User-aligned and Robust Bipedal Locomotion

Thesis by
Kejun Li

In Partial Fulfillment of the Requirements for the
Degree of
PhD in Computation and Neural Systems

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2026
Defended September 5th, 2025

Kejun Li
ORCID: 0000-0002-0823-9839

ACKNOWLEDGEMENTS

I would like to thank my committee members. In my first year, CNS required rotations, and I had the privilege of working in the labs of Drs. Aaron Ames, Yisong Yue, and Joel Burdick. Dr. Maegan Tucker introduced me to the Atalante exoskeleton, guided me through my very first project, and collaborated with me on many more afterward. In many ways, the four of you have been there from the beginning to the end, and I am deeply grateful for your support.

To my advisors: There were times when I was eager to pursue my own perspective, only to realize in hindsight that your guidance was right all along. I'm deeply grateful that you gave me the freedom to explore the directions I was most passionate about, while also providing the mentorship that kept me grounded.

To Dr. Ames, I believe I am your first CNS student. Knowing what I know now, I might not have recommended myself to join your lab back when I had no robotics experience. But you nevertheless gave me this opportunity, and because of how challenging it was, I feel I have grown tremendously as both a researcher and a person. I truly appreciate the chance you took on me. To Dr. Yue, thank you for your advice on both research and personal matters, and for always making yourself available, even during your busiest times. That accessibility has meant a lot.

Beyond my committee, I am indebted to Dr. Ellen Novoseller for her guidance, which was especially valuable during the challenges of the COVID lockdown. I am also grateful to Dr. Xiaobin Xiong for teaching me how to break down complex problems; to Dr. Jesseop Kim for your technical guidance during hardware deployment and meticulous attention to detail in manuscript preparation; and to Dr. Preston Culbertson for offering valuable exposure to different facets of robotics, enriching my perspective on the field.

I've been lucky to be part of not just one but two wonderful groups at Caltech: AMBER Lab and Yue crew. The supportive environment in both made even the hardest deadline pushes manageable. Thank you to everyone I have overlapped with.

To those who shared life with me during this journey, I will always cherish the memories we created together. Noel and Ryan, thank you for being unwavering sources of encouragement through every high and low, and for showing me that it's possible to be deeply in love with both work and life. Zach, your focus and ambition have been a constant source of inspiration, and I'll always remember the countless

to-do lists we built and cleared together. Yue, thank you for keeping me well-fed with endless snacks and shared meals. Min and Skylar, thank you for the badminton games, cat-petting breaks, and for letting me be part of your big life moments. Geeling, I'm grateful for the hobbies we shared, the time we spent together, and for being the best workout buddy I could have asked for. And Evelyn, Rebecca, and Rachel—thank you for reminding me to slow down, enjoy the present, and cherish life beyond research, and for all the support, laughter, meals, and trips we've shared over the years.

Finally, to my parents, thank you for your unwavering love, belief, and patience. None of this would have been possible without you.

ABSTRACT

Bipedal robots are uniquely positioned to operate in environments designed for humans. From humanoids traversing unstructured terrain to robotic exoskeletons assisting individuals with paralysis, these systems highlight the promise of legged locomotion. Yet enabling walking that is both robust and aligned with user needs remains a fundamental challenge, owing to hybrid dynamics, hardware limitations, and the variability across individuals in assistive devices.

The first part of this dissertation addresses user-aligned locomotion. A gait that is theoretically stable may still be rejected in practice if it feels unnatural, uncomfortable, or strenuous. To bridge this gap, we integrate musculoskeletal modeling with trajectory optimization to generate anthropomorphic, dynamically feasible walking gaits, and extend preference-based learning with an active learning formulation that efficiently elicits user feedback within a region of interest while maintaining comfort. Together, these methods enable systematic design of gaits that not only achieve stable walking but also capture the nuanced trade-offs users make between comfort, effort, and naturalness of movement.

The second part of this dissertation focuses on robust locomotion in the face of model mismatch, external disturbances, and environmental variability. We develop robustness strategies spanning multiple layers of the control hierarchy: offline trajectory design informed by robustness metrics grounded in hybrid forward invariance, online adaptation through a data-driven predictive framework, and feedback policies learned in massively parallel simulation using reinforcement learning guided by control Lyapunov functions. While independent, these approaches together provide complementary strategies for handling uncertainty, spanning from offline design to real-time adaptation.

Although motivated by the challenges of exoskeleton locomotion, the methods are validated on other bipedal platforms such as humanoids and lower-limb prostheses, highlighting their broad applicability to diverse bipedal platforms. Overall, this dissertation shows that principled integration of model-based and data-driven approaches enables locomotion strategies that are robust, adaptive, and aligned with human needs, advancing the deployment of bipedal robots and assistive devices.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Kejun Li, Jeeseop Kim, Maxime Brunet, Marine Pétriaux, Yisong Yue, and Aaron D Ames. "Hybrid Data-Driven Predictive Control for Robust and Reactive Exoskeleton Locomotion Synthesis." In: *IROS* (2025).
K.L. participated in the conception of the project, developed and implemented the framework, conducted the experiment, and participated in the writing of the manuscript.

[2] Kejun Li, Zachary Olkin, Yisong Yue, and Aaron D Ames. "CLF-RL: Control Lyapunov Function Guided Reinforcement Learning." In: *arXiv preprint arXiv:2508.09354* (2025).
K.L. participated in the conception of the project, developed and implemented the framework, conducted the experiment, and participated in the writing of the manuscript.

[3] Raul Astudillo, Kejun Li, Maegan Tucker, Chu Xin Cheng, Aaron D Ames, and Yisong Yue. "Preferential Multi-objective Bayesian Optimization." In: *Transactions on Machine Learning Research* (2024).
K.L. participated in the conception of the project, setup simulation to collect data for the experiment, and participated in the writing of the manuscript. URL: https://openreview.net/pdf?id=mjsoESaWDH.

[4] Kejun Li, Jeeseop Kim, Xiaobin Xiong, Kaveh Akbari Hamed, Yisong Yue, and Aaron D Ames. "Data-driven Predictive Control for Robust Exoskeleton Locomotion." In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
K.L. participated in the conception of the project, developed and implemented the framework, conducted the experiment, and participated in the writing of the manuscript. IEEE. 2024, pp. 162–169. URL: https://doi.org/10.1109/IROS58592.2024.10802759.

[5] Maegan Tucker, Kejun Li, and Aaron D. Ames. "Synthesizing Robust Walking Gaits via Discrete-Time Barrier Functions with Application to Multi-Contact Exoskeleton Locomotion". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*.
K.L. contributed to implement low-level controller and setting up simulation environment, obtaining baseline result, conducted the experient, and participated in the writing of the manuscript. 2024, pp. 1136–1142. URL: https://doi.org/10.1109/ICRA57147.2024.10610537.

[6] Ryan K. Cosner, Maegan Tucker, Andrew J. Taylor, Kejun Li, Tamás Molnár, Wyatt Ubellacker, Anil Alan, Gábor Orosz, Yisong Yue, and Aaron D. Ames. "Safety-Aware Preference-Based Learning for Safety-Critical Control." In: *Learning for Dynamics and Control Conference*.

K.L. helped develop the preference-based learning framework, and participated in the writing. PMLR. 2022, pp. 1020–1033. URL: https://proceedings.mlr.press/v168/cosner22a.html.

[7] Kejun Li, Maegan Tucker, Rachel Gehlhar, Yisong Yue, and Aaron D Ames. "Natural Multicontact Walking for Robotic Assistive Devices via Musculoskeletal Models and Hybrid Zero Dynamics." In: *IEEE Robotics and Automation Letters* 7.2 (2022).
K.L. participated in the conception of the project, developed and implemented the framework, conducted the experiment, and participated in the writing of the manuscript., pp. 4283–4290. URL: http://dx.doi.org/10.1109/LRA.2022.3149568.

[8] Maegan Tucker, Kejun Li, Yisong Yue, and Aaron D Ames. "Polar: Preference Optimization and Learning Algorithms for Robotics." In: *arXiv preprint arXiv:2208.04404* (2022).
K.L. contributed to toolbox implementation, participated in the writing of the manuscript.

[9] Kejun Li, Maegan Tucker, Erdem Bıyık, Ellen Novoseller, Joel W Burdick, Yanan Sui, Dorsa Sadigh, Yisong Yue, and Aaron D Ames. "ROIAL: Region of Interest Active Learning for Characterizing Exoskeleton Gait Preference Landscapes." In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*.
K.L. participated in the conception of the project, developed and implemented the algorithm, conducted the experiment, and participated in the writing of the manuscript. IEEE. 2021, pp. 3212–3218. URL: http://dx.doi.org/10.1109/ICRA48506.2021.9560840.

RELEVANT VIDEO CONTENT

[1] *Supplementary video for "ROIAL: Region of Interest Active Learning for Characterizing Exoskeleton Gait Preference Landscapes."* https://www.youtube.com/watch?v=04lMJmKmZrQ.

[2] *Supplementary video for "Natural Multicontact walking for Robotic Assistive Devices via Musculoskeletal Models and Hybrid Zero Dynamics."* https://www.youtube.com/watch?v=gOhZlTypNIs.

[3] *Supplementary Video for "Safety-aware Preference-based Learning for Safety-critical Control."* https://youtu.be/QEuwRDTG7TE. 2022.

[4] *Supplementary Video for "Synthesizing Robust Walking Gaits via Discrete-Time Barrier Functions with Application to Multi-Contact Exoskeleton Locomotion."* https://youtu.be/6aXsBKMxDH0.

[5] *Supplementary Video for "Data-driven Predictive Control for Robust Exoskeleton Locomotion."* https://www.youtube.com/watch?v=rgs36YXRb4I,

[6] *Supplementary Video for "Hybrid Data-Driven Predictive Control for Robust and Reactive Exoskeleton Locomotion Synthesis."* https://www.youtube.com/watch?v=OmVjbQzUGQ0,

[7] *Supplementary video for "CLF-RL: Control Lyapunov Function Guided Reinforcement Learning."'* https://youtu.be/f8iuwgCZs3A.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

Background & Approaches

Bipedal locomotion has long stood as one of the central challenges in robotics. While humans execute locomotion effortlessly, robots struggle with the same task due to the underlying hybrid and nonlinear dynamics of legged movement. Each step involves alternating phases of swing and impact, transitions between continuous and discrete dynamics, and complex interactions with the environment. These features make walking inherently unstable and sensitive to disturbances such as terrain variations or external pushes. While advances in control, optimization, and mechanical design have enabled increasingly dynamic locomotion [1, 2], sustaining robust performance across diverse environments remains challenging.

Traditional approaches have leaned heavily on physics-based modeling, drawing on tools such as the Euler–Lagrange equations, hybrid systems theory, and trajectory optimization. These methods span a spectrum of model fidelities—from reduced-order abstractions like the linear inverted pendulum or single rigid body, to full-order rigid body dynamics—to generate stable walking gaits, either as periodic orbits or through model predictive control [3, 4, 5, 6]. Such approaches provide valuable structure and, in some cases, formal guarantees of stability. Yet they rely critically on the accuracy of the underlying models, which are inevitably imperfect when applied to real hardware and uncertain environments. As a result, controllers designed with perfect model assumption often require substantial tuning or adaptation before succeeding on physical systems.

With advances in computing and learning, the robotics community has increasingly turned to data-driven methods, particularly reinforcement learning (RL). These approaches leverage large-scale simulation to train controllers directly from experience, bypassing explicit system identification and enabling the discovery of agile behaviors. RL has achieved impressive results in simulation and, more recently, in hardware demonstrations for quadrupeds and humanoids [7, 8, 9, 10, 11, 12], showing robustness through domain randomization and promising integration of perception and high-level reasoning with larger foundation models. Beyond simulation, learning based methods are increasingly leveraging human demonstrations,

motion capture datasets, cross embodiment transfer, and internet scale video [13, 14], further enhancing generalizability and overall performance.

However, in practice, high-quality hardware data remain limited, and simulators still struggle to capture the full complexity of real-world dynamics. While world models may eventually help bridge this gap, structured, model-based approaches provide valuable grounding in physical principles and interpretability. Even though strict theoretical guarantees are rarely attainable on hardware, these analytical frameworks remain useful, not for their idealized precision, but for the structure they reveal: quantitative measures of stability, robustness, and feasibility that continue to guide learning-based control toward physically meaningful and safe behavior.

Assistive Devices: Constraints and Objectives

These trade-offs between structure and flexibility become even more pronounced in the context of assistive devices such as exoskeletons and prostheses. Unlike autonomous legged robots, these systems must achieve stability while operating in coordination with a human user. This coupling introduces unique constraints. First, user variability: exoskeletons must accommodate a wide range of body sizes, impairments, and walking preferences, adapting to individuals with diverse biomechanical needs rather than fixed parameters. Second, hardware limitations: to remain wearable, exoskeletons are constrained in weight, power, and range of motion—often more restricted than the human body itself for safety—narrowing the set of feasible control strategies. Third, experiential factors: beyond stability and efficiency, users value comfort, safety, and intuitiveness, qualities that are difficult to encode in traditional control formulations.

Despite extensive research on the biomechanics of non-disabled human locomotion [15], it remains unclear how to translate the principles of natural and efficient walking into robotic assistive devices. Prostheses, for instance, have long remained passive—favored for their reliability and insurance coverage—while the broader wearable robotics community has traditionally optimized for metabolic efficiency. Yet such criteria are irrelevant for fully paraplegic patients who cannot contribute biological effort [16], underscoring the difficulty of defining meaningful control objectives beyond purely mechanical or energetic metrics.

In parallel, humanoid robotics has advanced rapidly, with platforms demonstrating dynamic behaviors such as parkour, backflips, and running. Progress in lower-limb exoskeletons, however, has lagged behind. Devices for individuals with complete

motor paraplegia still struggle with autonomous balance, often relying on crutches or overhead supports [17, 18, 19]. These limitations constrain operation to slow walking speeds [20] and controlled environments, preventing natural upper-body movement and impeding widespread use.

Only recently have crutchless exoskeletons emerged, with systems such as XoMotion, REX, and Wandercraft's Atalante and Eve marking notable progress toward hands-free mobility. Yet their functionality remains modest, typically enabling only slow, deliberate walking with limited behavioral versatility.

These limitations highlight the unique position of exoskeleton control at the intersection of robotic locomotion and human interaction. Like autonomous robots, exoskeletons must maintain robustness under uncertainty; unlike them, they must also adapt to individual users' morphology, intent, and comfort. As such, assistive devices form a compelling testbed for locomotion strategies that fuse model-based structure with data-driven adaptability.

Specific Aims

Building on this perspective, two priorities emerge as central to advancing bipedal and assistive locomotion: user alignment and robustness. A gait that is theoretically stable may still fail in practice if it does not match the needs and preferences of the user, or if it cannot sustain performance under the uncertainty and variability of real-world environments.

In this thesis, the term user is interpreted in two complementary ways. For assistive devices such as exoskeletons, the user is the human wearer, for whom alignment requires comfort, safety, and biomechanical compatibility. For autonomous bipedal robots, the user is the system designer or operator, for whom alignment means tailoring locomotion to task-level objectives such as stability margins, energy efficiency, or coordination with other agents.

For wearable systems, alignment has been pursued along two lines: (i) human-inspired nominal gait generation that respects joint kinematics and ground-reaction force patterns [21], and (ii) sample-efficient personalization frameworks that adapt gait parameters to individual comfort and preference profiles [22, 23, 24]. Early personalization often relied on manual self-exploration—workable but time-consuming and subjective—motivating principled, data-driven methods that scale.

For task-driven autonomous systems, user alignment can involve tailoring locomotion to mission-specific requirements—for example, maximizing stability margins for hazardous terrain, conserving energy for long-duration missions, or producing motion patterns that coordinate effectively with other agents. Such adaptation may be achieved by tuning controller gains, adjusting constraint bounds, or modifying gait generation objectives to emphasize desired behaviors [25, 26].

Yet, alignment alone is not sufficient: once a gait is designed for a particular user or mission, it must remain effective under the uncertainty of real-world operation. This brings robustness to the forefront, ensuring that the selected locomotion strategy maintains stability and performance despite modeling errors, disturbances, and environmental variability. Existing approaches span offline analysis of hybrid dynamics and sensitivity to parameter variations [27], online replanning and adaptation to reject perturbations in real time [6, 28], and learning-based control with domain randomization to improve generalization to model mismatches and diverse operating conditions [29, 30].

This thesis addresses the priorities of user alignment and robustness through distinct but complementary contributions, validated on platforms ranging from powered exoskeletons to humanoid robots.

The contributions are organized into the following two major arcs:

- **User-aligned locomotion** (Chapters 3 to 4):
    1. Develop a nominal gait generation method that integrates musculoskeletal models into the Hybrid Zero Dynamics (HZD) framework, producing motions that are both dynamically stable and anthropomorphic.
    2. Design interactive learning approaches for user preference modeling and personalization. This includes characterizing the user's gait preference landscape via active learning within a region of interest (e.g., avoiding regions identified as uncomfortable). Follow-up work explores safety-aware preference-based optimization, multi-objective preference learning, and deployment of these frameworks on real-world systems.
- **Robust locomotion** (Chapter 5):
    1. Develop *offline robustness metrics* that evaluate existing controllers—validated by tracking nominal offline-generated trajectories—using a hybrid forward invariance framework to quantify disturbance tolerance.

2. Design *online adaptation* methods for rapid replanning via Data-Driven Predictive Control (DDPC) and its hybrid extension (HDDPC), which incorporate both continuous swing dynamics and discrete step-to-step transitions. These methods leverage hardware data to construct data-driven reduced-order models that capture user variability, while enabling real-time replanning to reject external perturbations.
3. Implement *learning-based generalization* through reinforcement learning policies trained in simulation with domain randomization and Control Lyapunov Function (CLF)-based reward shaping, enabling robust generalization beyond training conditions and reliable transfer to hardware.

While each contribution was developed in isolation, they address complementary components of the locomotion control stack and could in principle be integrated into unified frameworks. For example, offline robustness metrics could be used to shape the reference trajectories guiding reinforcement learning, or preference based personalization could be combined with any of the robustness strategies.

Through the integration of biomechanics, user informed modeling, principled robustness analysis, and learning based control, this thesis advances locomotion controllers that are stable, robust, and user-aligned, demonstrating applicability across both humanoid robots and assistive devices.

Chapter 2

HYBRID DYNAMICS OF BIPEDAL LOCOMOTION

Bipedal walking presents unique challenges for robotic and assistive systems due to its hybrid dynamics, underactuation, and sensitivity to variability in users and environments. To address these challenges, this chapter reviews the theoretical and algorithmic foundations that underpin the locomotion strategies developed in this thesis. We begin with the hybrid dynamical models that describe legged locomotion, covering both full-order representations and reduced-order templates. Building on these models, we review model-based motion synthesis techniques—ranging from periodic orbit design with Hybrid Zero Dynamics (HZD) to receding-horizon optimization and Lyapunov/barrier function methods. We then discuss reinforcement learning approaches that complement model-based control by enabling data-driven adaptation. Finally, we provide context in assistive devices and describe the robotic platforms used as experimental testbeds. Together, these sections establish the modeling, control, and learning tools that will be referenced throughout the dissertation.

Organization of this chapter. Section 2.1 introduces hybrid models of legged locomotion, including continuous dynamics, impact dynamics, step-to-step maps, and reduced-order templates. Section 2.2 then presents model-based motion synthesis methods, covering periodic orbit design (HZD, H-LIP), receding-horizon optimization, and stability and safety tools such as CLFs and CBFs. Section 2.3 reviews reinforcement learning approaches for locomotion, emphasizing their complementarity with model-based techniques. Section 2.4 discusses locomotion in assistive devices, highlighting challenges in stability, clinical use, and personalization. Finally, Section 2.5 describes the robotic platforms used in this work.

2.1 Modeling of Legged Locomotion

System Setup

We begin by defining the generalized coordinates and state variables used to describe the dynamics of a legged robot. Consider a system with generalized coordinates $q = [q_b^\top, q_a^\top]^\top \in \mathcal{Q} \subset \mathbb{R}^n$, where $q_b \in SE(3)$ represents the floating-base pose and

$q_a \in \mathbb{R}^m$ the actuated degrees of freedom (DoFs). The control input is $u \in \mathbb{R}^m$, and the full-order state is $x = [q^\top,\, \dot{q}^\top]^\top \in T\mathcal{Q}$, with $T\mathcal{Q}$ denoting the tangent bundle of the configuration manifold.

This formulation serves as the basis for modeling the hybrid dynamics of walking, in which continuous evolution of the state is interleaved with discrete events such as foot impacts and liftoff.

Hybrid Dynamics

This structure can be formalized as a hybrid control system, consisting of continuous domains that describe the dynamics within a fixed contact mode and discrete transitions (edges) that model contact events [4].

Let $\mathcal{D} \subset \mathcal{X}$ denote the admissible domain in which the continuous-time dynamics evolve, and let $\mathcal{S} \subset \mathcal{D}$ denote the *guard* (or *switching surface*) that triggers discrete transitions. For a continuously differentiable function[1] $h : \mathcal{X} \to \mathbb{R}$, these sets can be defined as

$$\mathcal{D} = \{x \in \mathcal{X} \mid h(x) \geq 0\}, \tag{2.1}$$

$$\mathcal{S} = \{x \in \mathcal{X} \mid h(x) = 0,\ \dot{h}(x) < 0\}. \tag{2.2}$$

The hybrid system $\mathcal{H}$ is then

$$\mathcal{H} : \begin{cases} \dot{x} = f(x) + g(x)u & x \in \mathcal{D} \setminus \mathcal{S}, \quad (2.3) \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S}, \quad (2.4) \end{cases}$$

where (2.3) describes the continuous full-order Lagrangian dynamics, $\Delta : \mathcal{S} \to \mathcal{D}$ is the *reset map* applied at discrete events, and the superscripts "$-$" and "$+$" indicate states immediately before and after the event, respectively. In practice, events such as impact are often detected when the swing foot height equals the ground height, while lift-off occurs when the normal ground reaction force becomes zero.

The following subsections examine these two components in more detail: the continuous-time dynamics that govern motion within a single contact mode, and the impact dynamics that capture discrete changes in velocity at contact events.

[1] h must be chosen so that it is not in the null space of the actuation matrix, i.e., $L_g h(x) \neq 0$.

Continuous-Phase Dynamics

Within each continuous domain of the hybrid system, the robot's motion evolves according to the Euler–Lagrange equations with holonomic contact constraints:

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_h(q)^\top F, \tag{2.5}$$

$$J_h(q)\ddot{q} + \dot{J}_h(q,\dot{q})\dot{q} = 0, \tag{2.6}$$

where $D(q) \in \mathbb{R}^{n\times n}$ is the mass–inertia matrix, $H(q,\dot{q}) \in \mathbb{R}^n$ collects Coriolis and gravitational terms, and $B \in \mathbb{R}^{n\times m}$ is the actuation matrix. The holonomic contact constraints are captured by the Jacobian $J_h(q) \in \mathbb{R}^{h\times n}$, with associated constraint wrench $F \in \mathbb{R}^h$. In single-support with a planar patched contact, there are typically $h = 6$ independent constraints.

While the same dynamics can equivalently be written in Newton–Euler form using momentum balances, the Euler–Lagrange representation offers a compact coordinate-based formulation that is particularly convenient for virtual constraints, trajectory optimization, and control design.

Impact Dynamics

At the end of a continuous phase, when the swing foot strikes the ground or a new contact is established, the system undergoes an instantaneous change in velocity due to the contact impulse. During this event, the configuration remains continuous ($q^+ = q^-$), while the velocities experience a discontinuity. The impact can be modeled using the impulse–momentum form of the Euler–Lagrange equations with holonomic constraints:

$$J_c(q^-)\dot{q}^+ = 0, \tag{2.7}$$

$$\begin{bmatrix} D(q^-) & -J_c(q^-)^\top \\ J_c(q^-) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \delta F \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix}, \tag{2.8}$$

where $J_c(q^-)$ is the contact Jacobian at impact, and δF is the net contact impulse over the infinitesimal impact interval.

Solving (2.8) for $\dot{q}^+$ yields the velocity reset map:

$$x^+ = \Delta(x^-) \triangleq \begin{bmatrix} Rq^- \\ R\big(-D^{-1}J_c^\top (J_c D^{-1} J_c^\top)^{-1} J_c + I\big)\,\dot{q}^- \end{bmatrix}, \tag{2.9}$$

where R is a relabeling matrix that accounts for changes in generalized coordinate indexing between domains.

Together with the continuous dynamics, this reset law fully specifies the hybrid model of walking. In the next subsection, we examine how these hybrid dynamics give rise to discrete step-to-step evolution through the Poincaré return map.

Step-to-Step (S2S) Dynamics and Poincaré Analysis

While continuous and impact dynamics describe motion within and across individual phases, it is often more insightful to analyze walking at the granularity of successive steps. This motivates a discrete-time perspective in which locomotion is characterized by the *Poincaré return map*, a classical tool in nonlinear dynamics for studying the stability of periodic motions. The idea is to define a section of the state space intersected once per cycle (e.g., the guard corresponding to foot impact in walking) and to track how the state evolves from one intersection to the next. In this way, the problem of analyzing the orbital stability of a continuous gait reduces to studying the stability of a fixed point of a finite-dimensional discrete-time map.

Formally, using the guard $\mathcal{S}$ of the hybrid system $\mathcal{H}$, the Poincaré map $P : \mathcal{S} \to \mathcal{S}$ is defined as

$$P(x^-) \triangleq \varphi_{T_I(x^-)}\big(\Delta(x^-)\big), \tag{2.10}$$

where $\varphi_t(\cdot)$ denotes the flow of the continuous closed-loop dynamics, Δ is the reset map, and $T_I : \mathcal{S} \to \mathbb{R}> 0$ is the *time-to-impact function* measuring the duration until the next guard crossing. The sequence x_k generated by $x_{k+1} = P(x_k)$ thus represents the hybrid system's state evaluated at successive impacts.

A periodic orbit $\mathcal{O}$ corresponds to a fixed point $x^* \in \mathcal{S}$ satisfying $P(x^*) = x^*$. Equivalently, the flow satisfies A n $\varphi_t(x_0)$ for some period period $T > 0$ and the orbit can be written as

$$\mathcal{O} := \{\varphi_t(\Delta(x^*)) \in \mathcal{D} \mid 0 \le t \le T_I\}. \tag{2.11}$$

As shown in Theorem 1 of [31], $\mathcal{O}$ is *exponentially stable* if and only if x^* is an exponentially stable fixed point of the discrete-time system. That is, there exist constants $M > 0$ and $\alpha \in (0, 1)$ such that

$$\|P^i(x) - x^*\| \le M\,\alpha^i\,\|x - x^*\|, \quad \forall x \in B_\delta(x^*), \tag{2.12}$$

with P^i denoting applying P i times in succession. This equivalence makes orbital stability of walking amenable to analysis using standard tools from discrete-time nonlinear systems theory.

Extended Poincaré Map. The domain of P can be locally extended beyond $\mathcal{S}$ by defining an *extended time-to-impact function* $T_e : B_\rho(x^) \subset \mathcal{D} \to \mathbb{R}$ via

$$h\big(\varphi_{T_e(x)}(\Delta(x))\big) = 0, \tag{2.13}$$

which is well-defined in a neighborhood $B_\rho(x^)$ of the fixed point under regularity conditions on h. The *extended Poincaré map* is then

$$P_0(x) \triangleq \varphi_{T_e(x)}(\Delta(x)), \quad x \in B_\rho(x^*), \tag{2.14}$$

enabling the analysis of robustness to perturbations for initial conditions not lying exactly on $\mathcal{S}$.

Robustness Considerations. The stability of a periodic orbit can be certified through the Poincaré map, but this analysis is inherently local: it guarantees only that sufficiently small perturbations decay in the vicinity of the orbit. Robustness, by contrast, addresses whether desirable behavior is preserved under more realistic sources of uncertainty, including modeling errors, parameter variations, timing discrepancies, and external disturbances. A gait that converges rapidly in the linearized sense may still fail under moderate perturbations if its basin of attraction is narrow. Thus, while stability characterizes local convergence under nominal dynamics, robustness quantifies a system's ability to maintain performance and safety in the presence of uncertainty. To formalize robustness, hybrid-systems extensions of input-to-state stability (ISS) and input-to-state safety (ISSf) have been proposed. Yet verifying such properties for high-dimensional legged robots remains computationally prohibitive, motivating the development of reduced-order models and tractable approximations, as discussed in the next subsection.

Reduced-Order Models

Reduced-order models (ROMs) approximate locomotion dynamics by restricting attention to low-dimensional structures that capture the dominant behaviors of the full system. Rather than simply neglecting higher-order effects, they abstract away detailed joint-level dynamics in favor of templates—such as inverted pendulum

or spring-mass models—that preserve the essential mechanics of balance, energy exchange, and periodicity. These abstractions vary in fidelity and computational cost, but provide tractable tools for analysis, controller design, and reasoning about robustness.

A common starting point is the centroidal momentum dynamics:

$$m(a_{\text{com}} + g) = \sum_{i \in \mathcal{C}} f_i, \tag{2.15}$$

$$\dot{L} = \sum_{i \in \mathcal{C}} (r_i - \mathrm{p}_{\text{com}}) \times f_i, \tag{2.16}$$

where m is the total mass, p_{com} and L_{com} denote the center of mass (CoM) position and angular momentum, and f_i, τ_i are the ground reaction forces and moments at contact i.

Assuming $\dot{L}_{\text{com}} = 0$ yields a model where balance is governed solely by the interplay between CoM motion and contact forces:

$$\mathrm{p}^{x,y}_{\text{cop}} = \mathrm{p}^{x,y}_{\text{com}} - \frac{\mathrm{p}^{z}_{\text{com}} - \mathrm{p}^{z}_{\text{cop}}}{a^{z}_{\text{com}} + g}\, a^{x,y}_{\text{com}}, \tag{2.17}$$

which expresses the center of pressure (CoP) as a function of CoM position and acceleration. Feasibility is enforced by requiring the CoP to remain within the convex hull of active contact points, ensuring that the corresponding ground reaction forces are physically realizable.

A further simplification assumes constant CoM height $\mathrm{p}^{z}_{\text{com}} = z_0$ and negligible vertical acceleration $a^{z}_{\text{com}} \approx 0$. In this case, the relation reduces to the *Linear Inverted Pendulum* (LIP) dynamics:

$$a^{x}_{\text{com}} = \omega_0^2 \big(\mathrm{p}^{x}_{\text{com}} - \mathrm{p}^{x}_{\text{cop}}\big), \tag{2.18}$$

$$a^{y}_{\text{com}} = \omega_0^2 \big(\mathrm{p}^{y}_{\text{com}} - \mathrm{p}^{y}_{\text{cop}}\big), \tag{2.19}$$

with natural frequency $\omega_0 = \sqrt{g/z_0}$. The LIP model has become a cornerstone in locomotion research, as it admits closed-form solutions for capture points and step adjustment strategies [32, 33], providing both practical tools for gait generation and theoretical insights into balance recovery.

Hybrid-LIP (H-LIP). To better reflect the hybrid structure of walking, the *Hybrid Linear Inverted Pendulum* (H-LIP) [34] partitions the motion into single-support

(SSP) and double-support (DSP) phases, while retaining the constant CoM height assumption:

$$a_{\text{com}}^{x,y} = \frac{g}{\text{p}_{\text{com}}^{z}} \text{p}_{\text{com}}^{x,y}, \tag{SSP}$$

$$a_{\text{com}}^{x,y} = 0, \tag{DSP}$$

where the $\text{p}_{\text{com}}^{z}$ and domain duration (T_{SSP}, T_{DSP}) are constant. Assuming no velocity jump, we can combine the two domains with step size $u^{\text{H-LIP}}$ equivalently as:

$$\Delta_{\text{SSP}^- \to \text{SSP}^+} : \begin{cases} \text{v}_{\text{com}}^{+} = \text{v}_{\text{com}}^{-} \\ \text{p}_{\text{com}}^{+} = \text{p}_{\text{com}}^{-} + \text{v}_{\text{com}}^{-} T_{\text{DSP}} - u^{\text{H-LIP}}. \end{cases}$$

Assuming smooth transitions between these domains, the dynamics can be compactly expressed in the step-to-step (S2S) format:

$$x_{k+1}^{\text{H-LIP}} = A x_k^{\text{H-LIP}} + B u_k^{\text{H-LIP}} \tag{2.20}$$

$$x^{\text{H-LIP}} \triangleq \begin{bmatrix} \text{p}_{\text{com}} \\ \text{v}_{\text{com}} \end{bmatrix}, \tag{2.21}$$

which provides a linear approximation of the Poincaré return map of the full-order system. Model discrepancies between the H-LIP predictions and the full-order dynamics can be treated as bounded disturbances. With state-feedback stepping controllers, these disturbances can be actively regulated, ensuring that the tracking errors converge to disturbance-invariant sets.

These reduced-order templates are particularly appealing for real-time MPC due to their computational efficiency and analytic tractability. At the same time, their simplifying assumptions (e.g., constant CoM height, fixed timing) limit their fidelity. This trade-off has motivated hybrid schemes that combine ROM-based planning for efficiency with full-order feedback control for accuracy and robustness.

Summary: Modeling of Legged Locomotion

The modeling of legged locomotion spans a spectrum of fidelity and abstraction. At the full-order level, hybrid models capture continuous evolution interleaved with discrete impacts, with Poincaré analysis providing rigorous tools for orbital stability. Reduced-order models, by contrast, distill locomotion into low-dimensional templates that enable rapid analysis and planning. Collectively, these approaches range from

exact but high-dimensional representations to simplified yet computationally efficient ones.

Yet beyond this technical spectrum lies a deeper trade-off that roboticists must navigate based on their specific applications. Explicit modeling of domain transitions improves accuracy but increases computational burden; smoothing approximations reduce complexity but risk losing critical hybrid effects. Reduced-order models are appealing when their assumptions align with the hardware, offering fewer tuning parameters, rapid replanning, and easier debugging. When those assumptions are violated, however, their representations may be so poor that computational efficiency no longer yields practical benefit.

Conversely, striving for full-order optimality often exposes fragility: exact optimal solutions demand near-perfect models, and small mismatches can break the strategy entirely. In practice, feasibility and robustness often matter more than strict optimality. Thus, modeling is ultimately a design choice, shaped by the platform and the goals of the application. While this thesis introduces methods with broad scope, each emphasizes resolving the concrete challenges encountered on the evaluated systems. These considerations motivate the next section, which turns from modeling to model-based motion synthesis for generating feasible and stable gaits.

2.2 Model-based Motion Synthesis

Building on the preceding models, we now consider motion synthesis: approaches that exploit representations of legged dynamics—ranging from reduced-order templates to full hybrid formulations—to generate gaits that satisfy physical constraints, stability criteria, and robustness demands.

Gait Generation via Periodic Orbit Design

As discussed in the modeling section, periodic orbits of the hybrid dynamics characterize walking gaits and their stability can be assessed through Poincaré analysis. Here, we shift from analysis to synthesis, focusing on how such orbits can be designed to realize desired locomotion behaviors.

In this design perspective, the *step-to-step* (S2S) Poincaré map

$$P : \mathcal{S} \to \mathcal{S},$$

provides a reduced representation of the hybrid dynamics by capturing the evolution of the system across impacts. A periodic gait corresponds to a fixed point x^* satisfying $P(x^*) = x^*$. Orbital (exponential) stability is determined by the eigenstructure of the linearization $DP(x^*)$: the gait is stable if and only if the spectral radius satisfies $\rho(DP(x^*)) < 1$, i.e., all eigenvalues lie strictly inside the unit disk [31, 35].[2]

The power of this formulation lies in its discreteness: it compresses the complexity of hybrid locomotion into a step-level description, turning the problem of stable walking into the problem of shaping the return map to admit a stable fixed point. From an optimal control perspective, this design effectively reduces the infinite-horizon stability problem to a finite-horizon one over a single step. The challenge, however, is that fixed-point stability alone does not guarantee convergence from arbitrary initial conditions or smooth transitions between distinct gaits. These limitations motivate a spectrum of design methods, from full-order approaches such as Hybrid Zero Dynamics (HZD), which enforce invariance conditions on the full model, to reduced-order approaches like the H-LIP, which trade fidelity for tractability while preserving the step-to-step structure.

