zeta}_{t^-}$ holds since the IMU biases are continuous.

4.2.3.3 Propagation

Based on the deterministic portion of the jump model in Eq. (4.21), the propagation of the state estimate at a jump event is: $\bar{\mathbf{X}}_{t^+} = \boldsymbol{\Delta}_{u_t}(\bar{\mathbf{X}}_t)$ and $\bar{\boldsymbol{\theta}}_{t^+} = \bar{\boldsymbol{\theta}}_t$. With the linear error equation of $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ across a jump, the propagation of the covariance matrix is expressed as: $\mathbf{P}_{t^+} = \mathbf{P}_t + \bar{\mathbf{Q}}_t^{\Delta}$, where $\bar{\mathbf{Q}}_t^{\Delta} = \begin{bmatrix} \mathbf{Ad}_{\bar{\mathbf{X}}}\mathrm{Cov}(\mathbf{w}_t^{\Delta})\mathbf{Ad}_{\bar{\mathbf{X}}}^T & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}$.

The complete algorithm of the proposed right-InEKF is summarized as Algorithm 1:

Algorithm 1 Proposed Right InEKF for Hybrid Models of DRS Locomotion

Initialize: i) $\bar{\mathbf{X}}_{t_0} \in SE3(3)$; ii) $\mathbf{P}_{t_0}$ is symmetric, positive-definite.
while *True* **do**
  if *a foot landing (i.e., a jump) is detected* **then**
    Propagation at a jump
    $\bar{\mathbf{X}}_{t^+} = \boldsymbol{\Delta}_{u_t}(\bar{\mathbf{X}}_t)$, $\bar{\boldsymbol{\theta}}_{t^+} = \bar{\boldsymbol{\theta}}_t$, $\mathbf{P}_{t^+} = \mathbf{P}_t + \bar{\mathbf{Q}}_t^{\Delta}$
  else
    Propagation for continuous phases
    $\dot{\bar{\mathbf{X}}}_t = \mathbf{f}_{u_t}(\bar{\mathbf{X}}_t, \bar{\boldsymbol{\theta}}_t)$, $\dot{\bar{\boldsymbol{\theta}}}_t = \mathbf{0}$, $\dot{\mathbf{P}}_t = \mathbf{A}_t\mathbf{P}_t + \mathbf{P}_t\mathbf{A}_t^T + \bar{\mathbf{Q}}_t$
    Update for continuous phases
    $\mathbf{S}_{t_n} = \mathbf{H}_{t_n}\mathbf{P}_{t_n}\mathbf{H}_{t_n}^T + \bar{\mathbf{N}}_{t_n}$
    $\mathbf{z}_{tn} = \left[(\bar{\mathbf{X}}_{t_n}\mathbf{Y}_{1,t_n} - \mathbf{d}_{1,t_n})^T,\ (\bar{\mathbf{X}}_{t_n}\mathbf{Y}_{2,t_n} - \mathbf{d}_{2,t_n})^T\right]^T$
    $\mathbf{L}_{t_n} = \left[(\mathbf{L}_{t_n}^{\xi})^T,\ (\mathbf{L}_{t_n}^{\zeta})^T\right]^T = \mathbf{P}_{t_n}\mathbf{H}_{t_n}^T\mathbf{S}_{t_n}^{-1}$
    $\mathbf{P}_{t_n}^{\dagger} = (\mathbf{I} - \mathbf{L}_{t_n}\mathbf{H}_{t_n})\mathbf{P}_{t_n}$
    $\bar{\mathbf{X}}_{t_n}^{\dagger} = \exp\left(\mathbf{L}_{t_n}^{\xi}\mathbf{z}_{t_n}\right)\bar{\mathbf{X}}_{t_n}$
    $\bar{\boldsymbol{\theta}}_{t_n}^{\dagger} = \bar{\boldsymbol{\theta}}_{t_n} + \mathbf{L}_{t_n}^{\zeta}\mathbf{z}_{t_n}$
  end
end

Remark 4 **(Imperfect InEKF):** In the presence of IMU biases, the proposed filter is no longer a "perfect" InEKF in the sense that the group affine and invariant form properties no longer hold for continuous phases. Although the linear equation in Eq. (4.15) is no longer

independent of the true state, it depends on the true state only through the bias terms while the remaining part of the Jacobian matrix $\mathbf{A}_t$ is still independent of the true state. Also, the measurement models are still independent of the true state $\mathbf{X}_t$ and $\boldsymbol{\theta}_t$ as highlighted in Remark 2. For these reasons, the linearization inaccuracy induced by the biases has a limited impact on the continuous-phase propagation and update. Thus, the "imperfect InEKF" with biases considered can still ensure rapid and accurate convergence under large errors, which is experimentally confirmed on DRS locomotion as reported in Sec. 4.4.

4.3 Observability and Convergence Analysis

4.3.1 Observability Analysis for Continuous Phases

As measurement update is performed during continuous phases, we only analyze the continuous-phase observability.

Recall that the deterministic continuous-phase dynamics in Eq. (4.14) is group affine in the absence of IMU biases $\boldsymbol{\theta}_t$ (Remark 1). Also, recall that the measurement models in Eqs. (4.17) and (4.18) are in the right-invariant observation form w.r.t. $\mathbf{X}_t$, regardless of the presence of biases (Remark 2). Then, by Theorem 20 in [118], the observability of $\mathbf{X}_t$ for the complete continuous-phase system, which has both $\mathbf{X}_t$ and $\boldsymbol{\theta}_t$ as its state, is the same as that of the simplified continuous-phase system without IMU biases. Thus, by Theorem 5 in [24], the local observability of $\mathbf{X}_t$ for the complete system can be determined by the couple ($\mathbf{A}$, $\mathbf{H}$), with $\mathbf{A}$ and $\mathbf{H}$ updated with bias-related terms removed. Their updated expressions are in B.4 of appendix.

With Δt the duration of one propagation step, the discrete state transition matrix $\boldsymbol{\Phi}$ is given by $\boldsymbol{\Phi} = \mathrm{expm}(\mathbf{A}_t \Delta t)$ [24] (see the Eq.(B.18) for detailed expression). Then, from the expression of the observability matrix, which is $\mathfrak{O} = \begin{bmatrix} (\mathbf{H})^T, & (\mathbf{H}\boldsymbol{\Phi})^T, & (\mathbf{H}\boldsymbol{\Phi}^2)^T, & \dots \end{bmatrix}^T$, we

have:

$$\mathfrak{O} = \begin{bmatrix} (\mathbf{R}^{DRS}\,[0,0,1]^T)_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{I}_3 \\ (\mathbf{R}^{DRS}\,[0,0,1]^T)_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\frac{1}{2}(\mathbf{g})_\times\Delta t^2 & -\mathbf{I}_3\Delta t & -\mathbf{I}_3 & \mathbf{I}_3 \\ (\mathbf{R}^{DRS}\,[0,0,1]^T)_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -2(\mathbf{g})_\times\Delta t^2 & -2\mathbf{I}_3\Delta t^2 & -\mathbf{I}_3 & \mathbf{I}_3 \\ \dots & \dots & \dots & \dots \end{bmatrix}. \tag{4.22}$$

As the first two columns of $(\mathbf{g})_\times$ are linearly independent, the base roll and pitch angles are observable. Because all columns in the second column block of $\mathfrak{O}$ are linear independent, the base velocity $\mathbf{v}_t$ is observable. Yet, as the last two column blocks are linearly dependant, the base position $\mathbf{p}_t^b$ and contact point position $\mathbf{p}_t^c$ are unobservable.

The third column of $(\mathbf{g})_\times$ is always zero because only its z-component is nonzero. Then, if the surface is non-horizontal, (i.e., the third column of $(\mathbf{R}^{DRS}\,[0,0,1]^T)_\times$ is not all zero), the yaw will be observable; otherwise, it is unobservable.

From the expression of $\mathfrak{O}$, we also know that: a) the contact velocity $\mathbf{v}_t^c$ does not affect observability; b) either measurement model ensures observable base roll and pitch; c) the proposed measurement in Eq. (4.17) renders base yaw observable when the ground is non-horizontal; and d) the previous measurement in Eq. (4.18) makes base velocity observable.

4.3.2 Convergence Property for Hybrid Error System

The proposed convergence analysis for the hybrid error system is built upon previous analysis of the InEKF as a deterministic observer for systems without state-triggered jumps [24]. Different from the previous work, this subsection analyzes the effects of the jumps on the error convergence for the overall hybrid error system.

We first analyze the error evolution across the deterministic discrete jump of the system. Analyzing the state evolution across discrete, state-triggered jumps (e.g., foot-landing impacts) is typically complex [6].

Yet, since the jump map $\mathbf{\Delta}_{u_t}$ is a group action, the error $\boldsymbol{\xi}_t$ does not jump under $\mathbf{\Delta}_{u_t}$

despite the jump of the true state $\mathbf{X}_t$. Also, the bias error $\boldsymbol{\zeta}_t$ is continuous across a jump event. Thus, the hybrid, deterministic error dynamics is essentially continuous for all time, and its error convergence is equivalent to that of the deterministic continuous phases.

For continuous phases, the proposed filter meets the group affine condition and invariant observation form without biases, as discussed in Sec. 4.2. Thus, by the theory of InEKF [24], the proposed filter is locally asymptotically convergent for the observable variables of the deterministic continuous phases without biases. Accordingly, the local asymptotic convergence of the hybrid, deterministic filter system is guaranteed in the absence of biases.

This analysis also supports the local asymptotic convergence of the existing InEKF [29] designed for static surface locomotion, because the jump model in [29] is a group action and its continuous-phase design also satisfies the group affine and invariant observation conditions without biases.

4.4 Experiments

4.4.1 Experimental setup

The setup for experimental data collection (Fig. 4-3) is:

Treadmill (i.e, the tested DRS). A split-belt Motek M-gait treadmill is used as a DRS. Its dimension is 2.3 m$\times$1.82 m$\times$0.5 m. To emulate a rocking ship in sea waves, it performs a whole-body pitching motion without belt translation.

Robot. The Digit bipedal robot is 1.6 m tall, and each leg's kinematic chain used by the filter has 12 joints. To validate the filter, different robot movements are tested: **(RM1)** stepping and **(RM2)** standing. The robot is about 0.8 m behind the treadmill center.

Treadmill motion profiles. To test filter performance under different relatively significant DRS motions, two different profiles of the treadmill's pitch angle θ^{DRS} (Fig. 4-4) are tested: **(TM1)** a non-periodic trapezoidal wave, $f_{trap}(t)$, and **(TM2)** a sine wave $2.5^\circ \sin(\pi t)$. Under (TM1) and (TM2), the maximum contact point speeds $\|\mathbf{v}_t^c\|$ are respectively 0.41 m/s and 0.11 m/s. To test the filter's robustness under surface motion inaccuracy, a fictitious profile is considered: **(TM3)** $\theta^{DRS}(t) = f_{trap}(t) + 1.7^\circ \sin(\pi t)$, with the actual profile (TM1) used

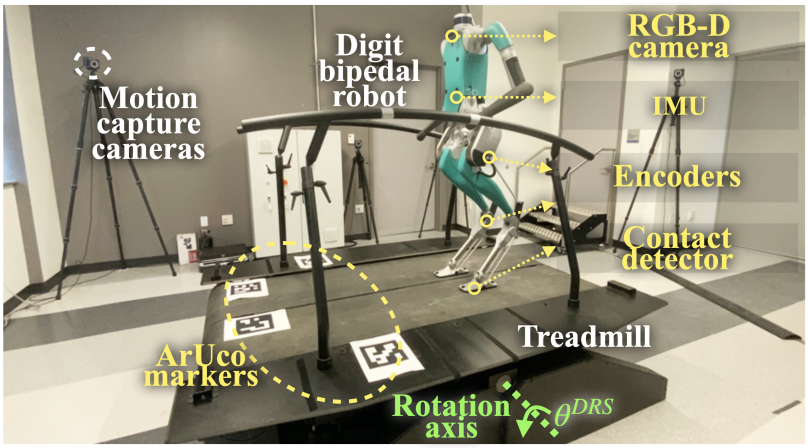

Figure 4-3: Experimental setup that includes a Digit bipedal humanoid robot and a pitching treadmill (i.e., DRS).

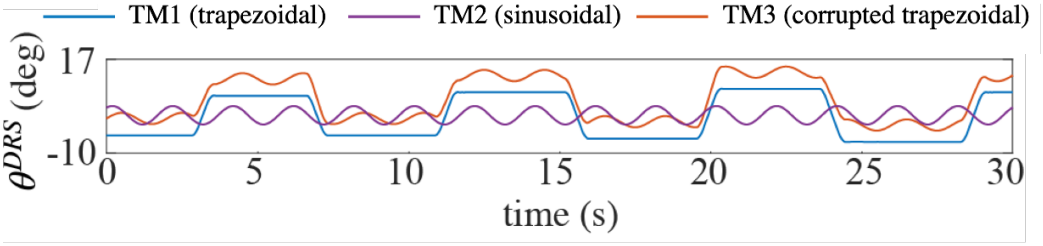

Figure 4-4: Profiles (TM1)-(TM3) of the treadmill pitch angle $\theta^{DRS}(t)$.

in experiments. The inaccuracy $1.7^\circ \sin(\pi t)$ is about 20% of the true profile in magnitude. Figure 4-4 shows the three profiles. The motion data streams at a rate of 60-90 Hz.

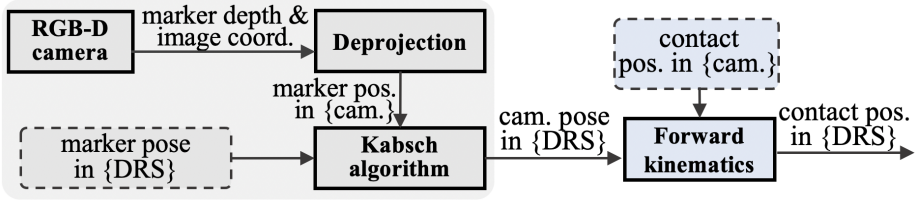

Figure 4-5: Procedure of obtaining the 3-D contact point position in the DRS frame using the ArUco markers and the robot's on-board RGB-D camera.

On-board sensors used. Digit's on-board sensors used (Fig. 4-3) are: an IMU, joint encoders, a RealSense RGB-D camera, and the robot's proprietary contact detector. The camera returns data at 15 Hz, and the remaining sensors stream data at the same rate within 60-90 Hz. Cortex motion capture cameras provide the ground truth. ArUco markers are attached to the treadmill, emulating the real-world scenario where legged robots that navigate

within a DRS (e.g., a vessel at sea) can only see landmarks attached to the DRS but not any landmarks on the earth's ground. The markers are sensed by the camera to obtain the camera pose in the treadmill frame, which is then used to compute contact point velocity as explained later.

Data collection cases. The proposed filter is simulated in MATLAB using four experimentally collected data sets under different robot and treadmill motions: Case A: Combination of (RM1) and (TM1); Case B: Combination of (RM1) and (TM2); Case C: Combination of (RM2) and (TM1); and Case D: Combination of (RM1) and (TM3), where the actual profile is (TM1) but the filter uses the inaccurate data (TM3).

4.4.2 Filter Setting

Filters compared. The proposed filter (denoted as "InEKF-DRS") is compared with an InEKF designed for locomotion on a static rigid surface [29] (denoted as "InEKF-SRS"). The InEKF-SRS models the deterministic contact point motion as $\dot{\mathbf{p}}^c = \mathbf{0}$, and uses the position measurement in Eq. (4.18) alone. It renders the base orientation (except for yaw) and velocity observable. It has realized substantially faster convergence under large errors during stationary surface locomotion, as compared with EKF-based method [32].

Contact point velocity computation. The contact point velocity $\tilde{\mathbf{v}}^c$ serves as an input to the continuous-phase process model of the proposed InEKF-DRS. To obtain the contact point velocity $\tilde{\mathbf{v}}^c$, as summarized in Fig. 4-5, we first obtain the camera pose in the DRS frame by processing the features of the ArUco markers in the camera images, which we then use to compute the 3-D contact point position in the DRS frame ($^{DRS}\mathbf{p}^c$) through forward kinematics. Next, we estimate the contact point velocity $\tilde{\mathbf{v}}^c$ based on Eq. (4.4) using the known treadmill motion data. Details of the procedure are given in Section B.5 of appendix. Results in Fig. 4-6 validate the accuracy of the proposed contact point velocity sensing.

Covariance settings. Table 4.1 shows the noise standard deviation (SD) of both filters. The SD for the accelerometer, gyroscope, and their corresponding biases are obtained from the manufacturer's manual with a slight adjustment for better performance. The SD for the joint encoder readings is adopted from the previous filter [29] designed for a similar robot. The SD for the contact-point velocity and orientation-based measurement are tuned for a

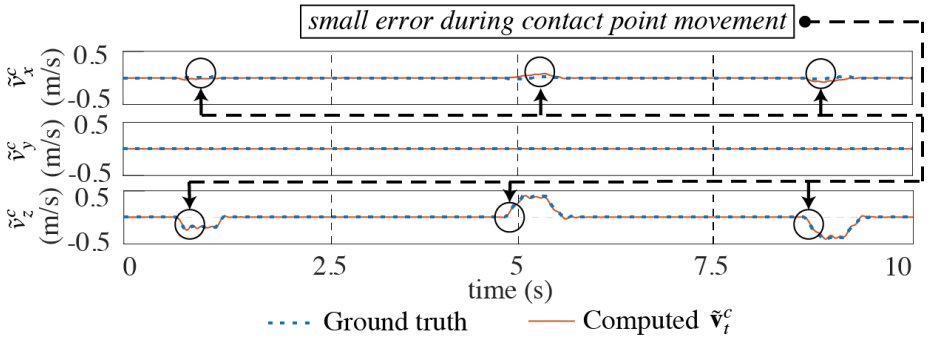

Figure 4-6: Validation results of the proposed method for obtaining the contact point velocity $\tilde{\mathbf{v}}_t^c \triangleq [\tilde{v}_x^c, \tilde{v}_y^c, \tilde{v}_z^c]^T$ under Case C. The velocity along the y-direction, $\tilde{v}_y^c$, is zero because the treadmill does not move in that direction.

Table 4.1: Noise standard deviation for inekf-srs and inekf-drs.

Measurement type	InEKF-SRS	InEKF-DRS (proposed)
Linear acceleration (m/s^2)	0.4	0.4
Angular velocity (rad/s)	0.01	0.01
Accelerometer bias (m/s^3)	0.001	0.001
Gyroscope bias (rad/s^2)	0.0001	0.0001
Contact velocity (m/s)	0.01	0.01
Encoder ($^\circ$)	1	1
DRS orientation ($^\circ$)	N/A	1

reasonable performance.

Initial estimation errors. For a fair comparison, the two filters are simulated under the same large range of initial estimation errors. The initial velocity and orientation errors in each direction are respectively uniformly distributed within $[-1.5,1.5]$ m/s and $[-1,1]$ rad.

4.4.3 Computational Time Comparison

In MATLAB, both filters take less than 1 ms to compute one estimation cycle (i.e., one propagation and one update step), confirming their validity for real-time estimation.

4.4.4 Convergence Rate and Yaw Observability Comparison

Figure 4-7 displays the estimation results of InEKF-DRS (proposed) and InEKF-SRS under Case A where the treadmill stays at a pitch angle of -8° for approximately 2.8 sec and then begins to pitch until reaching $+8^\circ$ in 0.5 sec.

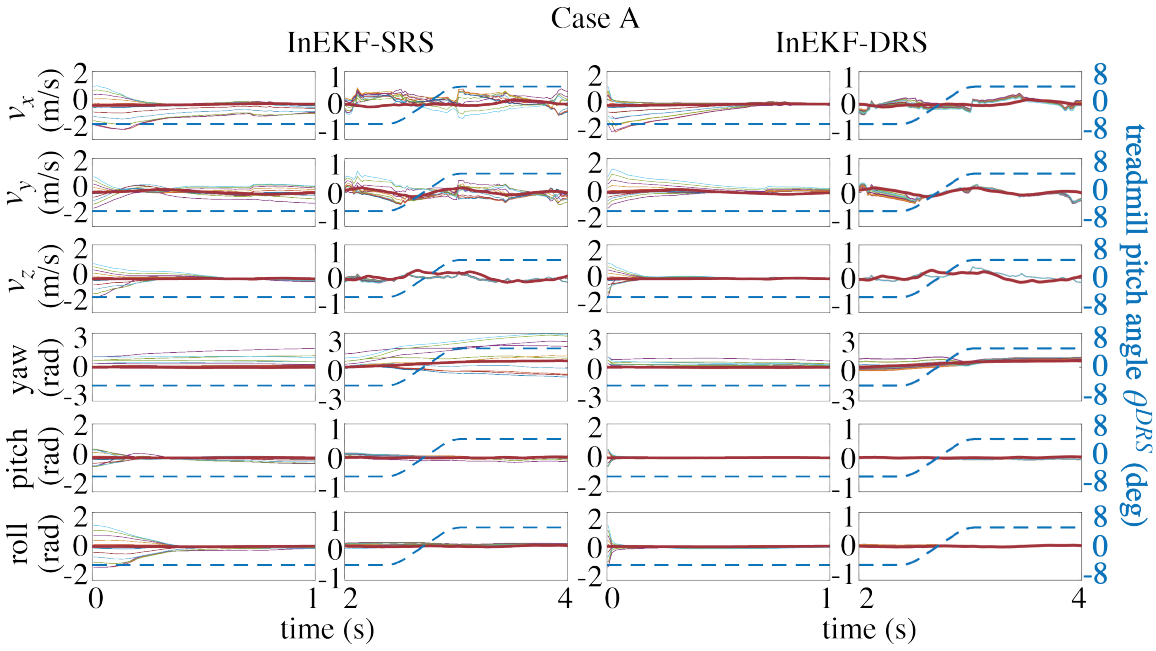

Figure 4-7: Base velocity and orientation estimation results of the two filters, InEKF-DRS (proposed) and InEKF-SRS, for Cases A. The thin solid and thick solid lines are the estimates and ground truth of the base velocity and orientation. The blue-dashed lines are the treadmill orientation profile.

Both filters drive the error of base roll, pitch, and velocity closer to zero, indicating their observability as predicted in Sec. 4.3 and previous work [32, 29]. In terms of the convergence rates for these variables, subplot a) shows that the proposed InEKF-DRS is faster than InEKF-SRS, driving the error close to zero within 1 sec. This is because InEKF-DRS considers the surface motion and has an additional measurement (Eq. (4.17)) that corrects estimates.

Under InEKF-DRS, the yaw estimate converges close to the ground truth in approximately 3 sec, which supports the observability analysis in Sec. 4.3 that the yaw angle is observable if the DRS/treadmill is not horizontal. Yet, the yaw convergence is slower than pitch and roll, possibly because both observations in Eqs. (4.17) and (4.18) help correct the roll and pitch estimates whereas only the former corrects the yaw estimate. Finally, as previously revealed [32], the yaw error divergence under InEKF-SRS confirms that the base yaw is indeed non-observable with InEKF-SRS.

Table 4.2 shows the comparison of the root-mean-square (RMS) estimation errors

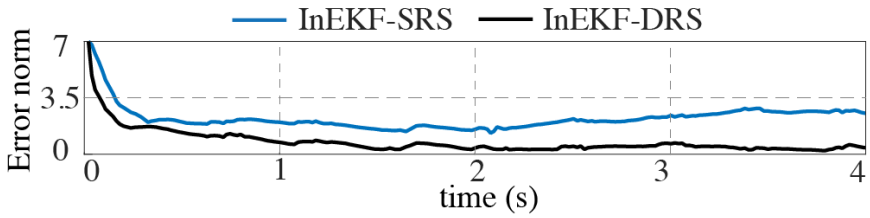

Figure 4-8: Accuracy comparison of InEKF-SRS and InEKF-DRS (proposed) for the estimation of base velocity and roll and pitch angles under Case A.

Table 4.2: RMS error comparison under Case A.

State variables	InEKF-SRS	InEKF-DRS (proposed)
v_x (m/s)	0.3320	0.2051
v_y (m/s)	0.2488	0.1955
v_z (m/s)	0.1438	0.1025
yaw (rad)	0.9294	0.2516
pitch (rad)	0.0897	0.0413
roll (rad)	0.1365	0.0318

for base orientation (including yaw) and velocity under Case A. Figure 4-8 shows the corresponding time evolution of the errors for base roll, pitch, and velocity under Case A. The table and the figure show that the proposed InEKF-DRS is more accurate in velocity and orientation estimation compared with InEKF-SRS.

4.4.5 Performance under Different DRS and Robot Motions

Figures 4-9 and 4-10 show the estimation results of the two filters under Case B (where the treadmill motion is different from Case A) and Case C (where the robot stands on the treadmill instead of walking as in Case A). The plots show that the performance comparison of the two filters under Cases B and C are similar to Case A (i.e., Fig. 4-7)), in terms of convergence rate, yaw observability, and accuracy, which indicates the effectiveness of the proposed InEKF-DRS in handling different DRS and robot movements.