Hybrid Zero Dynamics and Virtual Constraints

The Hybrid Zero Dynamics (HZD) framework is a model-based approach for designing stable walking gaits in legged robots. Its key insight is that exponentially stable periodic orbits of the zero dynamics—the lower-dimensional dynamics when selected outputs are driven to zero—correspond to exponentially stabilizable orbits of the full hybrid system [4]. In nonlinear control, the zero dynamics surface is the invariant manifold on which these internal dynamics evolve [36]. Extending this notion to systems with impacts yields the hybrid zero dynamics surface, which provides a rigorous foundation for analyzing and synthesizing stable periodic gaits.

We define the zero dynamics surface as

$$\mathcal{Z}_\alpha \triangleq \{x \in \mathcal{D} \mid y_\alpha(x) = 0,\ \dot{y}_\alpha(x) = 0\},$$

where $y_\alpha : \mathcal{X} \to \mathbb{R}^m$ denotes the *virtual constraints*—outputs that, when driven to zero, regulate the robot's motion to a desired trajectory. Virtual constraints are typically defined as

$$y_\alpha(x) = y^a(x) - y^d_\alpha(\tau(x)), \tag{2.22}$$

[2] In practice, neutral directions associated with invariances (e.g., absolute horizontal position) are excluded when assessing stability.

with $y^a : \mathcal{X} \to \mathbb{R}^m$ being the actual outputs (e.g., joint angles or end-effector positions) and y^d_α the desired outputs parameterized by a phase variable $\tau(x)$ that is monotonically increasing along the gait cycle. A stabilizing controller $u^*(x)$—for example via feedback linearization or a control Lyapunov function—drives $y_\alpha \to 0$, yielding the closed-loop dynamics $\dot{x} = f_{\text{cl}}(x) = f(x) + g(x)u^*(x)$.

A common choice for y^d_α is a set of Bézier polynomials solved via numerical optimization:

$$\text{Bez}(\tau, \alpha) \triangleq \sum_{k=0}^{B} \alpha[k] \frac{B!}{k!(B-k)!} \tau^k (1-\tau)^{B-k}, \tag{2.23}$$

where $\alpha_v \in \mathbb{R}^{B+1}$ are control points for output v. Bézier polynomials offer smoothness, boundedness by control points, and analytically tractable derivatives [37, 38]. The phase variable is often normalized to $\tau \in [0, 1]$:

$$\tau(x) \triangleq \frac{\theta(x) - \theta^+}{\theta^- - \theta^+}, \tag{2.24}$$

where $\theta(x)$ is a monotonic quantity (e.g., horizontal hip position) and θ^+, θ^- are its values at the start and end of the gait cycle.

Hybrid Invariance. For a periodic gait to be preserved under impacts, $\mathcal{Z}_\alpha$ must be invariant under the reset map Δ:

$$\Delta(\mathcal{Z}_\alpha \cap \mathcal{S}) \subset \mathcal{Z}_\alpha, \tag{2.25}$$

where $\mathcal{S}$ is the switching surface. When this *impact invariance* condition holds, the intersection $\mathcal{Z}_\alpha \cap \mathcal{S}$ corresponds to a fixed point of the step-to-step (Poincaré) return map, ensuring the gait is both periodic and stable by construction.

When this *impact invariance* condition holds, the intersection $\mathcal{Z}_\alpha \cap \mathcal{S}$ corresponds to a fixed point of the step-to-step (Poincaré) return map, ensuring the gait is both periodic and stable by construction. In practice, trajectory optimization is often carried out on a reduced set of coordinates via the *partial hybrid zero dynamics* (PHZD) formulation, which enables efficient numerical solution [39, 40]; we return to these details in later chapters when applying HZD to concrete gait synthesis problems.

Trajectory Optimization To obtain HZD trajectories, we solve an offline trajectory optimization problem to determine the Bézier coefficients α that define the virtual

constraints:

$$\{\alpha^*, X^*\} = \underset{\alpha, X}{\operatorname{argmin}}\ \Phi(X) \tag{2.26}$$

$$\begin{aligned} \text{s.t.} \quad & \dot{x} = f(x) + g(x)u && \text{(Dynamics)} \\ & \Delta(\mathcal{S} \cap \mathcal{Z}_\alpha) \subset \mathcal{Z}_\alpha && \text{(HZD Condition)} \\ & X_{\text{min}} \preceq X \preceq X_{\text{max}} && \text{(Decision Variables)} \\ & c_{\text{min}} \preceq c(X) \preceq c_{\text{max}}, && \text{(Physical Constraints)} \end{aligned}$$

where $X = (x_0, ..., x_N, T) \in \mathbb{R}^{n_d}$ denotes the collection of decision variables, with $n_d \in \mathbb{N}$. Each $x_i \in \mathcal{X}$ represents the state at node i, and $T \in \mathbb{R}_{>0}$ denotes the duration of the limit cycle. The cost function is given $\Phi : \mathbb{R}^{n_d} \to \mathbb{R}_{\geq 0}$, while the constraints are represented by $c : \mathbb{R}^{n_d} \to \mathbb{R}^{n_p}$ with $n_p \in \mathbb{N}$. These constraints encode the physical laws governing locomotion, such as friction cone conditions, workspace limits, and actuator capacity [5]. Solving this optimization yields a stable periodic solution to the walking dynamics, parameterized by a fixed set of Bézier coefficients $\alpha^* \in \mathbb{R}^{n_o \times B+1}$.

Here, $y(q)$ denotes the controlled outputs, and $\text{Bez}(\tau; \alpha)$ is the Bézier-parameterized desired trajectory, with $\tau(t) = \phi(x, t)$ serving as the phase variable that encodes progression along the gait. In practice, this problem is typically formulated as a nonlinear program. Toolboxes such as FROST [41] provide a convenient framework for setting up the optimization, automatically generating the dynamics, constraints, and Jacobians, and interfacing with solvers such as IPOPT through MATLAB. More recently, pipelines leveraging CasADi and Pinocchio have become attractive alternatives, providing automatic differentiation and symbolic modeling within Python. These capabilities support scalable trajectory optimization and facilitate integration with contemporary robotics software stacks [42].

Hybrid Linear Inverted Pendulum (H-LIP)

Building on the reduced-order model introduced in Section 2.1, the H-LIP provides a analytical gait library that has close form solution. Given a desired walking velocity v^d, one solves the step-to-step (S2S) map

$$x_{k+1}^{\text{H-LIP}} = A x_k^{\text{H-LIP}} + B u_k^{\text{H-LIP}} \tag{2.27}$$

for a fixed point x^* with corresponding nominal step length $u^* = v^d T$. This defines a periodic center-of-mass (CoM) orbit $x^* = [p_{\text{com}}, v_{\text{com}}]^\top$ from which the continuous CoM trajectory of each step can be generated analytically.

On top of this CoM motion, the swing foot is planned using a separate Bézier curve to guarantee smooth lift-off, clearance, and touchdown. Additional degrees of freedom, such as torso pitch, knee flexion, or arm swing, can be specified via simple periodic functions synchronized with the gait phase. Together, these trajectories form a set of reference signals that can be tracked by the full-order robot.

Because the linear S2S map allows closed-form solutions, the H-LIP readily supports simple stabilizing feedback laws. For example, a deadbeat foot-placement controller—reminiscent of Riabert's stepping strategy—can drive the system back to its nominal orbit within a single step. Moreover, the same framework naturally extends to multi-step regulation strategies, such as period-2 gaits, providing additional flexibility for achieving stability. These features make the H-LIP particularly attractive for online planning and control, even though it necessarily sacrifices some modeling fidelity compared to full-order methods like HZD.

Summary and Trade-offs. HZD and H-LIP both aim to synthesize stable periodic walking gaits, but they operate at different modeling levels with distinct computational consequences. HZD works directly with the full-order hybrid dynamics, enforcing virtual constraints whose hybrid invariance guarantees the existence and stability of a periodic orbit in the underlying step-to-step map, though the S2S perspective is less explicit. This provides strong guarantees on stability and physical feasibility but requires solving high-dimensional nonlinear programs, which are typically suited for offline trajectory design.

In contrast, H-LIP starts from a reduced-order hybrid template that approximates the Poincaré map with an affine linear step-to-step model. This structure admits closed-form foot-placement and CoM controllers, supporting lightweight online stabilization. By explicitly parameterizing the return map, H-LIP also captures multi-step periodic orbits (e.g., period-2 gaits), offering a practical mechanism to recover stability when immediate convergence is not possible.

Taken together, these methods illustrate a core trade-off between model fidelity and computational tractability. HZD offers rigorous guarantees but at high computational cost, while H-LIP sacrifices some physical realism for simplicity and real-time viability. Modern hybrid control strategies—such as MPC and learning-based approaches—often combine these perspectives, using reduced-order models for fast planning and full-order models for accurate stabilization.

Gait Generation via Receding-Horizon Optimization

While effective for steady-state behaviors, purely periodic gaits lack flexibility in dynamic or uncertain environments, where online adaptation and smooth transitions between behaviors are required. Moreover, naively tracking a precomputed offline trajectory cannot ensure satisfaction of essential constraints such as joint torque limits, contact conditions, or foot placement feasibility, nor can it robustly handle disturbances.

To overcome these limitations, gait generation can be posed within a receding-horizon *Model Predictive Control* (MPC) framework, which enables online replanning and provides stabilizing feedback. MPC formulates a finite-horizon optimal control problem:

$$\begin{aligned}
\min_{u(\cdot)} \quad & \phi(x(t_H)) + \int_0^{t_H} l(x(t), u(t), t)\,\mathrm{d}t, && \text{(2.28a)}\\
\text{subject to:} \quad & x(0) = x_0, && \text{(2.28b)}\\
& \dot{x} = f(x) + g(x)u, && \text{(2.28c)}\\
& c_{\text{eq}}(x, u, t) = 0, && \text{(2.28d)}\\
& c_{\text{in}}(x, u, t) \geq 0, && \text{(2.28e)}
\end{aligned}$$

where t_H is the horizon length, ϕ and l are the terminal and running costs, and c_{eq}, c_{in} are equality and inequality constraints. The optimization is resolved online at each control step using the measured state x_0.

By solving (2.28) in real time with full-order dynamics, MPC can adapt footstep placement, body motion, and contact forces online, while respecting physical constraints. However, full-order nonlinear MPC for underactuated bipeds faces the same challenges as directly running HZD online: the high dimensionality of the dynamics and contact constraints makes real-time solution difficult. Common strategies to alleviate this include replacing the full dynamics with a reduced-order model (at the cost of model fidelity), or retaining the full-order model but linearizing around a nominal trajectory and solving a sequence of quadratic programs (SQP) with line search. These approaches trade optimality for computational tractability, allowing higher replanning rates. A critical design choice lies in the terminal constraints: they play a role analogous to the periodic orbit perspective in gait design, since the quality of the generated trajectories depends heavily on whether the terminal conditions provide a good proxy for long-term stability and performance.

Summary: Model-Based Motion Synthesis

The approaches described above reflect a spectrum of strategies for generating walking gaits from first principles. HZD provides a full-order framework grounded in hybrid zero dynamics, ensuring stability through virtual constraints but requiring offline optimization. At the opposite end, reduced-order templates such as the H-LIP enable lightweight, closed-form regulation and online planning, though at the cost of strong simplifying assumptions. Receding-horizon formulations such as MPC strike a balance between these extremes, leveraging optimization to enforce feasibility and adaptability in real time, while relying on reduced-order models or linearizations to remain computationally tractable.

Other classical approaches have also played influential roles in shaping legged locomotion control. Raibert's foot-placement heuristics demonstrated that simple step-to-step laws could stabilize hopping and running robots with remarkable robustness. Zero-Moment Point (ZMP) methods provided an industry-standard framework for humanoid robots, ensuring balance by constraining the CoP within the support polygon, though typically assuming quasi-static motion. Passive dynamic walking offered the counterpoint that carefully designed morphology alone can generate stable gaits without active control, motivating reduced-order templates and energetic perspectives on locomotion.

What unifies these approaches is not a single shared mechanism but rather a spectrum of trade-offs. HZD emphasizes model fidelity through full-order dynamics, while H-LIP prioritizes computational efficiency with a reduced-order approximation. HZD trajectories are typically designed offline, whereas MPC enables online replan in real time. Similarly, HZD and H-LIP focus on orbit stability and analytic guarantees, while MPC highlights the flexible and generic structure. Viewed together, these strategies illustrate the strengths and limitations of model-based synthesis and motivate hybrid or learning-augmented methods that aim to combine principled guarantees with greater adaptability.

Control Lyapunov Functions (CLFs)

Whereas the previous methods focused on generating feasible periodic or receding-horizon trajectories, Control Lyapunov Functions (CLFs) provide a complementary tool for analyzing and enforcing stability in nonlinear systems. A CLF offers a scalar energy-like certificate whose decrease under suitable feedback control guarantees

asymptotic convergence of the system to a desired set or trajectory. More concretely, consider a control-affine system of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \mathbb{R}^n,\ \mathbf{u} \in \mathbb{R}^m. \tag{2.29}$$

A continuously differentiable, positive definite function $V : \mathbb{R}^n \to \mathbb{R}$ is called an *exponentially stabilizing control Lyapunov function* (ES-CLF) if there exist constants $k_1, k_2, k_3 > 0$ such that

$$k_1\|\mathbf{x}\|^2 \leq V(\mathbf{x}) \leq k_2\|\mathbf{x}\|^2, \tag{2.30}$$

$$\inf_{\mathbf{u}\in\mathbb{R}^m} \dot{V}(\mathbf{x}, \mathbf{u}) \leq -k_3 V(\mathbf{x}). \tag{2.31}$$

The inequality (2.31) ensures that there exists at least one admissible control input that causes V to decrease exponentially along system trajectories, implying convergence to the origin (or a desired manifold).

Quadratic CLFs from Linearization. For a linearized closed-loop system $\dot{\eta} = A_{\text{cl}}\eta$, the continuous-time Lyapunov equation

$$A_{\text{cl}}^\top P + P A_{\text{cl}} = -Q, \quad Q \succ 0 \tag{2.32}$$

admits a unique positive definite solution $P \succ 0$. The quadratic form

$$V(\eta) = \eta^\top P \eta \tag{2.33}$$

then satisfies (2.30) and

$$\dot{V}(\eta, \mathbf{v}) = L_F V(\eta) + L_G V(\eta)\mathbf{v} \leq -\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} V(\eta), \tag{2.34}$$

thus serving as an ES-CLF on the output coordinates.

Nonlinear Systems and Hybrid Dynamics. For a general nonlinear control-affine system, the ES-CLF condition can equivalently be written as

$$\inf_{\mathbf{u}\in\mathbb{R}^m} \nabla_{\mathbf{x}} V(\mathbf{x})\,[f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}] \leq -\lambda V(\mathbf{x}), \quad \lambda > 0, \tag{2.35}$$

which guarantees exponential decay of V along trajectories. In hybrid systems such as bipedal locomotion, CLFs can be applied within each continuous phase. If the decay rate is sufficiently fast, it can offset the destabilizing effects of discrete impacts [43], enabling stability guarantees across hybrid transitions.

Flexibility in Control Design. The CLF condition (2.35) specifies a set of admissible controls rather than a unique feedback law. This flexibility makes CLFs particularly attractive for reinforcement learning: instead of enforcing a fixed stabilizing controller, one can embed the CLF inequality into the reward design, encouraging learned policies to exhibit Lyapunov-stable behavior while retaining freedom in control structure.

Control Barrier Functions

In parallel with the Lyapunov-based notion of stability, barrier functions provide a formalism for certifying the safety of a dynamical system. For controlled systems, this concept is extended through *Control Barrier Functions* (CBFs), which guarantee that the state remains within a prescribed safe set.

Definition 1 (Control Barrier Function)**.** Let $\mathcal{C} \subset \mathbb{R}^n$ denote the *safe set*, defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} := \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}. \tag{2.36}$$

The function h is a *Control Barrier Function* for the control-affine system

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \tag{2.37}$$

if there exists an extended class $\mathcal{K}_\infty$ function $\alpha \in \mathcal{K}_\infty^e$ such that, for all $\mathbf{x} \in \mathbb{R}^n$,

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \left[L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} \right] \geq -\alpha\left(h(\mathbf{x})\right). \tag{2.38}$$

Intuitively, the CBF condition (2.38) ensures that, at every state in the domain, there exists at least one admissible control input capable of preventing the system from leaving $\mathcal{C}$, thereby guaranteeing *forward invariance* of the safe set.

Remark 1. While CBFs offer a powerful safety guarantee, verifying (2.38) globally is generally difficult, and the synthesis of valid CBFs for complex systems remains an open challenge. For a detailed treatment of CBF theory and applications, we refer the reader to [44, 45].

2.3 Learning-Based Motion Synthesis

The periodic and receding-horizon approaches described above both rely on explicit models of the robot's dynamics and contact schedule. While these methods can yield

strong performance when the model is accurate, they may struggle in the presence of model mismatch, unmodeled compliance, or complex contact interactions. To address these limitations, a broad class of data-driven approaches has emerged, leveraging measured trajectories or direct interaction data to improve gait generation. These methods reduce reliance on precise analytical models and instead exploit the rich statistical structure present in data.

Imitation and Demonstration Learning

Another paradigm uses expert data to shape locomotion strategies. Imitation learning leverages motion-capture datasets (human or animal locomotion) or demonstrations from existing controllers (e.g., HZD gaits) to train neural networks or regression models that map state to actions. These methods bypass the need for reward engineering, instead learning to reproduce expert trajectories directly. Extensions such as Generative Adversarial Imitation Learning (GAIL) or behavior cloning with regularization allow controllers to generalize beyond training demonstrations while retaining natural motion quality.

Residual and Hybrid Learning

Data-driven methods can also be used to enhance model-based controllers. For example, a reduced-order gait generator (e.g., H-LIP or HZD) can be augmented with a learned residual policy that compensates for unmodeled dynamics or hardware-specific effects. This hybridization retains interpretability and structure while exploiting the adaptability of learning. In practice, residual learning has been shown to improve terrain robustness, energy economy, and comfort by fine-tuning foot placement, compliance responses, or user-specific dynamics.

Reinforcement Learning

RL formulates locomotion control as a *Markov Decision Process* (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p(s'|s,a)$ defines the transition dynamics, $r(s,a)$ is the reward function, and $\gamma \in (0,1]$ is the discount factor. A stochastic policy $\pi_\theta(a|s)$, parameterized by θ, is trained to maximize the expected discounted return

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t)\right].$$

Depending on how dynamics are treated, RL methods can be model-free (learning a policy directly from data) or model-based (learning an explicit dynamics model to plan or improve sample efficiency). In bipedal locomotion, RL offers the ability to handle strong nonlinearities and uncertainties without precise modeling, to leverage high-dimensional observations (e.g., multi-axis force sensing, vision), and to flexibly encode task-specific objectives such as energy economy, comfort, or terrain agility. At the same time, RL faces challenges: training is often data-hungry, direct hardware learning is impractical without careful safety measures, formal stability guarantees are generally absent, and performance can be sensitive to reward design and exploration strategy. These issues motivate *simulation-to-real* transfer techniques and *structure-guided* rewards or constraints.

From an optimization perspective, RL adjusts policy parameters to maximize long-horizon cumulative reward, rather than explicitly planning trajectories as in MPC, enforcing virtual constraints to render a hybrid zero-dynamics manifold and synthesize periodic orbits as in HZD, or using reduced-order templates with explicit step-to-step maps (e.g., LIP/H-LIP) for foot-placement control. In this sense, RL is complementary to model-based methods: it can be combined with reduced-order references, hybrid-invariance designs, and control-theoretic ingredients (e.g., CLF-inspired objectives or CBF safety filters) to guide learning toward policies that are more stable, robust, and transferable to hardware.

2.4 Assistive Device

Robotic assistive devices, particularly powered lower-limb exoskeletons, hold significant promise for restoring or augmenting mobility in individuals with impaired motor function. While human locomotion has been extensively studied in biomechanics research for non-disabled individuals [15], translating the principles of natural, efficient walking to robotic platforms remains a challenge. This is especially true for assistive devices, which must achieve not only mechanical stability but also seamless physical cooperation with the user. Beyond mobility restoration, lower-body exoskeletons offer numerous clinical benefits for individuals with complete motor paraplegia, including pressure relief, improved circulation, enhanced bone density, and general physiological health gains associated with upright posture and walking [46, 47, 48].

Spinal cord injury (SCI) often leads to severe reductions in the ability to perform

regular exercise, resulting in physical deconditioning and secondary complications such as cardiovascular issues, osteoporosis, spasticity, and pressure ulcers [49, 50, 51]. Rehabilitation strategies have traditionally relied on clinician-led mobility exercises [52] or electrical stimulation [53, 54], but these approaches are largely clinic-bound. Lower-limb exoskeletons capable of full weight-bearing, dynamically stable locomotion could extend rehabilitation benefits beyond the clinic, enabling regular activity in daily environments.

Despite these benefits, most commercial exoskeletons for individuals with complete paraplegia face stability limitations. Current devices are generally restricted to slow walking speeds [20] and often require external stability aids such as forearm crutches [17, 55] or overhead weight support [18]. These constraints reduce upper-limb freedom, limit natural arm swing, and confine use to controlled settings. Intensive training—sometimes 20+ sessions—is also required for users to achieve independent walking, primarily to learn weight-shifting strategies for stability [56, 57]. More recent efforts have adapted dynamic walking control methods from bipedal robotics [21] to achieve crutch-less exoskeleton locomotion [58, 59], substantially reducing training time and enabling more natural arm use.

Designing robust locomotion strategies for lower-body exoskeletons has increasingly drawn on concepts from bipedal robotics, including the Hybrid Zero Dynamics (HZD) framework [21] (see Section X.X). HZD provides a mathematically principled approach to synthesizing dynamically stable walking gaits by enforcing impact-invariant periodic orbits. This methodology has enabled demonstrations of crutch-less exoskeleton walking that reduce the reliance on upper-body support and improve gait naturalness [58, 59, 60], with clinical studies reporting that many users can achieve independent locomotion after fewer training sessions compared to conventional designs [61].

Beyond stability, there is growing interest in tailoring exoskeleton walking to individual users. Personalization methods have included tuning gait parameters to body measurements and desired speeds [62, 63], minimizing metabolic cost [24, 64], and optimizing for subjective comfort [22, 23, 65]. Preference-based learning has emerged as a promising approach, where user feedback—often collected through pairwise comparisons—is used to guide gait optimization. Such feedback tends to be more reliable than absolute ratings [66], and can reveal individualized gait patterns that improve comfort and acceptance.

Challenges in preference-based personalization include the limited data available

from time-intensive human subject trials, the need to maintain user safety and comfort during exploration, and the difficulty of efficiently searching large gait parameter spaces. Some approaches prioritize finding an optimal gait for a specific user, while others aim to model the broader "preference landscape" to better understand how different gait parameters influence user satisfaction. Active learning methods have been proposed to improve sample efficiency by strategically selecting gait candidates that maximize expected information gain, while also avoiding prolonged exposure to highly undesirable walking patterns.

2.5 Robotic Platforms

This section describes the robotic platforms used in this work, covering their mechanical design, actuation, sensing capabilities, and control architecture. Each platform serves a distinct role in our experiments—from lower-limb assistive devices to full-body humanoid robots—providing diverse testbeds for locomotion control, learning algorithms, and biomechanical studies.

AMPRO3 Prosthesis

The AMPRO3 is a powered transfemoral prosthesis with actuated knee and ankle joints, designed to support 3D, multidirectional walking. Each joint is driven by electric actuators and instrumented with encoders for position and velocity measurement. Integrated load cells capture ground reaction forces, while inertial measurement units (IMUs) provide segment orientation feedback. The device was developed for studying powered, compliant prosthetic walking in both straight-line and multicontact locomotion. For further mechanical and design details, see [67, 68].

The Atalante Lower-Body Exoskeleton

The Atalante lower-body exoskeleton, developed by Wandercraft, features 12 actuated joints (Figure 2.2). Each leg has three actuators for spherical hip motion, one actuator for knee flexion/extension, and two actuators for ankle motion—covering inversion/eversion and dorsiflexion/plantarflexion. Hip and knee joints are powered by brushless DC motors, while the ankle uses a hybrid mechanism enabling sagittal-plane rotation and rotation about the Henke axis. Joint positions and velocities are measured with digital encoders.

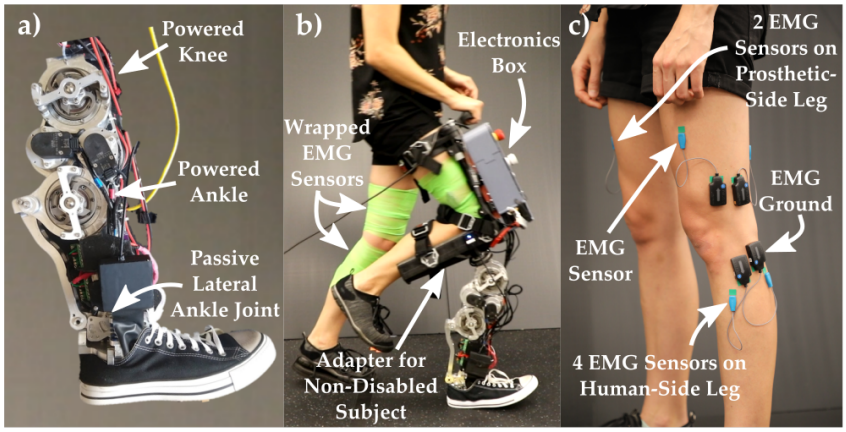


Figure 2.1: Overview of prosthesis, subject testing, and EMG electrode setup a) AMPRO3 prosthesis, b) Non-disabled subject wearing the device during multicontact locomotion, c) placement of the surface mount electrodes for electromyography (EMG).

For sensing, six inertial measurement units (IMUs) are mounted on the torso, pelvis, shanks, and feet to estimate body orientation. Ground contact is detected via four single-axis force sensors under each foot. An embedded real-time computer coordinates actuator control and sensor fusion.

Additional hardware includes overhead hoist loops, mode-selection buttons, a computer connection port, side handles for operator assistance, adjustable thigh and shank segments, and torso/thigh/shank harnesses to secure the user. These features allow fitting to a range of body dimensions and ensure safety during operation.

Unitree G1 Humanoid Robot

The Unitree G1 EDU is a compact humanoid robot with 29 actuated degrees of freedom, comprising 6 per leg, 5 per arm, 1 at the waist, and articulated hands (Figure 2.3. Its aluminum alloy and carbon fiber structure results in a total mass of approximately 35 kg. In this work, the EDU configuration is equipped with an NVIDIA Jetson Orin onboard computer, depth camera, 3D LiDAR, joint encoders, and inertial measurement units (IMUs). A custom front plate was added to mount an external laptop for policy inference, increasing the total mass by 0.616 kg.

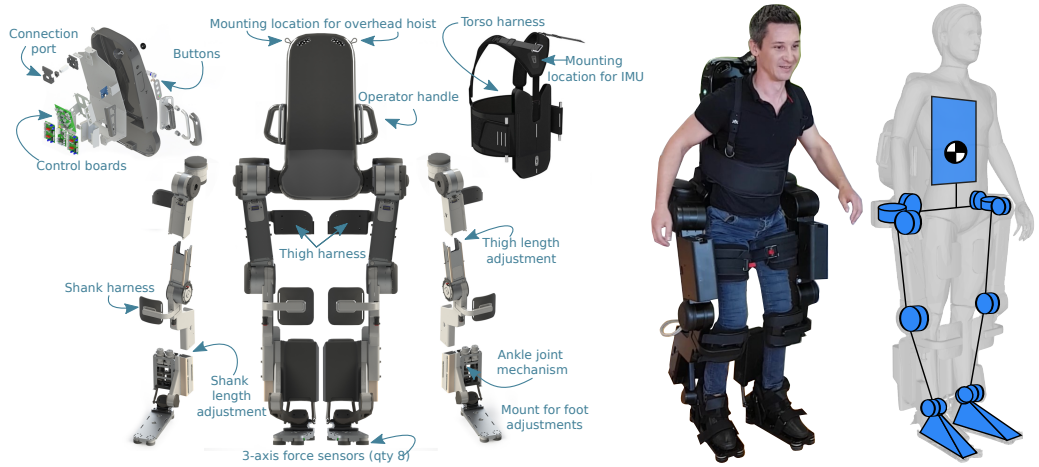


Figure 2.2: The Atalante lower-body exoskeleton designed by Wandercraft: (a) A breakdown of the Atalante exoskeleton components including patient-harnesses and electronics; (b) A patient inside the exoskeleton; and (c) A depiction of the locations of the 12 actuated joints.

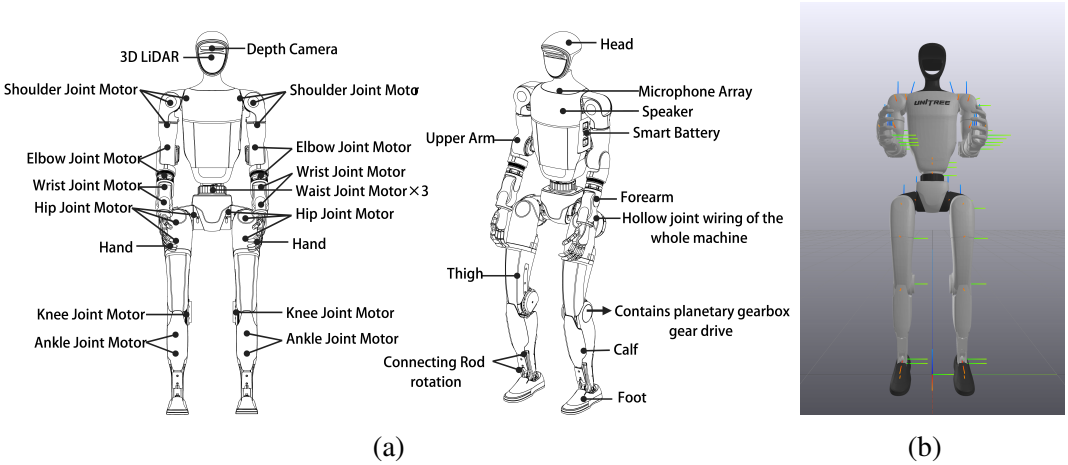


Figure 2.3: The Unitree G1 EDU humanoid robot used in this work: (a) component overview highlighting the onboard sensing and motor locations; (b) Reference frame, joint axis and zero position of the 29 actuated degrees of freedom.

Chapter 3

BIOMECHANICALLY-INSPIRED NOMINAL GAIT DESIGN

This chapter addresses the first stage of the thesis objective: synthesizing nominal gaits that are both robustly stable and user-aligned. For robotic assistive devices, this means not only guaranteeing stability on the robotic platform but also generating motions that are natural and anthropomorphic, so that the device can coordinate seamlessly with human users. Achieving this requires incorporating key biomechanical principles of non-disabled human walking [15] into the design of periodic gaits, while ensuring that these motions remain stabilizable under the robot's hybrid dynamics. Yet, translating the efficiency and naturalness of human locomotion into robotic systems remains an open challenge, particularly for assistive devices where user comfort and coordination are as critical as formal stability. In this chapter, we tackle this challenge by embedding musculoskeletal models into a multi-domain gait generation process, enabling assisted locomotion that is simultaneously stable, natural, and user-oriented.

This work extends the Hybrid Zero Dynamics (HZD) framework introduced earlier by integrating musculoskeletal models into the gait generation process. By embedding biomechanical constraints directly into the optimization, we aim to produce walking behaviors that retain HZD's formal stability guarantees while more closely matching human-like kinematics and dynamics—an essential step toward natural coordination with human users in robotic assistive devices.

While HZD yields provably stable walking, obtaining satisfactory gaits often requires extensive expert tuning of the cost and constraints. This challenge is amplified for assistive devices, which must produce locomotion that is not only stable but also natural to reduce user energy expenditure.

Other approaches toward natural walking include modifying HZD to match motion capture data [21, 67] and optimizing joint-level trajectories for experimental metrics such as electromyography (EMG) signals and metabolic expenditure [64, 69]. While effective, these methods are data-driven and depend heavily on the availability and quality of behavior-specific datasets. A separate approach is to control walking directly from real-time EMG feedback [70, 71, 72, 73, 74], which can yield natural locomotion but lacks formal stability guarantees and requires careful tuning of the

musculoskeletal model.

In this chapter, we present a framework that, to our knowledge, is the first to combine hybrid system models with musculoskeletal modeling. Since humans typically self-select gaits that are physiologically and mechanically energy efficient [75], we hypothesize that generating stable gaits subject to muscle model constraints will naturally lead to more anthropomorphic and efficient behavior that respects physiological limits. We evaluate this hypothesis by generating multicontact walking gaits with HZD and musculoskeletal models, and implementing them experimentally on a dual-actuated transfemoral prosthesis, AMPRO3. Performance is quantified via motion capture and EMG analysis.

3.1 Muscle Model

In this section, we introduce how a single muscle-tendon unit (MTU) is modeled. Later, in Sec. 3.3, we will provide details on how we extend these muscle models to multiple muscles and incorporate them into the Hybrid Zero Dynamics (HZD) gait generation framework.

Muscle-tendon Unit (MTU)

We model each muscle as a two-element Hill-type muscle-tendon unit [76] with a contractile element (CE) and a series elastic element (SE) as shown in Fig. 3.1a. The constant parameters of each muscle are defined in [77, 76].

MTU Length

The length of an individual MTU, denoted by $l_{mtu} \in \mathbb{R}$, is modeled as $l_{mtu} = l_{se} + l_{ce}$, where $l_{ce} \in \mathbb{R}$ is the length of the contractile element (CE), and $l_{se} \in \mathbb{R}$ is the length of the series elasticity element (SE). Since the relative change of l_{mtu} depends on the individual joint angle $\theta \in \mathbb{R}$, with the collection of d joint angles denoted $q \in \mathbb{R}^d$, in practice we model the MTU length as a function of q:

$$l_{mtu}(q) = l_{opt} + l_{slack} - \sum_{j=1}^{j_N} \Delta l_{mtu}(\theta_j), \tag{3.1}$$

where l_{opt}, $l_{slack} \in \mathbb{R}$ are respectively the reference lengths of CE and SE at the reference angle $\theta_{ref} \in \mathbb{R}$. These reference parameters are constants taken from [77].

We use $\sum_{j=1}^{j_N} \Delta l_{mtu}(\theta_j)$ to denote the total change in length of the MTU based on the joint angles of each joint spanned by the MTU, out of a total of $j_N \in \{1, 2\}$ joints. The joints spanned by each MTU are illustrated in Fig.3.1b. The individual change in length due to a single joint, $\Delta l_{mtu}(\theta) \in \mathbb{R}$, is given by:

$$\Delta l_{mtu}(\theta) = \begin{cases} \rho r_0(\theta - \theta_{ref}), & \text{for hip} \\ \rho r_0 \big[\sin(\theta - \theta_{max}) & \\ \quad - \sin(\theta_{ref} - \theta_{max}) \big], & \text{otherwise} \end{cases} \tag{3.2}$$

The constant $\rho \in \mathbb{R}$ is a parameter that ensures the fiber length is within the physiological limits and accounts for muscle pennation angles (the angle between the longitudinal axis of the entire muscle and its fibers that increases as the tension increases in the muscle), and $r_0 \in \mathbb{R}$ is a parameter denoting the constant contribution of the MTU lever-arm. For the MTUs that span two joints, $\Delta l_{mtu}(\theta)$ is calculated separately with different reference angles θ_{ref} for each joint.

MTU Force-Length and Force-Velocity Relationships

The velocity of the CE contraction is denoted by $v_{ce} \in \mathbb{R}$ and is constrained to satisfy the relationship $l_{ce} = \int v_{ce} dt$. Depending on an MTU's instantaneous value of l_{ce} and v_{ce}, the amount of force the MTU is capable of exerting differs. This is described by the following force-length (f_l) and force-velocity (f_v) relationships:

$$f_l(l_{ce}) = \exp\left(\log(c) \left| \frac{l_{ce} - l_{opt}}{l_{opt} w} \right|^3 \right), \tag{3.3}$$

$$f_v(v_{ce}) = \begin{cases} \frac{v_{max} - v_{ce}}{v_{max} + K v_{ce}}, & \text{if } v_{ce} < 0 \\ N + \frac{(N-1)(v_{max} + v_{ce})}{7.56 K v_{ce} - v_{max}}, & \text{if } v_{ce} \geq 0 \end{cases} \tag{3.4}$$

where the residual force factor $c = 0.05$ and N, v_{max}, w, $K \in \mathbb{R}$ are all muscle-dependent constants. Specifically, N is the eccentric force enhancement (modeling the increase in muscle force during active stretch), v_{max} is the maximum contractile velocity, and w and K are parameters that shape the force-length and force-velocity curves, respectively.