Comparing the convergence rate of the yaw estimate under the proposed InEKF-DRS in Cases A-C, we notice that the yaw estimate in Case C converges faster than Cases A and B. In Case C, the treadmill remains horizontal for the first 10 sec, during which the yaw estimate does not converge. Yet, once the treadmill begins to rock at $t = 10$ sec, the yaw estimate converges close to the ground truth within 1 sec, whereas it takes about 3 sec for

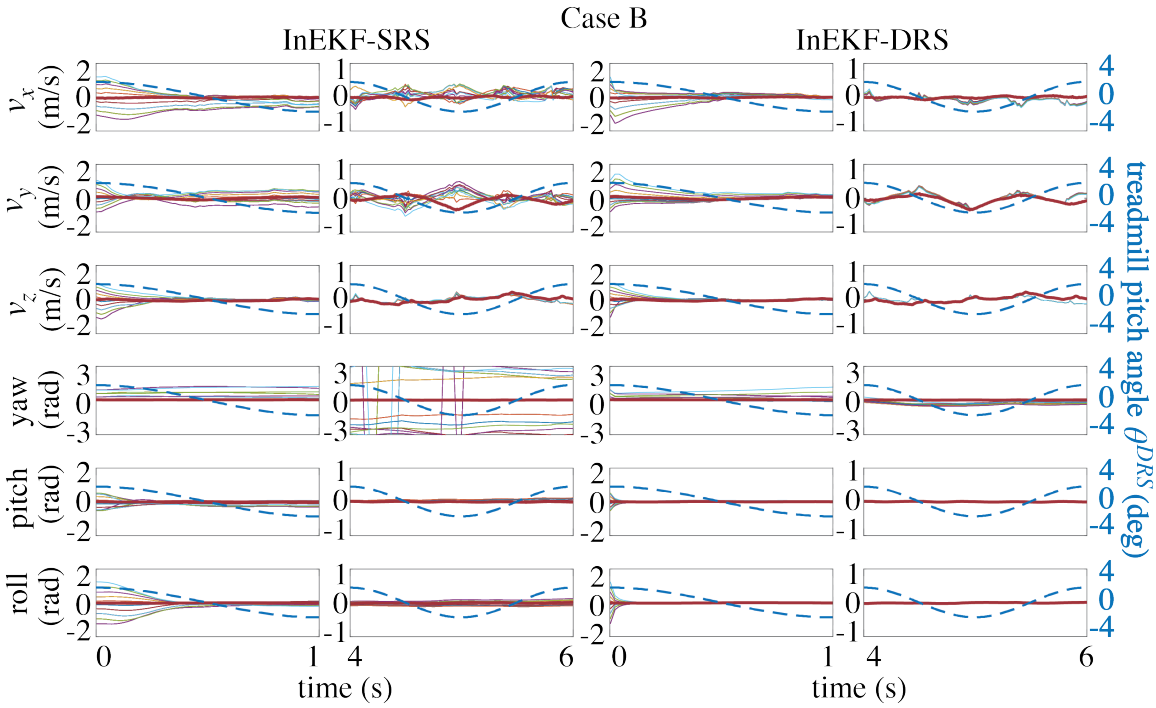

Figure 4-9: Base velocity and orientation estimation results of the two filters, InEKF-DRS (proposed) and InEKF-SRS, for Cases B. The thin solid and thick solid lines are the estimates and ground truth of the base velocity and orientation. The blue-dashed lines are the treadmill orientation profile.

the yaw estimate to enter into a similar neighborhood under Cases A and B. This might be due to the fact that in Case C, by the time the treadmill begins to pitch, the estimates of the rest observable state are already sufficiently accurate, making the correction of the yaw error faster than Cases A and B.

4.4.6 Robustness Assessment

Results from Cases A (Fig. 4-7 and D (Fig. 4-11) confirm the robustness of the proposed InEKF-DRS under inaccurate surface pose knowledge. Case D emulates the scenario where the DRS's motion monitoring system fails to provide accurate DRS pose. Subplots a and b show that the filter performance (e.g., convergence rate, accuracy, and yaw observability) under Case D is similar to that under Case A. Specifically, the velocity estimate under InEKF-DRS converges to the ground truth in all directions within 1 sec. Also, the orientation

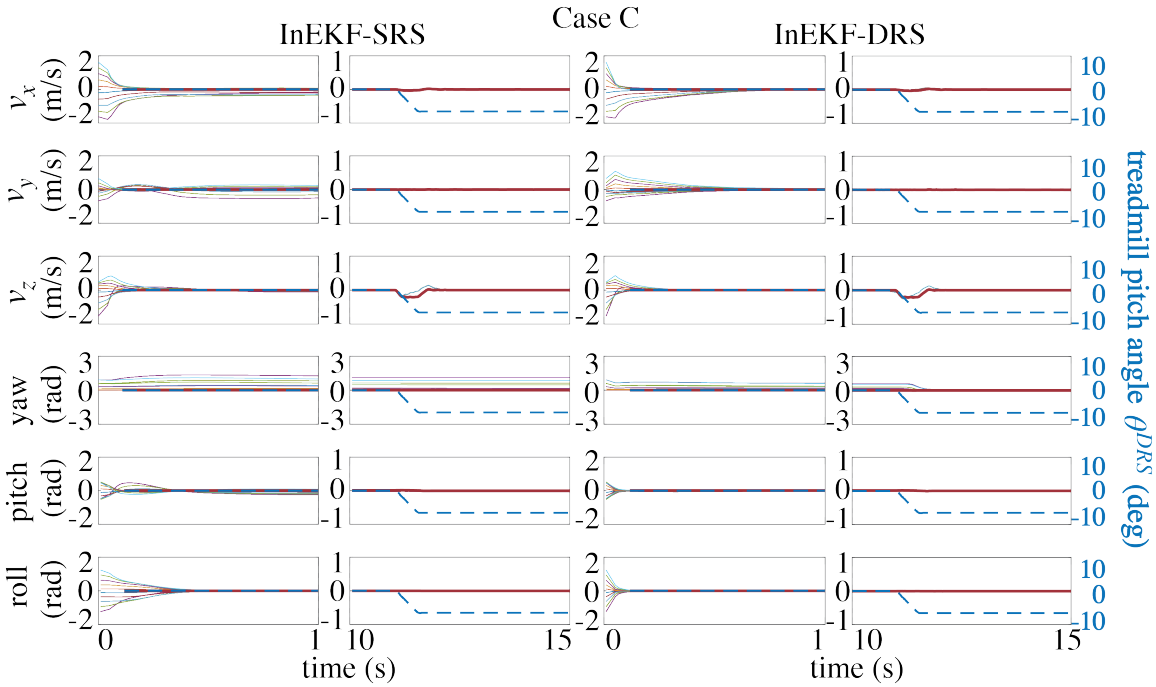

Figure 4-10: Base velocity and orientation estimation results of the two filters, InEKF-DRS (proposed) and InEKF-SRS, for Cases C. The thin solid and thick solid lines are the estimates and ground truth of the base velocity and orientation. The blue-dashed lines are the treadmill orientation profile.

convergence rates are similar: the roll and pitch estimates converge close to the ground truth within 0.3 sec, and the yaw angle converges within 3 sec.

4.5 Discussion

In this chapter, we have developed an InEKF for estimating a bipedal robot's orientation and velocity while it traverses a DRS with known substantial motion. Data from various sources, including the surface pose profile as well as the leg, visual, and inertial data, are integrated.

Our filter, similar to InEKF [29] and EKF [32] for stationary surfaces, uses IMU motion dynamics as the process model and leverages 3-D contact point position data and leg kinematics for the measurement model. However, it distinguishes itself by not assuming a static surface-foot contact point; instead, it accounts for its movement in the world frame.

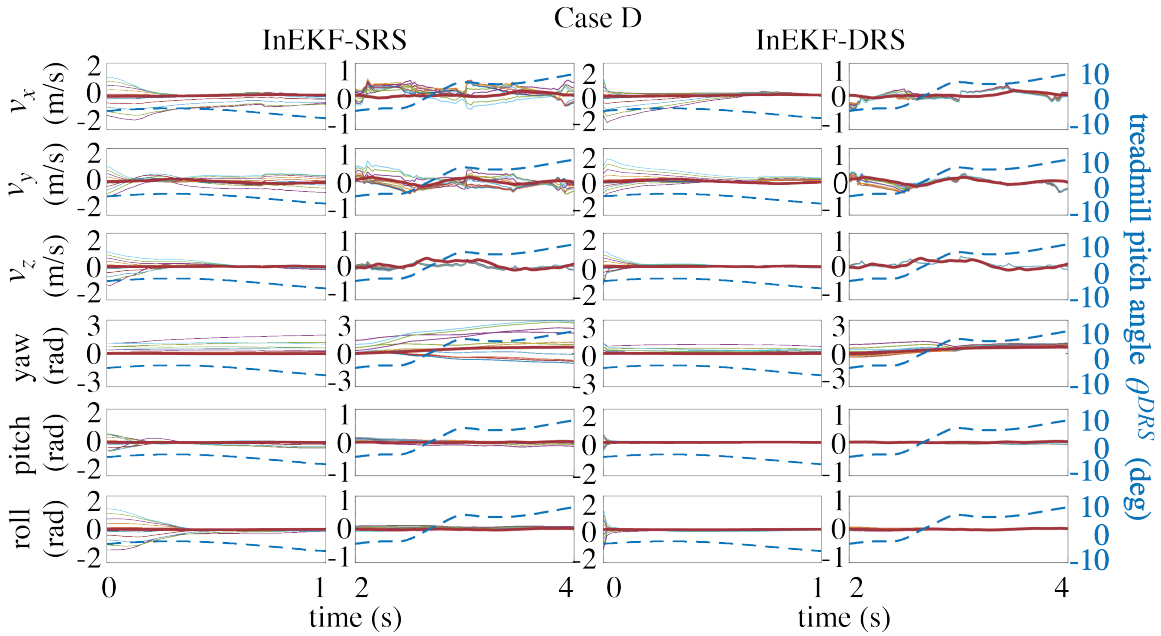

Figure 4-11: Base velocity and orientation estimation results of the two filters, InEKF-DRS (proposed) and InEKF-SRS, for Cases D. The thin solid and thick solid lines are the estimates and ground truth of the base velocity and orientation. The blue-dashed lines are the treadmill orientation profile.

We have also introduced a new right-invariant measurement model based on the rotational kinematic relationship between the surface and the support foot. These innovations ensure accurate estimation even with significant surface motion and estimation errors, as evidenced by RMS errors in Table 4.2 and state trajectories in Figs. 4-7,4-9,4-10, 4-11, and 4-8.

Our filter is well-suited for DRS scenarios with an accurately known surface pose but may not be as effective with highly inaccurate or entirely unknown profiles. We plan to extend it to estimate the surface pose by creating a matrix Lie group that incorporates this information into the state.

Lastly, we have assumed a relatively flat surface with minimal persistent slippage. In cases of uneven or slippery surfaces [120], where the robot's feet slip persistently, discrepancies can occur between actual robot movement and model predictions. Our method can be extended to address such situations by incorporating existing techniques such as using RGB-D sensors to measure base velocity under consistent foot slippage [30]. Figure 4-12 shows screenshots of experiments.

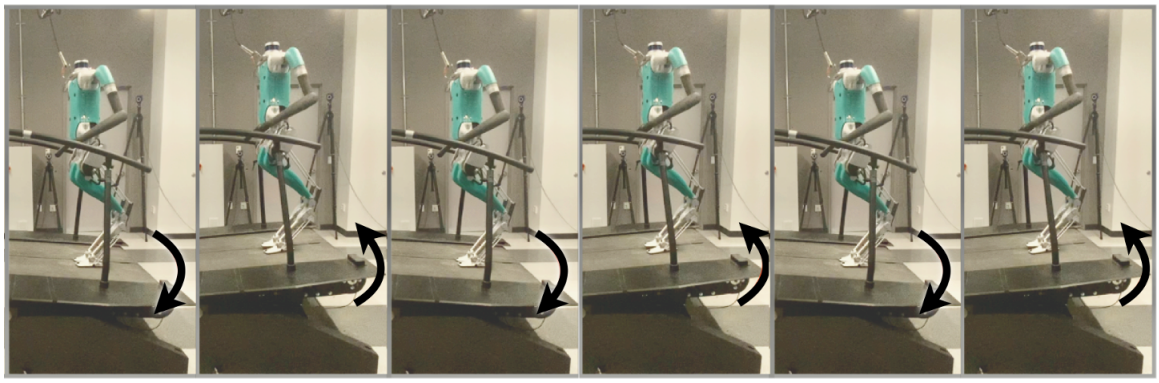

Figure 4-12: Time-lapse figures of Digit walking on a rocking treadmill. The black arrow indicates the treadmill's direction of rotation.

4.6 Summary

- A right-invariant extended Kalman filter that explicitly incorporates known DRS movement and hybrid robot behaviors is introduced.
- Observability analysis for the continuous locomotion phases shows that the robot's base velocity and roll and pitch angles are observable, and the base yaw angle becomes observable when the DRS is not horizontal.
- Stability analysis proves the asymptotic error convergence of these observable states for the hybrid deterministic system.
- Experimental results on a pitching treadmill demonstrate the enhanced accuracy and convergence of the proposed filter when compared to existing methods.
- Two papers on this topic [5, 121] have been published.

Chapter 5 Foot-Placement Control for Underactuated Bipedal Walking on Swaying Dynamic Rigid Surfaces

The aim of this chapter is to formulate a planning and control framework for achieving stable underactuated bipedal locomotion on a horizontally swaying rigid surface. Achieving this objective poses a significant challenge due to the nonlinear and hybrid dynamics of the robot [72, 122], the presence of underactuation [3, 82], and the time-varying movement of the robot-surface contact region [121] (Fig. 5-1).

5.1 Angular Momentum about the Contact Point

This section introduces the concept of angular momentum about the contact point during bipedal walking, along with its fundamental characteristics and associated advantages. Initially, we focus on the single support phase of walking, where only one leg maintains contact with the ground. We also investigate the discrete foot-switching event, which involves the instantaneous changing of the contact point. Finally, we summarize the benefits of considering angular momentum at the contact point.

5.1.1 Assumptions

Throughout this chapter, the following assumptions are considered:

A5.1 The CoM height remains constant above the contact point during walking.

A5.2 The terrain is flat; i.e., there is no vertical variation on the terrain.

A5.3 The role switching of the support and the swing legs is instantaneous.

A5.4 The DRS only moves horizontally, and its motion profile is periodic and accurately known for all time.

A5.5 Foot switching periodically occurs at fixed time instants.

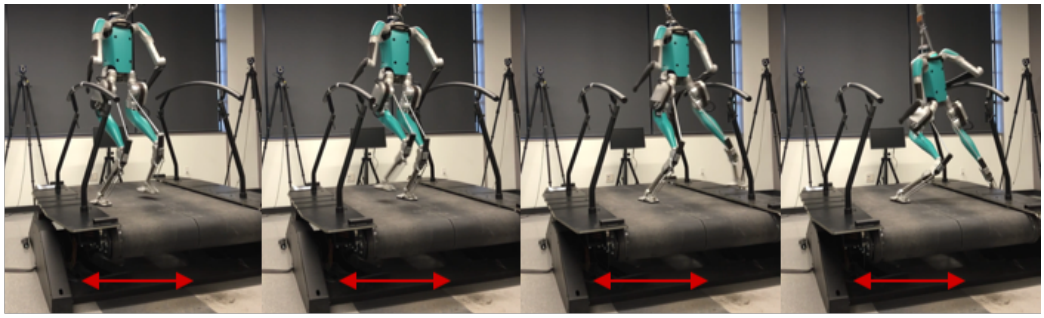

Figure 5-1: The default controller of the Digit humanoid robot exhibited fast lateral position drift on a DRS that oscillated at a frequency of 0.25 Hz and a magnitude of 5 cm.

A5.6 The support foot does not slip on the walking surface.

5.1.2 Single Support Phases

During the swing phase of walking, one foot of the biped maintains contact with the walking surface while the other remains in the air. We will refer to the contact point on the walking surface as S. In the case of a dynamic rigid surface (DRS), the point S moves in the world frame. Let A represent a fixed point in the world frame that coincides with S at a given moment in time.

In the single-support phase, the vector of angular momentum about point S, denoted as $\mathbf{L}_S = [L_{x,S}, L_{y,S}, L_{z,S}] \in \mathbb{R}^3$, is related to the angular momentum about point A, denoted as $\mathbf{L}_A \in \mathbb{R}^3$, as follows:

$$\mathbf{L}_S = \mathbf{L}_A + \mathbf{p}_{SA} \times (m\mathbf{v}_{CoM}), \tag{5.1}$$

where $\mathbf{p}_{SA} \in \mathbb{R}^3$ represents the position of point A relative to point S, described in the world frame. It is important to note that $\mathbf{p}_{SA}$ is equal to $\mathbf{0}_{3\times 1}$ since point A and point S coincide at the given time. The scalar constant m corresponds to the total mass of the robot. The vector $\mathbf{v}_{CoM}$ denotes the linear velocity of the center of mass (CoM) with respect to (w.r.t.) the world frame.

The relationship presented in Eq. (5.1) will be utilized to derive the dynamics of $\mathbf{L}_S$ for legged locomotion on a DRS, as described in Section 5.2.1.

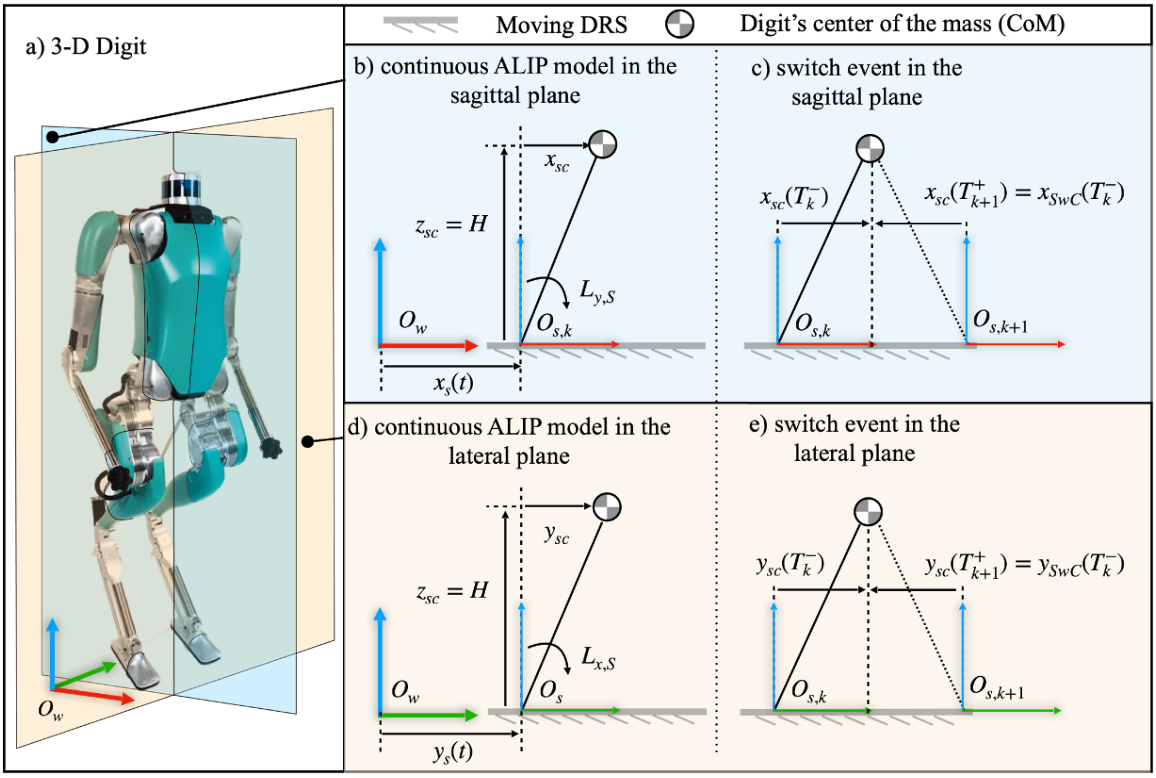

Figure 5-2: Illustrations of a) the sagittal and lateral plane of a 3-D robot, b) continuous and c) discrete ALIP dynamics in the sagittal plane, and d) continuous and e) discrete ALIP dynamics in the lateral plane.

5.1.3 Discrete Foot-Switching Event

Upon completing the swing phase, the swing foot makes contact with the walking surface. Assuming that the support foot begins its swing immediately after the swing foot touchdown (A5.3), we derive the robot dynamics associated with the foot landing event next.

At a foot landing event, there is a sudden change in the position of the contact point. Let $\mathbf{L}_{S,k}$ represent the angular momentum about the k^{th} contact point on the walking surface. The position vector $\mathbf{p}_{(k+1)\to k}$ points from the $(k+1)^{th}$ contact point to the k^{th} contact point.

To describe the angular momentum just before and after the k^{th} foot-landing instant, we use the notations $(\cdot)^-$ and $(\cdot)^+$, respectively. Thus, the relationship between the angular momentum about the new contact point, $\mathbf{L}_{S,k+1}^-$, and the previous contact point, $\mathbf{L}_{S,k}^-$, can be

expressed as follows:

$$\mathbf{L}^{-}_{S,k+1} = \mathbf{L}^{-}_{S,k} + \mathbf{p}_{(k+1)\to k} \times (m\mathbf{v}^{-}_{CoM}). \tag{5.2}$$

During the k^{th} foot-landing event, the impact between the surface and the foot at the new contact point results in zero impulse torque about the $(k+1)^{th}$ contact point. As a consequence, the angular momentum about the $(k+1)^{th}$ contact point, denoted as $\mathbf{L}_{S,k+1}$, remains unchanged across the foot-landing impact. In other words, $\mathbf{L}^{+}_{S,k+1}$ is equal to $\mathbf{L}^{-}_{S,k+1}$.

We can express $\mathbf{L}^{+}_{k+1}$ as follows:

$$\mathbf{L}^{+}_{S,k+1} = \mathbf{L}^{-}_{S,k} + \mathbf{p}_{(k+1)\to k} \times (m\mathbf{v}^{-}_{CoM}). \tag{5.3}$$

As the robot walks on a flat, horizontal terrain with zero vertical CoM velocity (A5.1 and A5.2), we have $\mathbf{p}_{(k+1)\to k} \times (m\mathbf{v}^{-}_{CoM}) = \mathbf{0}_{3\times 1}$ Thus, based on Eq. (5.3), we know that $\mathbf{L}^{+}_{S,k+1} = \mathbf{L}^{-}_{S,k}$ holds. This implies that the robot's angular momentum about the contact point remains constant across a foot switching event.

5.1.4 Advantages

To summarize, the advantages of using the angular momentum in controller design are [100]:

- In contrast to the linear CoM velocity, angular momentum provides additional information regarding a robot's inertial properties, such as mass and mass distribution.
- With the zero support-ankle torque, the time derivative of $\mathbf{L}_S$ is a function of the relative position of the CoM w.r.t. the support foot. This means the torque peaks weakly affect $\mathbf{L}_S$.
- Unlike the linear COM velocity, the $\mathbf{L}_S$ is invariant to the impact when the robot walks on a flat, horizontal surface with zero vertical CoM velocity. This property allows us to obtain $\mathbf{L}_S$ at a new contact point without involving the complex impact model.

5.2 ALIP Dynamics

This section presents the proposed ALIP model that describes the essential dynamics of a walking robot on a horizontally swaying surface during continuous phases and across contact switching. We first derive the three-dimensional (3-D) ALIP model and then we decompose it into the sagittal and the lateral planes to further simplify the model (see Fig. 5-2).

5.2.1 ALIP Dynamics During a Continuous Swing Phase

Recall that the point A is a static point in the world frame that instantaneously coincides with the contact point S (which is attached to the DRS) at the given time and that the 3-D angular momentum vectors $\mathbf{L}_A$ and $\mathbf{L}_S$ are related through Eq. (5.1). Taking the time derivative of both sides of Eq. (5.1) yields

$$\dot{\mathbf{L}}_S = \dot{\mathbf{L}}_A + \dot{\mathbf{p}}_{SA} \times (m\mathbf{v}_{CoM}) + \mathbf{p}_{SA} \times (m\dot{\mathbf{v}}_{CoM}). \tag{5.4}$$

Since $\mathbf{p}_{SA} = \mathbf{0}$ and $\dot{\mathbf{p}}_{SA} = -\dot{\mathbf{p}}_S$, Eq. (5.4) becomes

$$\dot{\mathbf{L}}_S = \dot{\mathbf{L}}_A - \dot{\mathbf{p}}_S \times (m\mathbf{v}_{CoM}), \tag{5.5}$$

where $\mathbf{p}_S = [x_S, y_S, z_S]^T \in \mathbb{R}^3$ is the position of the point S w.r.t. the world frame.

As the time derivative of the angular momentum $\mathbf{L}_A$ equals the sum of the external moments about point A, we have

$$\dot{\mathbf{L}}_A = \mathbf{p}_{AC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A, \tag{5.6}$$

where $\mathbf{p}_{AC} \in \mathbb{R}^3$ is the position of the CoM relative to point A, $\mathbf{g} = [0, 0, -g]^T \in \mathbb{R}^3$ is the gravitational acceleration, and $\boldsymbol{\tau}_A = [\tau_{A,x}, \tau_{A,y}, \tau_{A,z}]^T \in \mathbb{R}^3$ is the external torque that is applied to the contact point.