Similarly, the MTU force also depends on l_{se}. This is modeled using an additional force-length relationship:

$$f_{se}(l_{se}) = \begin{cases} \left(\frac{l_{se} - l_{slack}}{l_{slack}(\varepsilon_{ref})} \right)^2, & \text{if } l_{se} \geq l_{slack} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

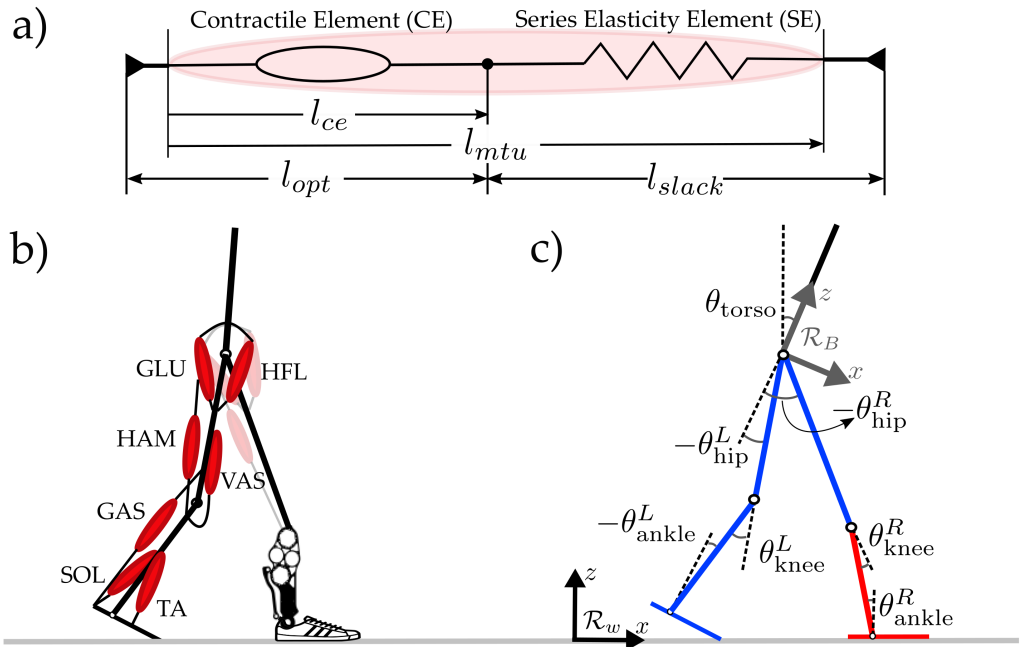


Figure 3.1: Muscle–Tendon unit model and human–prosthesis system. a) A single muscle tendon unit (MTU) consists of a contractile element (CE) and a series elasticity element (SE). The length of CE and SE is denoted by l_{ce} and l_{se}. At the reference angle (θ_{ref}), these lengths are equal to $l_{ce} = l_{opt}$ and $l_{se} = l_{slack}$. b) Human-prosthesis system with the following seven labeled muscles on the intact leg: gluteus (GLU), hamstrings (HAM), gastrocnemius (GAS), soleus (SOL), hip flexors (HFL), and vastus (VAS), and tibialis anterior (TA). Three muscles (GLU, HAM, HFL) are also considered on the prosthetic leg side. c) Illustration of system coordinates, including the base and world frames.

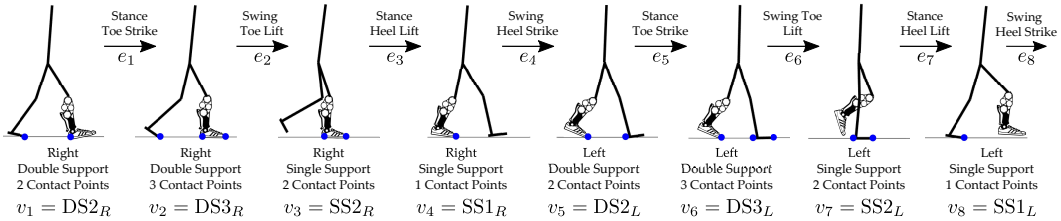


Figure 3.2: A complete gait cycle from right heel strike to right heel strike. The gait cycle is described using the directed cycle $\Gamma = (V, E)$ with the vertices $V = \{v_1, \ldots, v_8\}$ and edges $E = \{e_1, \ldots, e_8\}$ illustrated in the figure. The naming convention is based on the stance leg of the step and the number of contact points. If both legs are in contact, the domain is considered as a double support domain.

where the $\varepsilon_{ref} \in \mathbb{R}$ is a constant parameter denoting the MTU strain when $f_{se}(l_{se}) = 1$. Note that in the actual implementation, we used a continuous function, fitted via least squares regression, to replace the piece-wise functions for f_{se} and f_v since continuous functions are required for the implementation of a nonlinear optimization

program.

MTU Force

Because the SE and CE are in series, we model their respective forces, $F_{se} \in \mathbb{R}$ and $F_{ce} \in \mathbb{R}$, as equal to the total force exerted by the MTU, denoted by $F_m \in \mathbb{R}$. Explicitly, we enforce $F_m = F_{se} = F_{ce}$. We independently model the individual element forces as depending on the previously defined force-length and force-velocity relationships:

$$F_{ce}(l_{ce}, v_{ce}, s) = s\, F_{max}\, f_l(l_{ce})\, f_v(v_{ce}), \tag{3.6}$$

$$F_{se}(l_{se}) = F_{max}\, f_{se}(l_{se}), \tag{3.7}$$

where $s \in [0, 1]$ is the activation level of the muscles, and $F_{max} \in \mathbb{R}$ is a constant parameter dictating the maximum allowable force of the MTU. Note that we assume muscle activation to be instantaneous.

MTU Force-Torque Relationship

The torque provided by the MTU, denoted by $u_m \in \mathbb{R}$, is calculated individually for each joint it spans using the following equations:

$$u_m = r(\theta)\, F_m, \tag{3.8}$$

$$r(\theta) = \begin{cases} r_0, & \text{for hip} \\ r_0\, cos(\theta - \theta_{max}) & \text{otherwise.} \end{cases} \tag{3.9}$$

where $r(\theta) \in \mathbb{R}$ is the length of the MTU lever-arm based on r_0 (previously defined in Eq. 3.2), and $\theta_{max} \in \mathbb{R}$ is the reference angle at maximum lever contribution. For MTUs that span two joints, the muscle torque of each joint is calculated using different muscle-specific maximum lever contribution reference angles θ_{max}. For details see [77].

3.2 Application of Hybrid Zero Dynamics to the AMPRO3 Prosthesis

Next, we present a high-level introduction of the HZD method (without the inclusion of muscle models) applied to the AMPRO3 prosthesis. For more details, we refer the

reader to [67]. Additionally, information on the mechanical design of AMPRO3 is outlined in [68].

Human-Prosthesis Model for AMPRO3

The human-prosthesis system is modeled as a seven-link planar model, illustrated in Fig. 3.1c, with anthropomorphic parameters for the human segments (shown in blue), and parameters specific to the AMPRO3 prosthesis for the prosthetic segments (shown in red). Since the human user considered in this paper is not amputated, the model is asymmetric, with the knee of the prosthesis necessarily lower than the human knee.

The configuration space of the AMPRO3 prosthesis, assuming a floating-base convention [78], is defined as $\mathcal{Q} \subset \mathbb{R}^n$, where $n = 9$ is the planar unconstrained degrees of freedom of AMPRO3. The base frame is defined as $q_B = (p, \theta_{\text{torso}}) \in SE(2)$ with $p \in \mathbb{R}^2$ and $\theta_{\text{torso}} \in SO(2)$ being the position and rotation of the floating base frame R_B with respect to the world frame R_w.

We assume that the left leg is the intact leg and the right leg is the prosthetic leg. Hence, the human coordinates $q_h = (\theta_{\text{torso}}, \theta^L_{\text{hip}}, \theta^L_{\text{knee}}, \theta^L_{\text{ankle}}, \theta^R_{\text{hip}})^T$ consist of the torso angle and the joint angles of the human leg segments (left leg segments and right leg hip). The prosthetic coordinates $q_p = (\theta^R_{\text{knee}}, \theta^R_{\text{ankle}})^T$ include the joint angles of the prosthetic segments. The generalized coordinate of the system is then defined as $q = (p, q_h^T, q_p^T)^T$ and the state space as $\mathcal{X} = T\mathcal{Q} \subset \mathbb{R}^{18}$ with coordinates $x = (q^T, \dot{q}^T)^T$.

Multi-Domain Hybrid System

To capture the intrinsic nature of human walking, a multi-domain hybrid system is constructed for the human-prosthesis system model, with the goal of matching the temporal domain pattern observed in natural human walking [79]. Briefly, hybrid is used to refer to the involvement of both *time-driven* and *event-driven* events. Multicontact hybrid systems use multiple time-driven domains to describe different contact configurations. While such systems have been demonstrated to successfully yield multicontact locomotion [67, 68], the inclusion of multiple domains significantly increases the complexity of the nonlinear optimization problem, making it more challenging to constrain the search space to achieve desire behaviors. In

our chapter, we leverage muscle models to guide the optimization problem towards *natural* multicontact walking gaits.

As illustrated in Fig. 3.2, we construct a domain pattern with eight distinct domains (four in each step), and eight transitions between domains. Note that since our model is asymmetric, we need to consider an entire gait cycle from right heel strike to the next right heel strike, consisting of two individual steps. The domains within each step are named according to the contact points as: Double Support 2 ($\text{DS2}_{\{L,R\}}$), Double Support 3 ($\text{DS3}_{\{L,R\}}$), Single Support 2 ($\text{SS2}_{\{L,R\}}$), and Single Support 1 ($\text{SS1}_{\{L,R\}}$), where the subscript $\{L, R\}$ denotes either the left or right stance leg step. These domains are similar to the breakdown in [5].

Equipped with the domain definitions, we construct a directed cycle $\Gamma = (V, E)$ to describe our multi-domain hybrid system, with the vertices $V = \{v_1, \ldots, v_8\}$ and edges $E = \{e_1, \ldots, e_8\}$ illustrated in Fig. 3.2.

We denote the set of admissible domains by $\mathcal{D} = \{\mathcal{D}_v\}_{v \in V}$. The transitions between these domains are triggered by the set of guards, $S = \{S_e\}_{e \in E}$. The discrete dynamics of these transition events are denoted by $\Delta = \{\Delta_e\}_{e \in E}$.

We can then formally define our full hybrid system as a tuple $\mathscr{HC} = (\Gamma, \mathcal{D}, \mathcal{U}, \mathcal{S}, \Delta, FG)$, where $\mathcal{U} = \{\mathcal{U}_v\}_{v \in V}$ is the set of admissible inputs and $FG = \{(f_v, g_v)\}_{v \in V}$ is the set of control systems with (f_v, g_v) defining the continuous dynamics $\dot{x} = f_v(x) + g_v(x)\, u_v$ for each domain with inputs $u_v = [u^L_{\text{hip}}, u^L_{\text{knee}}, u^L_{\text{hip}}, u^R_{\text{ankle}}, u^R_{\text{knee}}, u^R_{\text{ankle}}]^T$. The continuous dynamics can be obtained using the Euler-Lagrangian equation as explained in [67].

Virtual Constraints

The behavior of the hybrid system can be shaped using *virtual constraints*, defined as the difference between the actual system outputs $y^a(q)$ and the desired outputs $y^d(q, \alpha)$. In our chapter, we describe the desired outputs using Bézier polynomials with coefficients α. To allow for discontinuities in the outputs between domains (necessitated by impact events), we describe the outputs using domain-specific Bézier polynomials with coefficients α_v.

For non-underactuated domains (DS2, DS3, SS2), a relative degree one output is explicitly included in the virtual constraints in order to regulate the forward

progression of the system:

$$y_v(q,\alpha) = \begin{bmatrix} y_{1,v}(q,\dot{q},\alpha) \\ y_{2,v}(q,\alpha) \end{bmatrix} = \begin{bmatrix} y^a_{1,v}(q,\dot{q}) - v_{\text{hip}} \\ y^a_{2,v}(q) - y^d_{2,v}(\tau(q),\alpha) \end{bmatrix}, \tag{3.10}$$

Here $y_{1,v} \in \mathbb{R}$ denotes the domain-specific relative degree one output, defined as the difference between the actual hip velocity $y^a_{1,v}(q,\dot{q})$ and the desired hip velocity v_{hip}. The virtual constraints, $y_{2,v}(q,\alpha)$, denote the relative degree two output. Since the forward hip velocity is approximately constant during the progress of each step cycle, we define our phase variable $\tau(q) = \frac{\delta_{p_{\text{hip}}}(q) - \delta^+_{p_{\text{hip}}}}{v_{\text{hip}}}$, where $\delta_{p_{\text{hip}}}(q)$ is the linearized forward hip position and $\delta^+_{p_{\text{hip}}}$ is the hip position at the beginning of the step. We select the virtual constraints for each domain within one step to be the following:

$$\begin{aligned}
y_{\text{DS2}} &= [v_{\text{hip}}, \theta^{\text{st}}_{\text{hip}}, \theta^{\text{sw}}_{\text{hip}}, \theta^{\text{sw}}_{\text{knee}}, \theta^{\text{sw}}_{\text{ankle}}]^T \\
y_{\text{DS3}} &= [v_{\text{hip}}, \theta^{\text{st}}_{\text{hip}}, \theta^{\text{sw}}_{\text{hip}}, \theta^{\text{sw}}_{\text{knee}}]^T \\
y_{\text{SS2}} &= [v_{\text{hip}}, \theta^{\text{st}}_{\text{hip}}, \theta^{\text{st}}_{\text{knee}}, \theta^{\text{sw}}_{\text{hip}}, \theta^{\text{sw}}_{\text{knee}}, \theta^{\text{sw}}_{\text{ankle}}]^T \\
y_{\text{SS1}} &= [\theta^{\text{st}}_{\text{hip}}, \theta^{\text{st}}_{\text{knee}}, \theta^{\text{st}}_{\text{ankle}}, \theta^{\text{sw}}_{\text{hip}}, \theta^{\text{sw}}_{\text{knee}}, \theta^{\text{sw}}_{\text{ankle}}]^T
\end{aligned}$$

where the superscripts (st, sw) denote either left (L) or right (R) for the stance and swing leg of the corresponding step. Note that the number of virtual constraints in each domain is dependent on the number of contact points.

Lastly, the virtual constraints $y_v(q,\alpha)$ are driven to zero using a feedback linearizing controller $u^*(x)$, resulting in the closed loop dynamics $\dot{x} = f_{cl,v}(x) = f_v(x) + g(x)u^*(x)$.

Impact-Invariance Condition

While the closed-loop dynamics of the designed trajectory may be stable, the system can destabilize at impact events. Thus, it remains to construct desired trajectories that are *impact-invariant*. Since our hybrid system is a multi-domain system with both fully-actuated and under-actuated domains, the entire system is impact invariant if the following individual *impact-invariance conditions* are met for each transition:

$$\begin{cases}
\Delta_e(S_e \cap \mathcal{Z}_{\alpha_v}) \subseteq \mathcal{P}\mathcal{Z}_{\alpha_v}, & e = \{3,7\}, \\
\Delta_e(S_e \cap \mathcal{P}\mathcal{Z}_{\alpha_v}) \subseteq \mathcal{Z}_{\alpha_v}, & e = \{4,8\}, \\
\Delta_e(S_e \cap \mathcal{Z}_{\alpha_v}) \subseteq \mathcal{Z}_{\alpha_v}, & \text{otherwise.}
\end{cases} \tag{3.11}$$

Here we use $\mathcal{PZ}_{\alpha_v}$ and $\mathcal{Z}_{\alpha_v}$ to denote the partial hybrid zero dynamics (PHZD) surface and HZD surface, respectively:

$$\mathcal{Z}_{\alpha_v} = \{(q, \dot{q}) \in \mathcal{D}_v : y_v(q, \alpha_v) = 0,\ \dot{y}_v(q, \dot{q}, \alpha_v) = 0\},$$
$$\mathcal{PZ}_{\alpha_v} = \{(q, \dot{q}) \in \mathcal{D}_v : y_{2,v}(q, \alpha_v) = 0,\ \dot{y}_{2,v}(q, \dot{q}, \alpha_v) = 0\}.$$

The system evolves on these surfaces when the virtual constraints are driven to zero.

Note that $\mathcal{PZ}_{\alpha_v}$ is a restriction of $\mathcal{Z}_{\alpha_v}$ for fully-actuated domains, in which the relative degree one output is used to regulate the system. In practice, impact-invariant periodic orbits are synthesized as solutions to a nonlinear optimization problem with constraints on the closed-loop dynamics and impact-invariance conditions.

3.3 Gait Generation with the Integrated Framework

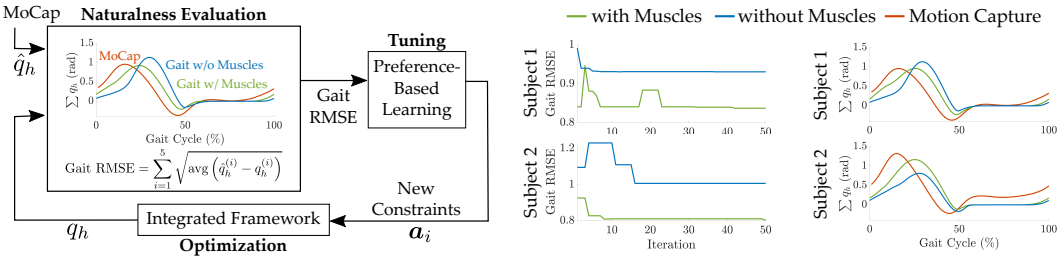
Next, we present an integrated framework that enforces the various muscle-tendon unit properties introduced in Section 3.1 directly into the HZD gait generation framework introduced in Section 3.2. First, we will present the details of the integrated framework. Then, we demonstrate its effect on the gait generation process by comparing gaits obtained with and without the inclusion of the musculoskeletal model.

Integrated Framework

To generate stable impact-invariant periodic orbits, with the inclusion of the muscle models presented in Sec. 3.1, we construct a nonlinear optimization problem of the form:

$$\{\boldsymbol{\alpha}^*, X^*\} = \underset{\boldsymbol{\alpha}, X}{\operatorname{argmin}}\ \Phi_{\text{mCoT}}(X)$$
$$\begin{aligned} \text{s.t.}\quad & \textbf{C1.} && \text{(Closed-loop Dynamics)} \\ & \textbf{C2.} && \text{(Impact-Invariance Conditions)} \\ & \textbf{C3.} && \text{(Decision Variable Bounds)} \\ & \textbf{C4.} && \text{(Physical Constraints)} \\ & \textbf{C5-C12.} && \text{(Muscle Model Constraints)} \end{aligned}$$

where $\boldsymbol{\alpha} = \{\alpha_v \mid v = 1, \ldots, 8\}$ is our collection of Bézier coefficients for each domain, and X is the collection of all decision variables $X = [X_{\text{NLP}}, X_{\text{MUSC}}]^\top$



a) Gait Generation and Tuning Procedure b) Results of 50 iterations c) Final Gaits

Figure 3.3: Results of gait generated with and without the muscle models. a) Gait generation and tuning procedure. Note that the MoCap data are taken from [80] and matched to subjects by height and weight. b) Gait RMSE of the optimal action identified by the algorithm at each iteration. c) The summed human joints angles of final gaits obtained after tuning.

separated into the nominal variables, X_{NLP}, and the additional muscle model decision variables, X_{MUSC}. The nominal decision variables are constructed as $X_{\text{NLP}} = (x_0, \ldots, x_N, T)$ with x_i being the system state at the i^{th} discretization for the duration T. The muscle model decision variables are similarly defined for the muscle states x^{musc} as $X_{\text{MUSC}} = (x_0^{\text{musc}}, \ldots, x_N^{\text{musc}}, T)$. Here, the muscle states include the MTU variables for each muscle $x^{\text{musc}} = \{[l_{ce}^{(i)}, l_{se}^{(i)}, F_{ce}^{(i)}, v_{ce}^{(i)}, s^{(i)}]^\top \mid i = 1, \ldots, 10\}$.

While the objective function can be arbitrarily defined, we intentionally select ours to be the mechanical cost of transport (mCoT), $\Phi_{\text{mCoT}} = \int \frac{P(t)}{mgv} dt$, since prior work has found it to yield natural and efficient locomotion [81].

The first four constraints (C1-C4) of our framework are standard to the HZD method: C1 enforces the closed-loop dynamics of the system; C2 enforces the impact-invariance conditions described by Eq. 3.11; C3 constrains the decision variables as $X_{\text{min}} \preceq X \preceq X_{\text{max}}$; and C4 enforces real world constraints such as contact constraints, as well as joint and torque limits. The remaining constraints (C5-C12) are muscle model constraints, explicitly defined as:

---

where $i = 1, \ldots, 10$ denotes a specific muscle out of the ten muscles we consider, illustrated in Fig. 3.1b. These muscles consist of seven muscles on the intact leg (hamstring (HAM), glutes (GLU), hip flexor (HFL), gastrocnemius (GAS), vastus (VAS), soleus (SOL), tibialis anterior (TA)), and three muscles on the prosthetic leg (HAM, GLU, HFL).

The first four muscular constraints (C5-C8) can be interpreted as dynamic and kinematics constraints acting on each MTU. The final four constraints (C9-C12) ensure that the actual human joint torque is equal to the sum of individual muscle

Muscle Model Constraints:

$$\textbf{C5.}\ \{F_m^{(i)} = F_{ce}^{(i)}(l_{ce}^{(i)}, v_{ce}^{(i)}, l_{se}^{(i)}, s^{(i)}),\ \forall i = 1, \ldots, 10\}$$
$$\textbf{C6.}\ \{F_m^{(i)} = F_{se}(l_{se}^{(i)}),\ \forall i = 1, \ldots, 10\}$$
$$\textbf{C7.}\ \{l_{ce}^{(i)} + l_{se}^{(i)} = l_{mtu}(q)^{(i)},\ \forall i = 1, \ldots, 10\}$$
$$\textbf{C8.}\ \{l_{ce}^{(i)} = \int v_{ce}^{(i)} dt,\ \forall i = 1, \ldots, 10\}$$
$$\textbf{C9.}\ u_{\text{hip}}^{L} = u_m^{(1h)} + u_m^{(2)} + u_m^{(3)}$$
$$\textbf{C10.}\ u_{\text{knee}}^{L} = u_m^{(1k)} + u_m^{(4k)} - u_m^{(5)}$$
$$\textbf{C11.}\ u_{\text{ankle}}^{L} = u_m^{(4a)} + u_m^{(6)} - u_m^{(7)}$$
$$\textbf{C12.}\ u_{\text{hip}}^{R} = u_m^{8} + u_m^{9} - u_m^{10}$$

torques. Depending on whether it is an extensor or flexor muscle, the torque is either applied towards the positive or negative direction. Note that since the HAM muscle span both the hip and knee joints, we use $u_m^{(1h)}$ and $u_m^{(1k)}$ to denote the torque HAM has on the these two joints, respectively. Similarly, we use $u_m^{(4k)}$ and $u_m^{(4a)}$ to denote the knee and ankle joint torques resulting by GAS muscle. The explicit calculation can be found in Eq. 3.8 with different reference angles in Eq. 3.9.

3.4 Evaluation of the Integrated Framework

Optimization setup

To evaluate our hypothesis that enforcing muscle model constraints would naturally lead to more anthropomorphic behavior, we synthesized two variants of the optimization problem for comparison: 1) with muscles, which includes constraints C1-C12; and 2) without muscles, which only includes constraints C1-C4. In both variants, the optimization problem is constructed using FROST [41].

We evaluated the naturalness of the gaits generated by the two variants via a custom metric defined as:

$$\text{Gait RMSE} = \sum_{i=1}^{5} \sqrt{\text{avg}\left(\hat{q}_h^{(i)} - q_h^{(i)}\right)}, \tag{3.12}$$

where $q_h^{(i)}$ denotes the angles of the i^{th} joint of the human coordinates and $\hat{q}_h^{(i)}$ the corresponding joint angles recorded by MoCap. Specifically, the MoCap data used here are from [80] and matched to subjects by height and weight.

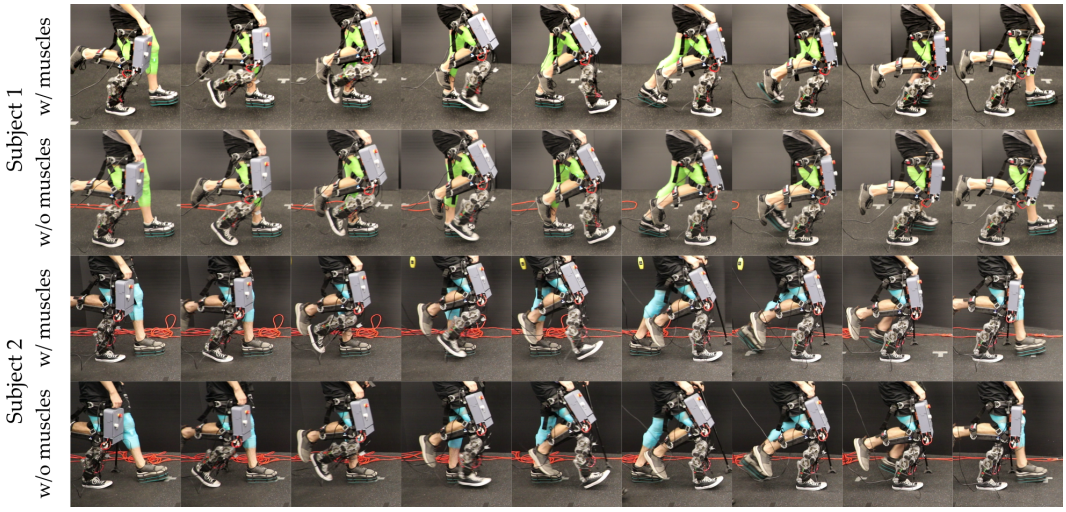


Figure 3.4: Gait tiles of experimental demonstration on AMPRO3 for gaits generated without or with muscle model for two subjects

Constraint Tuning via Preference-Based Learning

The bounds of C3 and C4 are commonly tuned in order to sufficiently constrain the optimization problem for convergence and to achieve desired behavior. Thus, to fairly compare gaits generated with and without the inclusion of the musculoskeletal model, we leverage preference-based learning to systematically identify the constraints that lead to the lowest Gait RMSE. The procedure of this framework is illustrated in Fig. 3.3a. We specifically use the LineCoSpar [82] algorithm since it can navigate high-dimensional spaces and is robust to noisy feedback, but other Bayesian optimization techniques could also be used.

In each iteration, we warm-start the optimization with the solution from the current

Table 3.1: PBL Constraint Search Space

Constraint Name	Constraint Values	lengthscales
$\lvert\dot{x}\rvert < a_1$	a_1:[15, 20]	5
$\lvert\ddot{x}\rvert < a_2$	a_2:[70,80,90]	10
$v_{\text{hip}} > a_3$	a_3:[0.3,0.4,0.5] (m/s)	0.1
$v_{\text{hip}} < a_4$	a_4:[1.2,1.3,1.4] (m/s)	0.1
Min. Foot Clearance	a_5:[0, 0.013, 0.026, 0.039] (m)	0.013
$\lvert\theta_{\text{torso}}\rvert < a_6$	a_6:[0,0.1,0.2,0.3,0.4,0.5] (rad.)	0.1
$\lvert\theta_{\text{hip}}\rvert < a_7$	a_7:[20,35,50] (deg.)	15
$\lvert\theta_{\text{ankle}}\rvert < a_8$	a_8:[20,30,40] (deg.)	10

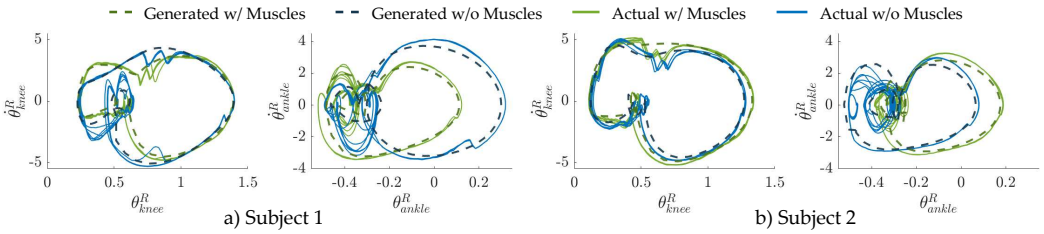


Figure 3.5: Limit cycles illustrating the periodic stability achieved during experimental multicontact locomotion (10s of data plotted).

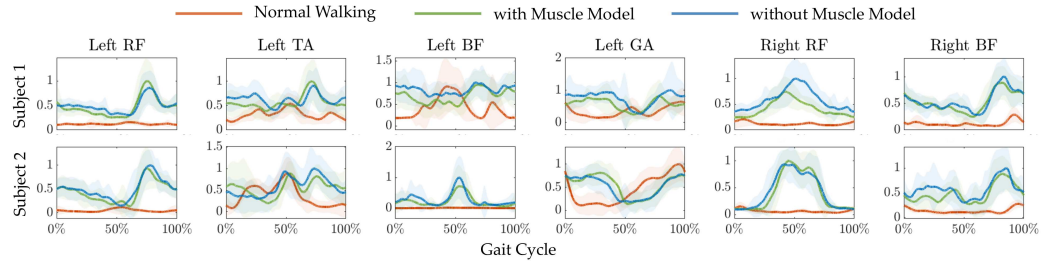


Figure 3.6: EMG activity normalized over a full gait cycle for normal walking, prosthetic walking with gaits generated with or without the muscle model.

best action according to the learning algorithm. To streamline the process, two types of feedback are automatically given to the algorithm. First, an ordinal label corresponding to either 'converged' or 'non-converged' is given based on the algorithm convergence status. Second, a pairwise preference is determined based on the Gait RMSE, where a lower RMSE gait would be preferred. We construct the search space of the algorithm with the following dimensions as in Table 3.1.

Comparison of generated gaits

This learning procedure was repeated for two subjects: subject 1 (Female, 172.7cm 65.7kg), subject 2 (Male, 180.3cm, 75kg). We plotted the Gait RMSE of gaits generated by the current best constraint parameters according to the algorithm at each iteration in Fig. 3.3b. The inclusion of muscle models led to a smaller Gait RMSE compared with the ones generated by the non-muscle version throughout the tuning process (Fig. 3.3b-c). This highlights the advantage of including muscle models in the gait generation, as it guided the optimization to find more natural solutions.

3.5 Experimental Demonstration on AMPRO3

We experimentally deployed the two gaits obtained in the automated tuning procedure as having the lowest Gait RMSE (with and without the inclusion of muscle model constraints) on the dual-actuated transfemoral prosthesis, AMPRO3. This experiment was conducted for each of the two subjects, with the results highlighted in the supplemental video [83].

Experiment Procedure

During the experiments, a non-disabled human user wore AMPRO3 using an adapter on the right leg (Fig. 2.1b). The joint-level trajectories of the gaits were tracked on the prosthesis with a PD controller. For an in-depth presentation of the hardware and control, see [67].

First, the subject was asked to walk without the prosthesis over a self-selected speed, followed by walking with the prosthesis for the two prosthetic gaits. At the end of the testing, the subject was queried for a single pairwise preference. Note that the order of the gaits was randomized and the subject was not informed of the order. During all tests, electromyography (EMG) signals were recorded. Before recording, the subject was given enough time to adjust to the walking. In total, the activity of four muscles on the left leg, including rectus femoris (RF), tibialis anterior (TA), bicep femoris (BF), and gastrocnemius (GAS), and two muscles on the left leg (RF and BF) was recorded with the Trigno wireless biofeedback system (Delsys Inc.), as illustrated in Fig. 2.1c.

Experiment Results

A visualization of the experimental behaviors is provided in Fig. 3.4 via gait tiles spanning a complete gait cycle. Both subjects strongly preferred the gait generated with the inclusion of the musculoskeletal model. The stability of the executed gaits is portrayed in Fig. 3.5 by the periodicity of the limit cycles. It is important to note that achieving this experimentally stable multicontact locomotion is a direct result of leveraging the HZD method to formally generate impact-invariant output trajectories.

The average EMG data over one gait cycle for each muscle after preprocessing is shown in Fig. 3.6. We also calculated the RMSE between the EMG activity of the

generated gaits and normal walking, defined as:

$$\text{EMG RMSE} = \sum_{i=1}^{6} \sqrt{\text{avg}\left(\hat{s}_{\text{EMG}}^{(i)} - s_{\text{EMG}}^{(i)}\right)}, \tag{3.13}$$

where $s_{\text{EMG}}^{(i)}$ denotes the muscle activation reflected by EMG signals for the i^{th} muscle during the prosthetic walking and $\hat{s}_{\text{EMG}}^{(i)}$ denotes the corresponding muscle activation during normal unassisted walking. The EMG RMSE are 1.58 and 1.84 for the gaits generated with muscles, and 1.80 and 2.34 for the gaits generated without muscles, for subject 1 and subject 2, respectively. The lower EMG RMSE suggests that the inclusion of the muscle model led to more *natural* behavior. In addition, the inclusion of muscle model also results in less muscle activation on average. Lastly, we observe that all prosthetic gaits yielded higher muscle activity than normal walking, which could be caused by factors such as the extra weight of the prosthesis or the misaligned knee joints.

However, when designing gaits for an amputee user, the human-prosthesis system would be more symmetric, which would likely to result in even more natural muscle activation.

3.6 Summary: Biomechanically-Inspired Gait Generation

This chapter demonstrates the first formal synthesis of stable multicontact locomotion using musculoskeletal models. Specifically, we directly enforce muscle model constraints in the HZD framework to experimentally realize both stable and natural robotic-assisted locomotion on the dual-actuated prosthesis AMPRO3 with two non-disabled users. We find that incorporating the muscle model guides the optimization problem towards uncovering periodic orbits that resemble natural bipedal locomotion.

Our proposed framework is advantageous since it results in more natural behavior as compared to state-of-the-art. Additionally, it can be applied to a wide range of behaviors and/or robotic platforms, without relying on the availability of experimental data from human subjects or human-in-the-loop testing. Lastly, even though the presented results are limited to planar locomotion, the framework can be extended to 3D locomotion by including muscles that act in the frontal plane (hip abductor and adductor [84]).

Since all physiological parameters (reference lengths, angle, etc.) were from [77], which was intended for a non-disabled subject with different height and weight, it might be beneficial to calibrate these parameters of the muscle model to account for individual differences (especially for amputee users) and improve the prediction accuracy of the embedded muscle models, using methods similar to those in [65]. Such prediction accuracy would further allow for targeted muscle behavior of the user for rehabilitation applications.

Chapter 4

USER-ALIGNED GAIT PREFERENCE LEARNING

The previous chapter focused on generating nominal gaits that are stable and user-aligned in an anthropomorphic sense by leveraging biomechanical models. However, anthropomorphism alone does not fully capture what makes a gait suitable for an individual user. Depending on the context, natural-looking motion may be insufficient—or even misaligned with the user's comfort or intent. A complementary avenue is to incorporate direct feedback from users, such as verbal input or preference-based comparisons, to shape gait generation in ways that reflect subjective comfort and individual needs. For robotic assistive devices, this highlights a key point: stability and naturalness are necessary foundations, but walking must ultimately be tailored to each user's physiology, intentions, and comfort thresholds. Personalization is therefore essential not only for functional mobility but also for long-term usability and adoption.

The partial hybrid zero dynamics (PHZD) method of gait generation has been successfully demonstrated in achieving dynamically stable crutch-less locomotion on the Atalante exoskeleton. While originally developed for bipedal robots [58, 59, 60], PHZD produces stable walking but does not address optimization for user comfort—an aspect that should be a critical objective in exoskeleton gait design. Although prior methods can generate human-like walking gaits for bipedal robots [21], they are unlikely to match the individual preferences of users receiving robotic assistance.