Since $\mathbf{p}_{AC} = \mathbf{p}_{SC}$, where $\mathbf{p}_{SC} = [x_{SC}, y_{SC}, z_{SC}]^T \mathbb{R}^3$ is the CoM position relative to point S,

Eq. (5.6) can be rewritten as:

$$\dot{\mathbf{L}}_A = \mathbf{p}_{SC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A. \tag{5.7}$$

Combining Eqs. (5.5) and (5.7), we obtain the 3-D dynamic model for $\mathbf{L}_S$:

$$\dot{\mathbf{L}}_S = \mathbf{p}_{SC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A - \dot{\mathbf{p}}_S \times (m\mathbf{v}_{CoM}). \tag{5.8}$$

Because the CoM of the robot has zero vertical velocity (A5.1) and DRS only moves horizontally (A5.4), we have $\dot{\mathbf{p}}_S \times (m\mathbf{v}_{CoM}) = \mathbf{0}_{3\times 1}$. Thus, Eq. (5.8) becomes

$$\dot{\mathbf{L}}_S = \mathbf{p}_{SC} \times (m\mathbf{g}) + \boldsymbol{\tau}_A. \tag{5.9}$$

The relative CoM velocity $\dot{\mathbf{p}}_{SC}$ can be expressed as:

$$\dot{\mathbf{p}}_{SC} = \mathbf{v}_{CoM} - \dot{\mathbf{p}}_S, \tag{5.10}$$

where the expression of $\mathbf{v}_{CoM}$ can be obtained through the following relationship between $\mathbf{L}_S$ and the robot's angular momentum about the CoM, denoted as $\mathbf{L}_{CoM}$:

$$\mathbf{L}_S = \mathbf{L}_{CoM} + \mathbf{p}_{SC} \times (m\mathbf{v}_{CoM}). \tag{5.11}$$

Supposing that the toe joint of the support foot is disabled (discussed in Section 5.3.2), we have $\tau_{A,x} = \tau_{A,y} = 0$. Recall that the DRS only moves horizontally (A5.4), and the CoM does not demonstrate any vertical velocity (A5.1). By putting Eqs. (5.9) and (5.10) into a scalar form, we have

$$\begin{bmatrix} \dot{L}_{x,S} \\ \dot{L}_{y,S} \\ \dot{L}_{z,S} \end{bmatrix} = \begin{bmatrix} -mgy_{SC} \\ mgx_{SC} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tau_{A,z} \end{bmatrix} \tag{5.12}$$

and

$$\begin{bmatrix} \dot{x}_{SC} \\ \dot{y}_{SC} \\ \dot{z}_{SC} \end{bmatrix} = \begin{bmatrix} \frac{1}{mH} L_{y,S} \\ \frac{-1}{mH} L_{x,S} \\ 0 \end{bmatrix} - \begin{bmatrix} \dot{x}_S(t) \\ \dot{y}_S(t) \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{mH} L_{y,CoM} \\ \frac{-1}{mH} L_{x,CoM} \\ 0 \end{bmatrix}, \tag{5.13}$$

where $L_{x,CoM}$ and $L_{y,CoM}$ are the x and y component of L_{CoM}, and H is the CoM height. Because the whole body does not demonstrate significant angular momentum about the CoM during walking, we approximately make $\mathbf{L}_{CoM} = \mathbf{0}_{3\times 1}$.

By splitting the first two rows of Eqs (5.12) and (5.13), we have the ALIP dynamics in the sagittal plane, i.e.,

$$\underbrace{\begin{bmatrix} \dot{x}_{SC} \\ \dot{L}_{y,S} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & \frac{1}{mH} \\ mg & 0 \end{bmatrix}}_{=:\mathbf{A}_x} \underbrace{\begin{bmatrix} x_{SC} \\ L_{y,S} \end{bmatrix}}_{\mathbf{x}} - \underbrace{\begin{bmatrix} \dot{x}_S(t) \\ 0 \end{bmatrix}}_{=:\mathbf{f}_x(t)}, \tag{5.14}$$

and in the lateral plane, i.e.,

$$\underbrace{\begin{bmatrix} \dot{y}_{SC} \\ \dot{L}_S \end{bmatrix}}_{\dot{\mathbf{y}}} = \underbrace{\begin{bmatrix} 0 & -\frac{1}{mH} \\ -mg & 0 \end{bmatrix}}_{=:\mathbf{A}_y} \underbrace{\begin{bmatrix} y_{SC} \\ L_S \end{bmatrix}}_{\mathbf{y}} - \underbrace{\begin{bmatrix} \dot{y}_S(t) \\ 0 \end{bmatrix}}_{=:\mathbf{f}_y(t)}. \tag{5.15}$$

It is important to note that the proposed continuous-phase ALIP model in Eqs. (5.14) and (5.15) exhibits explicit time-varying characteristics due to the presence of the time-varying forcing terms $\mathbf{f}_x(t)$ and $\mathbf{f}_y(t)$. This time-varying property sets it apart from the standard time-invariant LIP model [8]. The explicit time dependence of the forcing terms $\mathbf{f}_x(t)$ and $\mathbf{f}_y(t)$ arise from the time-varying velocity of the support point S. Throughout this study, we assume that the horizontal surface motion $\dot{x}_S(t)$ and $\dot{y}_S(t)$ are continuous, differentiable, and periodic such that $x_S(t) = x_S(t + T_{x,DRS})$ and $y_S(t) = y_S(t + T_{y,DRS})$ hold for any $t > 0$, where $T_{x,DRS}$ and $T_{y,DRS}$ are real, positive constants that represent the periodicity of the DRS motion in the respective plane.

5.2.2 ALIP Dynamics at Foot Landing

When a biped undergoes a foot landing event, the roles of its support and swing feet instantaneously switch, leading to a sudden jump in both the contact point position (x_S,y_S) and the relative CoM position (x_{SC},y_{SC}) (see Figs. 5-2-c and e).

Let the variables Δx_{SC} and Δy_{SC} represent the changes in x_{SC} and y_{SC}, respectively, across a foot landing event. They are defined as: $\Delta x_{SC} = x_{SC}^{+} - x_{SC}^{-}$ and $\Delta y_{SC} = y_{SC}^{+} - y_{SC}^{-}$. Here, x_{SC}^{+} and x_{SC}^{-} are the values of x_{SC} just after and before the foot landing event, respectively. Similarly, y_{SC}^{+} and y_{SC}^{-} denote the values of y_{SC} just after and before the foot landing event, respectively. These quantities represent the changes in the CoM position w.r.t. the support foot in the sagittal (x) and lateral (y) directions during the foot landing event.

As highlighted earlier and discussed in [9], $\mathbf{L}_S$ remains unaffected by foot landing events due to its impact invariance property. Consequently, the change in $\mathbf{L}_S$ during a foot landing event satisfies $\Delta L_{y,S} := L_{y,S}^{+} - L_{y,S}^{-} = 0$ and $\Delta L_{x,S} := L_{x,S}^{+} - L_{x,S}^{-} = 0$.

Let $\Delta\mathbf{x}$, $\Delta\mathbf{y}$ represent the difference between the pre- and post-switching states in the sagittal and lateral planes, respectively. The compact expression of the jump/impact map of $\Delta\mathbf{x}$ in the sagittal plane is given by:

$$\underbrace{\begin{bmatrix} \Delta x_{SC} \\ \Delta L_{y,S} \end{bmatrix}}_{\Delta\mathbf{x}} = \begin{bmatrix} x_{SC}^{+} - x_{SC}^{-} \\ 0 \end{bmatrix}, \tag{5.16}$$

and in the lateral plane is given by:

$$\underbrace{\begin{bmatrix} \Delta y_{SC} \\ \Delta L_{x,S} \end{bmatrix}}_{\Delta\mathbf{y}} = \begin{bmatrix} y_{SC}^{+} - y_{SC}^{-} \\ 0 \end{bmatrix}. \tag{5.17}$$

5.3 Hierarchical Planning and Control Framework

This section presents the proposed hierarchical planning and control framework that enables stable underactuated bipedal walking on a DRS with a known, periodic, horizontal

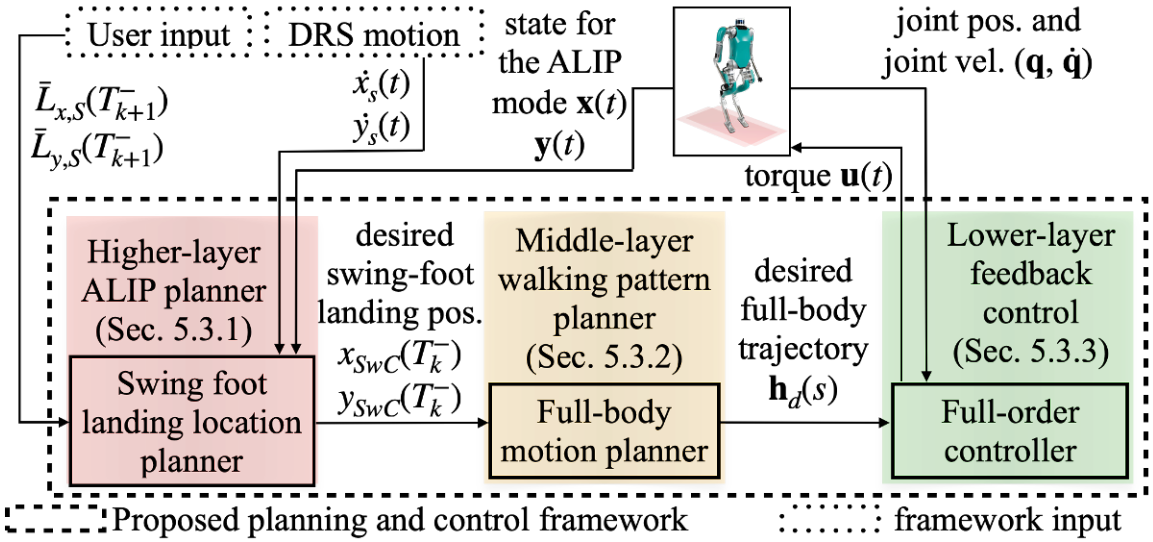

Figure 5-3: Overview of the proposed hierarchical control approach. The ALIP-based planner generates the desired foot landing locations based on the current state of the full-order model. The middle-layer walking pattern generator adjusts the desired swing foot trajectories based on the full-order model and the higher-layer planning result. In the lower layer, the input-output linearizing controller reliably tracks the full-order reference trajectories.

motion. As depicted in Fig. 5-3, the higher-layer ALIP-based planner is a component of the proposed three-layer control approach designed to achieve stable walking on a full-order, underactuated robot. The main challenge lies in effectively dealing with the complexity of the hybrid, time-varying, and nonlinear dynamics inherent in underactuated robots. The middle and lower layers serve two key purposes in achieving the overall control objective. Firstly, the middle layer generates full-body reference trajectories that are compatible with the ALIP model, taking into account the model simplifying assumptions underlying the proposed ALIP model, such as the constant CoM height and the zero angular momentum about the CoM, to reduce the discrepancies between the real robot and the ALIP model. Secondly, the middle layer translates the desired footstep locations provided by the higher layer into reference swing foot trajectories that can be reliably tracked by the lower layer controller. This ensures that the actual robot faithfully executes the planned foot placements, thereby ensuring stability for the underactuated dynamics of the full-order robot.

5.3.1 Higher-Layer ALIP-based Planner

Within this subsection, we present the higher-level planner component of the proposed hierarchical control method. This planner aims to generate the desired CoM position trajectory and desired foot placement locations.

The design of the proposed ALIP-based planner begins with the derivation of a discrete-time foot placement controller based on the hybrid, time-varying ALIP model, as explained next. As the ALIP model shares the same form in the sagittal and lateral planes, the derivation focuses on the sagittal plane alone.

5.3.1.1 Discrete-Time Foot Placement Control

The proposed discrete-time foot placement controller regulates the angular momentum at the end of the next step by choosing the swing foot landing location at the end of the current step.

From the linear system theory, the solution of Eq. (5.14) between $[t_1, t_2]$ is expressed as follows:

$$\begin{bmatrix} x_{SC}(t_2) \\ L_{y,S}(t_2) \end{bmatrix} = \underbrace{\begin{bmatrix} cosh(l(t_2-t_1)) & \frac{sinh(l(t_2-t_1))}{mHl} \\ mHlsinh(l(t_2-t_1)) & cosh(l(t_2-t_1)) \end{bmatrix}}_{e^{\mathbf{A}_x(t_2-t_1)}} \begin{bmatrix} x_{SC}(t_1) \\ L_{y,S}(t_1) \end{bmatrix} + \underbrace{\int_{t_1}^{t_2} e^{\mathbf{A}_x(t_2-\tau)}\mathbf{f}_x(\tau)d\tau}_{\mathbf{V}_x(t_1,t_2)=[V_{x,1}(t_1,t_2),V_{x,2}(t_1,t_2)]^T}, \tag{5.18}$$

where $l = \sqrt{\frac{g}{H}}$. It is worth highlighting that the computation of $\mathbf{V}_x(t_1,t_2)$ can be readily carried out when t_1, t_2, and $\mathbf{f}_x(\tau)$ are known. Since we assume that the DRS motion is known for all time, we can evaluate $\mathbf{f}_x(\tau)$ at any given time τ.

From the second row of Eq.(5.18), the angular momentum at the end of the next step $L_{y,S}(T_{k+1}^-)$ is related to the angular momentum at the beginning of the next step $L_{y,S}(T_k^+)$

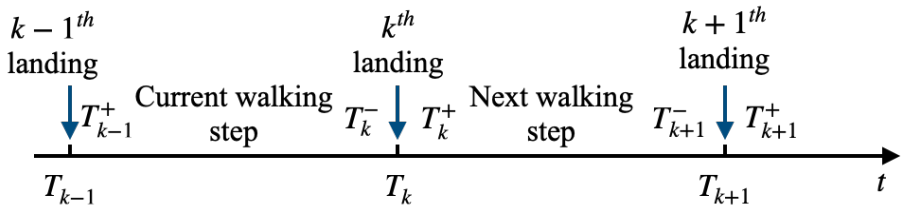

Figure 5-4: Time line illustration. T_{k-1} is the desired start instant of the current walking step, T_k is the desired timing for the end of the current step, and T_{k+1} is the desired timing for the end of the next step. The superscript $+$ and $-$ indicates the right and left limits.

and the relative CoM position at the beginning of the next step $x_{SC}(T_k^+)$ (see Fig. 5-4):

$$\begin{aligned} L_{y,S}(T_{k+1}^-) = mHlsinh(lT_{step})x_{SC}(T_k^+) \\ + cosh(lT_{step})L_{y,S}(T_k^+) + V_{x,2}(T_k^+, T_{k+1}^-), \end{aligned} \tag{5.19}$$

where $T_{step} = T_{k+1}^- - T_k^+$ is the desired step duration.

Given that the position of the support foot at T_k^+ equals that of the swing foot at T_k^- (A5.3), we obtain:

$$x_{SC}(T_k^+) = x_{SwC}(T_k^-), \tag{5.20}$$

where $x_{SwC}(T_k^-)$ is the horizontal element of the position vector from the swing foot to the CoM at T_k^-(see Fig. (5-2)). As discussed in Section 5.1, L_S remains invariant under impact. As a result, we obtain:

$$L_{y,S}(T_k^+) = L_{y,S}(T_k^-). \tag{5.21}$$

Combining Eqs. (5.19), (5.20), and (5.21), we have

$$\begin{aligned} L_{y,S}(T_{k+1}^-) = mHlsinh(lT_{step})x_{SwC}(T_k^-) \\ + cosh(lT_{step})L_{y,S}(T_k^-) + V_{x,2}(T_k^+, T_{k+1}^-). \end{aligned} \tag{5.22}$$

From Eq. (5.22), we are able to obtain the expression of the swing foot landing location at the end of the current step T_k^-:

$$x_{SwC}(T_k^-) = \frac{L_{y,S}(T_{k+1}^-) - V_{x,2}(T_k^+, T_{k+1}^-) - cosh(lT_{step})L_{y,S}(T_k^-)}{mHlsinh(lT_{step})}. \tag{5.23}$$

In Eq. (5.23), $V_{x,2}(T_k^+, T_{k+1}^-)$ can be directly computed, and $cosh(lT_{step})$ and $mHlsinh(lT_{step})$ are known constants.

Let us introduce the desired angular momentum $\tilde{L}_{y,S}(T_{k+1}^-)$, which will be treated as user input. Replacing $L_{y,S}(T_{k+1}^-)$ with $\tilde{L}_{y,S}(T_{k+1}^-)$, Eq. (5.23) can be rewritten as:

$$x_{SwC}(T_k^-) = \frac{\tilde{L}_{y,S}(T_{k+1}^-) - V_{x,2}(T_k^+, T_{k+1}^-) - cosh(lT_{step})L_{y,S}(T_k^-)}{mHlsinh(lT_{step})}. \tag{5.24}$$

The variable $L_{y,S}(T_K^-)$ can be computed by using the second row of Eq. (5.18) with $t_1 = T_{k-1}^+$ and $t_2 = T_k^-$.

Figure 5-5 illustrates the procedure for computing the swing foot landing location.

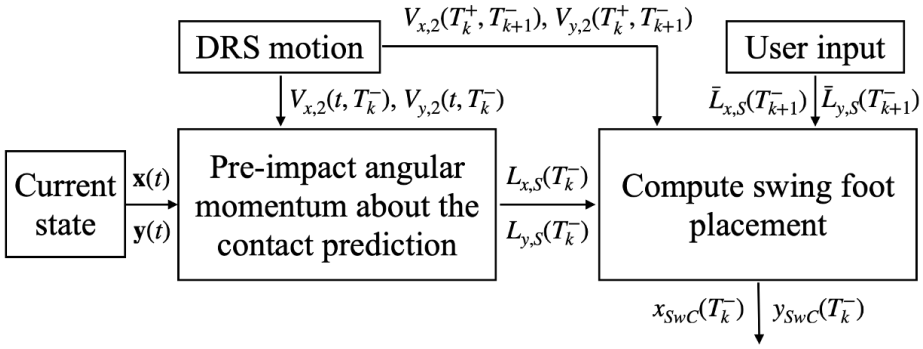

Figure 5-5: Illustration of computing the swing foot landing location

5.3.1.2 Hybrid ALIP Model

Given the combination of continuous-time dynamics and discrete-time behaviors during walking, it is appropriate to represent the robot dynamics as a hybrid system. To this end, we formulate the proposed ALIP as a linear hybrid system. This formulation enables us to use the existing theorems and analysis techniques to study the stability properties of the ALIP model.

By combining Eqs. (5.16) and (5.24), we obtain:

$$\begin{aligned}\underbrace{\begin{bmatrix}\Delta x_{SC}\\ \Delta L_S\end{bmatrix}}_{\Delta\mathbf{x}} &= \underbrace{\begin{bmatrix}-1 & -\frac{cosh(lT_{step})}{mHlsinh(lT_{step})}\\ 0 & 0\end{bmatrix}}_{=:\mathbf{B}_x}\underbrace{\begin{bmatrix}x_{SC}(T_k^-)\\ L_S(T_k^-)\end{bmatrix}}_{\mathbf{x}^-}\\ &+\underbrace{\begin{bmatrix}\frac{\tilde{L}_S(T_{k+1}^-)-V_2(T_k^+,T_{k+1}^-)}{mHlsinh(lT_{step})}\\ 0\end{bmatrix}}_{=:\mathbf{G}_x}, \text{ if } t=T_k^- \ (k=1,2,...).\end{aligned} \tag{5.25}$$

Combining Eq. (5.14) and Eq. (5.25), our hybrid linear system has the following format:

$$\begin{cases}\dot{\mathbf{x}}=\mathbf{A}_x\mathbf{x}^- -\mathbf{f}_x(t), & t\neq T_k^- \ (k=1,2,...);\\ \Delta\mathbf{x}=\mathbf{B}_x\mathbf{x}^-+\mathbf{G}_x, & t=T_k^- \ (k=1,2,...).\end{cases} \tag{5.26}$$

5.3.1.3 Stability Analysis

The stability condition for the periodic solution of linear time-varying nonhomogeneous hybrid systems, which includes the proposed hybrid ALIP model, is presented in [123]. To analyze the stability of a linear time-varying nonhomogeneous hybrid system, we need to consider the corresponding linear time-varying homogeneous hybrid system that is expressed as:

$$\begin{cases}\dot{\mathbf{z}}=\mathbf{A}_x\mathbf{z}^-, & t\neq T_k^- \ (k=1,2,...);\\ \Delta\mathbf{z}=\mathbf{B}_x\mathbf{z}^-, & t=T_k^- \ (k=1,2,...),\end{cases} \tag{5.27}$$

where $\mathbf{z}\in\mathbb{R}^2$ is the state variable. This homogeneous equation corresponds to the ALIP dynamics without the DRS motion.

Theorem 2. *(***Closed-loop stability conditions of the Hybrid ALIP model***) If all eigenvalues of the monodromy matrix of the hybrid homogeneous system (Eq.* (5.27)*) have magnitudes less than one, then the hybrid homogeneous system is exponentially stable. Further, the periodic solution $\boldsymbol{\psi}(t)$ of the overall nonhomogeneous hybrid system (Eq.* (5.26)*) with a periodicity of T_{sys}, i.e., $\boldsymbol{\psi}(t+T_{sys})=\boldsymbol{\psi}(t)$, is also exponentially stable.*

The proof of this theorem can be found in [123].

A Specific Scenario: Identical Periodicity of DRS Motion and One Walking Step:

When the periodicity of the DRS motion, denoted as $T_{x,DRS}$, is equal to the periodicity of the walking motion, denoted as T_{step}, it can be shown that $T_{x,DRS} = T_{step} = T_{sys}$, the matrix $\mathbf{M}_x$ can be calculated using the following expression [123]:

$$\begin{aligned}\mathbf{M}_x &= (\mathbf{I}+\mathbf{B}_x)e^{\mathbf{A}_x T_{step}} \\ &= \begin{bmatrix} -cosh(lT_{step}) & -\frac{cosh^2(lT_{step})}{mHlsinh(lT_{step})} \\ mHlsinh(lT_{step}) & cosh(lT_{step}) \end{bmatrix}.\end{aligned} \tag{5.28}$$

To compute the eigenvalues of the monodromy matrix $\mathbf{M}_x$, we start by obtaining the characteristic equation of $\mathbf{M}_x$. Let λ_x be an eigenvalue of $\mathbf{M}_x$, and $\mathbf{I}$ be the identity matrix with a proper dimension. The characteristic equation is given by $\det(\mathbf{M}_x - \lambda_x \mathbf{I}) = 0$. Solving this equation yields the eigenvalues $\lambda_{x,1}$ and $\lambda_{x,2}$. These eigenvalues can then be used to analyze the stability of the hybrid system.

The eigenvalue can be computed from

$$\mathbf{M}_x - \lambda \mathbf{I} = \begin{bmatrix} -cosh(lT_{step}) - \lambda & -\frac{cosh^2(lT_{step})}{mHlsinh(lT_{step})} \\ mHlsinh(lT_{step}) & cosh(lT_{step}) - \lambda \end{bmatrix} \tag{5.29}$$

as

$$\lambda_{x,i} = 0 \ (i = 1,2). \tag{5.30}$$

Thus, the hybrid system Eq. (5.27) is stable.

Remark 5 (Physical insights on the monodromy matrix): The monodromy matrix $\mathbf{M}_x$ characterizes the relationship between the state $\mathbf{x}(T_{K-1}^+)$ and the state $\mathbf{x}(T_K^+)$, as mentioned earlier. This relationship is described by the equation:

$$\mathbf{x}(T_k^+) = \mathbf{M}_x \mathbf{x}(T_{k-1}^+) + \mathbf{V}_x(T_{k-1}^+, T_k^-). \tag{5.31}$$

An Extended Scenario: Periodicity of the Overall System (Eq. (5.26)**) as the Least Common Multiple of DRS Motion and Robot Walking**:

Now consider a general case (see Fig. 5-6) where $N_1 T_{step} = N_2 T_{x,DRS} = T_{sys}$, where N_1

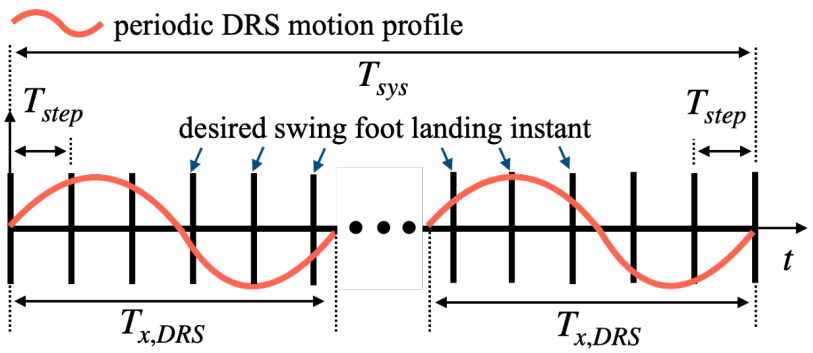

Figure 5-6: Illustration of the periodicity of robot walking T_{step}, DRS motion T_{DRS}, and the solution of the hybrid linear system T_{sys}. This highlights that the proposed formulation does not require T_{step} should equal T_{DRS}.

and N_2 are any positive integers. By using Eq.(5.31) recursively, we have

$$\begin{aligned}
\mathbf{x}(T^+_{k+N_1}) &= \mathbf{M}_x\mathbf{x}(T^+_{k+N_1-1}) + \mathbf{V}_x(T^+_{k+N_1-1}, T^-_{k+N_1}) \\
&= \mathbf{M}_x\left[\mathbf{M}_x\mathbf{x}(T^+_{k+N_1-2}) + \mathbf{V}_x(T^+_{k+N_1-2}, T^-_{k+N_1-1})\right] \\
&+ \mathbf{V}_x(T^+_{k+N_1-1}, T^-_{k+N_1}) \\
&= \mathbf{M}_x^2\mathbf{x}(T^+_{k+N_1-2}) + \mathbf{M}_x\mathbf{V}_x(T^+_{k+N_1-2}, T^-_{k+N_1-1}) \\
&+ \mathbf{V}_x(T^+_{k+N_1-1}, T^-_{k+N_1}) \\
&\ldots \\
&= \underbrace{\mathbf{M}_x^{N_1}}_{\bar{\mathbf{M}}_x}\mathbf{x}(T^+_{k-1}) + \sum_{i=1}^{N_1}\mathbf{M}_x^{N_1-i}\mathbf{V}_x(T^+_{k+i-2}, T^+_{k+i-1})
\end{aligned} \tag{5.32}$$

Let $\bar{\mathbf{M}}_x$ denote the monodromy matrix for this specific scenario. In contrast to the previous scenario, the monodromy matrix in this case can be described by the following equation:

$$\bar{\mathbf{M}}_x = \mathbf{M}_x^{N_1}. \tag{5.33}$$

The i^{th} eigenvalues of $\bar{\mathbf{M}}_x$, which are denoted as $\bar{\lambda}_{x,i}$, can be evaluated by the following expression:

$$\bar{\lambda}_{x,i} = \lambda_{x,i}^{N_1} = 0 \ (i = 1,2). \tag{5.34}$$

Similarly, the hybrid system Eq. (5.27) is stable.

5.3.1.4 Hybrid ALIP Planner on the Lateral Plane

The ALIP dynamics in the lateral plane is expressed as Eq. (5.15), and its solution between $[t_1, t_2]$ is expressed as:

$$\begin{bmatrix} y_{SC}(t_2) \\ L_{x,S}(t_2) \end{bmatrix} = \underbrace{\begin{bmatrix} cosh(l(t_2-t_1)) & -\frac{sinh(l(t_2-t_1))}{mHl} \\ -mHlsinh(l(t_2-t_1)) & cosh(l(t_2-t_1)) \end{bmatrix}}_{e^{\mathbf{A}_y(t_2-t_1)}} \begin{bmatrix} y_{SC}(t_1) \\ L_{x,S}(t_1) \end{bmatrix} + \underbrace{\int_{t_1}^{t_2} e^{\mathbf{A}_y(t_2-\tau)}\mathbf{f}_y(\tau)d\tau}_{\mathbf{V}_y(t_1,t_2)=[V_{y,1}(t_1,t_2),V_{y,2}(t_1,t_2)]^T}, \tag{5.35}$$

By following a similar approach as we have discussed, the corresponding landing location of the swing foot at the end of the current step in the lateral plane is expressed as:

$$y_{SwC}(T_k^-) = \frac{-\tilde{L}_{x,S}(T_{k+1}^-) + V_{y,2}(T_k^+, T_{k+1}^-) + cosh(lT_{step})L_{x,S}(T_k^-)}{mHlsinh(lT_{step})}. \tag{5.36}$$

To determine the value of $\bar{L}_{x,S}(T_{k+1}^-)$ and minimize lateral motion, we can adopt the approach used for walking on static terrain [9]. Thus, the expression for $\bar{L}_{x,S}(T_{k+1}^-)$ can be formulated as:

$$\bar{L}_{x,S} = \begin{cases} \frac{1}{2}mHW\frac{lsinh(lT_{step})}{1+cosh(lT_{step})}, \text{ right support foot} \\ -\frac{1}{2}mHW\frac{lsinh(lT_{step})}{1+cosh(lT_{step})}, \text{ left support foot} \end{cases} \tag{5.37}$$

where $W \in \mathbb{R}$ is the desired step width.

Remark 6 (Stability result in the lateral plane): The stability analysis for the lateral plane yields the same result as the stability analysis for the sagittal plane. Specifically, the two eigenvalues of the monodromy matrix in this case are also equal to 0.

Remark 7 (Walking periodicity in the lateral plane): The walking periodicity in the lateral plane differs from the walking periodicity in the sagittal plane due to the distinction between the left support leg and right support leg. We use the term "gait periodicity" (T_{gait}) to denote the periodicity of a complete walking cycle, which includes multiple steps. In the case of an even step periodicity, we have $T_{gait} = 2T_{step}$.

5.3.2 Middle-Layer Walking Pattern Planner

This subsection explains the middle-layer planner. The main role of this layer is to generate full-body reference motions that are compatible with both the ALIP model and the desired global behaviors determined by the higher-layer planner. These reference trajectories serve as input for the lower-layer controller described later.

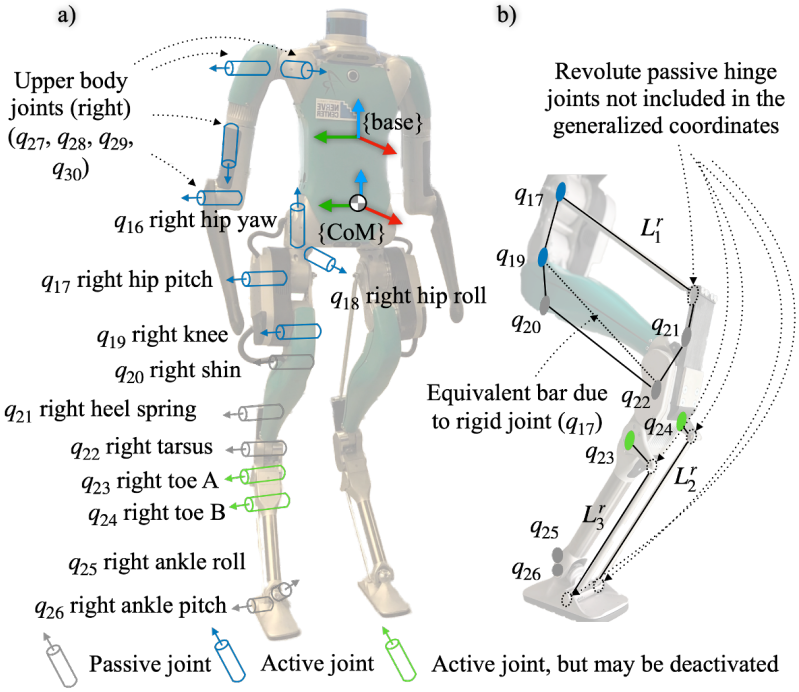

Figure 5-7: a) Illustration of the joints of Digit, with the highlighted joints on the right side (q_{16}-q_{30}). Although the joints on the left side (q_1-q_{15}) are not shown in this figure, they are symmtric to the right side. The green joints indicate that they are active when the corresponding leg is in the swinging phase, but they are deactivated when the corresponding leg serves as the support leg. The joints label are listed as follows: right upper body joints (RUPJ), right hip yaw (RHY), right hip pitch (RHP), right hip roll (RHR), right knee (RK), right shin (RS), right heel spring (RHS) , right tarsas (RTA), right toe A (RTO-A), right toe B (RTO-B), right ankle roll (RAR), right ankle pitch (RAP) b) Illustration of the three closed-loop linkage systems that can be characterized via holonomic constraints. This visualization aids in understanding how passive joints are structured in the system.

5.3.2.1 Generalized coordinates and Holonomic constraints

Let $\mathbf{q}$ denote the vector of generalized coordinates for a full-order 3-D Digit robot. In general, $\mathbf{q}$ is defined as $\mathbf{q} := [\mathbf{p}_b^T, \boldsymbol{\gamma}_b^T, q_1, ..., q_{30}]$, where $\mathbf{p}_b$ and $\boldsymbol{\gamma}_b$ represent the pose of

the robot's base, including its position and orientation in the world frame (recall Chap. 2 Sec 2.1.2). Additionally, q_i (where $i = 1, ..., 30$) corresponds to the angle of the i^{th} revolute joint of the Digit (Fig. 5-7 a)). The base frame $\{base\}$ of the robot is fixed and rigidly attached to a specific point in the torso. Whereas the center of mass frame $\{CoM\}$ has its origin located at the center of mass of the robot. The $\{CoM\}$ frame shares the same yaw angle as the $\{base\}$ frame. However, the roll and pitch angles of the $\{CoM\}$ frame remain fixed at 0 degree w.r.t. the world frame (See Fig. 5-7 a)).

Considering the floating base joints, the Digit robot is equipped with a total of 36 joints, out of which 20 are directly actuated. During each step, the support foot maintains full contact with the ground, enforcing 6 holonomic constraints that ensure the robot's static position and orientation w.r.t. the DRS. The passive joints q_{20} (right shin) and q_5 (left shin) are treated as rigid joints due to their inherent rigidity. Similarly, the heel springs (q_{21} and q_6) on both legs are also treated as rigid links due to their rigidity.

On each leg, there are three closed-loop bar systems implemented to actuate the passive joints, as depicted in Fig. 5-7 b). These holonomic constraints are induced by the constant length of each bar in the closed-loop chain. The 10 scalar holonomic constraints associated with the passive joints on both legs can be compactly expressed as:

$$\mathbf{p}_{pj} = \begin{bmatrix} q_5 \\ q_6 \\ L_1^l(\mathbf{q}) \\ L_2^l(\mathbf{q}) \\ L_3^l(\mathbf{q}) \\ q_{20} \\ q_{21} \\ L_1^r(\mathbf{q}) \\ L_2^r(\mathbf{q}) \\ L_3^r(\mathbf{q}) \end{bmatrix} = \mathbf{a}, \tag{5.38}$$

where $\mathbf{a} \in \mathbb{R}^{10}$ is a constant vector, L_1^r, L_2^r, L_3^r, L_1^l, L_2^l, L_3^l are the length of the rigid bars on both legs, as depicted in Fig. 5-7.

During walking, the Toe-A and Toe-B joints of the support foot are deactivated. This enables Digit to emulate point contact, aligning with the ALIP model. Consequently, the overall degrees of freedom (DOF) of the Digit during walking are computed as:

$$\text{DOF} = 36 - 6 - 10 = 20.$$

Considering that the number of active actuators, denoted as n_a, is 18 (with two joints deactivated), we obtain $n_a < \text{DOF}$. Therefore, the Digit walking is underactuated, with two degrees of underactuation.

5.3.2.2 Full-Body Control Variable Selection

Consider the Digit robot with 30 revolute joints (i.e., $n = 30$) shown in Fig. 5-7 a). Since the robot has 18 independent actuators, the controller can directly command 18 independent variables of interest. Let $\mathbf{h}_c \in \mathbb{R}^{18}$ represent the control variables:

$$\mathbf{h}_c(\mathbf{q}) = [z_{CoM}(\mathbf{q}), \boldsymbol{\gamma}_b^T, x_{sw}(\mathbf{q}), y_{sw}(\mathbf{q}), z_{sw}(\mathbf{q}), \boldsymbol{\gamma}_{sw}^T(\mathbf{q}), \mathbf{q}_{upper}^T]^T. \tag{5.39}$$

Here, $z_{CoM} \in \mathbb{R}$ denotes the height of CoM. $x_{sw} \in \mathbb{R}$ and $y_{sw} \in \mathbb{R}$ respectively represent the position of the swing foot in the x- and y-directions w.r.t. the $\{CoM\}$ frame. $z_{sw} \in \mathbb{R}$ represents the position of the swing foot in the z-direction w.r.t the world frame. $\boldsymbol{\gamma}_{sw}(\mathbf{q}) \in \mathbb{R}^3$ is the orientation of the swing foot w.r.t the $\{CoM\}$ frame. $\mathbf{q}_{upper} \in \mathbb{R}^8$ is a vector that includes all the joints in the upper body.

To enhance the accuracy of the ALIP model, we choose to directly control the CoM height, z_{CoM}. It is crucial to ensure that the CoM height of the actual robot remains constant and aligns with the ALIP model. Additionally, we control the trunk orientation, $\boldsymbol{\gamma}_b$, to maintain an upright trunk posture and an approximately zero angular momentum about the CoM. By controlling the upper body joints, we ensure that there are no unexpected arm movements during walking. Finally, the control of the swing foot position (x_{sw}, y_{sw}, z_{sw}) enables the full-order robot to accurately execute the desired footstep locations provided by the higher-layer planner.

5.3.2.3 Full-Body Trajectory Generation

Let $\mathbf{h}_d$ denote the vector of the desired trajectories for the control variables $\mathbf{h}_c$. We define $\mathbf{h}_d$ as:

$$\mathbf{h}_d = \boldsymbol{\phi}, \tag{5.40}$$

where $\boldsymbol{\phi} = [\phi_1, \phi_2, \ ..., \ \phi_{18}] \in \mathbb{R}^{18}$ represents the desired trajectories for $\mathbf{h}_c$.

Trajectory Parameterization: To parameterize the desired trajectory ϕ_i $(i = 1, ..., 18)$, we utilize Bézier polynomials (recall Sec. 2.3). These polynomials are encoded using a time-based phase variable, s, which represents the progress of a step over time.

Let τ_{k-1} denote the actual switching instant at the beginning of the current walking step of the full-order robot. The phase variable s is defined as $s := \frac{t-\tau_{k-1}}{T_k-\tau_{k-1}}$, where t represents the current time. Therefore, s takes on a value of 0 at the beginning of a planned step and increases to 1 as the step progresses, reaching its maximum value at the end of the step.

Desired Full-order Trajectory Update:

The higher-level planner continually updates the Bézier coefficients of $\phi_5(s)$ (desired swing foot trajectory in the sagittal plane) and $\phi_6(s)$ (desired swing foot trajectory in the lateral plane) based on the robot's current states. The value of a variable at time step t is denoted by $(\cdot)_t$. The update procedure at time step t is outlined in Algorithm 2.

5.3.3 Lower-Layer Feedback Control based on the Full-Order Model

This subsection describes the feedback controller used to track the desired full-body motion by commanding the directly controlled variables of the actual robot. Drawing inspiration from the HZD approach [2], we develop a tracking controller that utilizes the full-order, hybrid, nonlinear dynamics model of the robot. We employ the exact linearization technique to approximate the nonlinear mapping between the joint torques and the tracking error being commanded.

In addition, we address the time-varying nature of the robot dynamics by incorporating modeling and control techniques, similar to our previous work on the provable stabilization of DRS walking [6]. By combining these methods, we aim to achieve accurate tracking of the desired motion while accounting for the complexities of the robot's dynamics over time.

Algorithm 2 Pseudocode for updating the Bézier polynomial coefficients at time step t based on the ALIP planner result

while *True* **do**
  At time step t, obtain the current generalized coordinates $\mathbf{q}_t$ of the full-order robot. **if** *s=0* **then**
    Respectively assign the current swing foot position in the x and y directions to $\alpha_{5,0}$ and $\alpha_{6,0}$:
    $\alpha_{5,0} \leftarrow x_{sw}(\mathbf{q}_t), \quad \alpha_{6,0} \leftarrow y_{sw}(\mathbf{q}_t)$.
  end
  if $0 \leq s \leq 1$ **then**
    Convert $\mathbf{q}_t$ into the current ALIP state:
    $\mathbf{x}_t \leftarrow \mathbf{x}(\mathbf{q}_t), \quad \mathbf{y}_t \leftarrow \mathbf{y}(\mathbf{q}_t)$.
    Compute the pre-impact angular momentum about the contact point $L_{y,S}(T_k)$, $L_{x,S}(T_k)$ based on the known DRS motion and the current state $\mathbf{x}_t$ and $\mathbf{y}_t$ using Eqs. (5.18) and (5.35).
    Compute $x_{SwC}(T_k^-)$ and $y_{SwC}(T_k^-)$ respectively based on Eqs. (5.24) and (5.36).
    Assign the planned swing-foot landing locations to $\alpha_{5,M}$ and $\alpha_{6,M}$:
    $\alpha_{5,M} \leftarrow x_{SwC}(T_k^-)$, $\alpha_{6,M} \leftarrow y_{SwC}(T_k^-)$.
    Assign $\alpha_{5,1}$,..., $\alpha_{5,M-1}$ as the linear interpolation between $\alpha_{5,0}$ and $\alpha_{5,M}$,
    Assign $\alpha_{6,1}$,..., $\alpha_{6,M-1}$ as the linear interpolation between $\alpha_{6,0}$ and $\alpha_{6,M}$.
  else
    No update is applied to $\alpha_{5,0}$,..., $\alpha_{5,M}$, $\alpha_{6,0}$,..., $\alpha_{6,M}$
  end
end

5.3.3.1 Full-Order Dynamic Model

Due to the combination of continuous dynamics and discrete behaviors, such as foot landings, bipedal robotic walking on DRS can be effectively modeled as a hybrid dynamical system. In the following content, we will introduce the full-order continuous dynamics and the full-order discrete dynamics of the system.

Full-order Continuous-phase Dynamics: During the continuous phase of walking, the support foot of the Digit robot maintains full contact with the ground. However, both toe joints of the support foot are deactivated to emulate the zero ankle torque from the ALIP model in both planes. Let $\mathbf{p}_F(\mathbf{q})$ denote the position of the robot's support foot w.r.t. the world frame. Under the assumption (A5.6) that the support foot does not slip on the DRS (i.e., $\mathbf{p}_F(\mathbf{q}) = \mathbf{p}_S(t)$), the overall holonomic constraint associated with the surface-foot contact and passive joints (as described in Eq. (5.38)) can be represented as follows:

$$\underbrace{\begin{bmatrix}\mathbf{J}_F\\ \mathbf{J}_{pj}\end{bmatrix}}_{\mathbf{J}_c}\ddot{\mathbf{q}} + \underbrace{\begin{bmatrix}\dot{\mathbf{J}}_F\\ \dot{\mathbf{J}}_{pj}\end{bmatrix}}_{\dot{\mathbf{J}}_c}\dot{\mathbf{q}} = \underbrace{\begin{bmatrix}\ddot{\mathbf{p}}_S(t)\\ \mathbf{0}_{10\times 1}\end{bmatrix}}_{\ddot{\mathbf{p}}_c(t)}, \tag{5.41}$$

where $\mathbf{J}_F := \frac{\partial \mathbf{p}_F}{\partial \mathbf{q}}(\mathbf{q})$, and $\mathbf{J}_{pj} := \frac{\partial \mathbf{p}_{pj}}{\partial \mathbf{q}}(\mathbf{q})$.

By incorporating the holonomic constraint, the overall equation of motion is expressed as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{c}}(t,\mathbf{q},\dot{\mathbf{q}}) = \bar{\mathbf{B}}_j\boldsymbol{\tau}_u, \tag{5.42}$$

where $\bar{\mathbf{B}}_j$ ($j \in \{\text{left}, \text{right}\}$) is the input-selection matrix. Note that $\bar{\mathbf{B}}_{\text{left}}$ and $\bar{\mathbf{B}}_{\text{right}}$ are different because toe joints are deactivated only on the support foot. Please refer to Section 2.2 for derivation details.

Switching Surface and Impact Dynamics: When the swing foot strikes the DRS, an impact occurs, causing jumps in the generalized velocities $\dot{\mathbf{q}}$ [2]. The switching surface that determines the occurrence of the impact is given by

$$S_q := \{(\mathbf{q},\dot{\mathbf{q}}) : z_{sw}(\mathbf{q}) = 0, \dot{z}_{sw}(\mathbf{q},\dot{\mathbf{q}}) < 0\}. \tag{5.43}$$

where z_{sw} is the swing foot height.

The velocity jump can be described by a reset map:

$$\dot{\mathbf{q}}^{+} = \mathbf{R}_{\dot{\mathbf{q}}}(\mathbf{q}^{-})\dot{\mathbf{q}}^{-}, \tag{5.44}$$

where $\mathbf{R}_{\dot{\mathbf{q}}}$ is the reset map (see Section 2.2).

Unlike the fixed-time switching of the ALIP model, the full-order switching surface is state-triggered, which accurately captures the actual robot behavior.

5.3.3.2 Input-output Linearizing Control

The tracking error $\mathbf{h} \in \mathbb{R}^{18}$ is defined as the difference between the $\mathbf{h}_c$ and $\mathbf{h}_d$:

$$\mathbf{h} = \mathbf{h}_c - \mathbf{h}_d. \tag{5.45}$$

The following control law:

$$\boldsymbol{\tau}_u = (\tfrac{\partial \mathbf{h}_c}{\partial \mathbf{q}}\mathbf{M}^{-1}\bar{\mathbf{B}}_j)^{-1}[(\tfrac{\partial \mathbf{h}_c}{\partial \mathbf{q}})\mathbf{M}^{-1}\bar{\mathbf{c}} + \mathbf{v} - \tfrac{\partial}{\partial \mathbf{q}}(\tfrac{\partial \mathbf{h}_c}{\partial \mathbf{q}}\dot{\mathbf{q}})\dot{\mathbf{q}} + \tfrac{1}{T^2}\tfrac{\partial \mathbf{h}_d}{\partial s^2}], \tag{5.46}$$

is an input-output linearizing control law that drives the tracking error to zero. By applying this control law, the dynamics of the output function $\mathbf{h}$ become $\ddot{\mathbf{h}} = \mathbf{v}$, where $\mathbf{v} = -\mathbf{K}_p\mathbf{h} - \mathbf{K}_d\dot{\mathbf{h}}$. Here $\mathbf{K}_p$ and $\mathbf{K}_d$ are the proportional and derivative (PD) gain matrices. We can stabilize this linear, time-invariant system by choosing proper values of $\mathbf{K}_p$ and $\mathbf{K}_d$.

Remark 8 (Effects of time-varying Bézier coefficients): The control law described in Eq. (5.46) assumes that the Bézier coefficients remain constant within a continuous phase, which allows for the complete cancellation of the nonlinearity present in the continuous-phase dynamics described by Eq. (5.42). However, the Bézier parameters are constantly updated by the middle-layer walking pattern planner (see Algorithm 2). Despite the time-varying nature of the Bézier parameters, the controller in Eq. (5.46) can still be effective as long as the variations in the Bézier parameters are sufficiently slow.

5.3.3.3 Inverse Kinematics Control

Alternatively, we can use a inverse kinematics approach by providing joint-space references for each actuated joint. These references are obtained via a Jacobian-based inverse

kinematics method that considers the closed-loop chains on the robot's kinematic tree. For instance, one of those closed-loop chains are formed by the joints $\mathbf{q}_{15}$ through $\mathbf{q}_{18}$ as shown in Fig. 5-7).

We express the inverse kinematic problem as a first order control law $\dot{\mathbf{h}}(\mathbf{q},\dot{\mathbf{q}}) = -\kappa\mathbf{h}(\mathbf{q})$ to drive $\mathbf{h}(\mathbf{q}) \to \mathbf{0}_{18\times 1}$. Here $\kappa \in \mathbb{R}^{18\times 18}$ is a positive-definite matrix. Then, choosing the joint velocities $\dot{\mathbf{q}}^r$ as the vector to regulate, the controller can be re-written as:

$$\mathbf{J}_h(\mathbf{q})\dot{\mathbf{q}}^r = \dot{\mathbf{h}}_d - \kappa\mathbf{h}(\mathbf{q}) \tag{5.47}$$

where, $\mathbf{J}_h(\mathbf{q}) = \frac{\partial \dot{\mathbf{h}}_c}{\partial \mathbf{q}} \in \mathbb{R}^{18\times 36}$.

Since $\mathbf{J}_h(\mathbf{q})$ is not a square matrix, we consider the collection of the Jacobians $\mathbf{J}_k(\mathbf{q}) = \left[\mathbf{J}_h(\mathbf{q})^T, \mathbf{J}_F(\mathbf{q})^T\right]^T \in \mathbb{R}^{24\times 36}$. Furthermore, we split $\mathbf{q}$ into the controlled and uncontrolled joints, with their angles respectively denoted as $\mathbf{q}_c \in \mathbb{R}^{24}$ and $\mathbf{q}_u \in \mathbb{R}^{12}$. Consequently, we have $\mathbf{J}_k(\mathbf{q})\dot{\mathbf{q}} = \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}}_c + \mathbf{J}_u(\mathbf{q})\dot{\mathbf{q}}_u$, with $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{24\times 24}$ and $\mathbf{J}_u(\mathbf{q}) \in \mathbb{R}^{24\times 12}$. This leads to the inverse kinematics solution:

$$\dot{\mathbf{q}}_c^r = \mathbf{J}_c(\mathbf{q})^{-1}\left(\begin{bmatrix} -\kappa\mathbf{h}(\mathbf{q}) \\ \mathbf{0}_{4\times 1} \end{bmatrix} - \mathbf{J}_u(\mathbf{q})\dot{\mathbf{q}}_u\right) \text{ and} \tag{5.48}$$

$$\mathbf{q}_c^r = \mathbf{q}_c + \dot{\mathbf{q}}_c^r \Delta t, \tag{5.49}$$

where, $\Delta t \in \mathbb{R}$ represents the period of the control loop.

Finally, we apply a PD controller at joint-level to obtain $\boldsymbol{\tau}_u$.

5.4 Simulation Validation

This section presents the results of the simulation validation for the proposed underactuated bipedal robot model, Digit, under various surface movements. The simulations were carried out using MATLAB.

5.4.1 Simulation Setup

5.4.1.1 Initial Condition of the Digit

The initial condition of Digit is set to a regular standing pose.

5.4.1.2 Implementation of Higher-layer Planner

In the ALIP model, the following parameters are set:

- Height of the CoM $H = 0.9$ m,
- Mass $m = 46.1$ kg,
- Step duration $T_{step} = 0.4$ s,
- Step width $W = 0.2$ m.

The value of $\bar{L}_{y,S}$ is set differently for each case, as specified in Table 5.1.