This gap is particularly important for lower-body exoskeletons, which aim to restore mobility to people with paralysis, a group with nearly 5.4 million people in the US alone [85]. Currently, the relationship between exoskeleton users' preferences and the exoskeleton's walking parameters is poorly understood. Scientifically, such an understanding could yield insight into the science of walking, for instance, why exoskeleton users prefer certain gaits to others. On the direct clinical side, identifying the gaits that users prefer is critical for rehabilitation and assistive device design. Existing approaches for customizing exoskeleton walking include optimizing factors such as body parameters and targeted walking speeds [62, 63], minimizing metabolic cost [24, 64], and optimizing user comfort [22, 23, 65]. More specifically, the work

in [22, 23, 65] demonstrated the notion of optimizing exoskeleton gaits based on user preferences to find the optimal gait for each exoskeleton user. Learning from preferences is beneficial because it has been shown that pairwise preferences (e.g., "Does the user prefer A or B?") are often more reliable than numerical scores [66].

However, learning user preferences for exoskeleton walking presents several challenges: limited data from time-intensive human subject experiments, the need to ensure comfort and safety, variability in feedback reliability, and the complexity of the high-dimensional gait space. Broadly, preference-based learning methods can be designed with two distinct goals:

1. **Preference optimization** — finding the optimal gait for a specific user, providing immediate personalization but requiring dedicated trials for each individual.
2. **Preference characterization** — mapping the broader relationship between gait features and user comfort, enabling predictions of optimal gaits without extensive per-user trials.

The choice between these goals directly shapes the data collection strategy. Prior work [22, 23] emphasized direct function optimization, effectively addressing preference optimization but without producing a complete model of the underlying preference landscape. In this chapter, we extend the scope to also capture the broader landscape through preference characterization, accepting that this comes at the cost of less dense sampling near the optimum.

To achieve personalization while ensuring safety and practicality in exoskeleton experiments, this chapter builds on three complementary preference-based learning algorithms: CoSpar [22], LineCoSpar [82], and ROIAL [86]. My contributions are as follows: 1) Development of ROIAL for preference characterization in a region of interest that ensures safety and comfort. 2) Extension of ROIAL to a safety-aware preference-based optimization framework for safety-critical control. 3) Evaluation of a unified framework combining ROIAL with CoSpar and LineCoSpar on the Atalante exoskeleton, demonstrating both preference optimization and preference characterization with paraplegic participants and tuning turning controller. 4) Development of a preferential multi-objective Bayesian optimization method.

The remainder of this chapter is organized as follows: Section 4.1 presents ROIAL; Section 4.2 details the integrated framework combining ROIAL, CoSpar, and

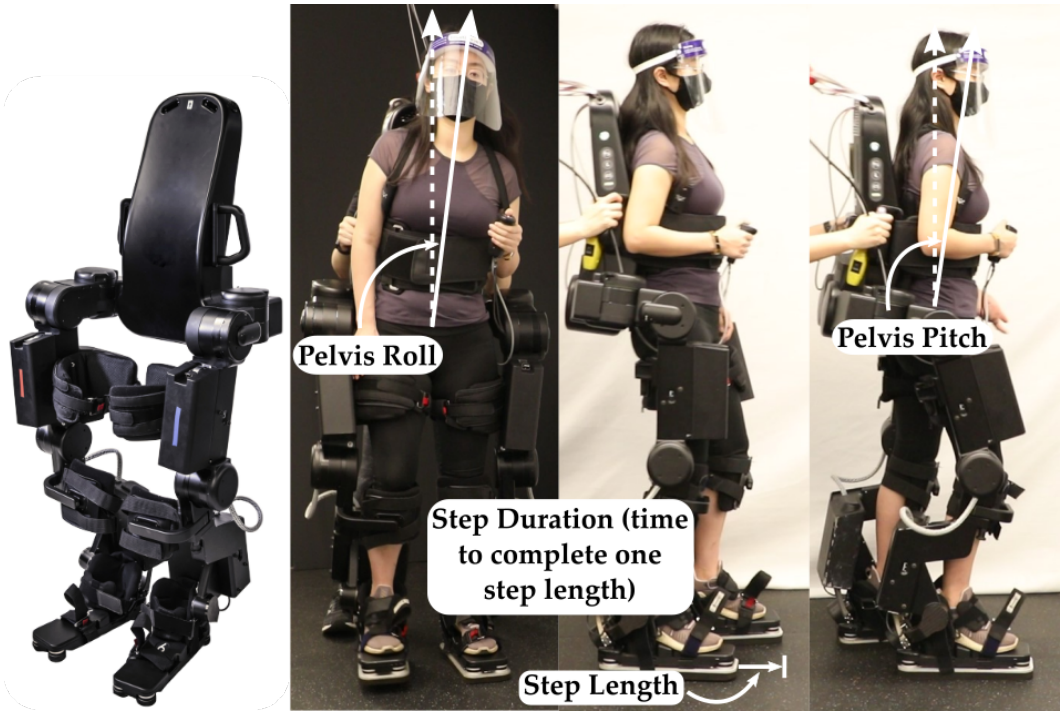


Figure 4.1: The Atalante exoskeleton, designed by Wandercraft, has 12 actuated joints, 6 on each leg. The experiments explore four gait parameters: step length, step duration, pelvis roll, and pelvis pitch.

LineCoSpar for clinical use; Section 4.3 describes its extension to safety-aware preference-based optimization; and Section 4.4 introduces a preferential multi-objective Bayesian optimization method for settings where user preferences span multiple competing objectives.

4.1 Region of Interest Active Learning (ROIAL)

In human-in-the-loop exoskeleton gait optimization, exploring the full range of gait parameters inevitably exposes users to walking patterns that feel unsafe or uncomfortable. Repeated exposure to such undesirable gaits not only reduces user comfort but can also lead to disengagement from the experiment. We refer to this set of undesirable gaits as the *Region of Avoidance* (ROA) and the remaining set of acceptable gaits as the *Region of Interest* (ROI).

Existing work on safe exploration [87, 88, 89, 90] treats unsafe actions as catastrophically bad, enforcing strict avoidance throughout learning. While appropriate in high-risk domains, these methods can be overly conservative in our setting, where occasional sampling from less-preferred regions is tolerable if it improves the overall

preference model.

To address this, we proposes the Region of Interest Active Learning (ROIAL) algorithm, a novel active learning framework which queries the user for qualitative or preference feedback to: 1) locate the ROI, and 2) estimate the utility function as accurately and quickly as possible over the ROI. The algorithm selects samples by modeling a Bayesian posterior over the utility function using Gaussian processes and maximizing the information gain (over the ROI) with respect to this posterior. Information gain maximization for preference elicitation is a sample-efficient, state-of-the-art approach that generates preference queries that are easy for users to answer accurately [91, 92, 93]. To our knowledge, our approach is the first to tackle such a region of interest active learning task.

The vast majority of prior work on preference learning obtains at most 1 bit of information per preference query [91, 92, 93, 22, 23, 94, 95, 65, 96, 97, 98, 99]. ROIAL additionally learns from ordinal labels [100], which assign actions to r discrete ordered categories such as "bad," "neutral," and "good." Ordinal feedback enables ROIAL to both: 1) locate the ROI by learning the boundary between the least-preferred category (ROA) and remaining actions (ROI), and 2) estimate the utility function more efficiently within the ROI. Compared to the 1 bit of information obtained per preference, each ordinal query yields up to $\log_2(r)$ bits of information. Since ordinal feedback is identical for actions within each ordinal category, preferences provide finer-grained information about the utility function's shape within the categories.

We validate ROIAL both in simulation and experimentally. We demonstrate in simulation that ROIAL estimates both the ROI and the utility function within the ROI with high accuracy. We experimentally demonstrate ROIAL on the lower-body exoskeleton Atalante (Fig. 4.1) to learn the utility functions of three non-disabled users over four gait parameters. The obtained landscapes highlight both agreement and disagreement in preferences among the users. Previous algorithms for exoskeleton gait optimization were incapable of drawing such conclusions; thus, this work represents progress towards establishing a better understanding of the science of walking with respect to exoskeleton gait design.

Problem Statement

We consider an active learning problem over a finite (but potentially-large) action space $\mathcal{A} \subset \mathbb{R}^d$ with $A = |\mathcal{A}|$. Each action $\boldsymbol{a} \in \mathcal{A}$ is assumed to have an underlying utility to the user, $f(\boldsymbol{a})$. The algorithm aims to learn the unknown utility function $f : \mathcal{A} \to \mathbb{R}$. The actions' utilities can be written in the vectorized form $\boldsymbol{f} := [f(\boldsymbol{a}^{(1)}), f(\boldsymbol{a}^{(2)}), ..., f(\boldsymbol{a}^{(A)})]^\top$, where $\{\boldsymbol{a}^{(k)} \mid k = 1, \ldots, A\}$ are the actions in $\mathcal{A}$. Let $\boldsymbol{a}_i \in \mathcal{A}$ be the action selected in trial i, where $i \in \{1, \ldots, N\}$. We receive qualitative information about f after each trial i, consisting of an ordinal label y_i and (possibly) a preference between $\boldsymbol{a}_i$ and $\boldsymbol{a}_{i-1}$ for $i \geq 2$. We use $\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2}$ to denote a preference for action $\boldsymbol{a}_{k1}$ over $\boldsymbol{a}_{k2}$, and following each trial i, collect these preferences into a dataset $\mathcal{D}_p^{(i)} = \{\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid k = 1, 2, ..., N_p^{(i)}\}$. Since preference feedback is not necessarily given for every trial, $N_p^{(i)} \leq i - 1$. The ordinal labels are similarly collected into $\mathcal{D}_o^{(i)} = \{(\boldsymbol{a}_k, y_k) \mid k = 1, 2, ..., N_o^{(i)}\}$. The full user feedback dataset after iteration i is defined as $\mathcal{D}_i := \mathcal{D}_p^{(i)} \cup \mathcal{D}_o^{(i)}$.

Ordinal feedback assigns one of r ordered labels to each sampled action. These (possibly-noisy) labels are assumed to reflect ground truth ordinal categories (e.g., "bad," "neutral," "good," etc.), which partition $\mathcal{A}$ into r sets $\mathbf{O}_j$, $j \in \{1, \ldots, r\}$. We define the ROA as $\mathbf{O}_1$; for instance, in the exoskeleton setting, it consists of gaits that make the user feel unsafe or uncomfortable. Similarly, the ROA could be defined as $\bigcup_{j=1}^{n} \mathbf{O}_j$ for $n > 1$, where the choice of n is task-specific given the ordinal category definitions. We define the ROI as the complement of the ROA, $\mathcal{A} \setminus \mathbf{O}_1$.

Defining $\hat{\boldsymbol{f}}_i := [\hat{f}_i(\boldsymbol{a}^{(1)}), \ldots, \hat{f}_i(\boldsymbol{a}^{(A)})]^\top$ as the maximum a posteriori (MAP) estimate of the utilities $\boldsymbol{f}$ given $\mathcal{D}_i$, we aim to adaptively select the N actions $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N \in \mathcal{A}$ that minimize the error in estimating $\boldsymbol{f}$ over the ROI. Defining $\boldsymbol{u} \in \{0, 1\}^A$ as a binary vector denoting which actions are within the ROI, we model the error as $\text{Error}(N) := \boldsymbol{u}^\top |\boldsymbol{f} - \hat{\boldsymbol{f}}_N|$, where the absolute value is taken element-wise.

Active Learning Algorithm

This subsection describes the ROIAL algorithm (Alg. 1), which leverages qualitative human feedback to estimate the ROI and utility function (code available at [101]). We first discuss Bayesian modeling of the utility function, and then explain the procedure for rendering it tractable in high dimensions. We then detail the process for estimating the ROI and approximating the information gain to select the most informative actions.

Algorithm 1 ROIAL Algorithm

Require: Utility prior parameters; ordinal thresholds $b_1, \ldots, b_{r-1}$; subset size M; confidence parameter λ

$\mathcal{D}_0 = \emptyset$, ▷ $\mathcal{D}_i$: user feedback dataset including iteration i
Select an action $\boldsymbol{a}_1$ at random
Add ordinal feedback to data to obtain $\mathcal{D}_1$
for i = 2,. . . , N **do**
    Update the model posterior $P(\boldsymbol{f} \mid \mathcal{D}_{i-1})$ ▷ Eq. (4.1)
    Determine $\mathcal{S}^{(i)}$ by randomly selecting M actions
    Determine $\mathcal{S}^{(i)}_{ROI} \subset \mathcal{S}^{(i)}$
    $\boldsymbol{a}_i \leftarrow \arg\max_{\boldsymbol{a} \in \mathcal{S}^{(i)}_{ROI}} I(\boldsymbol{f}; s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a})$
    Add preference and ordinal feedback to data to obtain $\mathcal{D}_i$
end for

Bayesian Posterior Inference. To simplify notation, this section omits the iteration i from all quantities. Given the feedback dataset $\mathcal{D} = \mathcal{D}_p \cup \mathcal{D}_o$, the utilities $\boldsymbol{f}$ have posterior:

$$P(\boldsymbol{f} \mid \mathcal{D}_p, \mathcal{D}_o) \propto P(\mathcal{D}_p \mid \boldsymbol{f}) P(\mathcal{D}_o \mid \boldsymbol{f}) P(\boldsymbol{f}), \tag{4.1}$$

where $P(\boldsymbol{f})$ is a Gaussian prior over the utilities $\boldsymbol{f}$:

$$P(\boldsymbol{f}) = \frac{1}{(2\pi)^{A/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{f}^\top \Sigma^{-1} \boldsymbol{f}\right),$$

in which $\Sigma \in \mathbb{R}^{A \times A}$, $\Sigma_{ij} = \mathcal{K}(\boldsymbol{a}_i, \boldsymbol{a}_j)$, and $\mathcal{K}$ is a kernel of choice. This work uses the squared exponential kernel.

Preference feedback. We assume that the users' preferences are corrupted by noise as in [102], such that:

$$P(\boldsymbol{a}_1 \succ \boldsymbol{a}_2 \mid \boldsymbol{f}) = g_p\left(\frac{f(\boldsymbol{a}_1) - f(\boldsymbol{a}_2)}{c_p}\right),$$

where $g_p : \mathbb{R} \to (0, 1)$ is a monotonically-increasing link function, and $c_p > 0$ quantifies noisiness in the preferences.

Ordinal feedback. We define thresholds $-\infty = b_0 < b_1 < b_2 < \ldots < b_r = \infty$ to partition the action space into r ordinal categories, $\mathbf{O}_1, \ldots, \mathbf{O}_r$. For any $\boldsymbol{a} \in \mathcal{A}$, if $f(\boldsymbol{a}) < b_1$, then $\boldsymbol{a} \in \mathbf{O}_1$, and $\boldsymbol{a}$ has an ordinal label of 1. Similarly, if $b_j \leq f(\boldsymbol{a}) < b_{j+1}$, then $\boldsymbol{a} \in \mathbf{O}_{j+1}$, and $\boldsymbol{a}$ corresponds to a label of $j+1$. We assume that the users' ordinal labels are corrupted by noise as in [100], such that:

$$P(y \mid \boldsymbol{f}, \boldsymbol{a}) = g_o\left(\frac{b_y - f(\boldsymbol{a})}{c_o}\right) - g_o\left(\frac{b_{y-1} - f(\boldsymbol{a})}{c_o}\right),$$

where $g_o : \mathbb{R} \to (0, 1)$ is a monotonically-increasing link function, and $c_o > 0$ quantifies the ordinal noise.

Assuming conditional independence of queries, the likelihoods $P(\mathcal{D}_p \mid \boldsymbol{f})$ and $P(\mathcal{D}_o \mid \boldsymbol{f})$ are:

$$P(\mathcal{D}_p \mid \boldsymbol{f}) = \prod_{k=1}^{N_p} g_p \left(\frac{f(\boldsymbol{a}_{k1}) - f(\boldsymbol{a}_{k2})}{c_p} \right),$$
$$P(\mathcal{D}_o \mid \boldsymbol{f}) = \prod_{k=1}^{N_o} \left[g_o\left(\frac{b_{y_k} - f(\boldsymbol{a}_k)}{c_o} \right) - g_o\left(\frac{b_{y_k - 1} - f(\boldsymbol{a}_k)}{c_o} \right) \right].$$

Our simulations and experiments fix the hyperparameters c_p, c_o, and $\{b_j \mid j = 1, \ldots, r-1\}$ in advance. One could also estimate them during learning using strategies such as evidence maximization, but this can be very computationally expensive, especially in high-dimensional action spaces.

Common choices of link function (g_p and g_o) include the Gaussian cumulative distribution function [102, 100] and the sigmoid function, $g(x) = (1 + e^{-x})^{-1}$ [23]. We model feedback via the sigmoid link function because empirical results suggest that a heavier-tailed noise distribution improves performance. We use the Laplace approximation to approximate the posterior as Gaussian: $P(\boldsymbol{f} \mid \mathcal{D}_i) \approx \mathcal{N}(\hat{\boldsymbol{f}}_i, \hat{\Sigma}_i)$ [103].

High-Dimensional Tractability. Calculating the model posterior is the algorithm's most computationally-expensive step, and is intractable for large action spaces. Most existing work in high-dimensional Gaussian process learning requires quantitative feedback [104, 105]. Previous work in preference-based high-dimensional Gaussian process learning [23] restricts posterior inference to one-dimensional subspaces. However, the approach in [23] is more amenable to the regret minimization problem because each one-dimensional subspace is biased toward regions of high posterior utility. Instead, to increase ROIAL's online computing speed over high-dimensional spaces, in each iteration i we select a subset $\mathcal{S}^{(i)} \subset \mathcal{A}$ of M actions uniformly at random, and evaluate the posterior only over $\mathcal{S}^{(i)}$.

Estimating the Region of Interest. Since we lack prior knowledge about the ROI, it must be estimated during the learning process. In each iteration i, we model the ROI as the set of actions $\{\boldsymbol{a}_k\}$ that satisfy the following criterion: $\hat{f}_{i-1}(\boldsymbol{a}_k) + \lambda \hat{\sigma}_{i-1}(\boldsymbol{a}_k) > b_1$, where $\hat{\sigma}_{i-1}(\boldsymbol{a}_k)$ is the posterior standard deviation associated with $\boldsymbol{a}_k$. The variable

λ is a user-defined hyperparameter that determines the algorithm's conservatism in estimating the ROI; positive λ's are optimistic, while negative λ's are more conservative in avoiding the ROA. We evaluate actions in the randomly-selected subset $\mathcal{S}^{(i)}$ and define $\mathcal{S}^{(i)}_{ROI} = \{\boldsymbol{a} \in \mathcal{S}^{(i)} \mid \hat{f}_{i-1}(\boldsymbol{a}) + \lambda\hat{\sigma}_{i-1}(\boldsymbol{a}) > b_1\}$ in each iteration i. Note that this definition is optimistic, whereas safe exploration approaches use pessimistic definitions [87, 88, 89, 90].

Action Selection via Information Gain Optimization. To learn the utility function in as few trials as possible, we select actions to maximize the mutual information between the utility function and the preference-based and ordinal human feedback. While optimizing the entire sequence of N actions is NP-hard [106], previous work has shown that a greedy approach which only optimizes the next immediate action achieves state-of-the-art data-efficiency [92]. Hence, we adopt the same approach to solve the following optimization in each iteration i:

$$\max_{\boldsymbol{a}_i \in \mathcal{S}^{(i)}_{ROI}} I(\boldsymbol{f}; s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i), \tag{4.2}$$

where s_i denotes the outcome of a pairwise preference elicitation between $\boldsymbol{a}_i$ and $\boldsymbol{a}_{i-1}$. One can rewrite (4.2) in terms of information entropy:

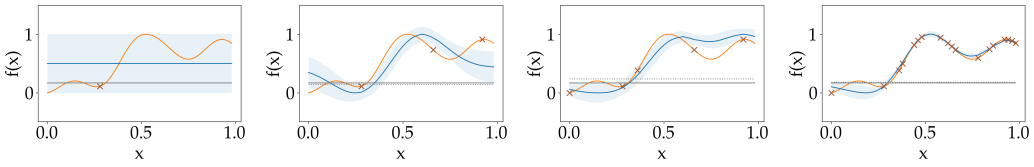
$$\max_{\boldsymbol{a}_i} H(s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i) - \mathbb{E}_{\boldsymbol{f}\mid\mathcal{D}_{i-1}}[H(s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i, \boldsymbol{f})]\,.$$

We can interpret the first term as the uncertainty about action $\boldsymbol{a}_i$'s ordinal label and preference relative to $\boldsymbol{a}_{i-1}$. We aim to maximize this term, because queries with high model uncertainty could potentially yield significant information. The second term is conditioned on $\boldsymbol{f}$, and so represents the user's expected uncertainty. If the user is very uncertain about their feedback, then the action $\boldsymbol{a}_i$ gives only a small amount of information. Hence, we aim to minimize this second term. In this way, information gain optimization produces queries that are both informative and easy for users.

The second term is estimated via sampling from the Laplace-approximated Gaussian posterior $P(\boldsymbol{f} \mid \mathcal{D}_{i-1})$. Computing the first term requires the probability $P(s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i)$. We derive it as:

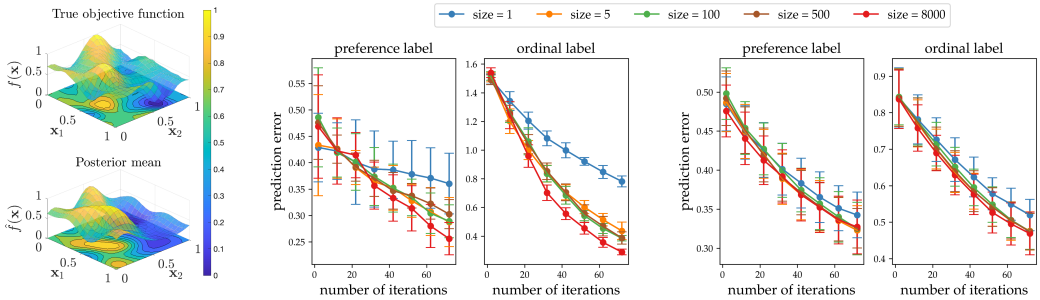
$$\begin{aligned} &P(s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i) \\ &\quad= \int_{\mathbb{R}^A} P(\boldsymbol{f} \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i) P(s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i, \boldsymbol{f}) d\boldsymbol{f} \\ &\quad= \mathbb{E}_{\boldsymbol{f}\mid\mathcal{D}_{i-1}}\left[P(s_i, y_i \mid \mathcal{D}_{i-1}, \boldsymbol{a}_i, \boldsymbol{f})\right], \end{aligned}$$

which we approximate with samples from $P(\boldsymbol{f} \mid \mathcal{D}_{i-1})$.



(a) Iteration 1 (b) Iteration 3 (c) Iteration 5 (d) Iteration 20

Figure 4.2: 1D posterior illustration. The true objective function is shown in orange, and the algorithm's posterior mean is blue. Blue shading indicates the confidence region for $\lambda = 0.5$. The solid grey line indicates the true ordinal threshold b_1: the ROI is above this threshold, while the ROA is below it. The dotted grey line is the algorithm's b_1 hyperparameter. The actions queried so far are indicated with "x"s. Utilities are normalized in each plot so that the posterior mean spans the range from 0 to 1.



(a) Synthetic function posterior (b) Hartmann3 function prediction error (c) Synthetic function prediction error

Figure 4.3: Impact of random subset size on algorithm performance. a) Example 3D synthetic objective function and posterior learned by ROIAL with subset size = 500 after 80 iterations. Values are averaged over the 3rd dimension and normalized to range from 0 to 1. b-c) Algorithm's error in predicting preferences and ordinal labels (mean $\pm$ std). Each simulation evaluated performance at 1000 randomly selected points; the model posterior was used to predict preferences between consecutive pairs of points and ordinal labels at each point.

Results

Simulation Results. We evaluate ROIAL's performance on the Hartmann3 (H3) function—which is a standard benchmark for learning non-convex, smooth functions—and on 3-dimensional synthetic functions, sampled from a Gaussian process prior over a $20 \times 20 \times 20$ grid. As evaluation metrics, we use the algorithm's errors in preference and ordinal label prediction; these allow us to quantify performance when the true utility function is unknown. The average ordinal prediction error is defined as $\overline{\text{Error}}(N) := \frac{1}{N}\sum_{k=1}^{N} |y_k^{pred} - y_k^{true}|$, and all simulations use 5 ordinal

categories.[1]

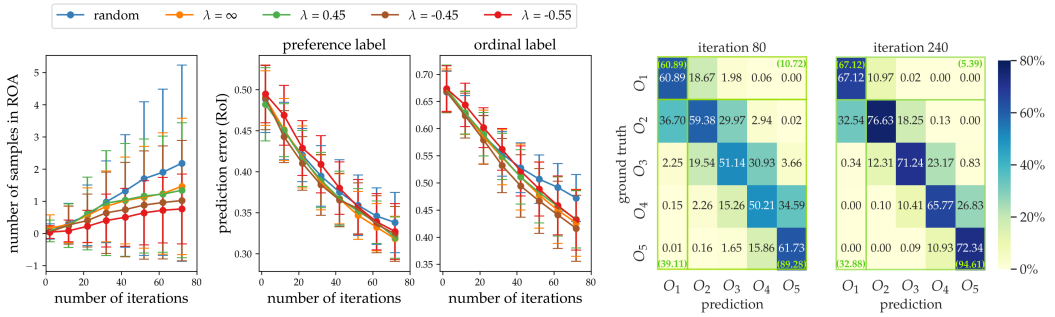
1D illustration of ROIAL. Fig. 4.2 illustrates the algorithm for a 1D objective function. Initially, ROIAL samples widely across the action space (Fig. 4.2(a)-4.2(c)). As seen by comparing iterations 5 and 20 (Fig. 4.2(c)-4.2(d)), the algorithm stops querying points in the ROA (actions in $\mathbf{O}_1$) because the upper confidence bound (top of the blue shaded region) there falls below the hyperparameter b_1 (dotted gray line).

Extending to higher dimensions. To characterize the impact of the random subset size on algorithmic performance, we compare performance of different sizes in simulation for both the H3 and synthetic functions. We calculate the posterior over the entire action space only every 10 steps to reduce computation time, and then use this posterior to evaluate the algorithm's error in predicting preference and ordinal labels. Fig 4.3(a) provides an example of a 3D posterior, Fig. 4.3(b) depicts the average performance for H3 over 10 simulation repetitions, and Fig. 4.3(c) shows the average performance over a set of 50 unique synthetic functions. We find that a subset size of at least 5 yields performance close to using all points.

Estimating the region of interest. We demonstrate the effect of the confidence parameter λ on the number of actions sampled from the ROA and on prediction error in the ROI. Fig. 4.4(a) demonstrates that across various values of λ, visits to the ROA decrease as λ decreases. To confirm that restricting queries to the estimated ROI does not harm performance, we also compare label prediction error in the ROI across values of λ. When $\lambda = -0.45$, ROIAL achieves similar preference prediction accuracy and slightly-improved ordinal label prediction within the ROI compared to $\lambda = \infty$, which permits sampling over the entire action space (Fig. 4.4(a)). Additionally, the confusion matrix (Fig. 4.4(b)) shows that the algorithm usually predicts either the correct ordinal label or an adjacent ordinal category. The ROI prediction accuracy (green text in Fig. 4.4(b)) indicates that ROIAL predicts whether points belong to the ROI with relatively-high accuracy.

Robustness to noisy feedback. Since user feedback is expected to be noisy, we evaluate the algorithm's robustness to noisy feedback generated from the distributions $P(y \mid \boldsymbol{f}, \boldsymbol{a}) = g_o\left(\frac{\tilde{b}_y - f(\boldsymbol{a})}{\tilde{c}_o}\right) - g_o\left(\frac{\tilde{b}_{y-1} - f(\boldsymbol{a})}{\tilde{c}_o}\right)$ and $P(\boldsymbol{a}_1 \succ \boldsymbol{a}_2 \mid \boldsymbol{f}) = g_p\left(\frac{f(\boldsymbol{a}_1) - f(\boldsymbol{a}_2)}{\tilde{c}_p}\right)$ for ordinal and preference feedback, respectively, with true ordinal thresholds $\{\tilde{b}_j | j = 1, \ldots, r-1\}$ and simulated noise parameters $\tilde{c}_p$ and $\tilde{c}_o$. We set $\tilde{c}_o > \tilde{c}_p$

[1] Unless otherwise stated, hyperparameters are held constant across simulations and experiments, and their values can be found in [101].



(a) Sample number in ROA & prediction error in ROI (b) Confusion matrices

Figure 4.4: Effect of the confidence interval. All simulations are run over 50 synthetic functions with a random subset size of 500. a) Left: cumulative number of actions in the ROA ($\mathbf{O}_1$) queried at each iteration (mean $\pm$ std). Note that as λ increases, more samples are required for the confidence interval to fall below the ROA threshold, at which point ROIAL starts avoiding the ROA. Middle and right: error in predicting preference and ordinal labels for different values of λ; predictions are over 1,000 random actions (mean $\pm$ std). b) Confusion matrices (column-normalized) of ordinal label prediction over the entire action space at iterations 80 and 240 with λ = -0.45. The 2×2 confusion matrices for ROI prediction accuracy are outlined in green. Prediction accuracy increases with the number of iterations.

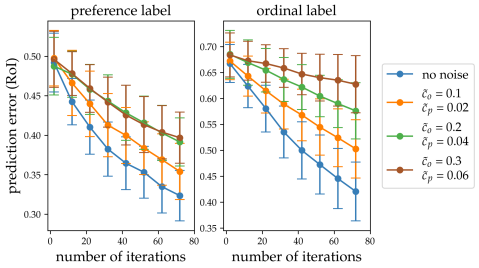


Figure 4.5: Effect of noisy feedback. The ordinal and preference noise parameters, $\tilde{c}_0$ and $\tilde{c}_p$, range from 0.1 to 0.3 and 0.02 to 0.06, respectively. All cases use a random subset size of 500 and $\lambda = -0.45$, and each simulation uses 1,000 random actions to evaluate label prediction. Plots show means $\pm$ standard deviation.

because we expect ordinal labels to be noisier than preferences, as they require users to recall all past experience to give consistent feedback, whereas a preference only involves the previous and current action. The algorithm learns more slowly with noisier feedback (Fig. 4.5).

Exoskeleton Experiments. After demonstrating ROIAL's performance in simulation, we experimentally deployed it on the lower-body exoskeleton Atalante, developed by Wandercraft (video: [107], ROIAL hyperparameters: [101]). Atalante, shown in Fig.

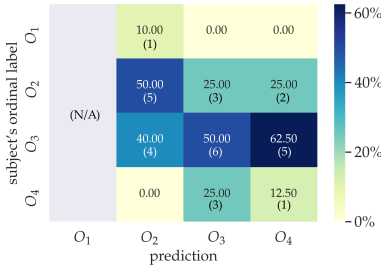


Figure 4.6: Confusion matrix of the validation phase results for all three subjects. The first column is grey because actions in the ROA ($\mathbf{O}_1$) were purposefully avoided to prevent subject discomfort. Percentages are normalized across columns. Parentheses show the numbers of gait trials in each case.

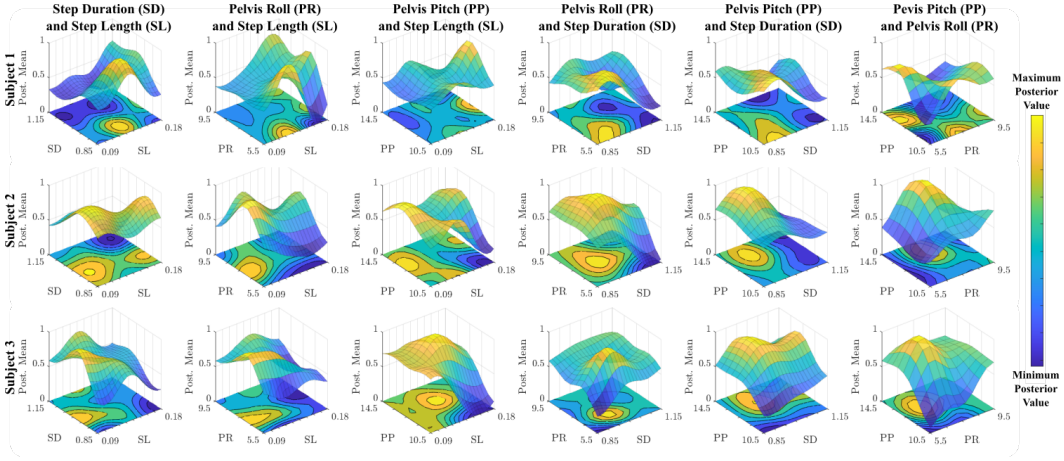


Figure 4.7: 4D posterior mean utility across exoskeleton gaits. Utilities are plotted over each pair of gait space parameters, with the values averaged over the remaining 2 parameters in each plot. Each row corresponds to a subject: Subject 1 is the most experienced exoskeleton user, Subject 2 is the second-most experienced user, and Subject 3 never used the exoskeleton prior to the experiment.

4.1, is an 18 degree of freedom robot designed to restore assisted mobility to patients with motor complete paraplegia through the control of 12 actuated joints: 3 joints at each hip, 1 joint at each knee, and 2 degrees of actuation in each ankle. For more details on Atalante, refer to [60, 58, 59].

Dynamically stable crutch-less exoskeleton walking gaits are generated through nonconvex optimization techniques (see Section II of [22]), based on the theory of hybrid zero dynamics (HZD) introduced by [21] and the HZD-based optimization method presented in [108]. These periodic gaits are parameterized by various features, and this studies focuses on four: step length (SL) in meters, step duration (SD) in seconds, maximum pelvis roll (PR) in degrees, and maximum pelvis pitch (PP) in degrees (Fig. 4.1). These parameters were selected because exoskeleton

users frequently suggested modifications to SL, SD, and PR in prior work [109], and we wanted to further study the relationship between PR and PP. We discretized these parameters into bins of sizes 10, 7, 5, and 5, respectively, resulting in 1,750 actions within a 4D action space. ROIAL randomly selected 500 actions in each iteration and used $\lambda = 0.45$ to estimate the ROI.

The experimental procedure was conducted for three non-disabled subjects and consisted of 40 trials divided into a *training phase* (30 trials) and a *validation phase* (10 trials). Subjects were not informed of when the validation phase began. Subjects provided ordinal labels for all 40 gaits, and optional pairwise preferences between the current and previous gaits for all but the first trial. Four ordinal categories were considered and described to the users as:

1. **Very Bad** ($\mathbf{O}_1$): User feels unsafe or uncomfortable to the point that the user never wants to repeat the gait.
2. **Bad** ($\mathbf{O}_2$): User dislikes the gait but does not feel unsafe or uncomfortable.
3. **Neutral** ($\mathbf{O}_3$): User neither dislikes nor likes the gait and would be willing to try the gait again.
4. **Good** ($\mathbf{O}_4$): User likes the gait and would be willing to continue walking with it for a long period of time.

While including additional ordinal categories could increase the potential information gain from each query, it also increases the cognitive burden for the users and thus makes the labels less reliable. Validation actions were selected so that at least two samples were predicted to belong to $\mathbf{O}_2, \mathbf{O}_3$, and $\mathbf{O}_4$, with the remaining four validation actions sampled at random. Actions predicted to belong in $\mathbf{O}_1$ were excluded because they are likely to make the user feel uncomfortable or unsafe, and actions sampled during the training phase were explicitly excluded from the validation trials.

Experimental results. Figure 4.6 depicts the results of the validation phase for all three subjects. These results show a reliable correlation between the predicted categories and the users' reported ordinal labels, in which the majority of the predicted ordinal labels are within one category of the true ordinal labels.

Since less than 2% of the action space was explored during the experiment, we expect that the prediction accuracy would increase with additional exoskeleton trials

as observed in simulation (Fig 4.4(b)). Overall, these results suggest that ROIAL can yield reliable preference landscapes within a moderate number of samples.

Figure 4.7 depicts the final posterior mean for each of the subjects. These utility functions highlight both regions of agreement and disagreement among the subjects. For example, all subjects strongly dislike gaits at the lower bound of PP and lower bound of PR. However, all subjects disagree in their utility landscapes across SL and SD. This type of insight could not be derived from direct gait optimization, which mostly obtains information near the optimum.

We also evaluated the effect of each gait parameter on the posterior utility using the permutation feature importance metric. The results of this test for each respective subject across the four gait parameters (SL, SD, PR, PP) are: (0.20, 0.30, 0.33, 0.27), (0.26, 0.36, 0.38, 0.29), and (0.23, 0.16, 0.21, 0.45). These values suggest that the preferences of more experienced users (Subjects 1 and 2) may be most influenced by SD and PR, while the least-experienced user's feedback may be most weighted by PP (Subject 3). The code for this test is available on GitHub [101]. These results demonstrate that ROIAL is capable of obtaining preference landscapes within relatively-few exoskeleton trials while avoiding gaits that make users feel unsafe or uncomfortable.

Summary

We present the ROIAL framework for actively learning utility functions within a region of interest from pairwise preferences and ordinal feedback. The ROIAL algorithm is experimentally demonstrated on the lower-body exoskeleton Atalante for three non-disabled subjects (video: [107]). In simulation, ROIAL predicts utilities in the ROI while learning to stay away from the ROA. In experiments, ROIAL typically predicts subjects' ordinal labels correctly to within one ordinal category. Furthermore, the results illustrate that gait preference landscapes vary across subjects. In particular, a feature importance test suggests that the two more-experienced users prioritized step duration and pelvis roll, while a new user prioritized pelvis pitch.

Making conclusive claims about gait preference landscapes requires conducting these experiments on patients with motor complete paraplegia, as well as scaling up the experiments. Another limitation of this work is the high noise in users' ordinal labels, which may depend on factors such as prior experience and bias due to the gait execution order. Thus, future work includes designing a study to directly

quantify the noise in exoskeleton users' ordinal labels. Future work also includes continuing the experiments over more trials, as prediction accuracy is expected to improve with additional data. To conclude, the ROIAL algorithm provides a principled methodology for characterizing exoskeleton users' preference landscapes in high-dimensional action spaces. This work contributes to better understanding the mechanisms behind user-preferred walking and optimizing future gait generation for user comfort.

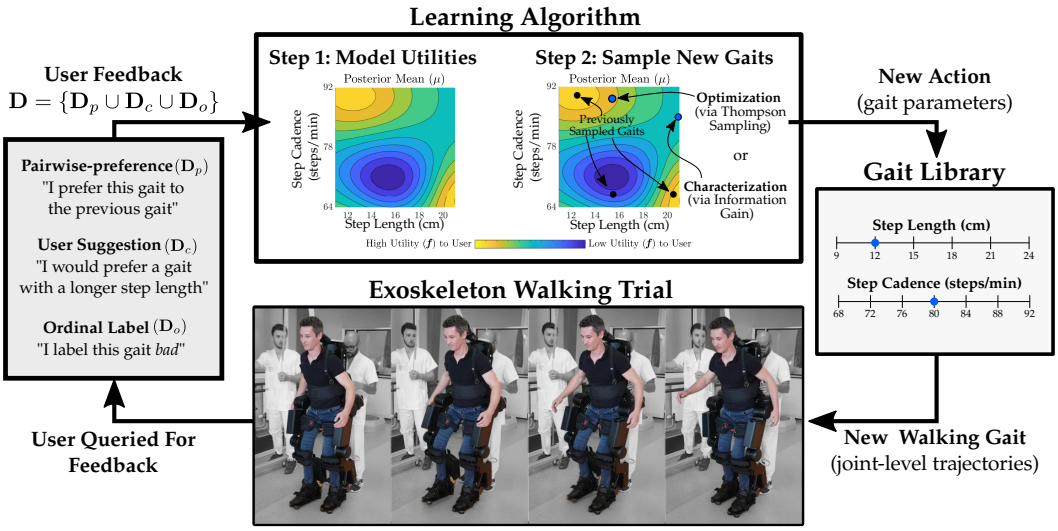


Figure 4.8: Illustration of the preference-based learning framework applied to exoskeleton gait optimization and characterization. The learning framework consists of four main components: 1) collecting subjective user feedback from exoskeleton users; 2) using the collected feedback to model the underlying preference landscape as a Gaussian Process and selecting new actions to sample from this GP; 3) translating the selected actions into a corresponding walking gait; and 4) allowing the user to experience the walking gait on the Atalante exoskeleton.

4.2 Unified Preference-based Learning Framework

In this section, we present experimental results from a unified preference-based learning framework that combines ROIAL with CoSpar [22] and LineCoSpar [82]. Together, these methods form a cohesive toolbox for preference-based optimization and characterization [110], offering a practical framework for applying such algorithms in clinical and experimental settings. The framework jointly supports *preference optimization* and *preference characterization*, enabling both efficient gait personalization and a richer understanding of the underlying preference landscape. We evaluate this approach on the Atalante exoskeleton with participants with paraplegia, incorporating not only preference and ordinal feedback but also coactive feedback that essentially are user suggestions on potential improvement of gaits. Beyond optimizing forward walking parameters, we further demonstrate its application to tuning a turning controller, thereby enhancing both maneuverability and overall walking comfort.

Before detailing the unified framework, we first review several key concepts and algorithms it builds upon that have not yet been introduced: coactive feedback, sampling strategies for regret minimization, dimensionality reduction techniques for

high-dimensional action spaces, and the LineCoSpar algorithm.

Background on Existing Methods

Coactive Feedback

User suggestions, also known as coactive feedback, can be incorporated into the learning framework by treating user-proposed improvements as implicit preferences. In this setting, a suggestion $\bar{\boldsymbol{a}}$ for improving a sampled action $\boldsymbol{a}$ is interpreted as $f(\bar{\boldsymbol{a}}) > f(\boldsymbol{a})$. This approach follows the *coactive learning* framework [111, 112], in which the user identifies an improved action as feedback to each presented action. Combining preference and coactive feedback is known as *mixed-initiative* learning [113, 114]. While coactive feedback can be noisier than pairwise preferences—especially in exoskeleton walking, where an improvement suggested before walking may feel worse in practice—it increases the information gained per trial and can improve sample efficiency.

Sampling for Regret Minimization

We use Thompson sampling [115] as our primary sampling strategy for regret minimization. Thompson sampling selects actions in proportion to their probability of being optimal under the posterior utility model, balancing exploration and exploitation. Compared to Relative Upper Confidence Bound (RUCB) [116], Thompson sampling tends to favor exploitation slightly, which has been shown to be advantageous in preference-based settings [22].

LineCoSpar and Dimensionality Reduction

In high-dimensional action spaces, updating the posterior over all possible actions can be computationally prohibitive. LineCoSpar [82] addresses this challenge by extending CoSpar [22] with a dimensionality reduction strategy inspired by [117]. Specifically, it restricts posterior updates to a subset $\mathbf{S} = \mathbf{L} \cup \mathbf{V}$, where $\mathbf{L}$ is a one-dimensional subspace passing through the current estimated optimum, and $\mathbf{V}$ is the set of previously visited actions. This approach reduces the complexity of posterior updates from scaling with the full action space dimension to scaling with the much smaller subset size, enabling efficient online learning without significant loss in optimization performance. By focusing exploration along lines through promising actions while retaining previously evaluated points, LineCoSpar, outlined in Alg.

Algorithm 2 LINECOSPAR

procedure LINECOSPAR(Utility prior parameters; m = granularity of discretization)
  $\mathbf{D} = \emptyset$, $\mathbf{V} = \emptyset$ ▷ $\mathbf{D}$: preference data, $\mathbf{V}$: visited actions
  Set $\boldsymbol{a}_1^*$, $\boldsymbol{a}_0$ to uniformly-random actions
  for t = 1, 2,. . . , T **do**
    $\mathbf{L}_t$ = random line through $\boldsymbol{a}_t^*$, discretized via m
    $\mathbf{S}_t = \mathbf{L}_t \cup \mathbf{V}$ ▷ Points over which to update posterior
    $\mathrm{Normal}(\boldsymbol{\mu}_t, \Sigma_t)$ = posterior over points in $\mathbf{S}_t$, given $\mathbf{D}$
    Sample utility function $f_t \sim \mathrm{Normal}(\boldsymbol{\mu}_t, \Sigma_t)$
    Execute action $\boldsymbol{a}_t = \mathrm{argmax}_{\boldsymbol{a} \in \mathbf{S}_t} f_t(\boldsymbol{a})$
    Add pairwise preference between $\boldsymbol{a}_t$ and $\boldsymbol{a}_{t-1}$ to $\mathbf{D}$
    Add coactive feedback $\boldsymbol{a}_t'$ to $\mathbf{D}$
    Set $\mathbf{V} = \mathbf{V} \cup \{\boldsymbol{a}_t\} \cup \{\boldsymbol{a}_t'\}$ ▷ Update actions in $\mathbf{V}$
    Set $\boldsymbol{a}_{t+1}^* = \mathrm{argmax}_{\boldsymbol{a} \in \mathbf{V}_t} \boldsymbol{\mu}_t(\boldsymbol{a})$
  end for
end procedure

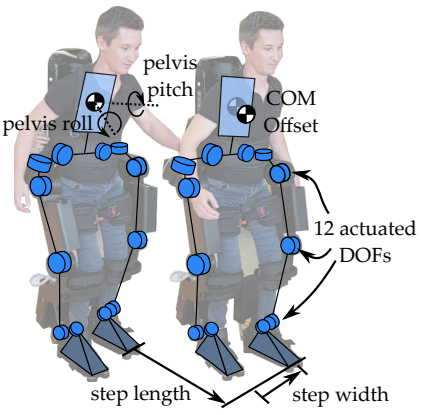
2,maintains strong optimization performance in high-dimensional gait parameter spaces with far fewer required user interactions.

Experimental Results

The components reviewed above—coactive feedback for richer interaction, Thompson sampling for regret minimization, LineCoSpar for scalable high-dimensional optimization, and ROIAL for preference characterization introduced in the previous section—form the foundation of the unified preference-based learning framework. We evaluate this framework on the Atalante exoskeleton across multiple action spaces (Fig. 4.9), with the goal of assessing its ability to support both preference optimization and preference characterization. We begin by presenting results from experiments conducted with participants with paraplegia.

Patients with Paraplegia

We demonstrate the entire preference-based learning framework for two subjects with complete motor paraplegia. These subjects will be referred to as Subject 9 and 10 since there was a total of eight non-disabled subjects. On the AISA impairment scale[118], the subjects both classify as completely impaired (AISA A), but differ in that Subjects 9 and 10 have T10 and T5 levels of injury, respectively. The subjects also differed in their prior experience with the Atalante exoskeleton, with Subject 9 having over 300 hours of prior experience and Subject 10 having fewer than 30.



	Parameter	$a_i^{\min}$	$a_i^{\max}$	d_i
a	a_1: Step Length (m)	0.08	0.18	0.01
	a_2: Step Cadence (steps/min)	52.2	70.6	3.067
	a_3: Step Width (m)	0.25	0.30	0.01
	a_4: Peak Step Height (m)	0.065	0.075	0.0025
	a_5: Max. Pelvis Roll (deg)	5.5	9.5	1
	a_6: Max. Pelvis Pitch (deg)	10.5	14.5	1
b	a_1: Step Length (m)	0.11	0.21	0.02
	a_2: Step Cadence (steps/min)	64	92	2
	a_3: Center of Mass (COM) Offset (cm)	-1	1	0.2
c	a_1: Rotation Angle (deg)	10	30	5
	a_2: Duration of Step 1 (s)	0.9	1.6	0.175
	a_3: Duration of Step 2 (s)	0.7	1.6	0.175
	a_4: Center of Mass (COM) Offset (mm)	-20	20	5
	a_5: Height of Step 1 (cm)	-15	40	5
	a_6: Height of Step 2 (cm)	-40	40	5

Figure 4.9: The preference-based learning framework aims to modify exoskeleton behavior through the selection of various gait parameters, such as those illustrated in the figure. The presented experiments demonstrate the methodology across three separate action spaces: a) the action space for experiments with non-disabled subjects; b) the action space for experiments with subjects with paraplegia; and c) the action space for exoskeleton turning experiments. Here, $a_i^{\min}$ and $a_i^{\max}$ are the minimum and maximum bounds, respectively, for each action space parameter, with d_i being the interval between neighboring actions.

As discussed earlier, these experiments explore a slightly different gait library than that considered in the first set of experiments. It is parameterized by three gait features: step length (cm), step cadence (steps/min), and center of mass offset (% offset), which shifts the user's center of mass either forward or backward during walking. We explore this gait library because it has CE certification in Europe for use in clinical settings. Figure 4.9 provides the action space definition associated with this gait library.

Unlike the experiments with non-disabled subjects, five minute breaks were taken every 20 minutes to prevent subjects from developing pressure sores. Due to the longer total duration, these experiments were broken over two days of testing. During the first session, the ROIAL algorithm was deployed for 15 learning iterations to coarsely characterize user comfort across the action space. As before, the subjects provided ordinal labels of "very bad," "bad," "neutral," and "good," with the "very bad" label defining the ROA. The preference landscapes learned in these first sessions are illustrated in the top row of Figure 4.10.

In the second session, we continued the learning process using the LineCoSpar algorithm to learn the parameters optimizing user comfort, with the final preference landscapes shown in the middle row of Figure 4.10 and the gaits identified as optimal illustrated via gait tiles in the bottom row. These second experimental sessions were

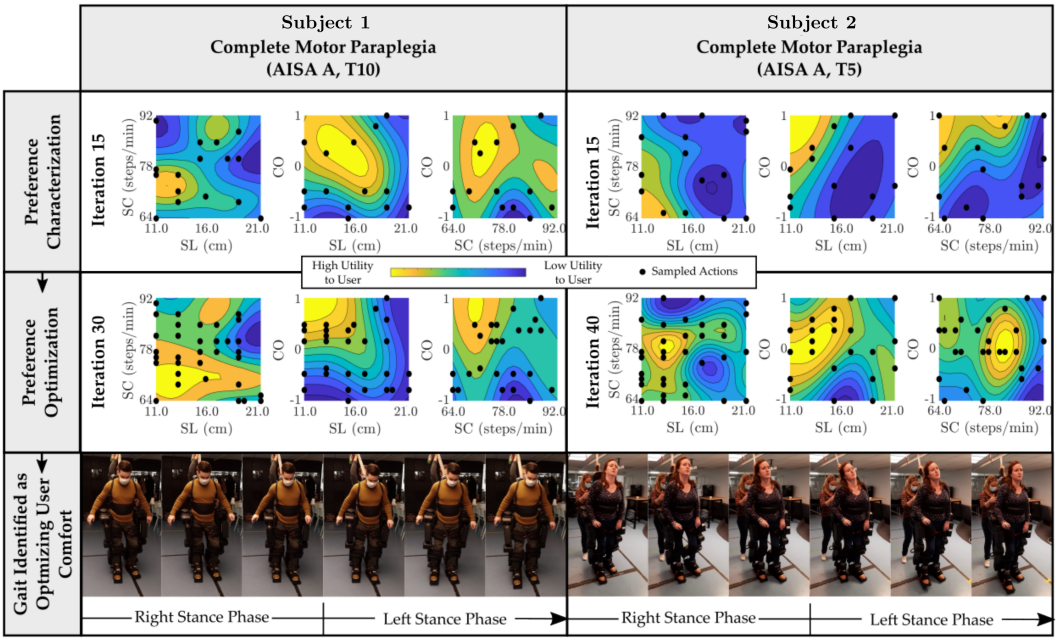


Figure 4.10: Experimental results of unified framework during exoskeleton walking for subjects with paraplegia. We illustrate the experimental results from applying the learning framework towards preference characterization and preference optimization for two subjects with complete motor paraplegia. Preference characterization experiments were first conducted via two-hour experimental sessions with the ROIAL algorithm. The landscapes obtained after these first sessions, shown in the top row, indicate that the two subjects have similar relationships between gait parameters and comfort. To identify the gait optimizing user comfort for each subject, we continued learning in additional two-hour experimental sessions using the LineCoSpar algorithm. The landscapes obtained after these second sessions are shown in the middle row of the figure. These updated landscapes indicate that while the subjects had similar gait characterization results, the gaits optimizing user comfort differ between these users. The step length (SL), step cadence (SC), and center of mass offset (CO) for the gaits identified as optimal, as depicted in the gait tiles in the bottom row, were [0.11 cm, 74 steps/min, 0.5 cm] and [0.13 cm, 80 steps/min, 0cm]. Lastly, it can be seen that actions are sampled more uniformly during preference characterization (sampled actions are marked with a black circle), and actions with higher underlying utility values were sampled more frequently during preference optimization.

conducted for 15 and 25 iterations for Subjects 9 and 10, respectively. Sessions terminated when the operator felt that the algorithm had identified an exoskeleton gait that sufficiently optimized user comfort. With additional iterations, it is likely that the algorithm would continue to converge to optimal behavior, but at a slower rate.

Since experiment time with the subjects was limited, we only conducted one

evaluation trial at the conclusion of each second session. During this trial, the subject was unknowingly given the gait that optimized the final posterior mean. The subjects were queried for feedback as usual, and both labeled the optimal gait as "good" (the highest ordinal category). This feedback indicates that after obtaining a general preference landscape across the entire gait library, the framework could successfully learn to identify gaits that optimized user comfort.

Exoskeleton Turning

To emphasize the application-agnostic nature of the preference-based learning framework, it is further applied towards optimizing user-comfort during exoskeleton turning. Similar to walking, turning is achieved by generating stable joint-level trajectories via the PHZD method. In this work, turning behavior consists of rotating the exoskeleton about its vertical axis through two distinct steps, with each step lifting and rotating either the left or right foot. As in the walking experiments, we define each unique turning behavior via several user-defined parameters. In our experiment, we chose six parameters to explore: rotation angle (degrees), duration of the first step (seconds), duration of the second step (seconds), center of mass offset (mm), height of the first step (cm), and height of the second step (cm). This action space definition is detailed in Figure 4.9, and illustrated in the top row of Figure 4.11.

First, we conducted a preference characterization phase using ROIAL, in which 50 learning iterations were performed over a coarse action space to obtain a general preference landscape. This rough landscape is illustrated for four two-dimensional cross-sections in the top row of Figure 4.11. Following these initial 50 iterations, an additional 10 iterations of preference optimization were conducted over the coarse action space, with the resulting posteriors illustrated in the middle row of Figure 4.11. Finally, to fine-tune the action predicted to maximize user comfort, we conducted an additional 40 iterations of preference optimization over a more finely discretized action space. The final posterior over user utilities learned from all 100 iterations is illustrated in the bottom row of Figure 4.11.

To evaluate the experimental results, we compared the parameters identified as optimizing user comfort to hand-tuned parameters. The optimal action identified by the learning framework after completion of the 100 iterations has the following values: $[20 \text{ deg}, 0.9 \text{ s}, 0.875 \text{ s}, 15 \text{ mm}, 5 \text{ cm}, 5 \text{ cm}]$. In comparison, the optimal action identified by the expert operator after approximately 2 months of manual tuning was $[22.5 \text{ deg}, 0.92 \text{ s}, 0.86 \text{ s}, -80 \text{ mm}, 0 \text{ cm}, 0 \text{ cm}]$. Aside from center of

mass offset (CO), these actions are very close, especially considering the wide action space range outlined in Figure 4.9. This is striking, considering that the action space (defined in Figure 4.9) contains a total of 275,400 discrete parameter combinations. Notably, the CO values likely differ because the action space only included CO values between -20 and 20 mm; since these values of CO are small, the effect of CO was negligible on the final turning behavior. The expert operator also noted that the algorithm-identified parameters and the manually-tuned parameters resulted in comparable turning behaviors. This indicates that the learning framework successfully identified user-preferred parameters. While this success demonstrates the extensibility of our method, it is important to note that this extension relies on the ability to parameterize the desired behavior. For locomotive behaviors, gait libraries are a common method of parameterization. However, for other human-robot interactive behaviors, our framework requires defining a parameterization that describes the space of all desired behaviors.

Summary

To systematically explore the space of possible design choice for gait parameters, a preference-based learning framework was developed to both directly optimize user comfort (preference optimization) and characterize the underlying preference landscape (preference characterization). Importantly, this framework leverages subjective feedback mechanisms (pairwise preferences, user suggestions, and ordinal labels), which have been shown to be more reliable compared to numerical scores. This framework was demonstrated towards preference optimization and characterization for non-disabled subjects on the Atalante lower-body exoskeleton, as well as for two subjects with complete motor paraplegia.

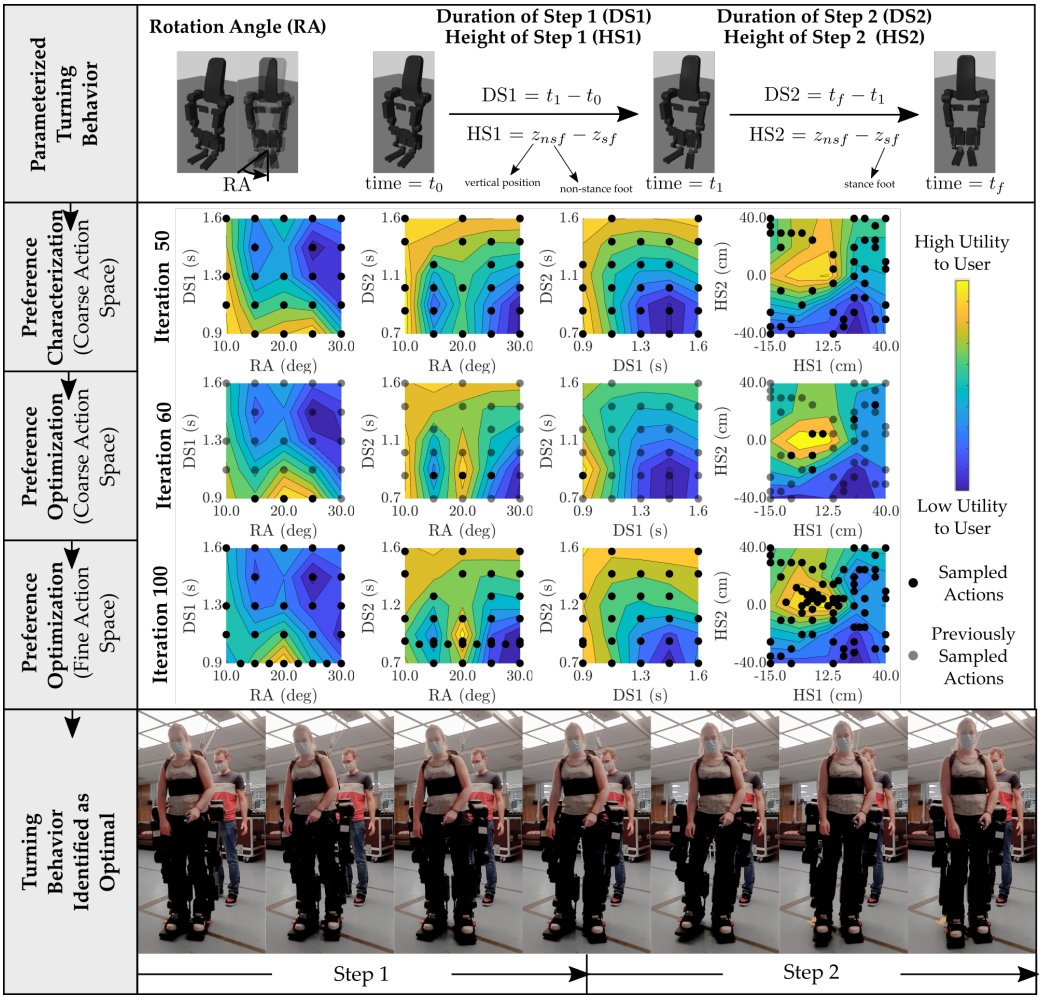


Figure 4.11: Experimental results of unified framework during exoskeleton turning for a non-disabled subject. To demonstrate the learning framework's application-agnostic nature, we applied it to sequentially characterize and optimize user comfort during exoskeleton turning. First, we defined the action space over five parameters of exoskeleton turning behavior: rotation angle (RA) in seconds, duration of the first and second steps (DS1, DS2) in seconds, and height of the first and second steps (HS1, HS2) in centimeters. The experiment was conducted in three separate phases. The ROIAL algorithm was first deployed to characterize user preferences for 50 iterations. Then, we used the LineCoSpar algorithm to find the optimal gait within a coarse action space for an additional 10 iterations. Finally, we fine-tuned the predicted optimal action by using LineCoSpar for another 40 iterations with a more finely-discretized action space.

4.3 Safety-Aware LineCoSpar

In the previous sections of this chapter, we reviewed existing preference-based learning (PBL) methods, including LineCoSpar [82], as well as the ROIAL algorithm for preference characterization within a region of interest (ROI). While LineCoSpar provides an efficient framework for high-dimensional preference optimization, its original formulation does not incorporate safety-awareness, limiting its suitability for safety-critical control systems. To address this limitation, we introduce Safety-Aware LineCoSpar, which augments LineCoSpar with the ROI-based safety-awareness of ROIAL. This directs exploration toward regions unlikely to violate safety constraints while still enabling efficient optimization of user-preferred behaviors.

This challenge reflects a broader issue in modern control design. As systems become increasingly complex and modular—integrating perception, planning, and low-level control—each subsystem must balance safety and performance. When subsystems are designed under conservative worst-case assumptions, overall performance at the system level is sacrificed [119, 120]. Conversely, manually tuning the safety–performance trade-off of each component can improve outcomes [121], but the process is qualitative, time-consuming, and heavily reliant on expert intuition. Embedding safety-awareness directly into the optimization process provides a systematic alternative for balancing these competing demands in user-aligned behavior personalization.

For safety-critical systems, Control Barrier Functions (CBFs) provide a principled model-based framework for enforcing safety guarantees [122, 123, 124]. While extensions exist to handle model uncertainty [125, 126, 127, 128], disturbances [129, 130, 131, 132, 120, 133], and measurement errors [134, 135, 136], combining such robust components typically compounds conservatism. As a result, deployment in practice typically requires loosening theoretical guarantees and labor-intensive manual tuning of controller parameters—a process that underscores the need for more systematic, data-driven approaches.

PBL offers a complementary alternative by leveraging user feedback to guide controller tuning. Instead of manually adjusting parameters, PBL infers a user's latent utility function from qualitative feedback—such as pairwise preferences, ordinal ratings, or coactive suggestions—and optimizes parameters accordingly. This approach has been applied in domains such as exoskeleton gait optimization [22], locomotion [25], and trajectory planning [137, 92], with safety-aware variants

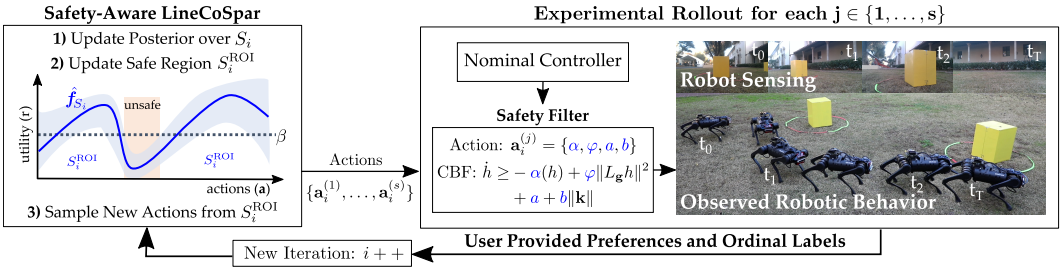


Figure 4.12: An overview of the Safety-Aware Preference-Based Learning design paradigm. Safety-Aware LineCoSpar is used to generate actions which are rolled out in experiments as parameters of the CBF-based safety filter to obtain user preferences and safety ordinal labels which are then used to update the user's estimated utility and generate new actions.

ensuring that unsafe actions are avoided during learning [87, 90, 89]. However, these algorithms often rely on worst-case approximations, again leading to conservative solutions that exclude many feasible and high-performing behaviors.

Safety-Aware LineCoSpar extends preference-based learning to safety-critical domains in a way that respects both safety constraints and user input. By avoiding overly conservative worst-case assumptions, it preserves a broader set of candidate actions for users to shape through feedback. This shifts the tuning process from expert-driven trial-and-error toward user-driven adaptation, enabling systems to more directly reflect user preferences while remaining safety-aware. Ultimately, it offers a scalable paradigm for aligning complex control systems with user needs in real-world settings.

Safety-Aware Sampling

It is important to avoid unsafe actions during sequential decision making in certain applications, such as learning robotic controllers on hardware, where low-reward actions might lead to physical damage of the platform. Safe exploration algorithms [87, 90, 89] considered the setting where actions below a prespecified safety threshold are catastrophic and must be avoided at all cost. In our work, since we construct controllers that account for safety, we adopt a more optimistic learning approach called *safety-aware*. In this case, actions labeled by a human as "unsafe" are not catastrophic but undesirable. Thus, the algorithm *avoids* these actions; whereas the safe exploration algorithms guarantee that no such actions are sampled which can be sometimes exceedingly conservative in settings like ours.

To achieve this safety-awareness, we leverage the approach introduced in [86], which

uses ordinal labels to identify a *region of interest* (ROI) in $\mathcal{A}$. In this work, the ROI is defined to be the actions labeled as "safe". In each iteration i we estimate an ROI within the set S_i as:

$$S_i^{\text{ROI}} = \{\boldsymbol{a} \in S_i \mid \hat{\boldsymbol{f}}_{S_i}(\boldsymbol{a}) + \lambda \boldsymbol{\sigma}_{S_i}(\boldsymbol{a}) > \beta\}, \tag{4.3}$$

where $\hat{\boldsymbol{f}}_{S_i}(\boldsymbol{a})$ and $\boldsymbol{\sigma}_{S_i}(\boldsymbol{a})$ are the posterior mean and standard deviation, respectively, evaluated at the action $\boldsymbol{a} \in S_i$. The variable $\lambda \in \mathbb{R}$ determines how conservative the algorithm would be in estimating the safety region, as illustrated in Figure 4.13. We see that lower values of λ result in fewer unsafe actions being sampled, with only a slight effect on sample-efficiency. The restriction to S_i^{ROI} is added to LineCoSpar by only considering actions in S_i^{ROI} during Thompson sampling. We refer to this as Safety-Aware LineCoSpar (SA-LineCoSpar), with the full algorithm outlined in Alg. 3.

Algorithm 3 Safety-Aware LineCoSpar

Require: s uniformly random initial actions $\mathbf{V}_1 \subset \mathcal{A}$ and corresponding feedback $\mathbf{D}_1$

for $i = 2$ to N **do**
  Update posterior over $\mathbf{V}_{i-1}$
  $\hat{\boldsymbol{a}}_{i-1}^* \leftarrow \arg\max_{\boldsymbol{a} \in \mathbf{V}_{i-1}} \hat{\boldsymbol{f}}_{\mathbf{V}_{i-1}}(\boldsymbol{a})$
  $L_i \leftarrow$ new 1D subspace intersecting $\hat{\boldsymbol{a}}_{i-1}^*$
  $S_i \leftarrow L_i \cup \mathbf{V}_{i-1}$
  Update posterior over S_i
  Determine region of interest S_i^{ROI}
  for $j = 1$ to s **do**
    $f^{(j)} \sim \mathcal{N}(\hat{\boldsymbol{f}}_{S_i}, \boldsymbol{\Sigma}_{S_i})$
    $\boldsymbol{a}_i^{(j)} \leftarrow \arg\max_{\boldsymbol{a} \in S_i^{\text{ROI}}} f^{(j)}$
  end for
  Deploy $\{\boldsymbol{a}_i^{(1)}, \ldots, \boldsymbol{a}_i^{(s)}\}$ on the system
  $\mathbf{V}_i \leftarrow \mathbf{V}_{i-1} \cup \{\boldsymbol{a}_i^{(1)}, \ldots, \boldsymbol{a}_i^{(s)}\}$
  $\mathbf{D}_i \leftarrow \mathbf{D}_{i-1} \cup$ new preferences $\cup$ new ordinal labels
end for

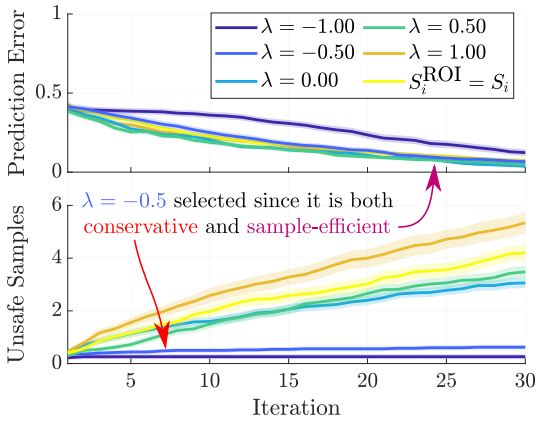


Figure 4.13: Comparison of SA-LineCoSpar and standard LineCoSpar on a synthetic utility function (drawn from the Gaussian prior), averaged over 50 runs. Shaded regions indicate standard error. The safety-aware criteria significantly reduces the number of sampled unsafe actions while maintaining similar prediction error, defined as $|\hat{\boldsymbol{a}}_i^* - \boldsymbol{a}^*|$, where $\hat{\boldsymbol{a}}_i^* \triangleq \text{argmax}_{\boldsymbol{a}}\, \hat{\boldsymbol{f}}_{S_i}$ and $\boldsymbol{a}^* \triangleq \text{argmax}_{\boldsymbol{a}}\, f(\boldsymbol{a})$.

Integrating Safety-Aware Preference-Based Learning with Safety-Critical Control

The nominal safety-critical controller used in this section is synthesized using Control Barrier Functions (CBFs). Notably, the specific formulation of the CBF yields parameters are able to be modified with SA-LineCoSpar to tune the overall performance-robustness trade off. In this subsection, we will outline the utilized controller before presenting the results of the overall learning framework towards tuning its parameters.

Consider the following nonlinear control-affine system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{v} + \mathbf{d}(t)), \tag{4.4}$$

with state $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{v} \in \mathbb{R}^m$, functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ assumed to be locally Lipschitz continuous on their domains, and piecewise continuous disturbance signal $\mathbf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ for which we define $\|\mathbf{d}\|_\infty \triangleq \sup_{t\geq 0} \|\mathbf{d}(t)\|$. Specifying the input via a controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ that is locally Lipschitz continuous on its domain yields the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{k}(\mathbf{x}) + \mathbf{d}(t)). \tag{4.5}$$

We assume for any initial condition $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$ and disturbance $\mathbf{d}$, this system has a unique solution $\mathbf{x}(t)$ for all $t \in \mathbb{R}_{\geq 0}$. We consider this system safe if its state

$\mathbf{x}(t)$ remains in a *safe set* $\mathcal{C} \subset \mathbb{R}^n$, defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, \boldsymbol{\rho}) \geq 0\}, \tag{4.6}$$

where $\boldsymbol{\rho} \in \mathbb{R}^p$ are constant application-specific parameters. We say the set $\mathcal{C} \subset \mathbb{R}^n$ is *forward invariant* if for every $\mathbf{x}_0 \in \mathcal{C}$ the solution $\mathbf{x}(t)$ to (4.5) satisfies $\mathbf{x}(t) \in \mathcal{C}$ for all $t \geq 0$. The system (4.5) is *safe* with respect to $\mathcal{C}$ if $\mathcal{C}$ is forward invariant. Ensuring the safety of the set $\mathcal{C}$ in the absence of disturbances and measurement error can be achieved through *Control Barrier Functions (CBFs)*:

Definition 2 (Control Barrier Functions (CBF) [122])**.** The function h is a Control Barrier Function (CBF) for (4.4) on $\mathcal{C}$ if there exists $\alpha \in \mathcal{K}_\infty^{\mathrm{e}}$[2] such that for all $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{v} \in \mathbb{R}^m} \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, \boldsymbol{\rho})\mathbf{f}(\mathbf{x})}_{L_{\mathbf{f}} h(\mathbf{x}, \boldsymbol{\rho})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, \boldsymbol{\rho})\mathbf{g}(\mathbf{x})}_{L_{\mathbf{g}} h(\mathbf{x}, \boldsymbol{\rho})} \mathbf{v} > -\alpha(h(\mathbf{x}, \boldsymbol{\rho})). \tag{4.7}$$

While it may be possible to synthesize controllers that render a given set $\mathcal{C}$ safe in the presence of disturbances [129], this may result in overly-conservative behavior. Instead, we consider how safety properties degrade with disturbances via the following definition.