5.4.1.3 Implementation of Middle-layer Planner

In the middle layer of the controller, specific coefficients of the Bézier polynomial are chosen as follows:

- ϕ_1: set to H so that the height of the CoM of the full-order Digit is close to the corresponding ALIP model.
- ϕ_2, ϕ_3, ϕ_4: set to 0 to maintain a constant orientation of the robot trunk.
- ϕ_5, ϕ_6: see Algorithm 2.
- ϕ_7: set to $[0, 0.02, 0.07, 0.15, 0.07, 0.02, 0]^T$ to minimize the desired swing foot height and reduce swing foot motion.
- ϕ_8, ϕ_9, and ϕ_{10}: set to 0 so that the swing foot is always ready for landing.
- ϕ_{11} to ϕ_{18}: set to 0 to keep both arms static during walking.

5.4.1.4 Implementation of Low-layer Planner

In MATLAB, the proposed full model based controller is implemented using Eq.(5.46). The matrix $\mathbf{M}$ and the vector $\mathbf{c}$ are obtained using FROST[124]. For all cases, the control gains are chosen as $\mathbf{K}_p = K_p\mathbf{I}$ ($K_p = 2500$) and $\mathbf{K}_d = K_d\mathbf{I}$ ($K_d = 100$), where $\mathbf{I}$ is a 18×18 identity matrix. These PD gains are selected to ensure the exponential convergence of the output function $\mathbf{h}$ within a continuous phase.

5.4.1.5 DRS Motions

The study focuses on horizontally and periodically moving surfaces, and it tests four different horizontal DRS motions. The DRS motions are described as cosine curves with distinct amplitudes and periodicities. The brief description for each case is as follows:

DRS 1: The DRS motion exists only in the sagittal plane and shares the same periodicity as the walking.

DRS 2: The DRS motion exists only in the sagittal plane, but its periodicity is 10 times that of the walking.

DRS 3: The DRS motion exists only in the lateral plane, and the periodicity of the DRS motion is related to the walking periodicity as $9T_{gait} = 10T_{y,DRS}$. Please note that the periodicity of walking in the lateral plane is twice as long as that of walking in the sagittal plane.

DRS 4: The DRS motion exists in both the sagittal and lateral planes and each direction has a unique amplitude and periodicity.

The detailed expressions of the DRS motion are listed in the Table 5.1.

5.4.1.6 Simulation Cases

The overall simulation cases are summarized in Table 5.1.

Table 5.1: Simulation Cases

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

5.4.2 Comparison Results

In this section, we performed a comprehensive performance comparison between our proposed framework, referred to as the DRS framework, and a previous framework [100], which we call the static rigid surface (SRS) framework here. The comparison focused on trajectory convergence, velocity regulation, accuracy of angular momentum prediction, and overall walking stability. Figure 5-8 illustrates the results of this comparison.

5.4.2.1 Trajectory Convergence and Overall Walking Stability Comparison

The full-body trajectories generated by the DRS framework exhibits significantly closer tracking of the desired trajectories compared to the SRS framework. This accurate tracking of the desired trajectory helps ensure the stability of the full-order model. Notably, in Case C, the DRS framework demonstrates the ability to maintain walking stability, while the SRS planner failes to do so, resulting in the robot falling over in the lateral direction.

5.4.2.2 Velocity Regulation Comparison

When the robot's upper body does not exhibit aggressive angular motion, regulating the desired angular momentum $\bar{L}_{y,S}$ is similar to regulating the linear CoM velocity. Table 5.2 presents a performance comparison between the two frameworks. It is evident that the DRS framework outperforms the SRS one in terms of linear velocity regulation.

5.4.2.3 Angular Momentum Prediction Comparison

Compared with the SRS framework, the DRS approach demonstrates higher accuracy in predicting the angular momentum at the end of the current step, as depicted by the yellow

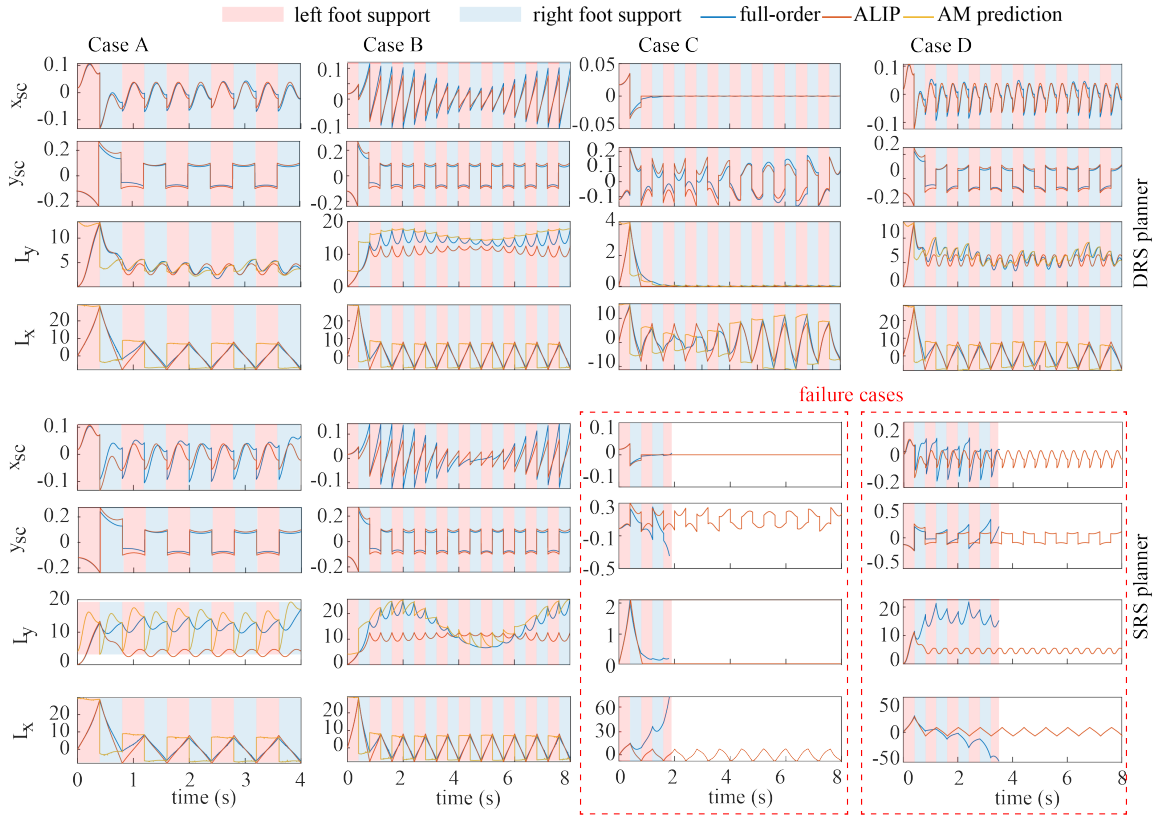

Figure 5-8: Performance comparison between the proposed DRS framework with the previous SRS framework. The pink background indicates the left foot support phase and the blue background indicates the right foot support phase. As we can observe, within all cases, the full-body trajectory generated by our proposed framework is much closer to the desired ALIP trajectory. Also, the angular momentum prediction at the end of the current step (yellow curve) shows much higher accuracy even in the presence of the DRS motion. In contrast, the SRS framework performs poorer with two failure cases.

Table 5.2: Linear velocity regulation comparison

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

curve in Fig. 5-8. This improved prediction accuracy leads to less variation in the planned stepping location for the swing foot during each step. Consequently, there is a reduced burden on the lower-layer controller as the coefficients of the Bézier polynomial exhibit less variation, resulting in better tracking performance.

5.4.2.4 Global-position Tracking Comparison

By adjusting the desired angular momentum $\bar{L}_{y,S}$ and $\bar{L}_{x,S}$, our framework allows the robot to track a planned global position trajectory on DRS. This capability is particularly useful in real-world applications where DRS environments, such as ships or ferries, often have narrow walkways. By closely following the planned global position trajectory, the robot can minimize the risk of colliding with walls or other individuals, thereby enhancing safety and preventing potential accidents. $\bar{L}_{y,S}$ and $\bar{L}_{x,S}$ are adjusted by the following expressions:

$$\begin{aligned} \bar{L}_{y,S} &= K_x^{GPT}(x_{DRS}^{ref} - p_{b,x}^{DRS}) + mHv_{x,DRS}^{ref} \\ \bar{L}_{x,S} &= K_y^{GPT}(y_{DRS}^{ref} - p_{b,y}^{DRS}) + mHv_{y,DRS}^{ref} \end{aligned} \tag{5.50}$$

where x_{DRS}^{ref} and y_{DRS}^{ref} are the desired position w.r.t. the $\{DRS\}$ frame. $p_{b,x}^{DRS}$ and $p_{b,y}^{DRS}$ are the actual position w.r.t. the $\{DRS\}$ frame. $v_{x,DRS}^{ref}$ and $v_{y,DRS}^{ref}$ are the desired linear velocity of the global position trajectory. K_x^{GPT} and K_y^{GPT} are gains that can be assigned by the user.

In Figure 5-9, we set up an simulated situation where Digit navigates through a narrow pathway inside a ship with the same motion as Case D. The proposed DRS framework enables the Digit to closely follow the desired walking path while maintaining the desired forward speed. In contrast, the SRS approach shows limitations in consistently staying close to the desired path. This becomes problematic in narrow pathways, as it increases the risk of

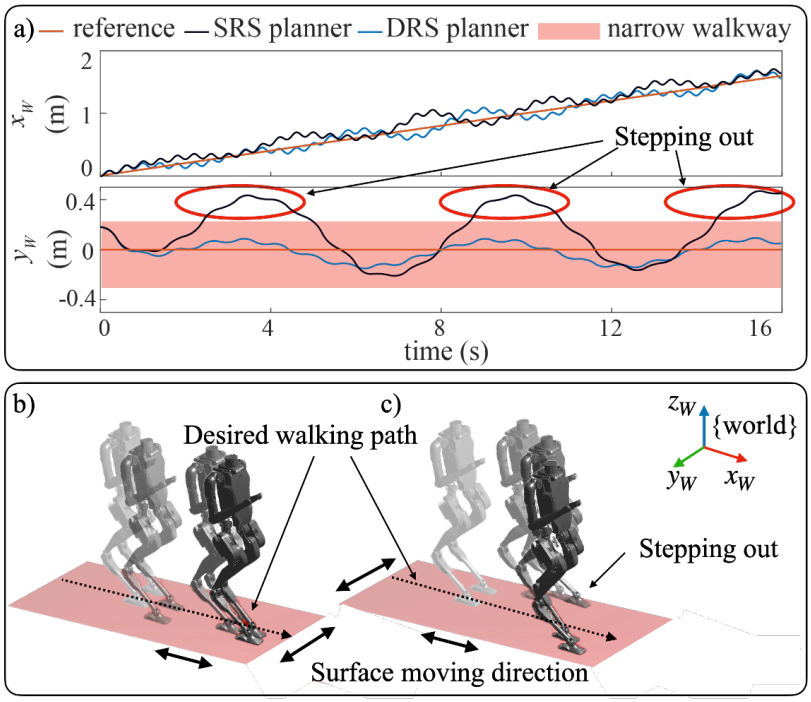

Figure 5-9: Comparison of global position tracking capability between the SRS and the DRS planner. a) Plots show superior tracking performance of the DRS planner in the forward direction, with the Digit staying within the designated walking area. b) Digit walking on a narrow DRS walkway under the DRS planner. c) Digit walking on the same walkway under the SRS planner, resulting in stepping outside the walking surface.

collisions with walls or obstacles.

5.4.3 Robustness

We investigate the robustness of our proposed work by applying sudden pushes and an unknown load to the robot.

5.4.3.1 Sudden Push

A sudden horizontal force of 200 N is applied to the Digit robot during its walking under Case A. As shown in Figure 5-10, the Digit is initially heavily affected by this unexpected push. However, after a few steps, the Digit is able to regain stability and maintain its walking on the DRS. This demonstrates the robustness of our proposed framework, as it can effectively handle and recover from external disturbances, such as sudden pushes, which are commonly encountered in real-world applications.

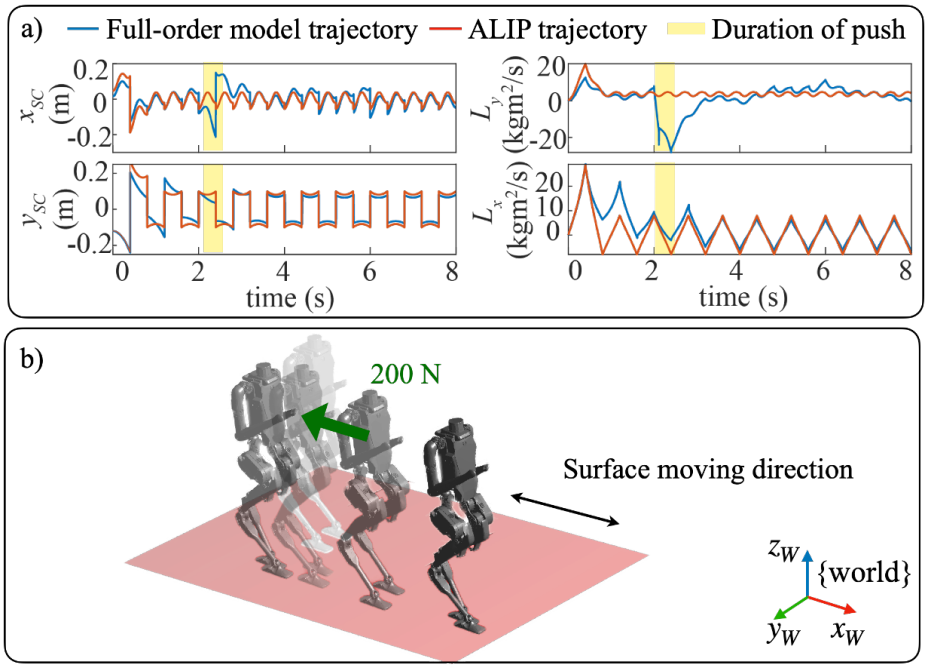

Figure 5-10: Simulation results of Case A showing the response of the Digit robot under an unknwon sudden push lasting 0.1 second. a) Time evolution of the CoM position and the angular momentum trajectories. b) Time lapse images of a walking Digit robot during push recovery. The plots demonstrate that the robot maintains stability and effectively regulates its desired forward speed after the push.

5.4.3.2 Unknown Load

Figure 5-11 showcases the performance of our proposed framework under Case C when the robot is carrying a 10 kg box, whose weight is unknown to the planner. In Case C, the intended behavior was for the robot to walk in place. However, due to the presence of the external load, the robot exhibits a forward drift during the walking process. Despite this external disturbance, our framework demonstrates robustness, as the walking motion remains stable throughout the simulation. This further highlights the effectiveness of our proposed framework in accommodating and adapting to unknown external disturbances.

5.5 Experiment Validation

This section presents the results from the experimental validation of the proposed DRS framework on a physical Digit robot. Due to hardware constraints, such as the limited region

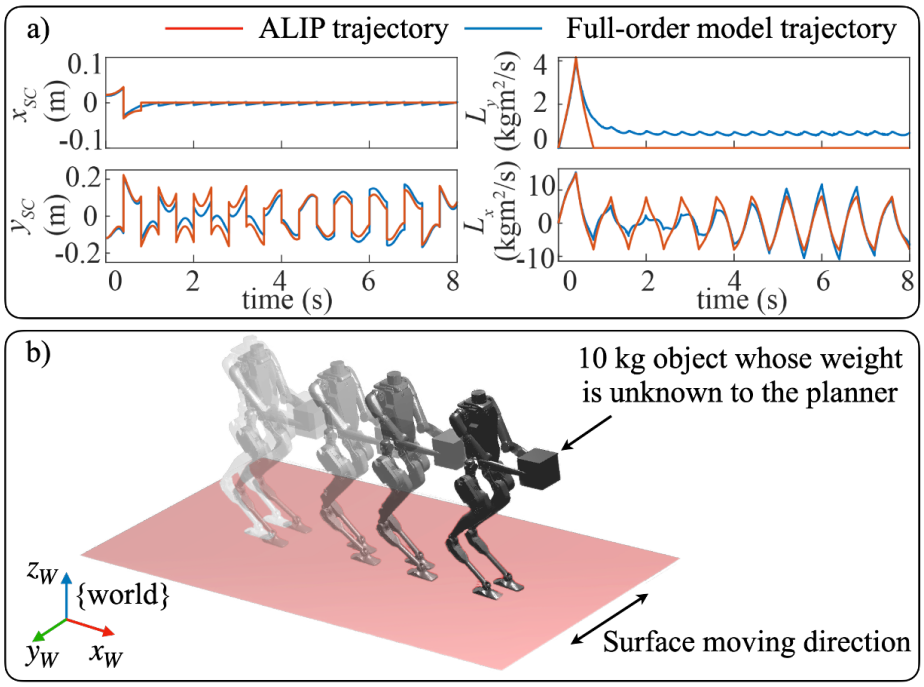

Figure 5-11: Simulation results of Case C were obtained when the Digit robot was carrying a 10 kg box. The weight of the box was unknown to the planner. The results are presented in the following two figures. a) shows the trajectory evolution along the time axis. b) illustrates the walking motion of the robot. It is evident that the robot maintains its stability throughout the walking process, even with the presence of the external load causing forward drift.

and motion capability of the tested DRS, it is not possible to implement all the simulation scenarios presented earlier. Instead, the main objective of the experiments is to validate the validity and effectiveness of our proposed framework in the real world.

In this experiment, a modified split-belt Motek M-gait treadmill is used as the DRS (Fig. 5-12)). Unlike in simulations, the robot's states are not completely known during experiments, introducing uncertainties stemming from incomplete state information. Additionally, during the experiments, the Digit robot exhibits position position drifting, which could be attributed to modeling errors and hardware imperfections.

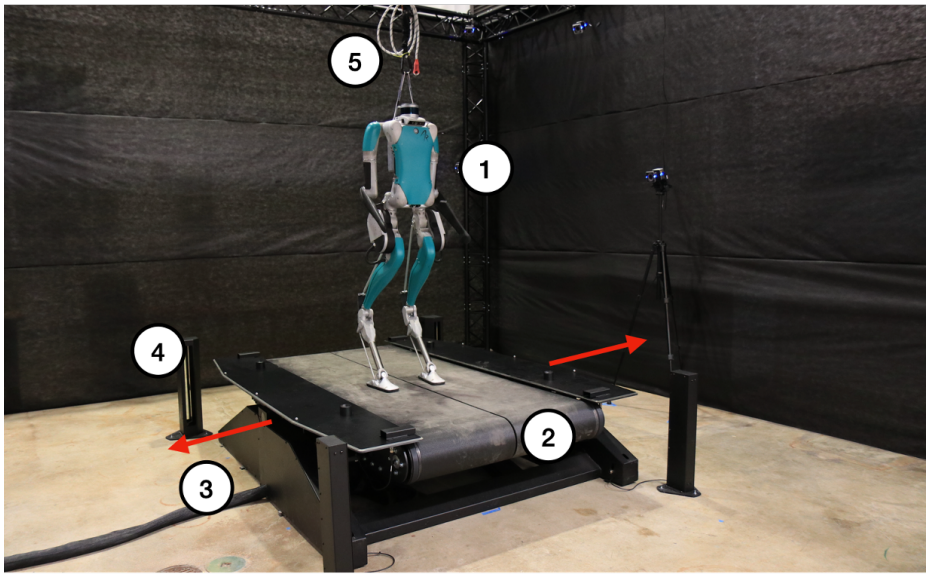

Figure 5-12: Experiment setup with the following elements: 1) a Digit robot, 2) a split-belt Motek M-gait treadmill (i.e., DRS), 3) the moving axis of the treadmill, 4) a laser safety guard, 5) a safety harness.

5.5.1 Experiment Setup

5.5.1.1 DRS motions

The DRS motion in the experiment will be less aggressive compared to the simulations due to the limited motion capabilities of the split-belt Motek M-gait treadmill. It is important to note that the treadmill can only generate horizontal motion along a single axis. As a result, we will focus on examining a specific type of DRS motion: lateral plane motion with a periodicity multiple times that of the gait (i.e., $T_{y,DRS} = N_1 T_{gait}$).

5.5.1.2 Experiment Cases

The experiment cases are listed in Table 5.3. To confine the robot's movement within

Table 5.3: Experiment Cases

Cases	DRS expressions (m)	$\bar{L}_{y,S}$ (kgm^2/s)
Case A	$y_S(t) = 0.04cos(\frac{2\pi}{6.8}t)$	0
Case B	$y_S(t) = 0.03cos(\frac{2\pi}{5.6}t)$	0
Case C	$y_S(t) = 0.04cos(\frac{2\pi}{5.6}t)$	0

the limited space of the treadmill, $\bar{L}_{y,S}$ is set to 0, resulting in the robot walking in place.

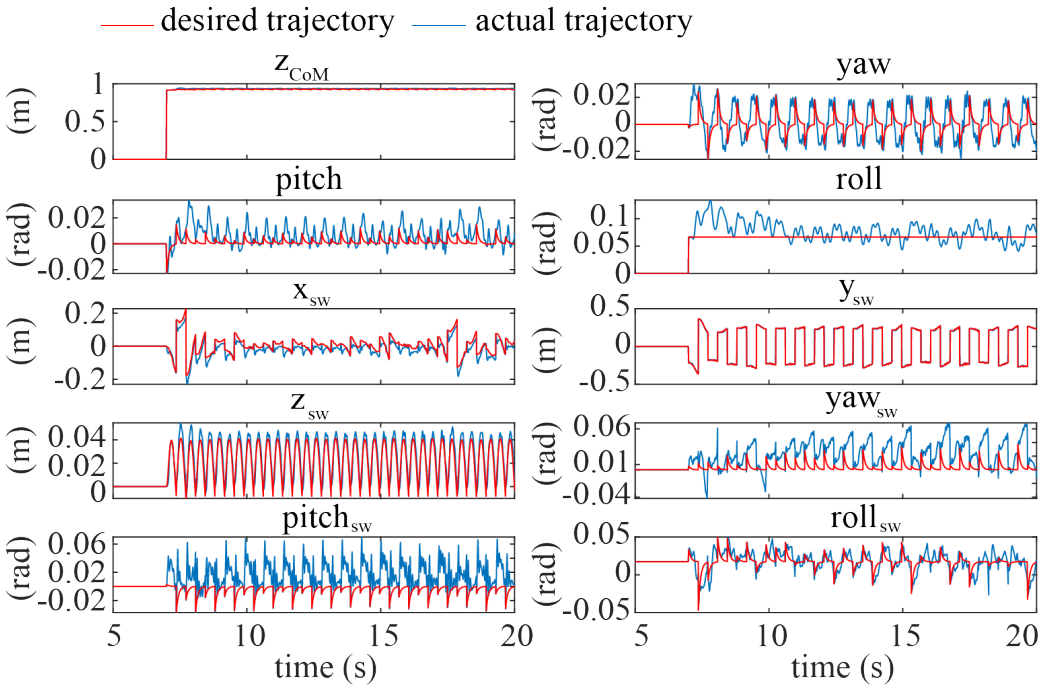

Figure 5-13: Experimental results showing the task space tracking performance under Case B. The results reveal accurate CoM height tracking, facilitating the close match between the full-order robot model and the ALIP model. Additionally, the position tracking of the swing foot is satisfactory, ensuring accurate placement at the intended destination. While mild fluctuations are observed in the orientation tracking of both the base and swing foot, their small magnitude prevents them from compromising the overall walking performance of Digit.

5.5.1.3 State Estimation

In this experiment, we utilize Digit's default state estimator since the DRS motion was relatively mild. To assess its performance, we conduct validation against the ground truth data obtained from a motion capture system. The validation results indicate that the default state estimator is reasonably accurate for the experiment.

5.5.1.4 Implementation of the Hierarchical Control Approach

The proposed three-layer hierarchical control approach, known as the DRS framework, was applied to hardware using C++ within the ROS platform. Notably, instead of employing input-output linearizing control, inverse-kinematics control was utilized. This choice was made because inverse-kinematics control is less impacted by model uncertainty, a common

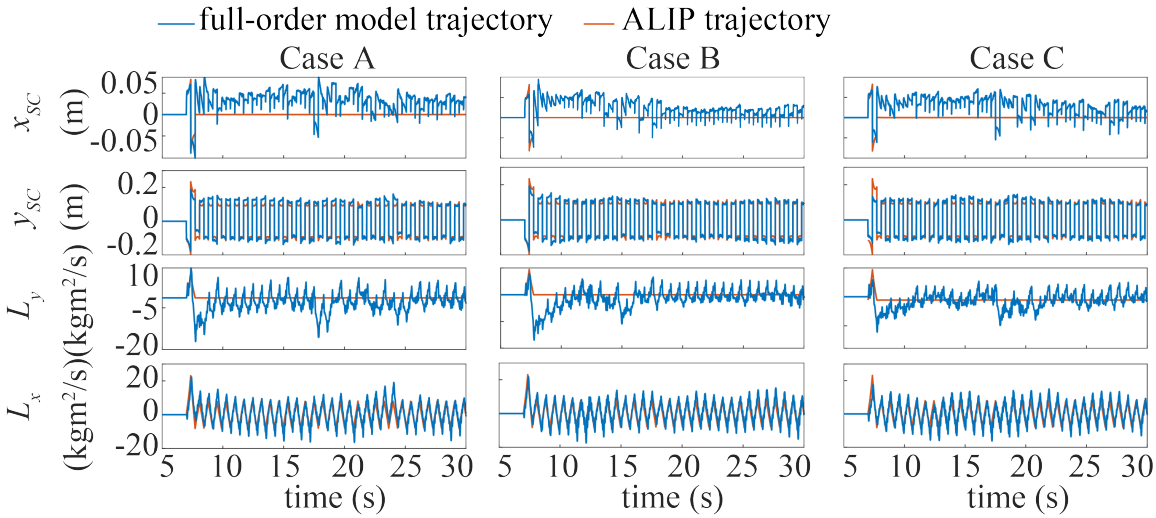

Figure 5-14: Experimental trajectory tracking performance across Cases A-C. The desired ALIP trajectories of x_{SC} and L_y feature a level zero line due to a desired forward speed of 0, preventing the Digit from stepping out of the confined testing area on the treadmill. Examining y_{SC} and L_x, it becomes evident that the Digit effectively adheres to its desired trajectory, thereby ensuring its stable locomotion.

issue shared between the theoretical model and the actual system.

5.5.2 Experiment Results

Trajectory Tracking: Figure 5-14 confirms the stable walking of the Digit robot on the treadmill, demonstrating the successful deployment of our proposed framework on hardware without extensive parameter tuning. The performance of full-body trajectory tracking in the task space is shown in Fig. 5-13. The height of the center of mass (CoM) in the experiment exhibits slight oscillations around the desired height of 0.9 m, indicating close tracking. The swing foot position tracking performs exceptionally well in all directions. Although the swing foot orientation tracking is not as accurate, the robot maintains stable walking on the DRS.

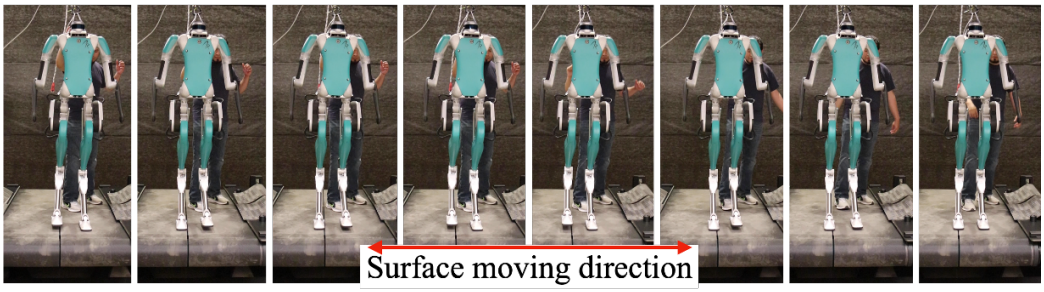

Figure 5-15: Time-lapse figures of Digit walking on a horizontally moving treadmill under case B.

5.6 Discussion

This section discusses the limitations of our approach despite its effectiveness demonstrated through simulations and experiments. The limitations stem from both the proposed theoretical derivation and experimental validation.

5.6.1 Limitations of the ALIP Model

One limitation of our approach lies in its assumptions, including that the DRS motion is confined to the horizontal direction, the walking surface is flat, and the CoM height is constant. However, in real-world environments such as ships and airplanes, vertical vehicle motions and stairs or uneven terrain inside the vehicles are prevalent. Neglecting these factors may limit the general applicability of our approach. To overcome this limitation, we will study controller designs for both vertical [125, 126, 127, 128, 129, 130, 131] and horizontal DRS movements, as well as variations in surface height.

Furthermore, our approach assumes an accurate prior knowledge of the DRS motion, which may not be realistic in real-world environments. One promising way to address this issue is to estimate the DRS motion, allowing the robot system to dynamically adapt to the varying DRS motion.

5.6.2 Limitations of the Experimental Validation

While the simulations extensively evaluate the performance of our planner in various scenarios along with a baseline framework that does not consider the DRS motion, it is

important to note that the experiments primarily focus on assessing the walking performance on a DRS with a relatively small surface and limited range of motion, which limits the disturbances, such as unknown weight loads, that can be safely tested on the robot.

To ensure the safety of the experiment, we made the decision to restrict the robot's movement to walking in place within the limited walking area of the DRS. This cautious approach was necessary to prevent the robot from stepping out of the DRS and potentially causing collisions or accidents. Consequently, we were unable to fully assess the speed regulation performance of our planner, as it required the robot to achieve forward, backward, or sideways movement. However, it is important to highlight that our experimental setup was conducted in a controlled laboratory environment, which provided valuable insights into the robot's walking performance on the DRS.

To further enhance the evaluation of our approach, future experimental tests could be arranged in more practical settings, such as on a ship or mobile offshore platform. This would provide a more realistic and dynamic environment, allowing for comprehensive assessments of the robot's reaction to disturbances, evaluation of speed regulation performance, and exploration of its robustness in the presence of uncertainties.

Overall, while our experiments were limited in scope, they have provided valuable insights into the walking performance of the robot on the DRS and highlighted the potential for future investigations in real-world applications.

5.7 Summary

- A new control approach for achieving stable walking of underactuated bipedal robots on a horizontally oscillating DRS is presented.
- A discrete feedback control strategy is synthesized to stabilize the periodic solution of the hybrid ALIP model, addressing the instability and uncontrollability of the continuous-phase ALIP dynamics.
- A hierarchical control approach is developed to stabilize the unactuated robot dynamics based on the hybrid ALIP model and its footstep controller.

- Stability analyses are conducted for the reduced ALIP model.
- Simulations, as well as experiments, demonstrate the effectiveness of the proposed framework in stabilizing underactuated walking on a swaying DRS under various cases.
- We have published one conference paper on this topic [95]. We are also in the process of preparing a journal article for submission.

Chapter 6 Conclusion and Future Work

6.1 Conclusions

In conclusion, this dissertation has delved into the fascinating and challenging realm of bipedal locomotion, addressing critical aspects, such as motion control and state estimation, of legged robotics in diverse real-world scenarios. We have ventured into three core domains: global-position tracking (GPT) control, state estimation for locomotion on dynamic rigid surfaces (DRSes), and linear inverted pendulum (LIP) based locomotion control for underactuated bipedal walking on a DRS.

Firstly, we developed a continuous tracking control law that guarantees accurate global-position tracking for multi-domain bipedal robotic walking. This control law, based on input-output linearization and proportional-derivative control, ensures exponential stability within each continuous phase of the hybrid walking process. Stability conditions were established using multiple Lyapunov functions for control gain tuning. We investigated three-domain and two-domain gaits to demonstrate the approach's effectiveness. Additionally, the control law was cast into a quadratic program (QP) to handle the actuator torque saturation and friction cone constraint that are commonly encountered in real-world applications. Simulations on a three-dimensional (3-D) bipedal humanoid robot validated the method across various scenarios. Finally, we compared the performance of the input-output linearizing control law with and without the QP formulation to emphasize its effectiveness in mitigating torque saturation while maintaining stability and trajectory tracking accuracy.

Secondly, we turned our attention to the challenge of state estimation on a DRS. DRSes, such as ships and aircraft, present unique complexities due to the non-zero surface-foot contact region velocity. This led to the development of an invariant extended Kalman filter tailored for bipedal humanoid locomotion on moving surfaces. The filter was explicitly designed to account for known surface motion and hybrid robot behaviors while maintaining key advantages such as satisfying the attractive group-affine condition and invariant observa-

tion form in the absence of the sensor biases of the inertial measurement unit. Observability analysis confirmed the observability of base velocity, roll, and pitch angles during continuous locomotion phases, with base yaw angle observable when the DRS is not horizontal. Stability analysis demonstrated asymptotic error convergence for these observable states for the hybrid deterministic system. Experimental results with humanoid walking on a pitching treadmill verified the filter's improved accuracy and convergence rate compared to existing methods, even under substantial estimation errors and moderate DRS motion.

Lastly, we introduced a control approach for stable underactuated bipedal walking on horizontally oscillating DRSes, involving an analytically derived angular momentum-based linear inverted pendulum (ALIP) model that accounts for the DRS's time-varying motion and hybrid robot behaviors. A discrete feedback control strategy was devised to stabilize the periodic solution of the hybrid ALIP model. We developed a hierarchical control strategy for stabilizing unactuated robot dynamics based on the hybrid ALIP model and its footstep controller. Stability analyses were conducted for the reduced ALIP model. Simulations with a 3-D Digit robot demonstrated the framework's efficacy in stabilizing underactuated walking on a swaying DRS across various scenarios, with hardware experiments further confirming practical viability.

In essence, this dissertation has pushed the boundaries of bipedal leg locomotion by addressing key challenges in global position tracking, state estimation on DRS, and locomotion control on DRS. These contributions represent significant progress in legged robotics, offering potential applications in dynamic real-world scenarios. The research provides foundational knowledge for the development of versatile legged robots suitable for diverse environments.

6.2 Future Work

While this dissertation research has made substantial progress on bipedal locomotion control, several exciting directions for future work emerge.

Firstly, concerning global-position tracking, one limitation of the current approach is the potential non-feasibility of meeting the proposed stability conditions if the duration of the

underactuation phase becomes overly large. This limitation necessitates a careful choice of the nominal duration of the UA domain, ensuring that it does not extend too long for practical implementation. Moreover, to enhance the controller's robustness for real-world applications, it is imperative to address model parametric errors, external disturbances, and hardware imperfections, such as sensor noise. Incorporating robust control techniques into the global-position tracking control law can effectively tackle these uncertainties. Additionally, the integration of online footstep planning offers the potential to dynamically adjust the robot's behavior in real-time, enhancing its ability to cope with modeling errors and external disturbances. These combined enhancements will make the controller more reliable and adaptable in complex environments.

Secondly, in the territory of DRS state estimation, the current filter design is well-suited for scenarios where the surface pose is reasonably accurately known. However, it may not perform as effectively in situations with highly inaccurate or entirely unknown surface profiles. A promising avenue for future work is to extend the filter's capabilities to estimate the surface pose. This can be achieved by introducing a matrix Lie group that incorporates surface pose information into the state estimation process. Such an extension will not only enhance the filter's applicability in more challenging environments but also contribute to a broader understanding of state estimation on dynamic rigid surfaces.

Lastly, in the domain of LIP-based planning, significant strides have been made in developing a planning and control framework for walking on horizontally oscillating DRS. However, there are promising avenues for further research and improvement. Firstly, optimizing the coefficients for full-body walking parameters holds the potential to significantly impact overall walking performance. While initial coefficient selections within the middle layer were made based on intuition and heuristics, a more systematic fine-tuning and optimization process can provide a deeper understanding of how these coefficients affect factors, such as energy consumption, landing impact, and robustness against external disturbances. Secondly, exploring the utilization of ankle joints in underactuated robots presents an intriguing research area. Enabling controlled ankle movements could enhance the robot's ability to recover from external disturbances and improve stability, potentially lead to better balance maintenance. Lastly, the integration of locomotion with manipulation tasks on DRS

is a promising direction. Coordinating locomotion and manipulation actions requires developing control strategies and addressing challenges in perception, planning, and coordination. However, this fusion can lead to versatile and robust robotic systems capable of complex tasks in dynamic settings. This integration is particularly relevant in scenarios such as search and rescue, manufacturing, and maintenance, and inspection as the execution of these tasks naturally demands manipulation. Exploring these directions will undoubtedly contribute to the development of more practical, adaptable, and capable bipedal robots.

References

[1] "ROBOTIS Co., Ltd.." https://www.robotis.us/. Accessed: 2023-01-20.

[2] J. W. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE T AUTOMAT CONTR*, vol. 46, no. 1, pp. 51–64, 2001.

[3] Y. Gu, B. Yao, and C. G. Lee, "Time-dependent orbital stabilization of underactuated bipedal walking," in *Proc. of American Control Conference*, pp. 4858–4863, 2017.

[4] Y. Gu, B. Yao, and C. G. Lee, "Feasible center of mass dynamic manipulability of humanoid robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5082–5087, 2015.

[5] Y. Gao, C. Yuan, and Y. Gu, "Invariant extended Kalman filtering for hybrid models of bipedal robot walking," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 290–297, 2021. Modeling, Estimation and Control Conference MECC 2021.

[6] A. Iqbal, Y. Gao, and Y. Gu, "Provably stabilizing controllers for quadrupedal robot locomotion on dynamic rigid platforms," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 2035–2044, 2020.

[7] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, 2010.

[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. of IEEE/RSJ International Conference on Intelligent Robots and Systems.*, vol. 1, pp. 239–246, 2001.

[9] 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.

[10] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11, pp. 1217–1227, 1989.

[11] W. J. Schwind, *Spring loaded inverted pendulum running: A plant model*. University of Michigan, 1998.

[12] B. Han, H. Yi, Z. Xu, X. Yang, and X. Luo, "3d-slip model based dynamic stability strategy for legged robots with impact disturbance rejection," *Scientific Reports*, vol. 12, no. 1, p. 5892, 2022.

[13] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[14] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Proc. IEEE-RAS International Conference on Humanoid Robots*, pp. 295–302, 2014.

[15] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *Proceedings 2003 ieee/rsj international conference on intelligent robots and systems (iros 2003)(cat. no. 03ch37453)*, vol. 2, pp. 1644–1650, IEEE, 2003.

[16] Z. Xie, X. Da, B. Babich, A. Garg, and M. v. de Panne, "Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," in *International Workshop on the Algorithmic Foundations of Robotics*, pp. 523–539, Springer, 2022.

[17] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization," in *Proc. of IEEE-RAS International Conference on Humanoid Robots*, pp. 936–940, 2015.

[18] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[19] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[20] F. Dellaert, M. Kaess, *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[21] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," tech. rep., Georgia Institute of Technology, 2012.

[22] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems [j]," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[23] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[24] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE T AUTOMAT CONTR*, vol. 62, no. 4, pp. 1797–1812, 2017.

[25] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent ekf slam estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[26] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–193, International Society for Optics and Photonics, 1997.

[27] N. Y. Ko, W. Youn, I. H. Choi, G. Song, and T. S. Kim, "Features of invariant extended kalman filter applied to unmanned aerial vehicle navigation," *Sensors*, vol. 18, no. 9, p. 2855, 2018.

[28] E. R. Potokar, K. Norman, and J. G. Mangelson, "Invariant extended kalman filtering for underwater navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5792–5799, 2021.

[29] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *Int J Rob Res*, vol. 39, no. 4, pp. 402–430, 2020.

[30] S. Teng, M. W. Mueller, and K. Sreenath, "Legged robot state estimation in slippery environments using invariant extended kalman filter with velocity update," *arXiv preprint arXiv:2104.04238*, 2021.

[31] G. Roston and E. Krotkov, "Dead reckoning navigation for walking robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 607–612, 1992.

[32] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, 2013.

[33] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2245–2252, 2018.

[34] S. Yang, H. Kumar, Z. Gu, X. Zhang, M. Travers, and H. Choset, "State estimation for legged robots using contact-centric leg odometry," *arXiv preprint arXiv:1911.05176*, 2019.

[35] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, "Legged robot state estimation using invariant kalman filtering and learned contact events," *arXiv preprint arXiv:2106.15713*, 2021.

[36] M. J. Powell, A. Hereid, and A. D. Ames, "Speed regulation in 3D robotic walking through motion transitions between human-inspired partial hybrid zero dynamics," in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 4803–4810, 2013.

[37] M. Vukobratović and B. Borovac, "Zero-Moment Point: thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[38] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa, "Zmp-based biped running control," *IEEE Robotics Automation Magazine*, vol. 14, pp. 63–72, June 2007.

[39] "Softbank robotics." https://www.softbankrobotics.com. Accessed: 2022-09-20.

[40] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[41] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[42] O. E. Ramos and K. Hauser, "Generalizations of the capture point to nonlinear center of mass paths and uneven terrain," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 851–858, 2015.

[43] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based pattern generation for walking with variable height," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 517–536, 2020.

[44] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based pattern generation for walking with variable height," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 517–536, 2019.

[45] R. W. Sinnet, M. J. Powell, R. P. Shah, and A. D. Ames, "A human-inspired hybrid control approach to bipedal robotic walking," *Proc. of IFAC*, vol. 44, no. 1, pp. 6904–6911, 2011.

[46] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1447–1454, 2016.

[47] M. J. Powell, W.-L. Ma, E. R. Ambrose, and A. D. Ames, "Mechanics-based design of underactuated robotic walking gaits: Initial experimental realization," in *Proc. of IEEE-RAS International Conference on Humanoid Robots*, pp. 981–986, 2016.

[48] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames, "Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control," in *Proc. of ACM International Conference on Hybrid Systems: Computation and Control*, pp. 263–272, 2014.

[49] H. Zhao, W. Ma, A. D. Ames, and M. B. Zeagler, "Human-inspired multi-contact locomotion with amber2," in *Proc. of ACM International Conference on Cyber-Physical Systems (ICCPS)*, pp. 199–210, 2014.

[50] 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," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[51] A. E. Martin and R. D. Gregg, "Stable, robust hybrid zero dynamics control of powered lower-limb prostheses," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3930–3942, 2017.

[52] X. Da, R. Hartley, and J. W. Grizzle, "Supervised learning for stabilizing underactuated bipedal robot locomotion, with outdoor experiments on the wave field," in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 3476–3483, 2017.

[53] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Hybrid zero dynamics inspired feedback control policy design for 3d bipedal locomotion using reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8746–8752, IEEE, 2020.

[54] J. Siekmann, S. Valluri, J. Dao, L. Bermillo, H. Duan, A. Fern, and J. Hurst, "Learning memory-based control for human-scale bipedal locomotion," *arXiv preprint arXiv:2006.02402*, 2020.

[55] 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 *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2811–2817, IEEE, 2021.

[56] 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 *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5136–5143, IEEE, 2021.

[57] S. J. Gershman, "Reinforcement learning and causal models," *The Oxford handbook of causal reasoning*, vol. 1, p. 295, 2017.

[58] H. Gao, R. Zhou, M. Tomizuka, and Z. Xu, "Reinforcement learning based online parameter adaptation for model predictive tracking control under slippery condition," in *2022 American Control Conference (ACC)*, pp. 2675–2682, IEEE, 2022.

[59] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Data-driven latent space representation for robust bipedal locomotion learning," *arXiv preprint arXiv:2309.15740*, 2023.

[60] Y. Gao and Y. Gu, "Global-position tracking control of a fully actuated NAO bipedal walking robot," in *Proc. of American Control Conference*, pp. 4596–4601, 2019.

[61] J. Stillwell, *Naive lie theory*. Springer Science & Business Media, 2008.

[62] H. Zhao, A. Hereid, W.-l. Ma, and A. D. Ames, "Multi-contact bipedal robotic locomotion," *Robotica*, vol. 35, no. 5, pp. 1072–1106, 2017.

[63] H.-H. Zhao, W.-L. Ma, A. D. Ames, and M. B. Zeagler, "Human-inspired multi-contact locomotion with amber2," in *Proc. of ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 199–210, 2014.

[64] M. Dai, J. Lee, and A. D. Ames, "Multi-domain walking with reduced-order models of locomotion," *arXiv preprint arXiv:2310.03179*, 2023.

[65] M. Tucker, K. Li, and A. D. Ames, "Synthesizing robust walking gaits via discrete-time barrier functions with application to multi-contact exoskeleton locomotion," *arXiv preprint arXiv:2310.06169*, 2023.

[66] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[67] E. R. Westervelt, C. Chevallereau, J. H. Choi, B. Morris, and J. W. Grizzle, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.

[68] 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," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[69] S. Veer, I. Poulakakis, *et al.*, "Input-to-state stability of periodic orbits of systems with impulse effects via poincaré analysis," *IEEE Transactions Automatic Control*, vol. 64, no. 11, pp. 4583–4598, 2019.

[70] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *Proc. of American Control Conference*, pp. 4559–4566, 2019.

[71] Y. Gu, B. Yao, and C. G. Lee, "Bipedal gait recharacterization and walking encoding generalization for stable dynamic walking," in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1788–1793, 2016.

[72] Y. Gu, *Time-dependent nonlinear control of bipedal robotic walking*. PhD thesis, Purdue University, 2017.

[73] Y. Gu, B. Yao, and C. G. Lee, "Straight-line contouring control of fully actuated 3-D bipedal robotic walking," in *Proc. of American Control Conference*, pp. 2108–2113, 2018.

[74] Y. Gu and C. Yuan, "Adaptive robust tracking control for hybrid models of three-dimensional bipedal robotic walking under uncertainties," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 143, no. 8, 2021.

[75] Y. Gu, Y. Gao, B. Yao, and C. G. Lee, "Global-position tracking control for three-dimensional bipedal robots via virtual constraint design and multiple Lyapunov analysis," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 11, p. 111001, 2022.

[76] Y. Gu, B. Yao, and C. George Lee, "Exponential stabilization of fully actuated planar bipedal robotic walking with global position tracking capabilities," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 5, p. 051008, 2018.

[77] Y. Zhao and Y. Gu, "A non-periodic planning and control framework of dynamic legged locomotion," *International Journal of Intelligent Robotics and Applications*, vol. 4, pp. 95–108, 2020.

[78] Y. Gu and C. Yuan, "Adaptive robust trajectory tracking control of fully actuated bipedal robotic walking," in *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1310–1315, 2020.

[79] A. Iqbal and Y. Gu, "Extended capture point and optimization-based control for quadrupedal robot walking on dynamic rigid surfaces," *IFAC-PapersOnLine*, vol. 54, no. 20, 2021.

[80] A. Iqbal, S. Veer, and Y. Gu, "Real-time walking pattern generation of quadrupedal dynamic-surface locomotion based on a linear time-varying pendulum model," *arXiv preprint arXiv:2301.03097*, 2023.

[81] H. K. Khalil, *Noninear systems*. No. 5, Prentice Hall, 1996.

[82] W. K. Chan, Y. Gu, and B. Yao, "Optimization of output functions with nonholonomic virtual constraints in underactuated bipedal walking control," in *Proc. of Annual American Control Conference*, pp. 6743–6748, 2018.

[83] M. S. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.

[84] H. K. Khalil, *Nonlinear control*. Prentice Hall, 1996.

[85] M. Rijnen, J. B. Biemond, N. Van De Wouw, A. Saccon, and H. Nijmeijer, "Hybrid systems with state-triggered jumps: Sensitivity-based stability analysis with application to trajectory tracking," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4568–4583, 2019.

[86] M. Rijnen, A. van Rijn, H. Dallali, A. Saccon, and H. Nijmeijer, "Hybrid trajectory tracking for a hopping robotic leg," *IFAC-PapersOnLine*, vol. 49, no. 14, pp. 107–112, 2016.

[87] Y. Gong and J. Grizzle, "Angular momentum about the contact point for control of bipedal locomotion: Validation in a LIP-based controller," *arXiv preprint arXiv:2008.10763*, 2020.

[88] J. P. Reher, A. Hereid, S. Kolathaya, C. M. Hubicki, and A. D. Ames, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS," in *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*, pp. 400–415, Springer, 2020.

[89] K. A. Hamed, W.-L. Ma, and A. D. Ames, "Dynamically stable 3D quadrupedal walking with multi-domain hybrid system models and virtual constraint controllers," in *Proc. of American Control Conference*, pp. 4588–4595, 2019.

[90] M. Yeatman, G. Lv, and R. D. Gregg, "Decentralized passivity-based control with a generalized energy storage function for robust biped locomotion," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 10, 2019.

[91] C. Hu, B. Yao, Q. Wang, Z. Chen, and C. Li, "Experimental investigation on high-performance coordinated motion control of high-speed biaxial systems for contouring

tasks," *International Journal of Machine Tools and Manufacture*, vol. 51, no. 9, pp. 677–686, 2011.

[92] J. Liao, Z. Chen, and B. Yao, "High-performance adaptive robust control with balanced torque allocation for the over-actuated cutter-head driving system in tunnel boring machine," *Mechatronics*, vol. 46, pp. 168–176, 2017.

[93] M. Yuan, Z. Chen, B. Yao, and X. Liu, "Fast and accurate motion tracking of a linear motor system under kinematic and dynamic constraints: an integrated planning and control approach," *IEEE Transactions on Control System Technology*, 2019.

[94] C. Yuan, Y. Gu, and W. Zeng, "Hybrid control of switched LFT uncertain systems with time-varying input delays," *International Journal of Control*, vol. 95, no. 12, pp. 3437–3448, 2022.

[95] 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. American Control Conference*, pp. 3282–3287, 2023.

[96] A. Iqbal, S. Veer, and Y. Gu, "Asymptotic stabilization of aperiodic trajectories of a hybrid-linear inverted pendulum walking on a dynamic rigid surface," in *Proc. of American Control Conference, to appear*, 2023.

[97] M. Dai, X. Xiong, and A. Ames, "Bipedal walking on constrained footholds: Momentum regulation via vertical com control," in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 10435–10441, 2022.

[98] X. Xiong and A. Ames, "3-D underactuated bipedal walking via H-LIP based gait synthesis and stepping stabilization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2405–2425, 2022.

[99] Q. Nguyen, X. Da, J. Grizzle, and K. Sreenath, "Dynamic walking on stepping stones with gait library and control barrier functions," in *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*, pp. 384–399, 2020.

[100] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 12, p. 121006, 2022.

[101] Y. Gao and Y. Gu, "Global-position tracking control of multi-domain planar bipedal robotic walking," in *Prof. of ASME Dynamic Systems and Control Conference*, vol. 59148, p. V001T03A009, 2019.