Definition 3 (Input-to-State Safety [130])**.** The system (4.5) is Input-to-State Safe (ISSf) with respect to $\mathcal{C}$ if there exists $\gamma \in \mathcal{K}_\infty$ such that for all $\delta \in \mathbb{R}_{\geq 0}$ and disturbances $\mathbf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ satisfying $\|\mathbf{d}\|_\infty \leq \delta$, the set $\mathcal{C}_\delta \subset \mathbb{R}^n$ defined as:

$$\mathcal{C}_\delta = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, \boldsymbol{\rho}) \geq -\gamma(\delta)\}, \tag{4.8}$$

is forward invariant. The function h is an Input-to-State Safe Control Barrier Function (ISSf-CBF) for (4.4) on $\mathcal{C}$ with parameter $\varphi \in \mathbb{R}_{\geq 0}$ if there exists $\alpha \in \mathcal{K}_\infty^{\mathrm{e}}$ such that for all $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{v} \in \mathbb{R}^m} L_{\mathbf{f}} h(\mathbf{x}, \boldsymbol{\rho}) + L_{\mathbf{g}} h(\mathbf{x}, \boldsymbol{\rho})\mathbf{v} - \varphi \|L_{\mathbf{g}} h(\mathbf{x}, \boldsymbol{\rho})\|^2 > -\alpha(h(\mathbf{x}, \boldsymbol{\rho})). \tag{4.9}$$

The parameter $\boldsymbol{\rho} \in \mathbb{R}^p$ contains information about the system's environment that affects safety, such as the location and size of obstacles. In novel environments

[2]We say that a continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is *class* $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_\infty$) if $\alpha(0) = 0$, α is strictly monotonically increasing, and $\lim_{r \to \infty} \alpha(r) = \infty$. We say that a continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is *class* $\mathcal{K}_\infty^e$ ($\alpha \in \mathcal{K}_\infty^{\mathrm{e}}$) if $\alpha(0) = 0$, α is strictly monotonically increasing, $\lim_{r \to \infty} \alpha(r) = \infty$, and $\lim_{r \to -\infty} \alpha(r) = -\infty$.

the system may need to generate estimates of $\boldsymbol{\rho}$ denoted by $\widehat{\boldsymbol{\rho}} \in \mathbb{R}^p$ from complex measurements, such as camera data. The process of converting complex measurements to environmental parameters $\widehat{\boldsymbol{\rho}}$ is often imperfect, leading to error between the estimated and true values (i.e., $\widehat{\boldsymbol{\rho}} \neq \boldsymbol{\rho}$), which can cause safety violations. In this setting, safety can be achieved via *Measurement-Robust Control Barrier Functions (MR-CBFs)*:

Definition 4 (Measurement-Robust Control Barrier Functions [135])**.** The function h is a *Measurement-Robust Control Barrier Function* (MR-CBF) for (4.4) on $\mathcal{C}$ with parameters $a, b \in \mathbb{R}_{\geq 0}$ if there exists $\alpha \in \mathcal{K}_\infty^{\mathrm{e}}$ such that for all $\widehat{\boldsymbol{\rho}} \in \mathbb{R}^p$ and $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{v} \in \mathbb{R}^m} L_{\mathbf{f}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}}) + L_{\mathbf{g}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}})\mathbf{v} - a - b\|\mathbf{v}\| > -\alpha(h(\mathbf{x}, \widehat{\boldsymbol{\rho}})). \tag{4.10}$$

The following theorem summarizes the safety results achieved with these various types of CBFs:

Theorem 1. *Consider the set $\mathcal{C}$ defined in (4.6).*

1. *If h is a CBF for* (4.4) *on $\mathcal{C}$, $\mathbf{d}(t) = \mathbf{0}$ for $t \in \mathbb{R}_{\geq 0}$ and $\widehat{\boldsymbol{\rho}} = \boldsymbol{\rho}$, then there exists a controller $\mathbf{k}$ such that* (4.5) *is safe with respect to $\mathcal{C}$.*
2. *If h is an ISSf-CBF for* (4.4) *on $\mathcal{C}$ with parameter φ and $\widehat{\boldsymbol{\rho}} = \boldsymbol{\rho}$, then there exists a controller $\mathbf{k}$ such that* (4.5) *is ISSf with respect to $\mathcal{C}$ with $\gamma(\delta) = -\alpha^{-1}(-\delta^2/(4\varphi))$ where $\alpha^{-1} \in \mathcal{K}_\infty^e$.*
3. *Assume $L_{\mathbf{f}}h$, $L_{\mathbf{g}}h$, and $\alpha \circ h$ are Lipschitz continuous on their domains, and assume that $\|\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}\| \leq \epsilon$ for some $\epsilon \in \mathbb{R}_{\geq 0}$. Then there exists $\underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ such that if h is an MR-CBF for* (4.4) *on $\mathcal{C}$ with parameters $a, b \in \mathbb{R}_{\geq 0}$ satisfying $a \geq \underline{a}$ and $b \geq \underline{b}$, and $\mathbf{d}(t) = \mathbf{0}$ for $t \in \mathbb{R}_{\geq 0}$, then there exists a controller $\mathbf{k}$ such that* (4.5) *is safe with respect to $\mathcal{C}$.*

In particular, consider the following cascaded nonlinear control-affine system resulting as a modification of (4.4):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\boldsymbol{\kappa}(\boldsymbol{\xi}), \qquad \dot{\boldsymbol{\xi}} = \mathbf{f}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}) + \mathbf{g}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi})\mathbf{u}, \tag{4.11}$$

with additional states $\boldsymbol{\xi} \in \mathbb{R}^{n_{\boldsymbol{\xi}}}$, control input $\mathbf{u} \in \mathbb{R}^{m_{\boldsymbol{\xi}}}$ and functions $\boldsymbol{\kappa} : \mathbb{R}^{n_{\boldsymbol{\xi}}} \to \mathbb{R}^m$, $\mathbf{f}_{\boldsymbol{\xi}} : \mathbb{R}^n \times \mathbb{R}^{n_{\boldsymbol{\xi}}} \to \mathbb{R}^{n_{\boldsymbol{\xi}}}$, and $\mathbf{g}_{\boldsymbol{\xi}} : \mathbb{R}^n \times \mathbb{R}^{n_{\boldsymbol{\xi}}} \to \mathbb{R}^{n_{\boldsymbol{\xi}} \times m_{\boldsymbol{\xi}}}$ assumed to be locally Lipschitz continuous on their domains. We note that the input $\mathbf{v}$ from (4.4) was replaced by $\boldsymbol{\kappa}(\boldsymbol{\xi})$. These dynamics may represent Euler-Lagrange systems such as robots, where

$\mathbf{x}$ reflects base position, $\boldsymbol{\xi}$ captures base velocities and joint positions and velocities, and the input $\mathbf{u}$ reflects the torques applied to the joints.

Given this cascaded system, we utilize the low-dimensional subsystem to ensure that $\mathcal{C}$ is ISSf by making two assumptions. First, we assume the safe set $\mathcal{C}$ can be described as in (4.6), such that it only depends on the states $\mathbf{x}$ and parameters $\boldsymbol{\rho}$, and not the states $\boldsymbol{\xi}$. For example, in the context of a robotic system, this assumption is justified if safety is described as keeping the base position of the robot away from obstacles. Second, we assume there exists a controller $\boldsymbol{\pi} : \mathbb{R}^n \times \mathbb{R}^{n_{\boldsymbol{\xi}}} \times \mathbb{R}^m \to \mathbb{R}^{m_{\boldsymbol{\xi}}}$ and $\mu_d \in \mathbb{R}_{\geq 0}$ such that for any continuous, bounded signal $\mathbf{s} : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$, the closed-loop system:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}) + \mathbf{g}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi})\boldsymbol{\pi}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{s}(t)), \tag{4.12}$$

satisfies the following implication:

$$\|\boldsymbol{\kappa}(\boldsymbol{\xi}(0)) - \mathbf{s}(0)\| \leq \mu_d \implies \|\boldsymbol{\kappa}(\boldsymbol{\xi}(t)) - \mathbf{s}(t)\| \leq \mu_d, \quad t \in \mathbb{R}_{\geq 0}. \tag{4.13}$$

This assumption reflects that a separate controller may be designed for the high-dimensional dynamics to track well-behaved reference signals synthesized via the low-dimensional model. In particular, if a continuous controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ is designed for the low-dimensional system (4.4) and $\|\boldsymbol{\kappa}(\boldsymbol{\xi}(0)) - \mathbf{k}(\mathbf{x}(0))\| \leq \mu_d$, then we have that the controller $\boldsymbol{\pi}$ ensures $\|\boldsymbol{\kappa}(\boldsymbol{\xi}(t)) - \mathbf{k}(\mathbf{x}(t))\| \leq \mu_d$ for $t \in \mathbb{R}_{\geq 0}$. With this assumption in mind, we may study the ISSf behavior of the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{k}(\mathbf{x}) + \mathbf{d}(t)), \qquad \dot{\boldsymbol{\xi}} = \mathbf{f}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}) + \mathbf{g}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi})\boldsymbol{\pi}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{k}(\mathbf{x})), \tag{4.14}$$

with the disturbance defined as $\mathbf{d}(t) = \boldsymbol{\kappa}(\boldsymbol{\xi}(t)) - \mathbf{k}(\mathbf{x}(t))$ satisfying $\|\mathbf{d}\|_\infty \leq \mu_d$.

Combined Robust CBFs for PBL

We now combine the robustness properties of MR-CBFs and ISSf-CBFs to account for measurement uncertainty and the disturbance, $\mathbf{d}$, allowing us to make robust safety guarantees for the full system (4.14). This is formalized in the following theorem:

Theorem 2. *Given the set $\mathcal{C}$ defined in* (4.6)*, suppose the functions $L_{\mathbf{f}}h$, $L_{\mathbf{g}}h$, $\|L_{\mathbf{g}}h\|^2$, and $\alpha \circ h$ are Lipschitz continuous on their domains, and assume that $\|\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}\| \leq \epsilon$ for some $\epsilon \in \mathbb{R}_{\geq 0}$. Then there exists $\underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ such that if h satisfies:*

$$\sup_{\mathbf{v} \in \mathbb{R}^m} L_{\mathbf{f}}h(\mathbf{x}, \widehat{\boldsymbol{\rho}}) + L_{\mathbf{g}}h(\mathbf{x}, \widehat{\boldsymbol{\rho}})\mathbf{v} - \varphi\|L_{\mathbf{g}}h(\mathbf{x}, \widehat{\boldsymbol{\rho}})\|^2 - a - b\|\mathbf{v}\| > -\alpha(h(\mathbf{x}, \widehat{\boldsymbol{\rho}})), \tag{4.15}$$

for all $\mathbf{x} \in \mathbb{R}^n$ *and some* $a, b \in \mathbb{R}_{\geq 0}$ *satisfying* $a \geq \underline{a}$ *and* $b \geq \underline{b}$*, then there exists a controller* $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ *such that* (4.14) *is ISSf with respect to* $\mathcal{C}$ *with* $\gamma(\delta) = -\alpha^{-1}(-\delta^2/(4\varphi))$.

The proof of this theorem can be found in the extended version of the corresponding publication[3]. As in [59], (4.15) can be incorporated as a constraint into a safety filter on a locally Lipschitz continuous nominal controller $\mathbf{k}_{\text{nom}} : \mathbb{R}^n \to \mathbb{R}^m$. We call this filter the Tunable Robustified Optimization Program (TR-OP) with tunable parameters α, φ, a, and b.

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{v} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \|\mathbf{v} - \mathbf{k}_{\text{nom}}(\mathbf{x})\|^2 \tag{TR-OP}$$
$$\text{s.t. } L_{\mathbf{f}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}}_i) + L_{\mathbf{g}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}}_i)\mathbf{v} - \varphi \|L_{\mathbf{g}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}}_i)\|^2 - a - b\|\mathbf{v}\| \geq -\alpha h(\mathbf{x}, \widehat{\boldsymbol{\rho}}_i),$$
$$\forall i \in \{1, \ldots, N_o\}.$$

Here we use a linear class $\mathcal{K}_\infty^{\text{e}}$ function with coefficient $\alpha \in \mathbb{R}_{>0}$. If we wish to enforce multiple safety constraints, such as in obstacle avoidance with several obstacles, $\widehat{\boldsymbol{\rho}}_i$ can be used to indicate the measured parameters of the i^{th} obstacle, with $N_o \in \mathbb{N}$ being the total number of obstacles. Enforcing this constraint for $N_o > 1$ can be viewed as Boolean composition of safe sets [138]. Additionally, this safety filter is a Second-Order Cone Program (SOCP) [139] for which an array of solvers exist including ECOS [140].

Integrating Learning to Tune the Control Barrier Function

The parameter selection process of TR-OP is particularly important, since the parameters $\underline{a}$ and $\underline{b}$ guaranteed to exist by Theorem 2 are worst-case approximations of the uncertainty generated using Lipschitz constants. Such approximations often lead to undesired conservatism and may render the system incapable of performing its goal (as seen in Figure 4.14). Thus, as illustrated in Figure 4.12, we propose utilizing SA-LineCoSpar to identify user-preferred parameters of TR-OP. This relaxes the worst-case over-approximation to experimentally realize performant and safe behavior. This design paradigm relies on the tunable construction of TR-OP, allowing us to define the actions for SA-LineCoSpar to $\boldsymbol{a} = (\alpha, \varphi, a, b)$. We note the construction of TR-OP assures that unsafe actions are not necessarily catastrophic, as any $\alpha, \varphi, a, b > 0$ endows the system with a non-zero degree of robustness to

[3]Extended Version: `https://arxiv.org/abs/2112.08516`.

hyperparameter	value
λ	-0.5
β	0

name	min.	max.	Δ
α	0.5	5	0.5
φ	0	1	0.1
a	0	1	0.1
b	0	0.05	0.005

Table 4.1: Preference-based learning setup. (Left) Hyperparameters dictating the algorithmic conservativeness when estimating if actions are within the region of interest. (Right) Control barrier function parameter bounds and discretizations (Δ) used to define the action space.

disturbances and measurement error. This assurance allows us to utilize a safety-aware approach where unsafe actions are considered undesirable as opposed to more conservative safety-critical approach to learning where unsafe actions are considered catastrophic.

Experimental Results on Unitree A1

Ultimately, the application of SA-LineCoSpar applied towards tuning the parameters of TR-OP was demonstrated for perception-based obstacle avoidance task with a Unitree A1 quadrupedal robot (Figure 4.12) in simulation and on hardware for both indoor and outdoor environments (see video: [141]). The action space $\mathcal{A}$ and learning hyperparameters are defined in Table 4.1. A unicycle model was used as the simplified model (4.4) with the nominal controller $\mathbf{k}_{\text{nom}}$:

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\cos\phi & 0\\ \sin\phi & 0\\ 0 & 1\end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \left(\underbrace{\begin{bmatrix}v\\ \omega\end{bmatrix}}_{\mathbf{v}} + \mathbf{d}(t) \right), \quad \mathbf{k}_{\text{nom}}(\mathbf{x}) = \begin{bmatrix} K_v d_{\text{g}} + C \\ -K_\omega(\sin\phi - (y_{\text{g}} - y)/d_g) \end{bmatrix}, \tag{4.16}$$

where (x, y) is the planar position of the robot, ϕ is the yaw angle, $(x_{\text{g}}, y_{\text{g}})$ is the goal position of the robot, $d_{\text{g}} = \|(x_{\text{g}} - x, y_{\text{g}} - y)\|$ is the distance to the goal, and K_v, K_ω, and C are positive constants. Obstacle avoidance is encoded via the 0-superlevel set of the function:

$$h(\mathbf{x}, \boldsymbol{\rho}_i) = d_{\text{obs},i} - r_{\text{obs}} - \zeta\cos(\phi - \theta_i), \tag{4.17}$$

where $\boldsymbol{\rho}_i = [x_{\text{obs},i}, y_{\text{obs},i}]$ is the location of the i^{th} obstacle, $d_{\text{obs},i} = \|(x_{\text{obs},i} - x, y_{\text{obs},i} - y)\|$ and $\theta_i = \arctan((y_{\text{obs},i} - y)/(x_{\text{obs},i} - x))$ are the distance and angle from the i^{th} obstacle, r_{obs} is the sum of the radii of the obstacle and robot, and $\zeta > 0$ determines the effect of the heading angle on safety. The controller used to drive the system is the TR-OP with the nominal controller $\mathbf{k}_{\text{nom}}$ from (4.16). In practice, infeasibilities

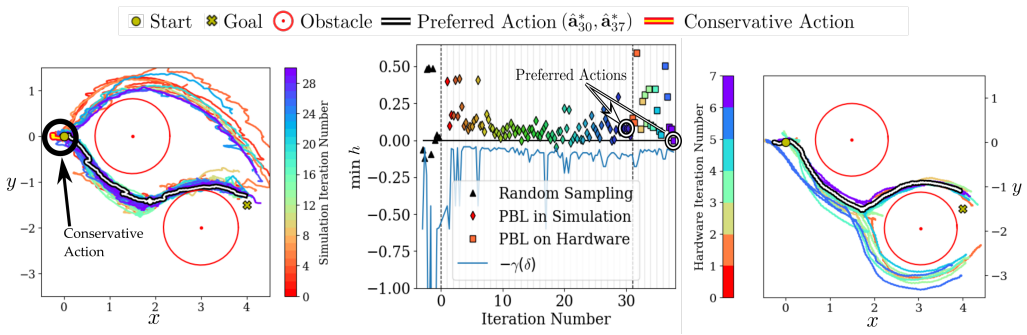


Figure 4.14: Illustration of the robotic behavior throughout the learning process. (Left) Actions sampled during simulation in 30 iterations with 3 new actions in each iteration. The preferred action, $\hat{\boldsymbol{a}}_{30} = (3, 0.6, 0.5, 0.015)$, is shown in black and white. A conservative action, $\boldsymbol{a} = (2, 0.5, 0.0651, 0.485)$, is indicated by the black circle, where a and b were determined by estimating the Lipschitz coefficients present in the proof of Theorem 2. The conservative action fails to progress whereas LineCoSpar provides an action which successfully navigates between obstacles. (Center) The minimum value of h that occurred in each iteration. Triangles, diamonds, and squares represent actions that are sampled randomly, by PBL in simulation and on hardware in an indoor setting, respectively. Colors correlate to iteration number. The lower bound $-\gamma(\delta)$ for the expanded set $\mathcal{C}_\delta$ with $\delta = 1$ is plotted. The preferred actions for simulation and hardware experiments are circled. (Right) Seven additional iterations of 3 actions executed indoors. The preferred action, $\hat{\boldsymbol{a}}^*_{37} = (4, 0.6, 0.4, 0)$, successfully traverses between the obstacles.

of this safety filter were considered unsafe and the inputs were saturated such that $v \in [-0.2, 0.3]$ m/s and $\omega \in [-0.4, 0.4]$ rad/s. The velocity command $\mathbf{v}$ is computed at 20 Hz and error introduced by this sampling scheme is captured by the tracking error $\mathbf{d}(t)$. Tracking of $\mathbf{v}$ is performed by an inverse dynamics quadratic program (ID-QP) walking controller designed using the concepts in [142], which realizes a stable walking gait for (4.14) at 1 kHz.

Simulation results

We simulated the quadruped executing the proposed controller with parameters provided by SA-LineCoSpar. The resulting trajectories and the position of the obstacles are shown in Figure 4.14. We ran 30 iterations, with 3 new actions sampled in each iteration ($s = 3$), and obtained user preferences and ordinal labels in between each set of actions. To simulate perception error, the measurements of the obstacles were shifted by -0.1 m in the y-direction. The parameters found with SA-LineCoSpar allow the robot to navigate between obstacles. For comparison, a conservative action is also shown, which is safe but fails to progress towards the goal.

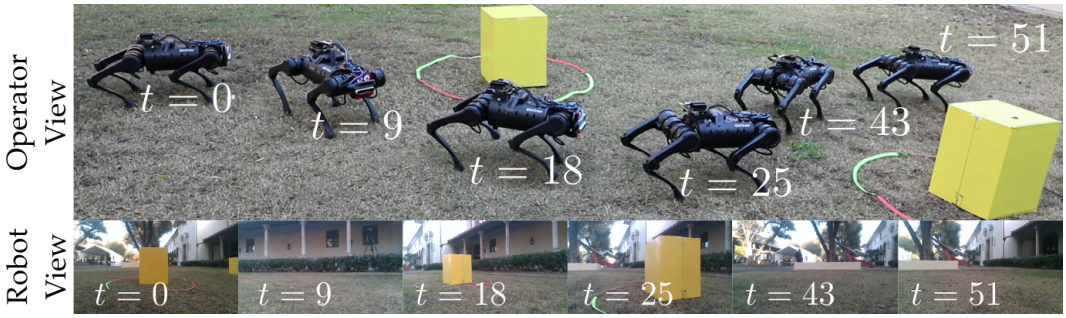


Figure 4.15: The preferred action, $\hat{\mathbf{a}}^*_{40} = (5, 0.1, 0.4, 0.02)$, after simulation, indoor experiments, and 3 additional iterations of 3 actions in an outdoor environment is shown alongside views from the onboard camera.

SA-LineCoSpar eliminates this conservatism with only minor safety violations and determines a parameter set which is both safe and performant.

Hardware results

After simulation, we continued learning on hardware experiments in a laboratory setting for 7 additional iterations until the user was satisfied with the experimental behavior. The robot and obstacle positions were estimated using Intel RealSense T265 and D415 cameras to perform SLAM and segmentation. Centroids of segmented clusters in the occupancy map were used as the measured obstacle positions $\widehat{\boldsymbol{\rho}}_i$. The true robot and obstacle positions were obtained for comparison using an OptiTrack motion capture system. The results of these experiments can be seen in Figure 4.14. Afterwards, three additional iterations were conducted outdoors on grass until again the user was satisfied with the experimental behavior. The resulting best trajectory can be seen in Figure 4.15. The preferred action was also tested on a variety of other obstacle arrangements to confirm its generalizability. The performance of the final preferred action for these obstacle configurations can be seen in the supplementary video [141].

4.4 Preferential Multi-Objective Bayesian Optimization

In the preceding sections, we examined preference-based learning (PBL) methods for single-objective settings, including ROIAL for preference characterization, LineCoSpar for preference optimization, and their integration into a unified framework for exoskeleton gait personalization. We also introduced a safety-aware extension of ROIAL combined with LineCoSpar for safety-critical control, enabling preference optimization while respecting operational constraints. While effective when user preferences can be represented by a single latent utility function, these approaches are limited in scenarios where multiple, often conflicting, objectives must be balanced. This motivates extending PBL to multi-objective settings.

Bayesian optimization (BO) provides a principled framework for optimizing expensive-to-evaluate objective functions and has been widely applied in domains where evaluation cost is high. A key variant, preferential Bayesian optimization (PBO), addresses cases where the objective is *latent*: rather than observing objective values directly, the algorithm infers structure from ordinal preference feedback provided by a decision-maker (DM).

While prior work in PBO has demonstrated success in various applications [143, 144, 22], existing methods operate under the assumption that preferences can be encoded by a single objective function. In practice, however, problems are often characterized by multiple conflicting objectives. This occurs, for instance, when multiple users with conflicting preferences collaborate in a joint design task, as illustrated in Figure 4.16, or when a user wishes to explore the trade-offs between multiple conflicting attributes before committing to a design.

To motivate the need for multi-objective PBO, we examine two illustrative applications. The first application involves an exoskeleton customization task that aims to enhance user comfort. In this situation, a user assisted by an exoskeleton experiences different gait designs and indicates the most comfortable option [82, 22]. In this and other robotic assistive personalization applications, users and clinical technicians often collaborate on a design task to maximize user comfort (the user's objective) while optimizing energy consumption and other metrics related to the exoskeleton's long-term functionality (the technician's objective) [61].

The second application is autonomous driving policy design, where a user is presented

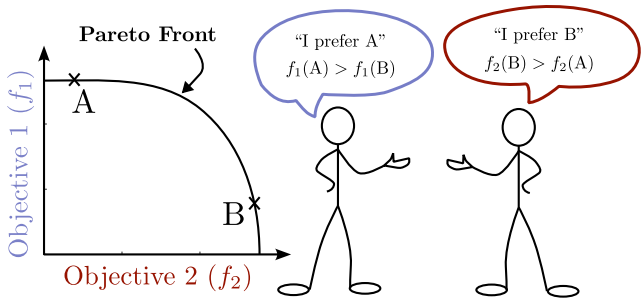


Figure 4.16: In this work, we extend preferential Bayesian optimization to the multi-objective setting. In contrast with existing approaches, our approach allows the decision-makers involved in the joint design task to efficiently explore optimal trade-offs between the conflicting objectives.

with multiple simulations of an autonomous vehicle under different driving policies, and the user indicates the one with better safety and performance attributes [93]. In such settings, policy-makers often seek to understand the trade-offs between multiple latent objectives, such as lane keeping and speed tracking, before committing to a specific policy [145].

Motivated by the applications described above, we propose a framework for multi-objective PBO. The specific contributions of this extension are:

- To the best of our knowledge, our work proposes the first framework for preferential Bayesian optimization with multiple objectives.
- We present *dueling scalarized Thompson sampling (DSTS)*, the first extension of dueling Thompson sampling (DTS) algorithms [98, 146, 147] to the multi-objective setting.
- We prove that DSTS is asymptotically consistent. Furthermore, we also provide the first convergence guarantee for DTS in single-objective PBO.
- We demonstrate our framework across six test problems, including simulated exoskeleton personalization and autonomous driving policy design tasks. Our results show that DSTS can efficiently explore the objectives' Pareto front using preference feedback.

Related work

To better situate the proposed multi-objective PBO framework, we first review related work in three areas: preference-based optimization, multi-objective optimization,

and additional relevant directions.

Preference-based optimization

Preference-based optimization has been actively studied across various frameworks, including multi-armed bandits [148, 99], reinforcement learning [149], and BO [143, 150, 151]. It has been successful in a broad range of applications, such as personalized medicine [144, 98, 82], robot control [93, 25, 152, 26] and, more recently, the alignment of large language models [153].

Most work in this area focuses on the single-objective setting. Two notable exceptions are the works of [145] and [154]. [145] considers one-shot preference-based optimization across multiple criteria over a finite design space. This study adopts a game-theoretic viewpoint and introduces the concept of a Blackwell winner, which implicitly requires the user to specify an acceptable trade-off between criteria, in contrast with our work. [154] considers multi-objective preference alignment of large language models. Like our work, these two works are motivated by the idea that preference-based optimization across multiple objectives is crucial for capturing richer human feedback.

Our work extends the dueling Thompson sampling algorithm for dueling bandits introduced by [98] (termed self-sparring), which has been adapted to preference-based reinforcement learning (termed dueling posterior sampling) [146] and PBO (termed batch Thompson sampling) [147]. To our knowledge, we provide the first multi-objective generalization of this algorithm.

Multi-objective optimization

The field of multi-objective optimization has been extensively studied, encompassing both theoretical advancements and applications across various engineering problems [155, 156, 157]. Literature within the BO framework is most closely related to our work [158, 159, 160, 161, 162].

Our algorithm draws inspiration from ParEGO [159], a multi-objective BO algorithm that employs augmented Chebyshev scalarizations to convert a multi-objective optimization problem into multiple single-objective problems. Unlike [159], our objectives are not observable, preventing direct modeling of scalarized values. Instead, we model each objective separately and scalarize samples drawn from these models, similar to [162]'s version of ParEGO.

Additionally, our work is related to research that incorporates user preferences into multi-objective optimization—a topic that has been actively studied both within and beyond the BO framework [163, 164, 165, 166]. In most of this prior work, all objectives are assumed to be directly observable, and user preferences are captured through a latent utility function that combines these objectives into a single score to guide optimization. In contrast, we do not assume access to the objective values. Instead, we receive binary preference feedback for each objective individually, without ever observing their actual values or requiring a predefined utility function to aggregate them.

Additional related work

Emerging from the operations research community, the field of multi-criteria decision analysis (MCDA) focuses on decision-making under multiple conflicting criteria [167, 168]. Although our work is related to this field, it diverges from the traditional MCDA approaches, which often involve aggregating preferences across criteria into a single performance measure [169, 170, 171, 145]. Such aggregation requires additional assumptions about the DM's desired trade-off. Additionally, methods in this field have been explored outside the PBO framework, making them not directly applicable in our setting.

Problem setting

Preferences Let $\mathcal{X}$ denote the space of designs. We assume there is a DM (which may represent one or multiple users collaborating on a design task) aiming to maximize their preferences over designs. We assume the DM's preferences can be encoded via m objective functions $f_1, \ldots, f_m : \mathcal{X} \to \mathbb{R}$ so that, for any given pair of designs $x, x' \in \mathcal{X}$, the DM prefers x over x' with respect to objective j if and only if $f_j(x) > f_j(x')$. For simplicity, we assume all m objectives are latent, but our approach can be easily adapted to settings where some objectives are observable, as discussed in Section 4.4.

Goal Let $f = [f_1, \ldots, f_m] : \mathcal{X} \to \mathbb{R}^m$ denote the concatenation of the m objective functions. The DM seeks to find designs that maximize each objective. This concept is formalized through the notion Pareto-dominance. For a pair of designs $x, x' \in \mathcal{X}$, x Pareto-dominates x', denoted by $x \succ_f x'$, if $f_j(x) \geq f_j(x')$ for $j = 1, \ldots, m$ with strict inequality for at least one index j. The DM seeks to find the Pareto-optimal set of

f, defined by $\mathcal{X}_f^* := \{x : \nexists\, x' \text{ such that } x' \succ_f x\}$. The set $\mathbb{Y}_f^* := \{f(x) : x \in \mathcal{X}_f^*\}$ is termed the Pareto front of f. Figure 4.17 depicts the Pareto front for one of our test problems; the light grey region is the set of feasible objective vectors, i.e., $\{f(x) : x \in \mathcal{X}\}$ and the dark grey curve indicates the Pareto front of f.

Feedback To assist the DM's goal, our algorithm collects preference feedback interactively (Algorithm 4). At each iteration, denoted by $n = 1, \ldots, N$, the algorithm selects a *query* constituted of q designs $X_n = (x_{n,1}, \ldots, x_{n,q}) \in \mathcal{X}^q$. The DM then indicates their most preferred design among these q designs for each objective. Let $r_{j,n} \in \{1, \ldots, q\}$ denote the DM's preferred design with respect to objective j. The collection of these responses is denoted by $r_n = [r_{1,n}, \ldots, r_{m,n}]$.

Algorithm 4 Dueling Scalarized Thompson Sampling

Input Initial dataset: $\mathcal{D}_0$, and prior on f: p_0.
for $n = 1, \ldots, N$ **do**
Compute p_n, the posterior on f given $\mathcal{D}_{n-1}$
Sample $\tilde{\theta}_n$ uniformly at random over Θ
Draw samples $\tilde{f}_{n,1}, \ldots \tilde{f}_{n,q} \overset{\text{iid}}{\sim} p_n$
Find $x_{n,i} \in \operatorname{argmax}_{x \in \mathbb{X}} s(\tilde{f}_{n,i}(x); \tilde{\theta}_n),\ i = 1, \ldots, q$
Set $X_n = (x_{n,1}, \ldots, x_{n,q})$, and observe r_n
Update dataset $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(X_n, r_n)\}$
end for

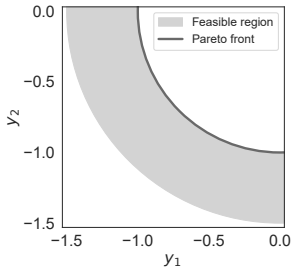


Figure 4.17: Feasible region and Pareto front of the DTLZ2 test function.

Dueling scalarized Thompson sampling

We introduce a novel algorithm termed *dueling scalarized Thompson sampling (DSTS)*, summarized in Algorithm 4. DSTS is obtained by adeptly combining ideas from preference-based and multi-objective optimization to derive a sound algorithm with strong performance and convergence guarantees. As is common in BO, our algorithm is comprised of a probabilistic model of the objective functions for predictions and uncertainty reasoning, along with a sampling policy that, informed by the probabilistic model, iteratively selects new queries, balancing exploration and exploitation.

Probabilistic model

The probabilistic model is encoded by a prior distribution over f, denoted by p_0. We assume p_0 consists of a set of independent Gaussian processes, each corresponding to an objective. However, our framework does not rely on this choice and can easily accommodate other priors as long as samples from the posterior distribution can be drawn.

As is standard in the PBO literature [150, 172, 151], we account for noise in the DM's responses by using a Logistic likelihood for each objective $j = 1, \ldots, m$ of the following form:

$$\mathbf{P}\left(r_{j,n} = i \mid f_j(X_n)\right) = \frac{\exp(f_j(x_{n,i})/\lambda_j)}{\sum_{i'=1}^{q} \exp(f_j(x_{n,i'})/\lambda_j)},\ i = 1, \ldots, q, \tag{4.18}$$

where $\lambda_j > 0$ is the noise-level parameter. We estimate λ_j along with the other hyperparameters via maximum likelihood. We assume noise is independent across objectives and interactions.

Let $\mathcal{D}_0$ denote the initial dataset and $\mathcal{D}_{n-1} = \mathcal{D}_0 \cup \{(X_k, r_k)\}_{k=1}^{n-1}$ denote the data available right before the n-th interaction with the DM. Let p_n denote he posterior over f given $\mathcal{D}_{n-1}$. The posterior cannot be computed in closed form but can be approximated using, e.g., a variational inducing point approach [172]. For observable objectives, the above model can be replaced by a standard Gaussian process model with a Gaussian likelihood (see Appendix in [173]).

Sampling policy

Our primary algorithmic contribution is our sampling policy, which extends the dueling Thompson sampling (DTS) algorithmic family to the multi-objective setting. This is achieved by leveraging augmented Chebyshev scalarizations, a technique from multi-objective optimization used to decompose a multi-objective optimization problem into multiple single-objective problems. We next explain augmented Chebyshev scalarizations and describe how we integrate them with DTS.

Augmented Chebyshev scalarizations Augmented Chebyshev scalarizations are widely used for multi-objective optimization [155]. In BO, in particular, they were employed by [159] and [161]. We also leverage them to derive a sound sampling policy in our setting.

For a given vector of scalarization parameters, $\theta \in \Theta := \{\theta \in \mathbb{R}^m : \sum_{j=1}^m \theta_j = 1 \text{ and } \theta_j \geq 0,\ j = 1, \ldots, m\}$, the augmented Chebyshev scalarization function is defined by

$$s(y; \theta) = \min_{j=1,\ldots,m} \{\theta_j y_j\} + \rho \sum_{j=1}^{m} \theta_j y_j, \tag{4.19}$$

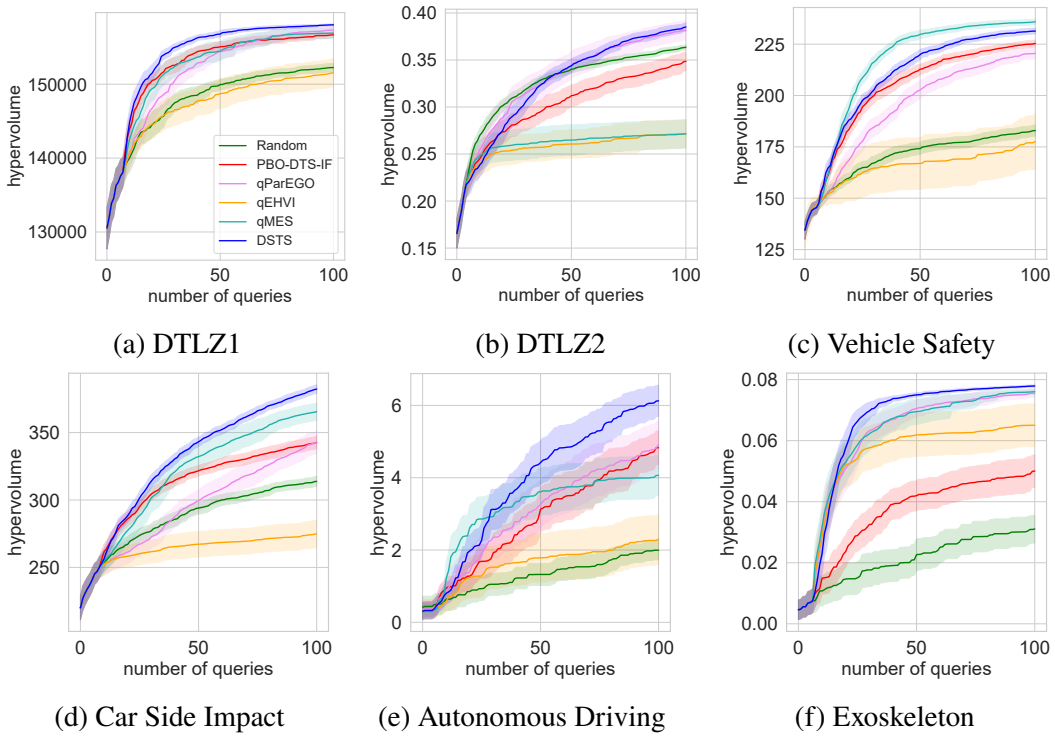
where ρ is a small positive constant. It can be shown that any solution of $\max_{x \in \mathcal{X}} s\left(f(x); \theta\right)$ lies in the Pareto-optimal set of f. Conversely, if ρ is small enough, every point in the Pareto-optimal set of f is a solution of $\max_{x \in \mathcal{X}} s\left(f(x); \theta\right)$ for some $\theta \in \Theta$ (Theorem 3.4.6, [155]).