[102] Y. Gao, K. Barhydt, C. Niezrecki, and Y. Gu, "Provably stabilizing global-position tracking control for hybrid models of multi-domain bipedal walking via multiple lyapunov analysis," *arXiv preprint arXiv:2304.13943*, 2023.

[103] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Trans. Rob.*, vol. 22, no. 5, pp. 932–943, 2006.

[104] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and Lidar sensing," in *Proc. IEEE-RAS Int. Conf. Humanoid Rob.*, pp. 112–119, 2014.

[105] S. Nobili, M. Camurri, V. Barasuol, M. Focchi, D. Caldwell, C. Semini, and M. F. Fallon, "Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots," in *Proc. Rob. Sc. Syst.*, 2017.

[106] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 952–958, 2014.

[107] P. Ramadoss, G. Romualdi, S. Dafarra, F. J. A. Chavez, S. Traversaro, and D. Pucci, "DILIGENT-KIO: A proprioceptive base estimator for humanoid robots using extended Kalman filtering on matrix Lie groups," *arXiv preprint:2105.14914*, 2021.

[108] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, "Deep multi-modal contact estimation for invariant observer design on quadruped robots," *arXiv preprint:2106.15713*, 2021.

[109] P. Chauchat, S. Bonnabel, and A. Barrau, "Invariant smoothing for localization: Including the imu biases," *arXiv preprint arXiv:2309.13903*, 2023.

[110] Z. Yoon, J.-H. Kim, and H.-W. Park, "Invariant smoother for legged robot state estimation with dynamic contact event information," *IEEE Transactions on Robotics*, 2023.

[111] X. Yu, S. Teng, T. Chakhachiro, W. Tong, T. Li, T.-Y. Lin, S. Koehler, M. Ahumada, J. M. Walls, and M. Ghaffari, "Fully proprioceptive slip-velocity-aware state estimation for mobile robots via invariant kalman filtering and disturbance observer," *arXiv preprint arXiv:2209.15140*, 2022.

[112] E. R. Potokar, K. Norman, and J. G. Mangelson, "Invariant extended kalman filtering for underwater navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5792–5799, 2021.

[113] M. Barczyk and A. F. Lynch, "Invariant observer design for a helicopter uav aided inertial navigation system," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 791–806, 2012.

[114] Z. Zhu, S. M. R. Sorkhabadi, Y. Gu, and W. Zhang, "Design and evaluation of an invariant extended Kalman filter for trunk motion estimation with sensor misalignment," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2158–2167, 2022.

[115] Z. Zhu, S. M. R. Sorkhabadi, Y. Gu, and W. Zhang, "Invariant extended kalman filtering for human motion estimation with imperfect sensor placement," in *2022 American Control Conference (ACC)*, pp. 3012–3018, 2022.

[116] "VECTORNAV." https://www.vectornav.com/. Accessed: 2022-09-20.

[117] M. Loueipour, M. Keshmiri, M. Danesh, and M. Mojiri, "Wave filtering and state estimation in dynamic positioning of marine vessels using position measurement," *IEEE Trans. Instr. Meas.*, vol. 64, no. 12, pp. 3253–3261, 2015.

[118] A. Barrau, *Non-linear state error based extended Kalman filters with applications to navigation*. PhD thesis, Mines Paristech, 2015.

[119] A. Barrau and S. Bonnabel, "Invariant Kalman filtering," *Annu Rev Control, Robotics, and Autonomous Systems*, vol. 1, pp. 237–257, 2018.

[120] M. Trkov, K. Chen, J. Yi, and T. Liu, "Inertial sensor-based slip detection in human walking," *IEEE Trans. Autom. Sc. Eng.*, vol. 16, no. 3, pp. 1399–1411, 2019.

[121] Y. Gao, C. Yuan, and Y. Gu, "Invariant filtering for legged humanoid locomotion on a dynamic rigid surface," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 1900–1909, 2022.

[122] Y. Gao, X. Da, and Y. Gu, "Impact-aware online motion planning for fully-actuated bipedal robot walking," in *2020 American Control Conference (ACC)*, pp. 2100–2105, 2020.

[123] D. Bainov and P. Simeonov, *Impulsive differential equations: periodic solutions and applications*, vol. 66. CRC Press, 1993.

[124] A. Hereid and A. D. Ames, "FROST: Fast robot optimization and simulation toolkit," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 719–726, 2017.

[125] A. Iqbal, S. Veer, and Y. Gu, "Analytical solution to a time-varying lip model for quadrupedal walking on a vertically oscillating surface," *Mechatronics*, vol. 96, p. 103073, 2023.

[126] A. Iqbal, S. Veer, C. Niezrecki, and Y. Gu, "Ht-lip model based robust control of quadrupedal robot locomotion under unknown vertical ground motion," *arXiv*, 2023.

[127] 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 *2023 American Control Conference (ACC)*, pp. 3030–3035, 2023.

[128] A. Iqbal, S. Veer, and Y. Gu, "Analytical approximate solution to Mathieu's equation enables real-time motion planning for legged robot walking on a vertically moving surface," in *Dynamic Walking Conference*, 2022.

[129] A. Iqbal and Y. Gu, "Extending capture point to dynamic rigid surfaces," *Dynamic Walking*, 2021.

[130] A. Iqbal and Y. Gu, "Extended capture point and optimization-based control for quadrupedal robot walking on dynamic rigid surfaces," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 72–77, 2021.

[131] A. Iqbal, Z. Mao, and Y. Gu, "Modeling, analysis, and control of slip running on dynamic platforms," *ASME L. Dyn. Syst. Contr.*, vol. 1, no. 2, 2021.

[132] "Realsense SDK." https://dev.intelrealsense.com/docs/projection-in-intel-realsense-sdk-20. Accessed: 2022-08-08.

[133] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.

Chapter A Appendix: Supplementary Materials of Chapter 3

A.1 Appendix: Proofs of Propositions and Theorem 1

A.1.1 Proof of Proposition 2

Integrating both sides of the UA closed-loop dynamics in Eq. (3.18) over time t yields

$$\mathbf{x}_\eta|_{3k-1}^- = \int_{T_{3k-2}}^{T_{3k-1}} \mathbf{f}_\eta(s, \mathbf{x}_\eta(s), \mathbf{x}_\xi(s))ds + \mathbf{x}_\eta|_{3k-2}^+. \tag{A.1}$$

Then,

$$\begin{aligned} \left\|\mathbf{x}_\eta|_{3k-1}^-\right\| &\leq \left\| \int_{T_{3k-2}}^{T_{3k-1}} \mathbf{f}_\eta(s, \mathbf{x}_\eta(s), \mathbf{x}_\xi(s))ds \right\| + \left\|\mathbf{x}_\eta|_{3k-2}^+\right\| \\ &\leq \int_{T_{3k-2}}^{T_{3k-1}} \left\|\mathbf{f}_\eta(s, \mathbf{x}_\eta(s), \mathbf{x}_\xi(s))\right\| ds + \left\|\mathbf{x}_\eta|_{3k-2}^+\right\|. \end{aligned} \tag{A.2}$$

Since the expression of $\mathbf{f}_\eta(\cdot)$ is obtained using the continuous-phase dynamics of the generalized coordinates in Eq. (2.5) and the expression of the output function $\mathbf{y}_U$ in Eqs. (3.11) and (3.12), we know $\mathbf{f}_\eta(t, \mathbf{x}_\eta, \mathbf{x}_\xi)$ is continuously differentiable in t, $\mathbf{x}_\eta$, and $\mathbf{x}_\xi$. Also, we can prove that there exists a finite, real, positive number r_η such that $\|\frac{\partial \mathbf{f}_\eta}{\partial t}\|$, $\|\frac{\partial \mathbf{f}_\eta}{\partial \mathbf{x}_\xi}\|$, and $\|\frac{\partial \mathbf{f}_\eta}{\partial \mathbf{x}_\eta}\|$ are bounded on $(T_{3k-2}, T_{3k-1}) \times B_{r_\eta}(\mathbf{0})$. Then, $\mathbf{f}_\eta(t, \mathbf{x}_\eta, \mathbf{x}_\xi)$ is Lipschitz continuous on $(T_{3k-2}, T_{3k-1}) \times B_{r_\eta}(\mathbf{0})$ [81], and we can prove that there exists a a real, positive number k_f such that

$$\left\|\mathbf{f}_\eta(t, \mathbf{x}_\eta(t), \mathbf{x}_\xi(t))\right\| \leq k_f \tag{A.3}$$

holds for any $t \times (\mathbf{x}_\eta, \mathbf{x}_\xi) \in (T_{3k-2}, T_{3k-1}) \times B_{r_\eta}(\mathbf{0})$.

Combining the two inequalities above, we have

$$\left\|\mathbf{x}_\eta|_{3k-1}^-\right\| \leq k_f(T_{3k-1} - T_{3k-2}) + \left\|\mathbf{x}_\eta|_{3k-2}^+\right\|. \tag{A.4}$$

The duration $(T_{3k-1} - T_{3k-2})$ of the UA phase can be estimated as:

$$\begin{aligned} |T_{3k-1} - T_{3k-2}| &= |T_{3k-1} - \tau_{3k-1} + \tau_{3k-1} - T_{3k-2}| \\ &\leq |T_{3k-1} - \tau_{3k-1}| + \delta_{\tau_U}, \end{aligned} \tag{A.5}$$

where $\delta_{\tau_U} := \tau_{3k-1} - T_{3k-2}$ is the expected duration of the UA phase and $|T_{3k-1} - \tau_{3k-1}|$ is the absolute difference between the actual and planned time instants of the UA→OA switching.

From our previous work [60], we know there exists small positive numbers ε_U and r_{U1} such that

$$|T_{3k-1} - \tau_{3k-1}| \leq \varepsilon_U \delta_{\tau_U} \tag{A.6}$$

holds for any $k \in \{1, 2, ...\}$ and $\mathbf{x}_U \in B_{r_{U1}}(\mathbf{0})$.

Thus, using Eqs. (A.4)-(A.6), we have

$$\left\| \mathbf{x}_\eta |_{3k-1}^- \right\| \leq k_f (1 + \varepsilon_U) \delta_{\tau_U} + \left\| \mathbf{x}_\eta |_{3k-2}^+ \right\|. \tag{A.7}$$

Substituting Eqs. (3.24) and (A.7) into Eq. (3.20) gives

$$\begin{aligned} V_U |_{3k-1}^- =& V_\xi |_{3k-1}^+ + \beta \left\| \mathbf{x}_\eta |_{3k-1}^+ \right\|^2 \\ \leq& e^{-c_{3\xi}(T_{3k-1} - T_{3k-2})} V_\xi |_{3k-2}^+ + 2\beta \left\| \mathbf{x}_\eta |_{3k-2}^+ \right\|^2 \\ &+ 2\beta k_f^2 (1 + \varepsilon_U)^2 \delta_{\tau_U}^2 \\ \leq& 2 V_U |_{3k-2}^+ + 2\beta k_f^2 (1 + \varepsilon_U)^2. \end{aligned} \tag{A.8}$$

Thus, for any $\mathbf{x}_U \in B_{r_{U2}}(\mathbf{0})$ with $r_{U2} := \min(r_\eta, r_{U1})$,

$$V_U |_{3k-1}^- \leq w_u (V_U |_{3k-2}^+)$$

holds, where $w_u(V_U |_{3k-2}^+) := 2 V_U |_{3k-2}^+ + 2\beta k_f^2 (1 + \varepsilon_U)^2$. It is clear that $w_u(V_U |_{3k-2}^+)$ is a positive-definite function of $V_U |_{3k-2}^+$. ■

A.1.2 Proof of Proposition 3

For brevity, we only show the proof for $... \leq V_F|_{3k}^{+} \leq V_F|_{3k-3}^{+} \leq ... \leq V_F|_{3}^{+} \leq V_F|_{0}^{+}$, based on which the proofs for the other two sets of inequalities in Eq. (3.28) can be readily obtained.

To prove that $V_F|_{3k}^{+} \leq V_F|_{3k-3}^{+}$ for any $k \in \{1,2,...\}$, we need to analyze the evoluation of the state variables for the k^{th} actual complete gait cycle on $t \in (T_{3k-3}, T_{3k})$, which comprises three continuous phases and three switching events.

Analyzing the continuous-phase state evolution: We analyze the state evolution during the three continuous phases based on the convergence and boundedness results established in Propositions 1 and 2.

Similar to the boundedness of the UA→OA switching time discrepancy given in Eq. (A.6), there exist small positive numbers ε_F, ε_O, r_{tF} and r_{tO} such that for any $\mathbf{x}_F \in \mathbf{B}_{r_{tF}}(\mathbf{0})$ and $\mathbf{x}_O \in \mathbf{B}_{r_{tO}}(\mathbf{0})$,

$$\left|T_{3k-2} - \tau_{3k-2}\right| \leq \varepsilon_F \delta_{\tau_F} \text{ and } \left|T_{3k} - \tau_{3k}\right| \leq \varepsilon_O \delta_{\tau_O} \tag{A.9}$$

hold, where δ_{τ_F} and δ_{τ_O} are the desired periods of the FA and OA phases of the planned walking cycle, with $\delta_{\tau_F} := \tau_{3k-2} - T_{3k-3}$ and $\delta_{\tau_O} := \tau_{3k} - T_{3k-1}$.

Substituting Eq. (A.9) into Eqs. (3.22) and (3.23) yields

$$\left\|\mathbf{x}_F|_{3k-2}^{-}\right\| \leq \sqrt{\tfrac{c_{2F}}{c_{1F}}} e^{-\frac{c_{3F}}{2c_{2F}}(1+\varepsilon_F)\delta_{\tau_F}} \left\|\mathbf{x}_F|_{3k-3}^{+}\right\| \tag{A.10}$$

and

$$\left\|\mathbf{x}_O|_{3k}^{-}\right\| \leq \sqrt{\tfrac{c_{2O}}{c_{1O}}} e^{-\frac{c_{3O}}{2c_{2O}}(1+\varepsilon_O)\delta_{\tau_O}} \left\|\mathbf{x}_O|_{3k-1}^{+}\right\| \tag{A.11}$$

for any $\mathbf{x}_i \in B_{\bar{r}_i}(\mathbf{0})$ ($i \in \{F,O\}$), with the small positive number $\bar{r}_i$ defined as $\bar{r}_i := \min\{r_i, r_{ti}\}$.

From the definition of the Lyapunov-like function V_U in Eq. (3.20), the continuous-phase boundedness of V_U in Eq. (A.8), and the continuous-phase convergence of V_ξ in Eq. (3.24), we obtain the following inequality characterizing the boundedness of the state variable $\mathbf{x}_U$

within the UA phase:

$$\left\|\mathbf{x}_U|_{3k-1}^-\right\|^2 \leq 2\frac{\tilde{c}_{2\xi}}{\tilde{c}_{1\xi}}\left\|\mathbf{x}_U|_{3k-2}^+\right\|^2 + \frac{2\beta k_f^2}{\tilde{c}_{1\xi}}(1+\varepsilon_U)^2 \tag{A.12}$$

where the real scalar constants $\tilde{c}_{1\xi}$ and $\tilde{c}_{2\xi}$ are defined as $\tilde{c}_{1\xi} := \min(c_{1\xi}, \beta)$ and $\tilde{c}_{2\xi} := \max(c_{2\xi}, \beta)$.

Since

$$\begin{aligned}
&2\frac{\tilde{c}_{2\xi}}{\tilde{c}_{1\xi}}\left\|\mathbf{x}_U|_{3k-2}^+\right\|^2 + \frac{2\beta k_f^2}{\tilde{c}_{1\xi}}(1+\varepsilon_U)^2 \\
&\leq \left(\sqrt{2\frac{\tilde{c}_{2\xi}}{\tilde{c}_{1\xi}}}\left\|\mathbf{x}_U|_{3k-2}^+\right\| + \sqrt{\frac{2\beta k_f^2}{\tilde{c}_{1\xi}}}(1+\varepsilon_U)\right)^2,
\end{aligned}$$

we rewrite Eq. (A.12) as:

$$\begin{aligned}
\left\|\mathbf{x}_U|_{3k-1}^-\right\| &\leq \sqrt{2\frac{\tilde{c}_{2\xi}}{\tilde{c}_{1\xi}}}\left\|\mathbf{x}_U|_{3k-2}^+\right\| + \sqrt{\frac{2\beta k_f^2}{\tilde{c}_{1\xi}}}(1+\varepsilon_U) \\
&=: \alpha_1\left\|\mathbf{x}_U|_{3k-2}^+\right\| + \alpha_2.
\end{aligned} \tag{A.13}$$

Analyzing the state evolution across a jump: Without loss of generality, we first examine the state evolution across the F→U switching event by relating the norms of the state variable just before and after the impact.

Using the expression of the reset map $\mathbf{\Delta}_{F\to U}$ at the switching instant $t = T_{3k-2}^-$ ($k \in \{1,2,...\}$), we obtain the following inequality

$$\begin{aligned}
\left\|\mathbf{x}_U|_{3k-2}^+\right\| =& \left\|\mathbf{\Delta}_{F\to U}(T_{3k-2}, \mathbf{x}_F|_{3k-2}^-)\right\| \\
=& \Big\|(\mathbf{\Delta}_{F\to U}(T_{3k-2}, \mathbf{x}_F|_{3k-2}^-) - \mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{x}_F|_{3k-2}^-)) \\
&+ (\mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{x}_F|_{3k-2}^-) - \mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{0})) \\
&+ \mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{0})\Big\| \\
\leq& \left\|\mathbf{\Delta}_{F\to U}(T_{3k-2}, \mathbf{x}_F|_{3k-2}^-) - \mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{x}_F|_{3k-2}^-)\right\| \\
&+ \left\|\mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{x}_F|_{3k-2}^-) - \mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{0})\right\| \\
&+ \left\|\mathbf{\Delta}_{F\to U}(\tau_{3k-2}, \mathbf{0})\right\|.
\end{aligned} \tag{A.14}$$

Next, we relate the three terms on the right-hand side of the inequality in Eq. (A.14)

explicitly with the norm of the state just before the switching (i.e., $\mathbf{x}_F|_{3k-2}^-$).

Recall that the expressions of $\boldsymbol{\Delta}_{F\to U}(t,\mathbf{x}_F)$ solely depends on the expressions of: (i) the impact dynamics $\boldsymbol{\Delta}_{\dot{q}}(\mathbf{q})\dot{\mathbf{q}}$, which is continuously differentiable on $(\mathbf{q},\dot{\mathbf{q}}) \in \mathcal{TQ}$; (ii) the output functions $\mathbf{y}_F(t,\mathbf{q})$, which is continuously differentiable on $t \in \mathbb{R}^+$ and $\mathbf{q} \in \mathcal{Q}$ under assumption (A7); and (iii) the time derivative $\dot{\mathbf{y}}_F(t,\mathbf{q},\dot{\mathbf{q}})$, which, also under assumption (A7), is continuously differentiable on $t \in \mathbb{R}^+$ and $(\mathbf{q},\dot{\mathbf{q}}) \in \mathcal{TQ}$. Thus, we know $\boldsymbol{\Delta}_{F\to U}$ is continuously differentiable for any $t \in \mathbb{R}^+$ (i.e., including any continuous phases) and state $\mathbf{x}_F \in \mathbb{R}^{2n}$.

Similarly, under assumption (A7), we can prove that there exists a small, real constant l_F such that $\|\frac{\partial \boldsymbol{\Delta}_{F\to U}}{\partial t}\|$ and $\|\frac{\partial \boldsymbol{\Delta}_{F\to U}}{\partial \mathbf{x}_F}\|$ are bounded for any $t \in \mathbb{R}^+$ (including all continuous FA phases) and $\mathbf{x}_F \in B_{l_F}(\mathbf{0})$. Thus, for any $k \in \{1,2,...\}$, the function $\boldsymbol{\Delta}_{F\to U}$ is Lipschitz continuous on for any $t \in [T_{3k-2};\tau_{3k-2}]$ and $\mathbf{x}_F \in B_{l_F}(\mathbf{0})$, where $[T_{3k-2};\tau_{3k-2}]$ equals $[T_{3k-2},\tau_{3k-2}]$ if $T_{3k-2} < \tau_{3k-2}$, and it equals $[\tau_{3k-2},T_{3k-2}]$ if $T_{3k-2} > \tau_{3k-2}$.

Thus, there exist Lipschitz constants L_{tF} and L_{xF} such that:

$$\begin{aligned}&\left\|\boldsymbol{\Delta}_{F\to U}(T_{3k-2},\mathbf{x}_F|_{3k-2}^-) - \boldsymbol{\Delta}_{F\to U}(\tau_{3k-2},\mathbf{x}_F|_{3k-2}^-)\right\| \\ \leq & L_{tF}|T_{3k-2}-\tau_{3k-2}|\end{aligned} \tag{A.15}$$

and

$$\begin{aligned}&\left\|\boldsymbol{\Delta}_{F\to U}(\tau_{3k-2},\mathbf{x}_F|_{3k-2}^-) - \boldsymbol{\Delta}_{F\to U}(\tau_{3k-2},\mathbf{0})\right\| \\ \leq & L_{xF}\left\|\mathbf{x}_F|_{3k-2}^-\right\|\end{aligned} \tag{A.16}$$

hold on $[T_{3k-2};\tau_{3k-2}]\times B_{l_F}(\mathbf{0})$ for any $k \in \{1,2,...\}$.

From condition (A2) and Eqs. (3.25), (A.9), and (A.14)-(A.16), we know that

$$\left\|\mathbf{x}_U|_{3k-2}^+\right\| \leq L_{xF}\left\|\mathbf{x}_F|_{3k-2}^-\right\| + L_{tF}\varepsilon_F\delta_{\tau F} + \gamma_\Delta. \tag{A.17}$$

Analogous to the derivation of the inequality in Eq. (A.17), we can show that there exist a real, positive number l_U and Lipschitz constants L_{tU} and L_{xU} such that:

$$\left\|\mathbf{x}_O|_{3k-1}^+\right\| \leq L_{xU}\left\|\mathbf{x}_U|_{3k-1}^-\right\| + L_{tU}\varepsilon_U\delta_{\tau_U} + \gamma_\Delta \tag{A.18}$$

holds for any $\mathbf{x}_U|_{3k-1}^- \in B_{l_U}(\mathbf{0})$.

As the robot has full control authority within the OA domain, we can establish a tighter upper bound on $\left\|\mathbf{x}_F|_{3k}^+\right\|$ than Eqs. (A.17) and (A.18) by applying Proposition 3 from our previous work [75]. That is, there exists a real, positive number l_O and Lipschitz constants L_{tO} and L_{xO} such that

$$\left\|\mathbf{x}_F|_{3k}^+\right\| \leq L_{tO}\sqrt{\frac{c_{2O}}{c_{1O}}}e^{-\frac{c_{3O}}{2c_{2O}}\delta_{\tau_O}}\left\|\mathbf{x}_O|_{3k-1}^+\right\| + L_{xO}\left\|\mathbf{x}_O|_{3k}^-\right\| \tag{A.19}$$

for any $\mathbf{x}_O|_{3k}^- \in B_{l_O}(\mathbf{0})$.

From Eqs. (A.10), (A.11), (A.13), and (A.17)-(A.19), we obtain:

$$\left\|\mathbf{x}_F|_{3k}^+\right\| \leq \bar{N} + \bar{L}\left\|\mathbf{x}_F|_{3k-3}^+\right\|. \tag{A.20}$$

where

$$\begin{aligned}\bar{N} :=& \left(L_{tU}\varepsilon_U\delta_{\tau_U} + \gamma_\Delta + L_{xU}(\alpha_1 L_{tF}\varepsilon_F\delta_{\tau_F} + \alpha_1\gamma_\Delta + \alpha_1\alpha_2)\right)\\ &\cdot\left(L_{tO}\sqrt{\frac{c_{2O}}{c_{1O}}}e^{-\frac{c_{3O}}{2c_{2O}}\delta_{\tau_O}} + L_{xO}\sqrt{\frac{c_{2O}}{c_{1O}}}e^{-\frac{c_{3O}}{2c_{2O}}(1+\varepsilon_O)\delta_{\tau_O}}\right)\end{aligned}$$

and

$$\begin{aligned}\bar{L} :=& L_{xU}\alpha_1 L_{xF}\sqrt{\frac{c_{2F}}{c_{1F}}}e^{-\frac{c_{3F}}{2c_{2F}}(1+\varepsilon_F)\delta_{\tau_F}}\\ &\cdot\left(L_{tO}\sqrt{\frac{c_{2O}}{c_{1O}}}e^{-\frac{c_{3O}}{2c_{2O}}\delta_{\tau_O}} + L_{xO}\sqrt{\frac{c_{2O}}{c_{1O}}}e^{-\frac{c_{3O}}{2c_{2O}}(1+\varepsilon_O)\delta_{\tau_O}}\right),\end{aligned}$$

Using Eqs. (3.21) and (A.20), we obtain

$$V_F|_{3k}^+ \leq 2c_{2F}\bar{N}^2 + \frac{2c_{2F}\bar{L}^2}{c_{1F}}V_F|_{3k-3}^+. \tag{A.21}$$

Note that the scalar positive parameters $\bar{N}$ and $\bar{L}$ in Eq. (A.21) are both dependent on the continuous-phase convergence rates of the Lyapunov-like functions within the OA and FA domains (i.e., c_{3F} and c_{3O}), Specifically, $\bar{N}$ and $\bar{L}$ (and accordingly $\frac{2c_{2F}\bar{L}^2}{c_{1F}}$ and $2c_{2F}\bar{N}^2$) will decrease towards zero as the continuous-phase convergence rates increase towards the infinity.

If condition (A3) holds (i.e., the PD gains can be adjusted to ensure a sufficiently high continuous-phase convergence rate), we can choose the PD gains such that $\frac{2c_{2F}\bar{L}^2}{c_{1F}}$ is less

than 1 and $2c_{2F}\bar{N}^2$ is sufficiently close to 0, which will then ensure $V_F|_{3k}^{+} \leq V_F|_{3k-3}^{+}$ for any $k \in \{1, 2, ...\}$.

■

A.1.3 Proof of Theorem 1

By the general stability theory based on multiple Lyapunov functions [83], the origin of the overall hybrid error system described in Eqs. (3.16) and (3.18) is locally stable in the sense of Lyapunov if the Lyapunov-like functions V_F, V_O, and V_U satisfy the following conditions:

(C1) The Lyapunov-like functions V_F and V_O exponentially decrease within the continuous FA and OA phases, respectively.

(C2) Within the continuous UA phase, the "switching-out" value of the Lyapunov-like function V_U is bounded above by a positive-definite function of the "switching-in" value of V_U; and

(C3) The values of each Lyapunov-like functions at their associated "switching-in" instants form a nonincreasing sequence.

If the proposed IO-PD control law satisfies condition (B1), then the control law ensures conditions (C1) and (C2), as established in Proposition 1 and 2, respectively. By further meeting conditions (B2) and (B3), we know from Proposition 3 that condition (C3) will hold. Thus, under conditions (B1)-(B3), the closed-loop control system meets conditions (C1)-(C3), and thus the origin of the overall hybrid error system described in Eqs. (3.16) and (3.17) is locally stable in the sense of Lyapunov. ■

Chapter B Appendix: Supplementary Materials of Chapter 4

B.1 Expression of Right-Invariant and Logarithmic Errors

The right-invariant error $\boldsymbol{\eta}_t$ is defined as:

$$\boldsymbol{\eta}_t = \bar{\mathbf{X}}_t \mathbf{X}_t^{-1} = \begin{bmatrix} \boldsymbol{\eta}_{R,t} & [\boldsymbol{\eta}_{v,t},\ \boldsymbol{\eta}_{p,t}, \boldsymbol{\eta}_{p^c,t}] \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix}, \tag{B.1}$$

with the individual terms expressed as:

$$\boldsymbol{\eta}_{R,t} = \bar{\mathbf{R}}_t \mathbf{R}_t^T, \quad \boldsymbol{\eta}_{v,t} = \bar{\mathbf{v}}_t - \bar{\mathbf{R}}_t \mathbf{R}_t^T \mathbf{v}_t, \quad \boldsymbol{\eta}_{p,t} = \bar{\mathbf{p}}_t - \bar{\mathbf{R}}_t \mathbf{R}_t^T \mathbf{p}_t, \quad \text{and} \quad \boldsymbol{\eta}_{p^c,t} = \bar{\mathbf{p}}_t^c - \bar{\mathbf{R}}_t \mathbf{R}_t^T \mathbf{p}_t^c. \tag{B.2}$$

Denote $\boldsymbol{\xi}_t = \left[(\boldsymbol{\xi}_t^R)^T,\ (\boldsymbol{\xi}_t^v)^T,\ (\boldsymbol{\xi}_t^p)^T,\ (\boldsymbol{\xi}_t^{p^c})^T\right]^T$. Then, the variable $\boldsymbol{\xi}_t^\wedge$ on the Lie algebra $\mathfrak{g}$ is $\boldsymbol{\xi}_t^\wedge = \begin{bmatrix} (\boldsymbol{\xi}_{R,t})_\times & [\boldsymbol{\xi}_{v,t},\ \boldsymbol{\xi}_{p,t},\ \boldsymbol{\xi}_{p^c,t}] \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}$, where $(\cdot)_\times$ is a skew-symmetric matrix.

B.2 Linearization of Continuous-Phase Error Propagation Equation

To obtain the linearized dynamic of the logarithmic error $\boldsymbol{\xi}_t$, we first apply the chain rule to obtain the time derivatives of the individual terms of the invariant error $\boldsymbol{\eta}_t$ as:

$$\begin{aligned} \dot{\boldsymbol{\eta}}_{R,t} &\approx (\bar{\mathbf{R}}_t(\mathbf{w}_t^\omega - \boldsymbol{\zeta}_t^\omega))_\times, \\ \dot{\boldsymbol{\eta}}_{v,t} &\approx (\mathbf{g})_\times \boldsymbol{\xi}_t^R + (\bar{\mathbf{v}}_t)_\times \bar{\mathbf{R}}_t(\mathbf{w}_t^\omega - \boldsymbol{\zeta}_t^\omega) + \bar{\mathbf{R}}_t(\mathbf{w}_t^a - \boldsymbol{\zeta}_t^a), \\ \dot{\boldsymbol{\eta}}_{p,t} &\approx \boldsymbol{\xi}_t^v + (\bar{\mathbf{p}}_t)_\times \bar{\mathbf{R}}_t(\mathbf{w}_t^\omega - \boldsymbol{\zeta}_t^\omega), \text{ and} \\ \dot{\boldsymbol{\eta}}_{p^c,t} &\approx (\bar{\mathbf{v}}_t^c)_\times \boldsymbol{\xi}_t^R + (\bar{\mathbf{d}}_t)_\times \bar{\mathbf{R}}_t(\mathbf{w}_t^\omega - \boldsymbol{\zeta}_t^\omega) + \bar{\mathbf{R}}_t \mathbf{w}_t^c, \end{aligned} \tag{B.3}$$

where $\boldsymbol{\zeta}_t = \left[(\boldsymbol{\zeta}_t^{\omega})^T,\ (\boldsymbol{\zeta}_t^{a})^T\right]^T$ is the IMU bias error.

Then, we use $\boldsymbol{\eta}_t = \exp(\boldsymbol{\xi}_t) = \text{expm}(\boldsymbol{\xi}_t^{\wedge})$ and its first-order approximation $\text{expm}(\boldsymbol{\xi}_t^{\wedge}) \approx \mathbf{I} + \boldsymbol{\xi}_t^{\wedge}$ to obtain the following approximations:

$$(\boldsymbol{\xi}_{R,t})_{\times} \approx \bar{\mathbf{R}}_t \mathbf{R}_t^T - \mathbf{I}_3,\ \boldsymbol{\xi}_{v,t} \approx \bar{\mathbf{v}}_t - \bar{\mathbf{R}}_t \mathbf{R}_t^T \mathbf{v}_t,\ \boldsymbol{\xi}_{p,t} \approx \bar{\mathbf{p}}_t - \bar{\mathbf{R}}_t \mathbf{R}_t^T \mathbf{p}_t,\ \text{and}\ \boldsymbol{\xi}_t^d \approx \bar{\mathbf{d}}_t - \bar{\mathbf{R}}_t \mathbf{R}_t^T \mathbf{d}_t. \tag{B.4}$$

By combining these equations and applying the chain rule to find the time derivative of $\boldsymbol{\xi}_t$, we obtain the linearized error equation in Eq. (4.15) of Chapter 4.

In Eq. (4.15) of Chapter 4, the expression of the adjoint matrix $\mathbf{Ad}_{\mathbf{X}_t}$ is obtained through its basic definition given in Chapter 2 (Preliminaries). Its specific expression in Eq. (4.15) is:

$$\mathbf{Ad}_{\bar{\mathbf{X}}_t} = \begin{bmatrix} \bar{\mathbf{R}}_t & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\bar{\mathbf{v}}_t)_{\times}\bar{\mathbf{R}}_t & \bar{\mathbf{R}}_t & \mathbf{0} & \mathbf{0} \\ (\bar{\mathbf{p}}_t)_{\times}\bar{\mathbf{R}}_t & \mathbf{0} & \mathbf{R}_t & \mathbf{0} \\ (\bar{\mathbf{p}}_t^c)_{\times}\bar{\mathbf{R}}_t & \mathbf{0} & \mathbf{0} & \bar{\mathbf{R}}_t \end{bmatrix}, \tag{B.5}$$

where the matrix $\mathbf{0}$ is a 3×3 zero matrix.

B.3 Linearization of Continuous-Phase Error Update Equation

This section explains the derivation of the linearized update equation for the errors $\boldsymbol{\xi}_{t_n}$ and $\boldsymbol{\zeta}_{t_n}$ in Eq. (4.20) of Chapter 4. We will first derive the linearized update equation for $\boldsymbol{\xi}_{t_n}$.

From the estimate update equation in Eq. (4.20) of Chapter 4, we have

$$\bar{\mathbf{X}}_{t_n}^{\dagger} = \exp\left(\mathbf{L}_{t_n}^{\xi} \begin{bmatrix} \bar{\mathbf{X}}_{t_n}\mathbf{Y}_{1,t_n} - \mathbf{d}_{1,t_n} \\ \bar{\mathbf{X}}_{t_n}\mathbf{Y}_{2,t_n} - \mathbf{d}_{2,t_n} \end{bmatrix}\right) \bar{\mathbf{X}}_{t_n}. \tag{B.6}$$

Multiplying $\mathbf{X}_{t_n}^{-1}$ on both sides of Eq. (B.6) gives:

$$\bar{\mathbf{X}}_{t_n}^{\dagger}\mathbf{X}_{t_n}^{-1} = \exp\left(\mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\bar{\mathbf{X}}_{t_n}\mathbf{Y}_{1,t_n} - \mathbf{d}_{1,t_n}\\ \bar{\mathbf{X}}_{t_n}\mathbf{Y}_{2,t_n} - \mathbf{d}_{2,t_n}\end{bmatrix}\right)\bar{\mathbf{X}}_{t_n}\mathbf{X}_{t_n}^{-1}. \tag{B.7}$$

Simplifying the above equation yields:

$$\boldsymbol{\eta}_{t_n}^{\dagger} = \exp\left(\mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\boldsymbol{\eta}_{t_n}\mathbf{d}_{1,t_n} + \bar{\mathbf{X}}_{t_n}\begin{bmatrix}\mathbf{V}_{1,t_n}\\ \mathbf{0}_{3\times 1}\end{bmatrix} - \mathbf{d}_{1,t_n}\\ \boldsymbol{\eta}_{t_n}\mathbf{d}_{2,t_n} + \bar{\mathbf{X}}_{t_n}\begin{bmatrix}\frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n})\mathbf{w}_{t_n}^{q}\\ \mathbf{0}_{3\times 1}\end{bmatrix} - \mathbf{d}_{2,t_n}\end{bmatrix}\right)\boldsymbol{\eta}_{t_n}. \tag{B.8}$$

By utilizing the approximations $\boldsymbol{\eta}_{t_n} \approx \mathbf{I} + \boldsymbol{\xi}_{t_n}^{\wedge}$ and $\exp(\mathbf{a}) \approx \mathbf{I} + \mathbf{a}^{\wedge}$ where $\mathbf{a}$ is the vector inside the parentheses of Eq. (B.8), and by neglecting higher-order terms, we obtain:

$$\begin{aligned}
\boldsymbol{\xi}_{t_n}^{\dagger} &\approx \boldsymbol{\xi}_{t_n} + \mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\boldsymbol{\xi}_{t_n}^{\wedge}\mathbf{d}_{1,t_n} + \mathbf{X}_{t_n}\begin{bmatrix}\mathbf{V}_{1,t_n}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\\ \boldsymbol{\xi}_{t_n}^{\wedge}\mathbf{d}_{2,t_n} + \mathbf{X}_{t_n}\begin{bmatrix}\frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n})\mathbf{w}_{t_n}^{q}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\end{bmatrix}\\
&= \boldsymbol{\xi}_{t_n} - \mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\tilde{\mathbf{H}}_{1,t_n}\boldsymbol{\xi}_{t_n}\\ \tilde{\mathbf{H}}_{2,t_n}\boldsymbol{\xi}_{t_n}\end{bmatrix} + \mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\begin{bmatrix}\mathbf{V}_{1,t_n}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\\ \begin{bmatrix}\frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n})\mathbf{w}_{t_n}^{q}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\end{bmatrix}\\
&= \boldsymbol{\xi}_{t_n} - \mathbf{L}_{t_n}^{\xi}\mathbf{H}_{t_n}\begin{bmatrix}\boldsymbol{\xi}_{t_n}\\ \boldsymbol{\zeta}_{t_n}\end{bmatrix} + \mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\begin{bmatrix}\mathbf{V}_{1,t_n}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\\ \begin{bmatrix}\frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n})\mathbf{w}_{t_n}^{q}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\end{bmatrix}\\
&= \boldsymbol{\xi}_{t_n} - \mathbf{L}_{t_n}^{\xi}\mathbf{H}_{t_n}\begin{bmatrix}\boldsymbol{\xi}_{t_n}\\ \boldsymbol{\zeta}_{t_n}\end{bmatrix} + \mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\begin{bmatrix}\boldsymbol{\eta}_{t_n}(\tilde{\mathbf{R}}_{t_n}^{DRS}\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{T})_{\times}\mathbf{w}_{t_n}^{DRS} + \bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n})\end{bmatrix}\\ \mathbf{0}_{3\times 1}\\ \begin{bmatrix}\bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n})\\ \mathbf{0}_{3\times 1}\end{bmatrix}\end{bmatrix}.
\end{aligned} \tag{B.9}$$

As $\boldsymbol{\eta}_{t_n}$ and $\mathbf{w}_{t_n}^{DRS}$ are small quantities, their product can be ignored. Thus, we have:

$$\boldsymbol{\xi}_{t_n}^{\dagger} \approx \boldsymbol{\xi}_{t_n} - \mathbf{L}_{t_n}^{\xi}\mathbf{H}_{t_n}\begin{bmatrix}\boldsymbol{\xi}_{t_n}\\ \boldsymbol{\zeta}_{t_n}\end{bmatrix} + \mathbf{L}_{t_n}^{\xi}\begin{bmatrix}\begin{bmatrix}\bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n})\\ \mathbf{0}_{3\times 1}\end{bmatrix}\\ \begin{bmatrix}\bar{\mathbf{R}}_{t_n}\frac{\partial \mathbf{h}_{p}}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n})\\ \mathbf{0}_{3\times 1}\end{bmatrix}\end{bmatrix}. \tag{B.10}$$

Next, we will derive the linearized update equation for $\boldsymbol{\zeta}_{t_n}$. From the definition of $\boldsymbol{\zeta}_t$, we know:

$$\boldsymbol{\zeta}_{t_n}^{\dagger} = \bar{\boldsymbol{\theta}}_{t_n}^{\dagger} - \boldsymbol{\theta}_{t_n}. \tag{B.11}$$

Also, from the estimate update equation in Eq. (4.19) of Chapter 4, we have:

$$\bar{\boldsymbol{\theta}}_{t_n}^{\dagger} = \bar{\boldsymbol{\theta}}_{t_n} + \mathbf{L}_{t_n}^{\zeta}\left(\begin{bmatrix}\bar{\mathbf{X}}_{t_n}\mathbf{Y}_{1,t_n} - \mathbf{d}_{1,t_n}\\ \bar{\mathbf{X}}_{t_n}\mathbf{Y}_{2,t_n} - \mathbf{d}_{2,t_n}\end{bmatrix}\right). \tag{B.12}$$

Combining Eqs. (B.11) and (B.12) yields:

$$\boldsymbol{\zeta}_{t_n}^{\dagger} = \boldsymbol{\zeta}_{t_n} + \mathbf{L}_{t_n}^{\zeta}\left(\begin{bmatrix}\boldsymbol{\eta}_{t_n}\mathbf{d}_{1,t_n} - \mathbf{d}_{1,t_n} + \bar{\mathbf{X}}_{t_n}\begin{bmatrix}\mathbf{V}_{1,t_n}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\\ \boldsymbol{\eta}_{t_n}\mathbf{d}_{2,t_n} - \mathbf{d}_{2,t_n} + \bar{\mathbf{X}}_{t_n}\begin{bmatrix}\frac{\partial \mathbf{h}_{p}}{\partial \mathbf{q}}(\mathbf{q}_{t_n})\mathbf{w}_{t_n}^{q}\\ \mathbf{0}_{3\times 1}\end{bmatrix}\end{bmatrix}\right). \tag{B.13}$$

By utilizing the approximation $\boldsymbol{\eta}_{t_n} \approx \mathbf{I} + \boldsymbol{\xi}^{\wedge}_{t_n}$ and neglecting higher-order terms, we obtain:

$$
\begin{aligned}
\boldsymbol{\zeta}^{\dagger}_{t_n} &\approx \boldsymbol{\zeta}_{t_n} + \mathbf{L}^{\zeta}_{t_n} \begin{bmatrix} \boldsymbol{\xi}^{\wedge}_{t_n} \mathbf{d}_{1,t_n} + \bar{\mathbf{X}}_{t_n} \begin{bmatrix} \mathbf{V}_{1,t_n} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \\ \boldsymbol{\xi}^{\wedge}_{t_n} \mathbf{d}_{2,t_n} + \bar{\mathbf{X}}_{t_n} \begin{bmatrix} \frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n}) \mathbf{w}^q_{t_n} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \end{bmatrix} \\
&= \boldsymbol{\zeta}_{t_n} + \mathbf{L}^{\zeta}_{t_n} \begin{bmatrix} -\tilde{\mathbf{H}}_{1,t_n} \boldsymbol{\xi}_{t_n} + \bar{\mathbf{X}}_{t_n} \begin{bmatrix} \mathbf{V}_{1,t_n} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \\ -\tilde{\mathbf{H}}_{2,t_n} \boldsymbol{\xi}_{t_n} + \bar{\mathbf{X}}_{t_n} \begin{bmatrix} \frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n}) \mathbf{w}^q_{t_n} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \end{bmatrix} \\
&= \boldsymbol{\zeta}_{t_n} - \mathbf{L}^{\zeta}_{t_n} \begin{bmatrix} \tilde{\mathbf{H}}_{1,t_n} \boldsymbol{\xi}_{t_n} \\ \tilde{\mathbf{H}}_{2,t_n} \boldsymbol{\xi}_{t_n} \end{bmatrix} + \mathbf{L}^{\zeta}_{t_n} \begin{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_{t_n} (\tilde{\mathbf{R}}^{DRS}_{t_n} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T)_{\times} \mathbf{w}^{DRS}_{t_n} + \bar{\mathbf{R}}_{t_n} \frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}}(\mathbf{q}_{t_n}) \end{bmatrix} \\ \mathbf{0}_{3\times 1} \\ \begin{bmatrix} \bar{\mathbf{R}}_{t_n} \frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n}) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \end{bmatrix} \\
&\approx \boldsymbol{\zeta}_{t_n} - \mathbf{L}^{\zeta}_{t_n} \begin{bmatrix} \tilde{\mathbf{H}}_{1,t_n} \boldsymbol{\xi}_{t_n} \\ \tilde{\mathbf{H}}_{2,t_n} \boldsymbol{\xi}_{t_n} \end{bmatrix} + \mathbf{L}^{\zeta}_{t_n} \begin{bmatrix} \begin{bmatrix} \bar{\mathbf{R}}_{t_n} \frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}}(\mathbf{q}_{t_n}) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \\ \begin{bmatrix} \bar{\mathbf{R}}_{t_n} \frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\mathbf{q}_{t_n}) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \end{bmatrix}.
\end{aligned}
\tag{B.14}
$$

Arranging Eqs.(B.10) and (B.14) into the matrix forms gives:

$$
\begin{bmatrix} \boldsymbol{\xi}^{\dagger}_{t_n} \\ \boldsymbol{\zeta}^{\dagger}_{t_n} \end{bmatrix} = (\mathbf{I} - \mathbf{L}_{t_n} \mathbf{H}_{t_n}) \begin{bmatrix} \boldsymbol{\xi}_{t_n} \\ \boldsymbol{\zeta}_{t_n} \end{bmatrix} + \mathbf{L}_{t_n} \begin{bmatrix} \bar{\mathbf{R}}_{t_n} \frac{\partial \mathbf{h}_{R,3}}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n}) \\ \mathbf{0}_{3\times 1} \\ \bar{\mathbf{R}}_{t_n} \frac{\partial \mathbf{h}_p}{\partial \mathbf{q}}(\tilde{\mathbf{q}}_{t_n}) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \mathbf{w}^q_{t_n}. \tag{B.15}
$$

B.4 Matrices Used in Observability analysis

In the observability analysis in the Section 4.3 of Chapter 4, the updated expressions of the matrices $\mathbf{A}_t$ and $\mathbf{H}_t$ in the absence of IMU biases are:

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ (\mathbf{g})_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ (\tilde{\mathbf{v}}_t^c)_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} \end{bmatrix} \tag{B.16}$$

and

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{H}_{1,t} \\ \mathbf{H}_{2,t} \end{bmatrix} = \begin{bmatrix} (\mathbf{R}_t^{DRS}\,[0,0,1]^T)_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{B.17}$$

In Eq. (4.22) of the observability analysis in the Chapter 4, the discrete state transition matrix $\mathbf{\Phi}$ is given by:

$$\mathbf{\Phi} = \text{expm}(\mathbf{A}_t \Delta t) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ (\mathbf{g})_\times \Delta t & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \frac{1}{2}(\mathbf{g})_\times \Delta t^2 & \mathbf{I}_3 \Delta t & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ (\tilde{\mathbf{v}}_t^c)_\times \Delta t & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}, \tag{B.18}$$

where Δt is the duration of one propagation step.

B.5 Contact Point Velocity Sensing

This section explains how to obtain the contact point velocity in the world frame, $\mathbf{v}_t^c$, which is briefly summarized in Section 4.4.2 of Chapter 4. For brevity, we drop the time t from the subscripts of all variables in this section.

We first obtain the 3-D contact point position in the DRS frame, ${}^{DRS}\mathbf{p}^c := [{}^{DRS}p_x^c,\ {}^{DRS}p_y^c,\ {}^{DRS}p_z^c]^T$

(see Fig. 4-1 in Chapter 4), by using the RGB-D camera to track features of the ArUco markers in the camera images, as summarized in Fig. 4-5 of Chapter 4. The detailed procedure of this step is:

(a) Obtain the pixel coordinates $(x^{img}_{marker,i}, y^{img}_{marker,i})$ of the i^{th} corner for all markers in the image frame;

(b) Extract the depth value $d^{img}_{mark,i}$ of the pixel $(x^{img}_{marker,i}, y^{img}_{marker,i})$;

(c) Apply deprojection [132] to obtain the 3-D coordinates $(x^{cam}_{marker,i}, y^{cam}_{marker,i}, z^{cam}_{marker,i})$ of the i^{th} corner with respect to (w.r.t.) the camera frame;

(d) With all the detected corners of the markers and their corresponding known position w.r.t. the treadmill, apply Kabsch algorithm [133] to obtain the optimal estimated camera orientation ${}^{DRS}\tilde{\mathbf{R}}^{cam}$ and position ${}^{DRS}\tilde{\mathbf{p}}^{cam}$ of the RGB-D camera w.r.t. the treadmill frame.

Second, we compute the surface-foot contact point position w.r.t. the treadmill frame ${}^{DRS}\mathbf{p}^c := [{}^{DRS}p^c_x,\ {}^{DRS}p^c_y,\ {}^{DRS}p^c_z]^T$ through forward kinematics $\mathbf{h}_{cam}(\mathbf{q})$. Here, $\mathbf{h}_{cam}(\mathbf{q})$ is the contact point position in the camera frame. Let ${}^{DRS}\tilde{\mathbf{p}}^c = [{}^{DRS}\tilde{p}^c_x,\ {}^{DRS}\tilde{p}^c_y,\ {}^{DRS}\tilde{p}^c_z]^T$ denote the computed value of ${}^{DRS}\mathbf{p}^c$. Then we have ${}^{DRS}\tilde{\mathbf{p}}^c = {}^{DRS}\tilde{\mathbf{R}}^{cam}\mathbf{h}_{cam}(\tilde{\mathbf{q}}) + {}^{DRS}\tilde{\mathbf{p}}^{cam}$.

Finally, we estimate the contact point velocity $\mathbf{v}^c$ as $\tilde{\mathbf{v}}^c = [\tilde{v}^c_x, \tilde{v}^c_y, \tilde{v}^c_z]^T$ based on the known treadmill pitch angle $\tilde{\theta}^{DRS}$ and velocity $\dot{\tilde{\theta}}^{DRS}$ and forward kinematics:

$$\|\tilde{\mathbf{v}}^c\| = (\dot{\tilde{\theta}}^{DRS})({}^{DRS}\tilde{p}^c_x), \quad \tilde{v}^c_x = \|\tilde{\mathbf{v}}^c\|\sin(\tilde{\theta}^{DRS}), \quad \tilde{v}^c_y = 0, \quad \text{and} \quad \tilde{v}^c_z = \|\tilde{\mathbf{v}}^c\|\cos(\tilde{\theta}^{DRS}). \tag{B.19}$$

Chapter C Author Biography

Yuan Gao was born on June, 1990 in Lanzhou, China. He has received a B.S degree in Thermal Energy and Engineering from China Jiliang University (Hangzhou, China) in 2014. After his undergraduate study, he join the Mechanical Engineering at Arizona State University (Tempe, AZ, US). This is where Yuan started his robotic journey. After 2 years study, Yuan worked as a robotic engineer at Diverse Automation located in South Carolina. In 2018, Yuan decided to pursue a Ph.D. degree in robotics at UMASS Lowell after getting exposed to the fascinating research field of robotics during his M.S. study and career. Yuan likes math and programming. He also enjoys skateboarding and playing electric guitar in his free time.