Dueling scalarized Thompson sampling At each iteration, n, we draw a sample from the scalarization parameters uniformly at random over Θ, denoted by $\tilde{\theta}_n$. We also draw q independent samples, denoted by $\tilde{f}_{n,1}, \ldots, \tilde{f}_{n,q}$, from the posterior distribution on f given $\mathcal{D}_{n-1}$. The next query is then given by $X_n = (x_{n,1}, \ldots, x_{n,q})$, where

$$x_{n,i} \in \operatorname*{argmax}_{x \in \mathcal{X}} s\left(\tilde{f}_{n,i}(x); \tilde{\theta}_n\right),\ i = 1, \ldots, q. \tag{4.20}$$

Intuitively, our sampling policy operates by first determining a subset of the Pareto-optimal set of f using $\tilde{\theta}_n$, denoted as $\mathcal{X}^*_{f;\tilde{\theta}_n} = \operatorname{argmax}_{x \in \mathcal{X}} s(f(x); \tilde{\theta}_n)$. Then, each $x_{n,i}$ is sampled according to the probability (induced by the posterior on f) that $x_{n,i} \in \mathcal{X}^*_{f;\tilde{\theta}_n}$, analogous to single-objective dueling posterior sampling [98]. The DM's responses provide information of the highest value point among $x_{n,1}, \ldots, x_{n,q}$ for each objective, which in turn allows us to learn about $\mathcal{X}^*_{f;\tilde{\theta}_n}$. Since $\tilde{\theta}_n$ is drawn independently at each iteration, we explore a diverse collection of subsets $\mathcal{X}^*_{f;\tilde{\theta}_n}$ within $\mathcal{X}^*_f$.

We note that our sampling policy is agnostic to the choice of the probabilistic model, provided that samples from the posterior can be drawn. In addition, our sampling policy is suitable for problems with mixed latent and observable objectives thanks to its dual interpretation as a policy for preference-based optimization [98] and traditional optimization with observable objectives [161]. Specifically, when all objectives are observable, our sampling policy can be interpreted as a batch generalization [174] of the scalarized Thompson sampling algorithm proposed by [161].



(a) DTLZ1 (b) DTLZ2 (c) Vehicle Safety

(d) Car Side Impact (e) Autonomous Driving (f) Exoskeleton

Figure 4.18: Our framework was demonstrated on six test problems: DTLZ1 (a), DTLZ2 (b), Vehicle Safety (c), Car Side Impact (d), Autonomous Driving (e), and Exoskeleton (f). Overall, our proposed method (DSTS) delivers the best performance. qMES and qParEGO exhibit a mixed performance, achieving good results in some test problems and poor results in others. The remaining methods, Random, PBO-DTS-IF, and qEHVI, consistently underperform DSTS.

Theoretical analysis

We now study the convergence properties of DSTS. We begin by analyzing the single-objective setting and establish the asymptotic consistency of DTS. To our knowledge, this is the first such result for DTS in PBO. The result is stated in Theorem 3, with a proof—based on a martingale argument—provided in Appendix of [173].

Theorem 3. *Suppose that $\mathcal{X}$ is finite, $m = 1$, and the sequence of queries $\{X_n\}_{n=1}^{\infty}$ is chosen according to the DTS policy. Then, for each $x \in \mathcal{X}$, $\lim_{n\to\infty} \mathbf{P}_n(x \in \operatorname{argmax}_{x' \in \mathcal{X}} f(x')) = \mathbf{1}\{x \in \operatorname{argmax}_{x' \in \mathcal{X}} f(x')\}$ almost surely for f drawn from the prior.*

Extending this result to the multi-objective setting requires a minor modification to the DSTS algorithm. Specifically, our proof introduces a small probability of comparing against a fixed reference design in each iteration. This modification is

required by our proof due to the non-linear nature of Chebyshev scalarizations and is not required in the single-objective case. The resulting convergence guarantee is stated in Theorem 4. The proof—which can be found in Appendix of [173] —again relies on a martingale argument and the fact that varying θ allows Chebyshev scalarizations to recover all Pareto-optimal points.

Before stating the result, we describe the modified DSTS policy under consideration. Assume $q = 2$, and fix any reference point $x_{\text{ref}} \in \mathcal{X}$ and $\delta \in (0, 1)$. At each iteration, the first design $x_{n,1}$ is selected as in Equation 4.20, while the second design $x_{n,2}$ is set to x_{ref} with probability δ, or otherwise selected via Equation 4.20.

Theorem 4. *Suppose that $\mathcal{X}$ is finite, $q = 2$, and the sequence of queries $\{X_n\}_{n=1}^{\infty}$ is chosen according to the modified DSTS policy described above. Then, for each $x \in \mathcal{X}$, $\lim_{n\to\infty} \mathbf{P}_n(x \in \mathcal{X}_f^*) = \mathbf{1}\{x \in \mathcal{X}_f^*\}$ almost surely for f drawn from the prior.*

We now place our results in context with prior theoretical work on DTS. [98] showed that DTS achieves asymptotic consistency and sublinear regret in the dueling bandits setting, assuming independent pairs of arms. However, their analysis does not extend to our setting, where arms may be correlated. Notably, the analysis of [98] relies on showing that all arms are chosen infinitely often, which may not be true in our context. Similarly, [146] showed analogous convergence results in a reinforcement learning setting under a Bayesian linear reward model. In contrast, our result holds for non-linear objectives. Moreover, the result of [146] also relies on showing that each arm is selected infinitely often. Finally, note that the results of [98] and [146] are only applicable in the single-objective setting; as discussed above, the multi-objective setting presents additional challenges.

Unlike prior work, we do not establish regret bounds for DSTS. Indeed, such bounds remain an open question even for DTS in single-objective PBO. While we see this as a valuable research direction, such analysis is beyond the scope of our work which primarily aims to introduce multi-objective PBO. Finally, it is important to recognize that the asymptotic consistency of data-driven algorithms like DSTS cannot be taken for granted. For instance, [151] showed that the adaptation of qEI proposed by [147] is not asymptotically consistent and can perform poorly in single-objective PBO, despite being one of the most widely used algorithms. In Theorem 5, we show that qEHVI [162], a multi-objective generalization of qEI, suffers from the same

limitation in our setting. A proof is provided in Appendix of [173]. Our empirical results support this finding, showing that qEHVI can perform very poorly.

Theorem 5. *There exists a problem instance with finite $\mathcal{X}$ and $q = 2$ such that if $X_n \in \operatorname{argmax}_{X \in \mathcal{X}^q} \mathrm{qEHVI}_n(X)$ for all n, then $\lim_{n\to\infty} \mathbf{P}_n(x \in \mathcal{X}_f^*) = t$ almost surely for some fixed $x \in \mathcal{X}$ and $t \in (0, 1)$.*

Numerical experiments

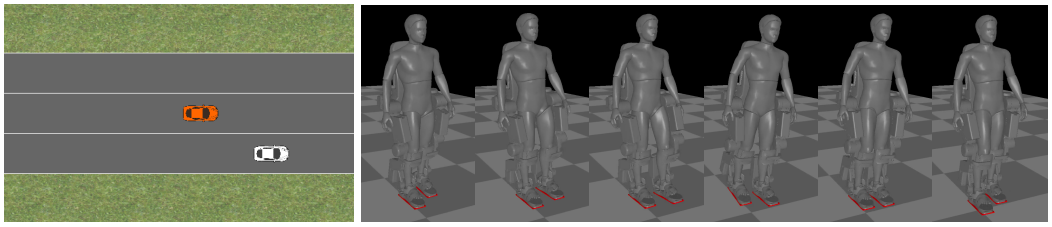
We evaluate our algorithm across six test problems and compare it with five other sampling policies. All algorithms are implemented using BoTorch [175]. Details on the performance metric, the benchmark sampling policies, and the test problems are provided below. The code for reproducing our experiments can be found at `https://github.com/RaulAstudillo06/PMBO`.

Performance metric

We quantify performance using the hypervolume indicator, which has been shown to result in good coverage of Pareto fronts when maximized [176]. Let $\widehat{\mathbb{Y}}^* = \{y_\ell\}_{\ell=1}^L$ be a finite approximation of the Pareto front of f. Its hypervolume is given by $\mathrm{HV}(\widehat{\mathbb{Y}}^*, r) = \mu\left(\bigcup_{\ell=1}^L [r, y_\ell]\right)$, where $r \in \mathbb{R}^m$ is a reference vector, μ denotes the Lebesgue measure over $\mathbb{R}^m$, and $[r, y_\ell]$ denotes the hyper-rectangle bounded by the vertices r and y_ℓ. We report performance by setting $\widehat{\mathbb{Y}}^*$ equal to the set of Pareto optimal points across designs shown to the DM.

Benchmarks

We compare our algorithm (DSTS) against uniform random sampling (Random), three adapted algorithms from standard multi-objective BO (qParEgo, qEHVI, qMES), and a standard PBO algorithm with inconsistent overall preference feedback (PBO-DTS-IF). Our experiments in this section use the regular version of DSTS. In Appendix in [173] we show that the modified version of DSTS used in Theorem 4 achieves virtually the same performance for small values of δ. All algorithms use the same priors, and the resulting posteriors are approximated via the variational inducing point approach proposed by [172]. Approximate samples from the posterior distribution used by DSTS and PBO-DTS-IF are obtained via 1000 random Fourier features [177].



(a) Autonomous Driving (b) Exoskeleton Walking

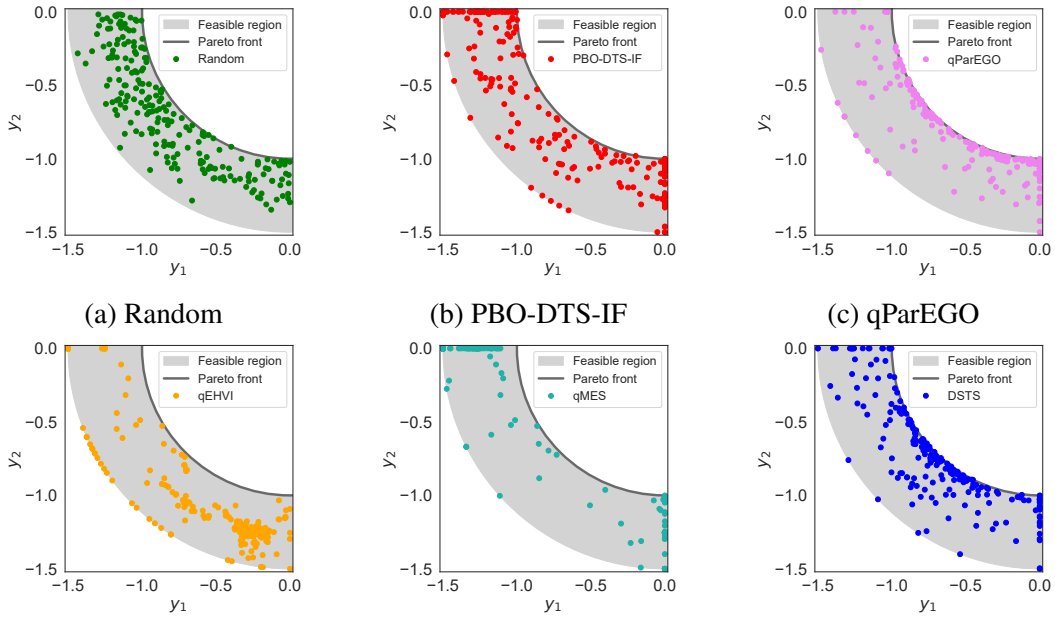
Figure 4.19: Simulation environments used in our test problems.

Adapted standard multi-objective BO methods A common approach in the PBO literature is to use a *batch* acquisition function designed for parallel BO with observable objectives, ignoring the fact that preference feedback is observed rather than objective values [147, 178]. Despite lacking the principled interpretations they enjoy in their original setting, they often deliver strong empirical performance. Following this principle, we adopt three batch acquisition functions from standard multi-objective BO as benchmarks: qParEGO [159, 162], qEHVI [162], and qMES [160]. Since these algorithms were not originally designed for latent objectives, they require minor adaptations that we describe in Appendix in [173]. These algorithms use the same probabilistic model as DSTS. Thus, any difference in performance is solely due to the use of different sampling policies.

Single-objective PBO with inconsistent aggregated preference feedback Single-objective PBO methods are often applied to problems characterized by multiple conflicting objectives. In such cases, DMs are expected to aggregate their preferences across objectives, which can be challenging for DMs and often results in inconsistent feedback. For example, in the context of exoskeleton personalization, this would require forcing the exoskeleton user and clinical technician to reach a unified response at every iteration, which can be challenging if the user's objective is to maximize comfort while the technician's objective is to ensure the exoskeleton's long-term energy efficiency. To understand the effect of using this approach, we include a standard single-objective PBO approach using inconsistent feedback. Additional details on this benchmark are provided in Appendix of [173].

Test problems

We report performance across four synthetic test problems (DTLZ1, DTLZ2, Vehicle Safety, and Car Side Impact), a simulated autonomous driving policy design task (Autonomous Driving), and a simulated exoskeleton gait design task (Exoskeleton)



(a) Random (b) PBO-DTS-IF (c) qParEGO

(d) qEHVI (e) qMES (f) DSTS

Figure 4.20: Illustration of sampled designs for the DTLZ2 test function. These figures show that our proposed method (DSTS) provides a better exploration of the Pareto front than its competitors.

using queries with $q = 2$ and $q = 4$ designs. Details of these test problems are provided below. In all problems, an initial dataset is obtained using $2(d + 1)$ queries chosen uniformly at random over $\mathcal{X}^q$, where d is the input dimension of the problem. After this initial stage, each algorithm was used to select 100 additional queries sequentially. Results for $q = 2$ are shown in Figure 4.18. Each plot shows the mean of the hypervolume of the designs included in queries thus far, plus and minus 1.96 times the standard error. Each experiment was replicated 30 times using different initial datasets. In all problems, the DM's responses are corrupted by moderate levels of Gumbel noise, which is consistent with the use of a Logistic likelihood (see Appendix in [173] for the details). Results for $q = 4$ can be found in Appendix as well.

DTLZ1 and DTLZ2 The *DTLZ1* and *DTLZ2* functions are standard test problems from the multi-objective optimization literature [179]. In our experiments, we configure DTLZ1 with $d = 6$ design variables and $m = 2$ objectives, and DTLZ2 with $d = 3$ design variables and $m = 2$ objectives. Results for these problems are shown in Figures 4.18(a) and 4.18(b), respectively. Our approach achieves the best performance in both problems, tied with qParEGO on DTLZ2.

Surprisingly, on the DTLZ2 problem, PBO-DTS-IF, qEHVI, and qMES underperform significantly, even being surpassed by Random. To understand this, we plot a representative set of objective vectors corresponding to the queried designs in Figure 4.20. As illustrated, Random offers a reasonable exploration of the Pareto front (likely due to the low dimensionality of DTLZ2). However, it exposes the user to many low-quality designs, which can potentially frustrate DMs. PBO-DTS-IF and qMES tend to favor designs where one of the objectives achieves its maximum possible value, which may be problematic for DMs seeking more balanced solutions. qEHVI fails to explore the Pareto front, concentrating its queries on a limited sub-optimal region instead. Finally, DSTS and qParEGO provide a more comprehensive exploration of the Pareto front.

Vehicle Safety and Car Side Impact The *Vehicle Safety* and *Car Side Impact* test functions are designed to emulate various metrics of interest in the context of crashworthiness vehicle design. Overall, these test problems emulate an expert's assessment based on expensive experiments where cars are intentionally crashed, and safety metrics are evaluated. Vehicle Safety has $d = 5$ design variables and $m = 3$ objectives. Car Side Impact has $d = 7$ design variables and $m = 4$ objectives. For further details, we refer the reader to [180]. Results for the Vehicle Safety and Car Side Impact experiments can be found in Figures 4.18(c) and 4.18(d), respectively. For the Vehicle Safety problem, qMES is the best-performing algorithm, followed by DSTS. For the Car Side Impact, DSTS performs the best, followed closely by qMES.

Autonomous Driving Policy Design To supplement the synthetic test functions, we further evaluate our algorithm on a simulated autonomous driving policy design task. For this problem, we use a modification of the Driver environment presented in [93]. A similar environment was also used by [145], providing empirical evidence that user preferences in this context are inherently governed by multiple latent objectives. In our modified environment, illustrated in Figure 4.19(a), an autonomous control policy is created to drive a trailing (red) vehicle forward to a goal location while maintaining some minimum distance with a leading (white) vehicle. The control policy switches between two modes, collision avoidance and goal-following, based on a minimum distance threshold. The behavior of the leading car is fixed by setting a pre-specified set of actions.

Using this simulation environment, we consider four objectives representing approximations of subjective notions of safety and performance: lane keeping, speed

tracking, heading angle, and collision avoidance. The design space is parameterized by four control variables: two parameters that account for how fast the vehicle approaches the goal or the other vehicle, respectively, one position gain that accounts for the adjustment on the desired heading, and the minimum distance threshold used to switch between the two modes. The results of this experiment are shown in Figure 4.18(e). As illustrated, our approach again delivers better performance than its competitors.

Exoskeleton Gait Customization Lastly, we evaluate our algorithm on an exoskeleton gait personalization task using a high-fidelity simulator of the lower-body exoskeleton Atalante [61], illustrated in Figure 4.19(b). This problem emulates the scenario discussed in the introduction, in which there are two conflicting objectives: subjective user comfort and energy efficiency. For simulation purposes, we approximate comfort as a linear combination of three attributes: average walking speed (faster speed is preferred), maximum pelvis acceleration (lower peak acceleration is preferred), and the center of mass tracking error (lower error is preferred). We approximate total energy consumption as the l^2-norm of joint-level torques, averaged over the simulation duration. We note that this is an observable objective. Thus, our approach is modified as discussed in Section 4.4 and further elaborated on Appendix in [173] to leverage direct observations of this objective.

The design space is parameterized by five gait features: step length, minimum center of mass position with respect to stance foot in sagittal and coronal plane, minimum foot clearance, and the percentage of the gait cycle at which minimum foot clearance is enforced. Each unique set of features corresponds to a unique gait. These gaits are synthesized using the FROST toolbox [41] and are simulated in Mujoco to obtain the corresponding objectives. Since simulations are time-consuming, we build surrogate objectives by fitting a (regular) Gaussian process to the objectives obtained from 1000 simulations, with each set of gait features drawn uniformly over the design space. As shown in Figure 4.18(f), DSTS achieves the best performance, followed closely by qParEGO and qMES.

Discussion

Across the broad range of problems considered, DSTS delivers the best overall performance. Specifically, DSTS yields the highest hypervolume in nearly all problems (except for the Vehicle Safety problem, where it is second to qMES).

Two of the standard multi-objective benchmarks, qParEGO and qMES, exhibit mixed results, highlighting the importance of developing algorithms designed to handle preference feedback as opposed to naively adapting algorithms intended for observable objectives. Notably, qEHVI is the worst-performing algorithm, even surpassed by Random. This is consistent with Theorem 5, which shows that qEHVI is not consistent in general, thus highlighting the value of our asymptotic consistency result for DSTS (Theorem 4). Lastly, PBO-DTS-IF consistently underperforms DSTS, confirming that a single-objective PBO approach is insufficient to explore the optimal trade-offs in problems with multiple conflicting objectives. The runtimes of all methods are discussed in Appendix in [173].

In summary, we proposed a framework for PBO with multiple latent objectives, where the goal is to help DMs efficiently explore the objectives' Pareto front guided by preference feedback. Within this framework, we introduced dueling scalarized Thompson sampling (DSTS), which, to our knowledge, is the first approach for PBO with multiple objectives. Our experiments demonstrate that DSTS provides significantly better exploration of the Pareto front than several benchmarks across six test problems, including simulated autonomous driving policy design and exoskeleton gait customization tasks. Moreover, we showed that DSTS is asymptotically consistent, providing the first convergence result for dueling Thompson sampling in PBO.

While our work provides a sound approach to tackling important applications not covered by existing methods, there are also a few limitations that suggest avenues for future exploration. Future work could include a deeper theoretical analysis of DSTS, such as investigating convergence rates and regret bounds, as well as the development of alternative sampling policies. For example, [151] provided an efficient approach to approximate a one-step lookahead Bayes optimal policy in single-objective PBO, demonstrating superior performance against various established benchmarks. Although their approach cannot be easily adapted to our context, exploring alternative mechanisms for computing non-myopic sampling policies in our setting would be valuable. Finally, it would be interesting to explore DSTS in other settings, such as preference-based reinforcement learning.

4.5 Summary

This chapter addressed the challenge of optimizing and understanding user preferences in exoskeleton gait generation, with the ultimate goal of improving user comfort and personalization. We first introduced ROIAL, a region of interest active learning framework for efficient preference characterization while avoiding gaits that make users feel unsafe or uncomfortable.

We then presented a unified preference-based learning framework that combines ROIAL with the existing CoSpar and LineCoSpar algorithms to support both preference optimization and characterization within a single pipeline. This unified framework was evaluated on the Atalante exoskeleton with paraplegic participants, incorporating both preference, ordinal and coactive feedback, and applied to tuning forward walking parameters and a turning controller.

Building on ROIAL, we proposed a safety-aware preference-based optimization extension designed for safety-critical control tasks, enabling regret-minimizing action selection while respecting region-of-interest constraints. Finally, we introduced a preferential multi-objective Bayesian optimization method to address settings where user preferences span multiple, potentially competing, objectives.

Overall, the methods developed in this chapter demonstrate how preference-based learning can be adapted and extended to safety- and comfort-critical human-in-the-loop robotics applications, while maintaining sample efficiency and accommodating various user feedback.

Chapter 5

ROBUST BIPEDAL LOCOMOTION

While preference alignment is essential for user adoption, robustness is equally critical for safe and reliable deployment. The next chapter examines strategies to achieve robust walking under uncertainty.

Achieving robust bipedal locomotion in real world environments remains a central challenge for both assistive devices and humanoid robots. Even when nominal walking gaits are dynamically stable in simulation, deployment introduces numerous sources of uncertainty: modeling errors, unmodeled human–robot interaction effects, terrain variability, and unexpected disturbances. Without mechanisms to handle these uncertainties, performance can degrade significantly, potentially compromising user safety and comfort in the case of wearable robots.

Robustness is not achieved through a single mechanism, but rather through strategies that act at different stages of the locomotion control pipeline. At the design stage, offline analysis can be used to evaluate and select gaits or controller parameters with strong stability margins and tolerance to modeling errors before deployment. During execution, online adaptation strategies can respond to disturbances or environmental changes by modifying the gait in real time to preserve stability. Finally, learning based generalization methods can produce control policies that inherently handle a wide range of conditions by training with diverse scenarios and disturbances in simulation.

This chapter presents three strategies that address robustness at these different stages. Each can be applied independently, but they also lend themselves to integration in specific contexts:

1. **Offline robustness metrics** — Evaluation of reference trajectory prior to deployment using a hybrid forward invariance approach. This method analyzes the full hybrid system by identifying forward invariant sets centered around the fixed point of the Poincaré return map. Larger forward invariant sets correspond to larger regions of attraction, providing a principled measure of disturbance tolerance (Section 5.1).

2. **Online adaptation** — Adjustment of gait execution in real time through data driven predictive control (DDPC) and hybrid DDPC (HDDPC), which use collected data to construct more accurate reduced order models and enable fast replanning of full order motions (Section 5.2).

3. **Domain randomization in reinforcement learning** — Training of control policies in simulation with randomized dynamics, terrain profiles, and disturbances to promote generalization to unseen conditions. These policies are guided by reference trajectories and reward shaping informed by Control Lyapunov Functions, enabling transfer to hardware with improved stability and robustness (Section 5.4).

These approaches span a spectrum from verification to adaptation to generalization, with distinct trade offs in performance guarantees, adaptability, and computational cost. Together, they form a toolkit for addressing robustness in general locomotion tasks and can be combined with one another or with user alignment strategies introduced in earlier chapters.

5.1 Robust Walking via Hybrid Forward Invariance

Bipedal locomotion has received increasing attention in recent years due to the growing potential of humanoid robots. This interest has led to numerous demonstrations of experimentally stable walking [181, 182, 183, 184, 1, 185, 186, 187, 188]. However, beyond achieving nominal stability in simulation or controlled laboratory settings, it remains challenging to systematically verify that walking behaviors are robust to the types of disturbances encountered in real environments, such as external pushes, terrain variability, and model uncertainty.

In general, locomotive robustness can be improved in two complementary ways. The first is through online planning, performed either on the full-order dynamics or on reduced-order models [189, 190]. Such planners can reject large disturbances by generating corrective motions in real time to drive the state toward a desired terminal condition. Since these methods rely on prespecified terminal constraints, the robustness of the chosen references is a critical factor in overall performance.

This motivates the second approach: improving the inherent robustness of the nominal reference trajectories themselves. More robust references reduce the control effort required for disturbance rejection, easing the demands on online planners.

Prior work has addressed this problem by using nonsmooth analysis [191] to produce limit cycles that are less sensitive to impact uncertainty [192, 193], and by optimizing for reduced sensitivity to early or late foot impacts [194]. While these methods can produce robust gaits in practice, they do not provide formal guarantees on the size of allowable disturbances.

Several tools exist for quantifying robustness, including computing regions of attraction [195] and verifying input-to-state stability [130, 196]. These techniques provide certificates of robustness, but their high computational cost often limits their practicality for gait synthesis.

In this section, we adopt a step-to-step perspective on locomotion dynamics [197, 34], representing the hybrid gait as a discrete-time Poincar'e return map. Robustness is assessed by identifying forward invariant sets around the fixed point of this map—larger invariant sets correspond directly to larger regions of attraction (Fig. 5.1. To approximate these sets, we employ discrete-time barrier functions combined with a sampling-based search [198, 199, 200].

The resulting robustness metric is general-purpose: 1) It directly quantifies disturbance tolerance for a nominal gait. 2) It remains computationally tractable for high-dimensional systems via reduced-order representations. 3) It is agnostic to the choice of control framework.

We further integrate this metric into a simulation-in-the-loop optimization process to synthesize nominal trajectories with enhanced robustness. By isolating the effect of the robustness metric, without conflating it with online replanning, we highlight the central role of reference trajectory design in overall walking stability.

The method is validated on both flat-foot and multi-contact (foot-rolling) gaits for the Atalante lower-body exoskeleton, a device aimed at restoring mobility to individuals with motor-complete paraplegia. Compared to a baseline heuristic stability optimization, our approach yields nominal gaits with substantially improved disturbance tolerance, while preserving computational efficiency suitable for practical deployment.

Preliminaries on Locomotion

In the background chapter, hybrid control systems were introduced using a single-domain symmetric locomotion example. While some earlier contributions already

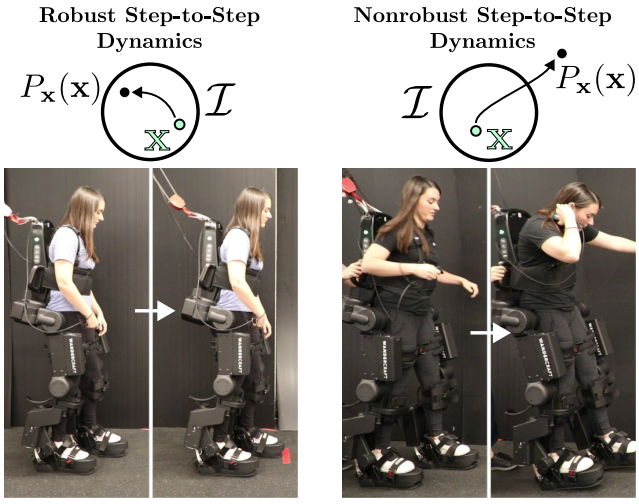


Figure 5.1: The framework developed in this paper optimizes locomotive robustness using forward invariance, certified via discrete-time barrier functions, as a metric for robustness.

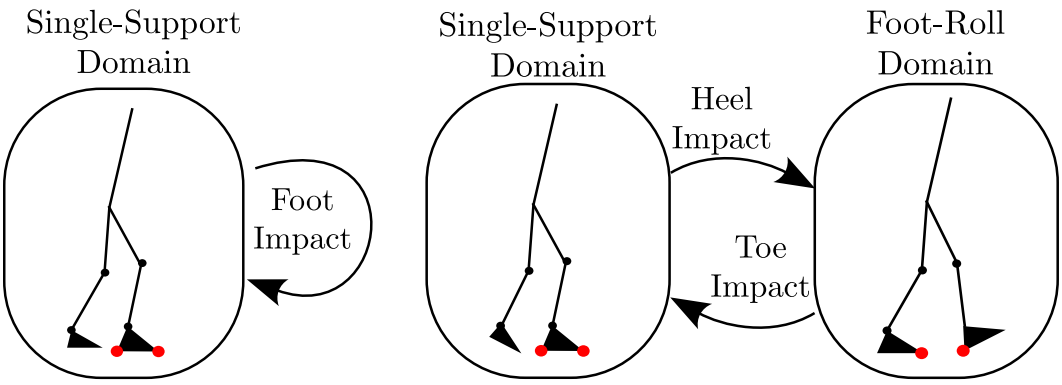
employed multi-domain models, here we further extend the structure by adding a foot-rolling domain to capture multi-contact walking, as shown in Fig. 5.2.

For fully actuated systems, such as powered exoskeletons, HZD extends to *Partial* HZD (PHZD) [201], in which only a subset of the degrees of freedom are directly actuated and the remaining evolve on a reduced-order zero dynamics manifold. In both cases, gait synthesis is formulated as a trajectory optimization problem enforcing physical feasibility, impact invariance, and task-specific constraints (e.g., step length, clearance), resulting in a nominal *limit cycle* $\mathcal{O}$ for the closed-loop hybrid system.

While stability of the periodic orbit guarantees convergence in the absence of disturbances, it does not directly characterize performance under real-world uncertainties, such as impact timing variations or external perturbations. Robustness notions such as input-to-state stability (ISS) and input-to-state safety (ISSf) provide a more complete picture, but verifying these properties for high-dimensional locomotion models remains computationally demanding.

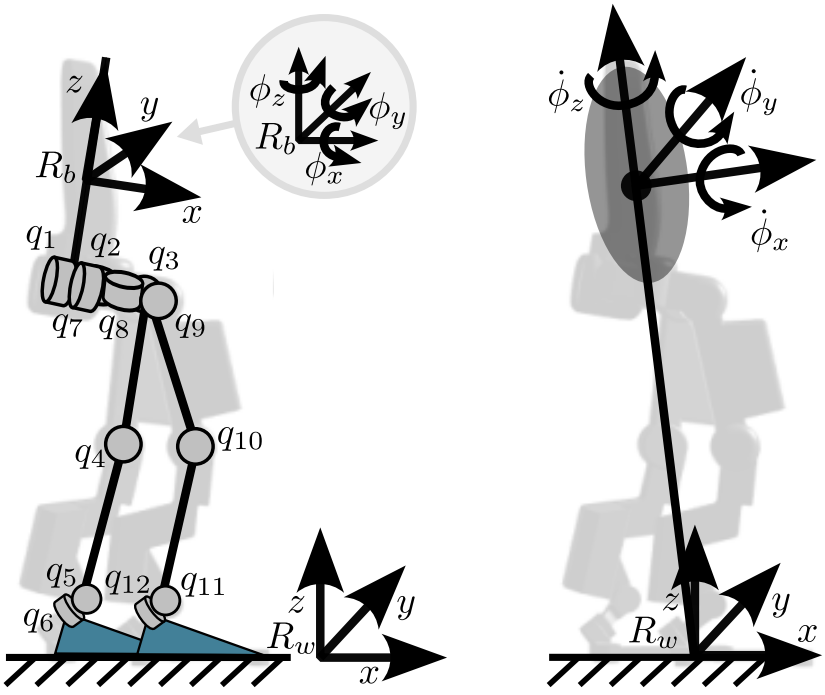
Reduced System Representation

While the following approach could be conducted for the full system state $x \in \mathcal{X}$, we propose the use of a lower-dimensional representation since certain coordinates often have a disproportional influence on overall system stability. For example, for bipedal robots, perturbations on actuated joints will influence the overall system much less than perturbations on the global coordinates.



(a) Flat-foot behavior (b) Multi-contact behavior

Figure 5.2: Directed graphs describing the hybrid system domain structure for the a) flat-foot and b) multi-contact walking.



(a) Full system (b) Reduced system

Figure 5.3: Model representations. a) The full system model is denoted by the generalized coordinates $x = (q_e^\top, \dot{q}_e^\top)^\top$ with $q_e := (p_b^\top, \phi_b^\top, q^\top)^\top \in \mathbb{R}^3 \times SO(3) \times \mathcal{Q}$. Here $p_b \in \mathbb{R}^3$ and ϕ_b respectively denote the euclidean position and orientation of the global base frame R_b relative to the world frame R_w. b) Here, the reduced-order representation of the model is illustrated, defined as the angular velocities of the global frame relative to the world frame, i.e., $\boldsymbol{x} := (\dot{\phi}_x, \dot{\phi}_y, \dot{\phi}_z)^\top$.

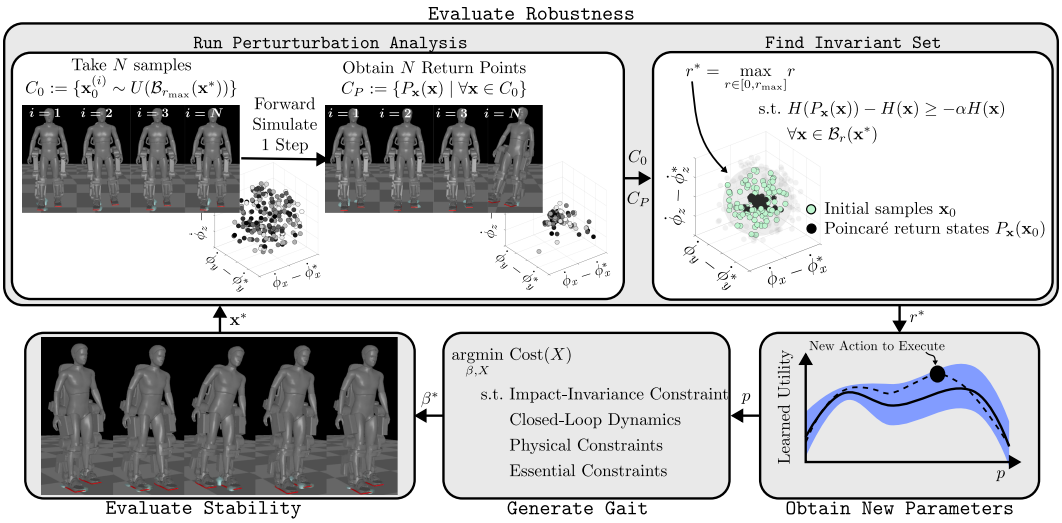


Figure 5.4: Diagram of sim-in-the-loop approach towards optimizing robustness.

This viewpoint is similar to that of reduced-order models, whereby the essence of the full-order dynamics can be captured by a few key states.

In general, the reduced state representation is denoted as $\boldsymbol{x} = \Phi(x) \in \mathbf{X}$, where $\mathbf{X}$ is a *reduced-order manifold*, i.e., $\dim(\mathbf{X}) \leq \dim(\mathcal{D})$ consisting of the lower-dimension representation of interest.

Assume that there exists a projection between our full system state and lower-dimension representation $\boldsymbol{x} = \Phi(x)$. For the Atalante lower-body exoskeleton, we restrict our attention to three specific global coordinates, denoted as the reduced system $\boldsymbol{x} := (\dot{\phi}_x, \dot{\phi}_y, \dot{\phi}_z)^\top \in \mathbb{R}^3$, representing the global angular velocity. This mapping is simply the labeling matrix $\boldsymbol{x} = \begin{bmatrix} \mathbf{0}_{3\times 21} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 12} \end{bmatrix} x$. The motivation for selecting these coordinates is the observation that exoskeleton users often add perturbations to the system that can be captured by instantaneous angular velocity changes. Note that in other applications, this reduced-order manifold could be the position of the center of the pressure in ZMP walking [32], centroidal dynamics [202, 203], or zero dynamics [204]. We can represent the discrete-time system associated with this reduced model as the Poincaré map restricted to $\mathbf{X}$:

$$\boldsymbol{x}_{k+1} = P_{\mathbf{X}}(\boldsymbol{x}_k) \tag{5.1}$$

with $P_{\mathbf{X}} := \Phi(P(\iota(\boldsymbol{x}_k)))$ defined as the Poincaré return map for the reduced system for states on the "reduced order" guard $\mathbf{X} \cap \mathcal{S}$. Here $\iota : \mathbf{X} \cap \mathcal{S} \to \mathcal{S}$ is a specific (non-unique) reconstruction of the full state, e.g., in the case when $\mathbf{X}$ are the zero dynamics, $\iota(\boldsymbol{x})$ can be obtained from the outputs and their derivatives [31].

Discrete-Time Barrier Functions

We will utilize barrier functions to guarantee forward-invariance of our reduced hybrid system. Thus, we will first present a brief overview. For more details, refer to [198, 123].

Consider a set $\mathcal{I} \subset \mathbf{X} \cap \mathcal{S}$ defined as:

$$\mathcal{I} := \{\boldsymbol{x}_k \in \mathbf{X} \cap \mathcal{S} \mid H(\boldsymbol{x}_k) \geq 0\} \tag{5.2}$$

$$\partial\mathcal{I} := \{\boldsymbol{x}_k \in \mathbf{X} \cap \mathcal{S} \mid H(\boldsymbol{x}_k) = 0\} \tag{5.3}$$

for a smooth function $H : \mathbb{R}^n \to \mathbb{R}$ associated with a Barrier function [122].

Definition 5. (Forward Invariance [123]). The set $\mathcal{I} \subset \mathbf{X} \cap \mathcal{S}$ is *forward invariant* with respect to (5.1) if:

$$\boldsymbol{x}_0 \in \mathcal{I} \implies \boldsymbol{x}_k \in \mathcal{I},\ \forall k \in \mathbb{N}.$$

The discrete-time hybrid system is *safe* with respect to perturbed initial conditions belonging to the set $\mathcal{I}$ if the set $\mathcal{I}$ is forward invariant with respect to (5.1).

While forward invariance is a desirable property for many systems, it can be a challenging property to check in practice. This motivates the use of Barrier functions as a tool for verifying forward invariance. Since we have a discrete-time system, we leverage discrete-time barrier functions, originally introduced in [198]. Specifically, we use the following definition.

Definition 6. (Discrete-time Barrier Function [199]). A function $H : \mathcal{I} \subset \mathbf{X} \cap \mathcal{S} \to \mathbb{R}$ is a *discrete-time barrier function* for the restricted Poincareé map $\boldsymbol{x}_{k+1} = P_{\mathbf{X}}(\boldsymbol{x}_k)$ on the set $\mathcal{I}$ defined by (5.2) if there exists $\alpha \in (0, 1)$ such that for all $\boldsymbol{x} \in \mathcal{I}$:

$$H(P_{\mathbf{X}}(\boldsymbol{x})) - H(\boldsymbol{x}) \geq -\alpha H(\boldsymbol{x}). \tag{5.4}$$

Note that this condition mimics the form of discrete-time Lyapunov functions, with important differences. Namely, we do not require H to be positive definite; this is a consequence that we only require set invariance, and not stability. Yet, because H takes values in the real line (rather than the positive reals), it does imply stability of the set. This is encoded in the following theorem (which is a straightforward application of the results from [199] to the setting of restricted Poincaré maps).

Theorem 6. *Let $\boldsymbol{x}_{k+1} = P_{\mathbf{X}}(\boldsymbol{x}_k)$ be the Poincaré map restricted to the set $\mathbf{X} \cap \mathcal{S}$. If there exists a discrete-time barrier function for the set $\mathcal{I}$, then the set $\mathcal{I}$ is forward invariant and exponentially stable. If $\dim(\mathbf{X} \cap \mathcal{S}) = \dim(\mathcal{S})$ and $\mathcal{I} = \{x^*\}$ then the point x^* is exponentially stable, i.e., the associated periodic orbit is exponentially stable.*

Practically, we choose to describe our set $\mathcal{I}$ as a ball of radius $r \in \mathbb{R}^+$, centered around the fixed point of our nominal periodic orbit $\mathcal{O}$ as defined in (2.11), but restricted to the reduced-order surface $\mathbf{X}$, i.e.:

$$\mathcal{I} := \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{B}_r(\boldsymbol{x}^*) \subset \mathbf{X} \cap \mathcal{S}\},$$

with $\boldsymbol{x}^* := \Phi(x^*)$. Thus, the statement of hybrid forward invariance of $\mathcal{I}$ is equivalent to the set-based condition $P_{\boldsymbol{x}}(\mathcal{B}_r(\boldsymbol{x}^*)) \subseteq \mathcal{B}_r(\boldsymbol{x}^*)$. Using this set definition, we can also explicitly construct our discrete-time Barrier function as:

$$H(\boldsymbol{x}) := r - \|\boldsymbol{x} - (\boldsymbol{x}^*)\|^2. \tag{5.5}$$

Simulation-Based Sampling

To identify the largest set $\mathcal{I}$ that is forward invariant for a given generated nominal limit cycle, we use a sampling-based approach to solve for the largest set $\mathcal{I}$ satisfying the forward-invariance condition. This is framed as an optimization problem of the form:

$$\begin{aligned} r^* = \max_{r \in [0, r_{\max}]} \; & r \\ \text{s.t.} \quad & H(P_{\mathbf{X}}(\boldsymbol{x})) - H(\boldsymbol{x}) \geq -\alpha H(\boldsymbol{x}), \\ & \forall \boldsymbol{x} \in \mathcal{B}_r(\boldsymbol{x}^*). \end{aligned} \tag{5.6}$$

While we were proposing the use of r_{opt} as a metric of robustness since it characterizes the set that is maximally forward-invariant, it is important to note that the parameter α is also indicative of robustness. In essence, the parameter α indicates how fast the step-to-step dynamics are allowed to approach the boundary of the forward-invariant set. When designing a control barrier function, the value of α is used as a parameter to control the rate at which the system is allowed to approach the boundary of the safe set, with the system becoming more conservative as $\alpha \to 1$. Since we are concerned with estimating the forward invariant set rather than controlling for safety, we set α close to 0.

Simulation-in-the-loop Optimization

Solving the aforementioned optimization problem not only provides us with the set $\mathcal{I}$ that is hybrid forward invariant, but it also provides us with a measure of robustness for the associated gait. It is important to note that this application of discrete-time barrier functions to measure locomotive robustness is independent of the choice of controller. For example, this metric could be used in a RL setting as part of the reward composition to optimize the learned policies for robustness.

Robust Walking Results

To show the benefits of the proposed metric, we specifically demonstrate its use towards optimizing offline gaits using a simulation-in-the-loop based framework. This procedure, as illustrated in Fig. 5.4, is repeated for both flat-foot and multi-contact walking on the Atalante lower-body exoskeleton. To further elucidate the efficacy of our approach, we also compare our results to gaits optimized only for stability. A video of the experimental results can be found at [205].

Implementation Details

We choose to generate gaits using the open-source toolbox FROST [41]. These gaits are parameterized using a set of *essential constraints*, as explained in [25]. Specifically, we define these essential constraints to enforce four gait features: step length, step duration, step width, and step height. The output of the gait generation problem is selected to be 7^{th}-order Bézier polynomials, with the phasing variable parameterized by time, for each of the 12 joints of the Atalante lower-body exoskeleton, i.e., $\beta^* \in \mathbb{R}^{12\times 8}$. These gaits are enforced on the robot by tracking the generated output trajectories.

Once generated, the gait is then provided to a MuJoCo simulation environment [206]. This simulation environment is first used to evaluate whether the simulated locomotion is periodically stable. If the resulting locomotion is stable (empirically evaluated as the total number of steps taken in simulation), this provides us with a fixed point, i.e. $\boldsymbol{x}^* = P_{\mathbf{X}}(\boldsymbol{x}^*)$, around which to conduct perturbation analysis. Specifically, a collection of $N = 200$ samples are uniformly drawn from $\mathcal{B}_{r_{\max}}(\boldsymbol{x}^*)$, with $r_{\max} = 2$. This set is denoted:

$$C_0 := \{\boldsymbol{x}_0^{(i)} \sim U(\mathcal{B}_{r_{\max}}(\boldsymbol{x}^*)) \mid i = [1, N]\}.$$

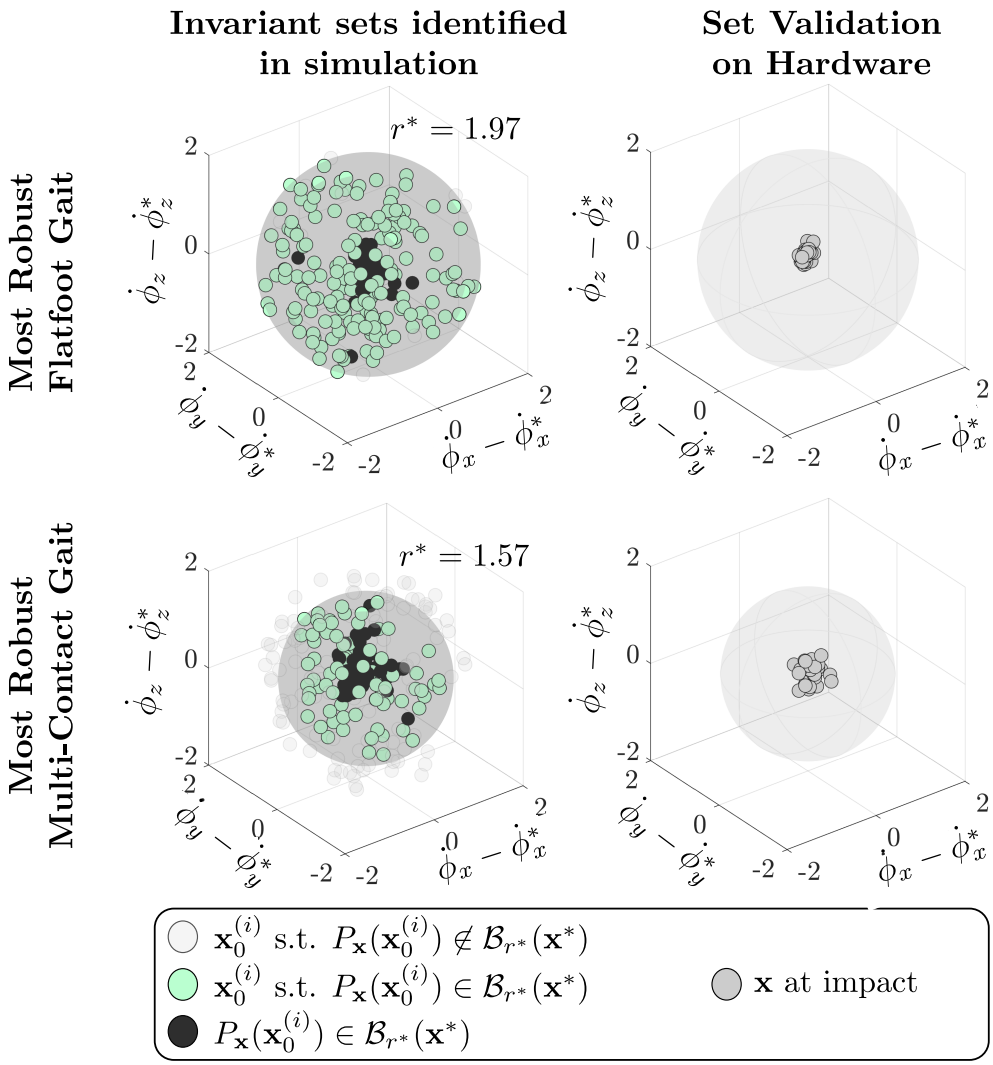


Figure 5.5: Invariant sets identified in simulation compared to the values seen on hardware during the experiments.

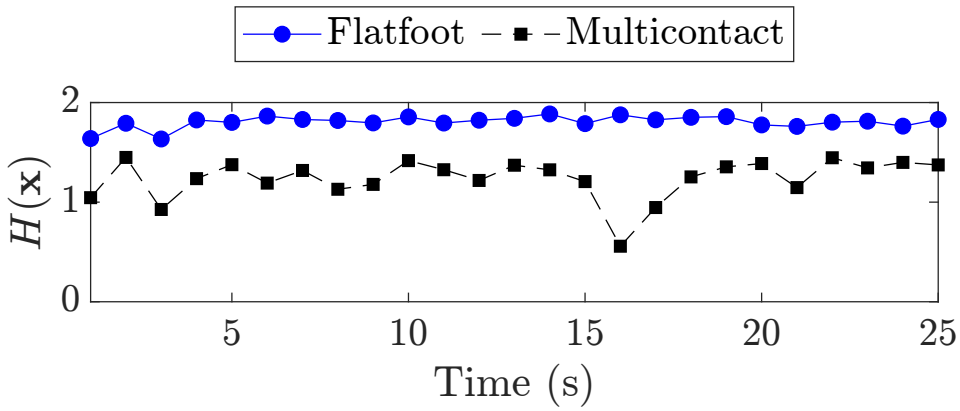


Figure 5.6: Barrier function evaluation using experimental data.

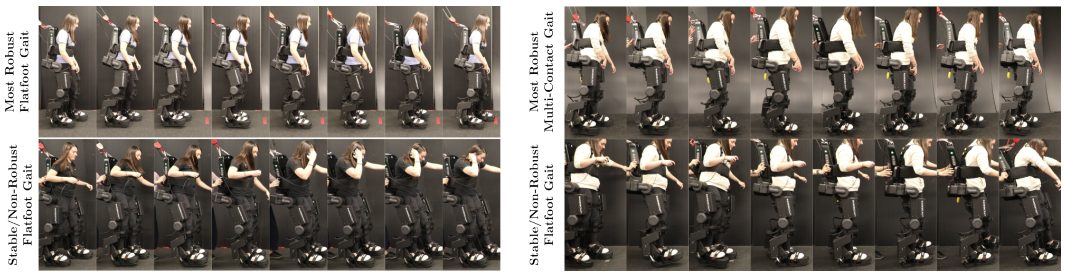


Figure 5.7: Experimental gait tiles.

To obtain the Poincaré return map of the samples (i.e., $C_P := \{P_{\mathbf{X}}(\boldsymbol{x}) \mid \forall \boldsymbol{x} \in C_0\}$), we define our reconstruction map $\iota(\boldsymbol{x})$ by enforcing the systems outputs and holonomic constraints. Each initial condition is then simulated forward for one gait cycle to obtain the Poincaré return set C_P. The maximum set that is forward invariant can then be solved for using the optimization problem (5.6).

The robustness measure r^*, along with the parameters associated with the generated gait, is then provided to a learning algorithm. Here, we choose to leverage preference-based learning since it is able to balance feedback regarding both stability (evaluated by number of steps taken) with robustness (evaluated by r^*) without explicit term weighting. The preference feedback is automatically determined to prefer gaits with higher values of r^*. For gaits that are not stable, the algorithm automatically prefers gaits with higher number of steps taken. The POLAR toolbox was used to implement the preference-based learning algorithm [110].

The entire simulation-in-the-loop procedure was conducted for 10 iterations, with 5 gaits being generated and ranked in each iteration. At the conclusion of the framework, the gait identified as being *maximally robust* was then experimentally demonstrated on the Atalante exoskeleton. The same c++ controller used in simulation is used in the experiments, with additional code to interface with Wandercraft API. For the flat-foot behavior a passivity-based controller [207] was implemented to track the generated gaits. For the multi-contact behavior, due to the uncertain domain transition resulted changing holonomic constraints, we switch the controller to a joint-level PD controller. The controllers are run directly on the Atalante on-board computer (i5-4300U CPU @1.90GHz with 8GB RAM at 1kHz).

Flat-Foot Walking

The framework was first conducted for flat-foot walking. This domain structure, as illustrated in Fig. 5.2(a), only has a single domain and a single edge. The domain

characterizes the continuous-time dynamics associated with single-support flat-foot walking, with the associated edge characterized by the impact of the non-stance foot. Specifically, the domain is described by the following holonomic constraints:

$$\eta_{SS}(q_e) := \begin{bmatrix} p_{st}(q_e) \\ \phi_{st}(q_e) \end{bmatrix}, \tag{5.7}$$

with $p_{st}(q_e) \in \mathbb{R}^3$ and $\phi_{st}(q_e) \in SO(3)$ denoting the position and orientation of the stance foot relative to the world frame. This holonomic constraint is imposed in the gait generation framework via the condition $\eta_{SS}(q_e) = \text{constant}$. We refer the reader to [78] for more details.

The entire learning procedure detailed in the previous subsection, was conducted to generate a total of 26 unique gaits. The invariant set associated with the gait being identified as most robust ($r^* = 1.97$ with $r_{\max} = 2$) is illustrated in Fig. 5.5. Additionally, the discrete-time barrier function, evaluated using experimental data, is illustrated in Fig. 5.6.

Multi-Contact Walking

To further demonstrate the proposed framework, we repeated the similar process for multi-contact walking. Multi-contact refers to behaviors that can be characterized by changing contact modes throughout a single stride. In the HZD framework, each domain is defined by a unique set of holonomic constraints and a corresponding impact event. While multi-contact walking can include as many 8 unique domains [208], we simplify our multi-contact domain structure to only have 2 domains, as illustrated in Fig. 5.2(b). The domains include the single-support phase (as in flat-foot walking) captured by the holonomic constraint (5.7), as well as an additional domain for the foot-rolling phase captured by the following holonomic constraint:

$$\begin{aligned} \eta_{DS}(q_e) := [&p_t(q_e)^\top, \phi_t^x(q_e), \phi_t^z(q_e), \dots \\ &p_h(q_e)^\top, \phi_h^x(q_e), \phi_h^z(q_e)], \end{aligned} \tag{5.8}$$

with $p_t(q_e) \in \mathbb{R}^3$ denoting the position of the back-foot toe frame with the frame's roll and yaw denoted by $\phi_t^y(q_e), \phi_t^z(q_e) \in \mathbb{R}$. Similarly, $p_h(q_e) \in \mathbb{R}^3$, $\phi_h^y(q_e), \phi_h^z(q_e) \in \mathbb{R}$ denote the position, roll, and yaw of the front-foot heel frame.

Again, the entire learning procedure was conducted for 10 iterations, resulting in a total of 21 unique gaits. The invariant set associated with the most robust gait ($r^* = 1.57$ with $r_{\max} = 2$) is illustrated in Fig. 5.5, with the experimental discrete-time barrier function evaluation shown in Fig. 5.6.

Comparison to Optimizing for Stability

To further test our proposed metric, we ran an additional set of experiments in which the gaits were optimized for stability. This was done by replacing the metric provided to the learning agent with the total number of steps taken in the simulation environment, rather than r^*. These experiments were conducted for both the flat-foot and multi-contact behaviors, with the resulting gaits 'optimized for stability' illustrated in Fig. 5.7. It is interesting to note that while these gaits were both stable in simulation, they were not able to successfully yield stable locomotion when translated to hardware. This indicates that our proposed approach is a successful metric for capturing real-world robustness.

Summary

This work explored the use of discrete-time barrier functions to synthesize robust walking gaits. The main idea was that locomotive robustness can be related to forward-invariance of the discrete-time step-to-step dynamics. Specifically, the size of these forward-invariant sets was proposed as a metric for locomotive robustness. Lastly, a simulation-based framework was outlined and demonstrated towards experimentally synthesizing robust nominal gaits for both flat-foot and multi-contact walking on the Atalante lower-limb exoskeleton. A limitation of this work includes the fact that robustness is not the only factor important for desirable robotic locomotion. As such, future work includes the combination of robustness with other important metrics such as user comfort and naturalness.

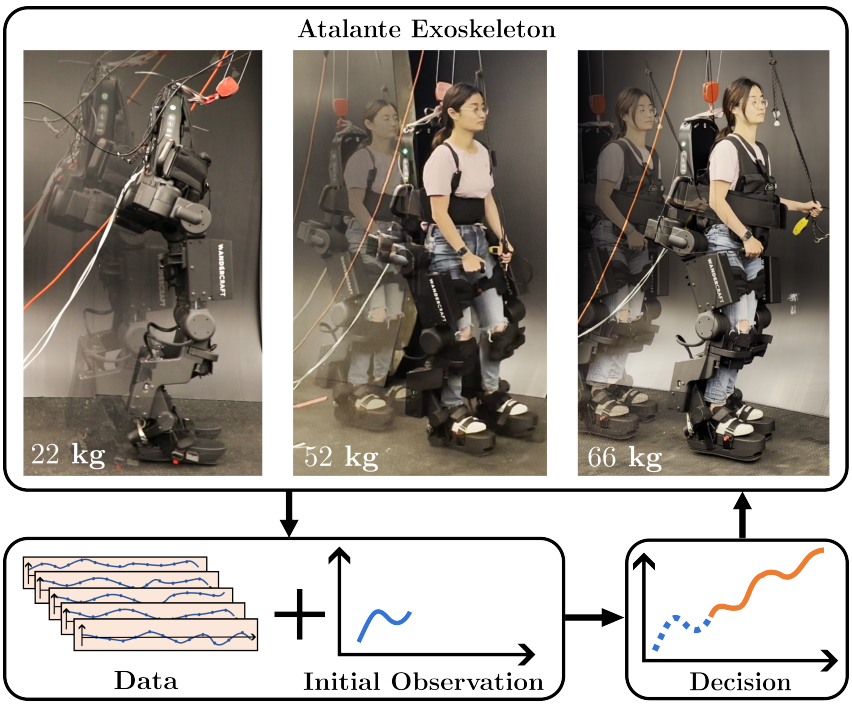


Figure 5.8: Illustration of data-driven predictive control for bipedal locomotion on lower-body exoskeleton Atalante with various payloads.

5.2 Online Planning for Dynamic Walking

The previous sections focused on designing robust nominal trajectories offline. While such trajectories improve baseline stability, they offer only limited robustness: once unexpected disturbances or user-induced variations push the system outside the predefined attraction basin, the gait cannot recover. Real-world walking, however, inevitably involves such variability—ranging from terrain changes to unmodeled dynamics—which cannot be fully anticipated during offline synthesis.

To address this, robustness must extend beyond static plan design to online adaptation. Online planning enables the gait to reconfigure in real time by adjusting footstep locations, timing, and body motion in response to disturbances. To enable online replan capability, the central trade-off is between accuracy and tractability: full-order models provide physical fidelity but are computationally demanding, while reduced-order models (e.g., LIP-based templates) enable fast replanning but require simplifying assumptions. Building on the trade-off discussion in Chapter 2, we pursue reduced-order approaches that remain computationally lightweight yet integrate with full-order dynamics to preserve feasibility and stability by leveraging data-driven approach.

With the advancement in modern computational power, pure data-driven approaches, such as reinforcement learning, have offered a model-free approach to train controllers by exploiting large amounts of data from simulators [30]. While offering robustness, these approaches often require extensive training data and are sensitive to reward design. This motivates leveraging data-driven approaches to learn a more accurate representation of the system dynamics, such as learning the residual dynamics either using Gaussian process [209] or deep neural network [210], to use in conjunction with classic control methods. In the context of locomotion, due to the high DoFs, to allow for online planning capability, existing work focuses on achieving robust locomotion [28, 211] via learning a reduced-order representation of the system to mitigate model mismatches.

Data-driven approaches based on behavioral systems theory [212] offer a middle ground between model-based methods and learning-based strategies. By representing linear time-invariant (LTI) systems directly from data, these methods can be incorporated into predictive control frameworks, commonly referred to as data-enabled predictive control (DeePC) or data-driven predictive control (DDPC) [213]. DDPC has demonstrated both computational efficiency and practical effectiveness, with applications ranging from underactuated quadrupeds to interconnected multi-robot systems and assistive devices [214, 215, 216]. Its ability to design policies from past feasible trajectories makes it well suited for real-time control in applications with constrained kinematic spaces. Unlike many modern legged robots that assume negligible leg mass [34, 183], the Atalante exoskeleton has a significant portion of its mass concentrated in the legs (see Fig. 5.8). This motivates our development of a data-driven dynamic model tailored for user–exoskeleton systems and its integration into trajectory planning.

Data-driven approaches, grounded in behavioral systems theory [212], provide a middle ground between model-based methods and learning-based approaches. These methods effectively learn linear time-invariant (LTI) system models, and have been successfully integrated into predictive control frameworks, such as data-enabled predictive control (DeePC) or data-driven predictive control (DDPC) [213]. DDPC is computationally efficient, and has been successfully applied to systems like quadrupeds, interconnected systems, and assistive devices [214, 215, 216]. Its ability to design policies based on past feasible trajectories makes it a more suitable choice for real-time control in environments with constrained kinematic spaces, where rapid and reliable control is critical.

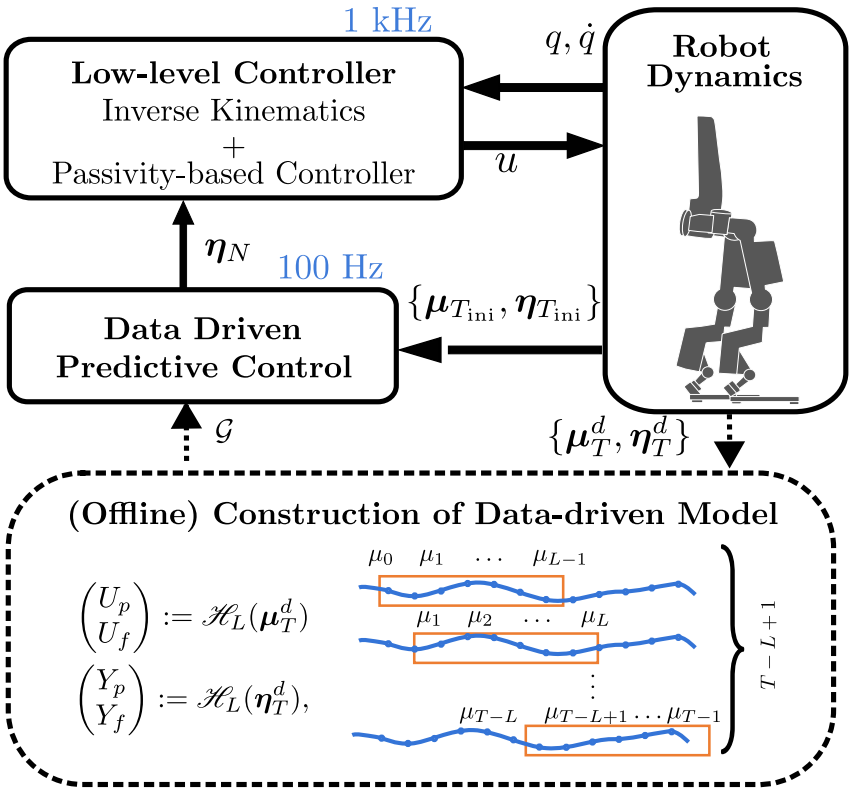


Figure 5.9: Overview of the proposed layered control framework composed of the DDPC as a planner with constructed data-driven model and low-level controller.

In this section, we present two online planning approaches that leverage offline data. Both Data-Driven Predictive Control (DDPC) and Hybrid Data-Driven Predictive Control (HDDPC) use Hankel matrices to represent system behavior directly from collected trajectories, but they differ in scope. DDPC only account for system continuous dynamics, where the Hankel matrix encodes feasible trajectories that evolve smoothly over time. This enables online trajectory generation and adaptation without requiring explicit model identification. HDDPC, in contrast, extends the framework to also consider contact schedule, where both continuous evolution and discrete step-to-step (S2S) transitions must be captured. By incorporating both continuous trajectory and S2S transition into the Hankel representation, HDDPC provides a natural mechanism for adapting gaits across steps, making it well suited for legged locomotion with impacts.

We begin with a brief introduction to behavioral systems theory and existing theoretical results in data-driven predictive control, before turning to the construction of data-driven models for the user–exoskeleton system.

Behavioral Systems Theory

This subsection briefly reviews some of the fundamental results of the behavioral systems theory. Let us consider a state representation of a discrete-time LTI system as follows:

$$\begin{aligned} \theta(t+1) &= A\,\theta(t) + B\,\mu(t) \\ \eta(t) &= C\,\theta(t) + D\,\mu(t), \end{aligned} \tag{5.9}$$

where $\theta(t) \in \mathbb{R}^{\beta}$, $\mu(t) \in \mathbb{R}^{\kappa}$, $\eta(t) \in \mathbb{R}^{\nu}$ represent the state vector, control inputs, and outputs, respectively, at time $t \in \mathbb{Z}_{\geq 0} := \{0, 1, \dots\}$, and $A \in \mathbb{R}^{\beta\times\beta}$, $B \in \mathbb{R}^{\beta\times\kappa}$, $C \in \mathbb{R}^{\nu\times\beta}$, $D \in \mathbb{R}^{\nu\times\kappa}$ denote the unknown state matrices. In behavioral systems theory, a dynamical system is defined as a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$, where $\mathbb{W}$ is a signal space and $\mathscr{B} \in \mathbb{W}^{\mathbb{Z}_{\geq 0}}$ is the behavior. In contrast with classical systems theory with a particular parametric system representation such as that of (5.9), behavioral systems theory focuses on the subspace of the signal space where system trajectories live. Let $\boldsymbol{\mu} := \text{col}(\mu_0, \mu_1, \dots, \mu_{T-1})$ be an input trajectory with length $T \in \mathbb{N} := \{1, 2, \cdots\}$ applied in $\mathscr{B}$ with the corresponding output trajectory $\boldsymbol{\eta} := \text{col}(\eta_0, \eta_1, \dots, \eta_{T-1})$. We can construct the Hankel matrix with $\boldsymbol{\mu}$ by concatenating trajectory with length $L \in \mathbb{N}$ and $T > L$ as follows:

$$\mathscr{H}(\boldsymbol{\mu}) := \begin{bmatrix} \mu_0 & \cdots & \mu_{T-L} \\ \vdots & \ddots & \vdots \\ \mu_{L-1} & \cdots & \mu_{T-1} \end{bmatrix} \in \mathbb{R}^{\kappa L\times(T-L+1)}.$$

Here we note that the Hankel matrix with $\boldsymbol{\eta}_T$, $\mathscr{H}(\boldsymbol{\eta}_T)$, can be constructed analogously.

Considering input sequences $\boldsymbol{\mu}^{\Lambda} := \{\boldsymbol{\mu}^{(i)} \mid \forall i = 1, \dots, \Lambda\}$, where Λ indicates the number of datasets, we can construct the Trajectory Hankel matrix as follows:

$$\mathscr{H}(\boldsymbol{\mu}^{\Lambda}) := \begin{bmatrix} \mu_0^{(1)} & \mu_0^{(2)} & \cdots & \mu_0^{(\Lambda)} \\ \mu_1^{(1)} & \mu_1^{(2)} & \cdots & \mu_1^{(\Lambda)} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{L-1}^{(1)} & \mu_{L-1}^{(2)} & \cdots & \mu_{L-1}^{(\Lambda)} \end{bmatrix} \in \mathbb{R}^{\kappa L\times\Lambda},$$

where $\mu_j^{(i)}$ represents the j-th sample from the i-th trajectory or dataset. The Trajectory Hankel matrix with $\boldsymbol{\eta}^{\Lambda}$, denoted by $\mathscr{H}(\boldsymbol{\eta}^{\Lambda})$, can be constructed analogously.

The signal $\boldsymbol{\mu}$ is said to be persistently exciting of order L if $\mathscr{H}(\boldsymbol{\mu})$ (or its multi-dataset extension $\mathscr{H}(\boldsymbol{\mu}^{\Lambda})$) is of full row rank [217]. If a dataset of input-output (I-O) trajectories $(\boldsymbol{\mu}, \boldsymbol{\eta})$ is persistently exciting of order $L + \beta$, then by the Fundamental Lemma and its multi-dataset extension [218, 213, 217], any trajectory of the LTI

system can be expressed as a linear combination of the columns of the corresponding Hankel matrices. Equivalently, any new trajectory pair $(\boldsymbol{\mu}_L, \boldsymbol{\eta}_L)$ of length L lies in the range space of the Hankel matrices, i.e., it is a valid trajectory if and only if there exists a coefficient vector γ such that

$$\begin{bmatrix} \mathscr{H}(\boldsymbol{\mu}^{\Lambda}) \\ \mathscr{H}(\boldsymbol{\eta}^{\Lambda}) \end{bmatrix} \gamma = \begin{bmatrix} \boldsymbol{\mu}_L \\ \boldsymbol{\eta}_L \end{bmatrix}. \tag{5.10}$$

The signal $\boldsymbol{\mu}$ is said to be persistently exciting of order L if $\mathscr{H}_L(\boldsymbol{\mu})$ or the multi dataset version $\mathscr{H}(\boldsymbol{\mu}^{\Lambda})$ is of full row rank [217]. If a given data input-output (I-O) trajectory of the system, denoted by the pair $(\boldsymbol{\mu}_T, \boldsymbol{\eta}_T)$, is persistently exciting of order $L + \beta$, from Fundamental Lemma [217, 218], any trajectory of the LTI system can be constructed using a linear combination of the columns of the Hankel matrices. In particular, one can use the columns of the Hankel matrices to develop a data-driven model to predict the system's future behavior.

To make this notion more precise, let us take L as $L = T_{\text{ini}} + N$, where T_{ini} and N denote the estimation horizon and control horizon, respectively. The estimation horizon is used to estimate the system's initial state from past I-O measurements. The control horizon is used for the predictive controller. We can partition the Hankel matrices into the past and future portions accordingly as follows:

$$\begin{bmatrix} U_p \\ U_f \end{bmatrix} := \mathscr{H}(\boldsymbol{\mu}_T) \qquad \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} := \mathscr{H}(\boldsymbol{\eta}_T), \tag{5.11}$$

where $U_p \in \mathbb{R}^{\kappa T_{\text{ini}} \times (T-L+1)}$, $U_f \in \mathbb{R}^{\kappa N \times (T-L+1)}$, $Y_p \in \mathbb{R}^{\nu T_{\text{ini}} \times (T-L+1)}$, $Y_f \in \mathbb{R}^{\nu N \times (T-L+1)}$. From the Fundamental Lemma [213, 217, 218], any new trajectory lies in the range space of the Hankel matrices, or equivalently, there exists $\gamma \in \mathbb{R}^{T-L+1}$ such that

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} \gamma = \begin{bmatrix} \boldsymbol{\mu}_{T_{\text{ini}}} \\ \boldsymbol{\eta}_{T_{\text{ini}}} \\ \boldsymbol{\mu}_N \\ \boldsymbol{\eta}_N \end{bmatrix}, \tag{5.12}$$

where $\boldsymbol{\mu}_{T_{\text{ini}}}$ and $\boldsymbol{\eta}_{T_{\text{ini}}}$ denote the past portions of the I-O trajectories over the estimation horizon of T_{ini}. Similarly, $\boldsymbol{\mu}_N$ and $\boldsymbol{\eta}_N$ represent the predicted (i.e., future) portions of the I-O trajectories over the control horizon of N. Since the dimensionality of the γ vector in formulating predictive controllers is huge, we aim to remove γ from (5.12). One way is to have an offline approximation for γ using least squares similar

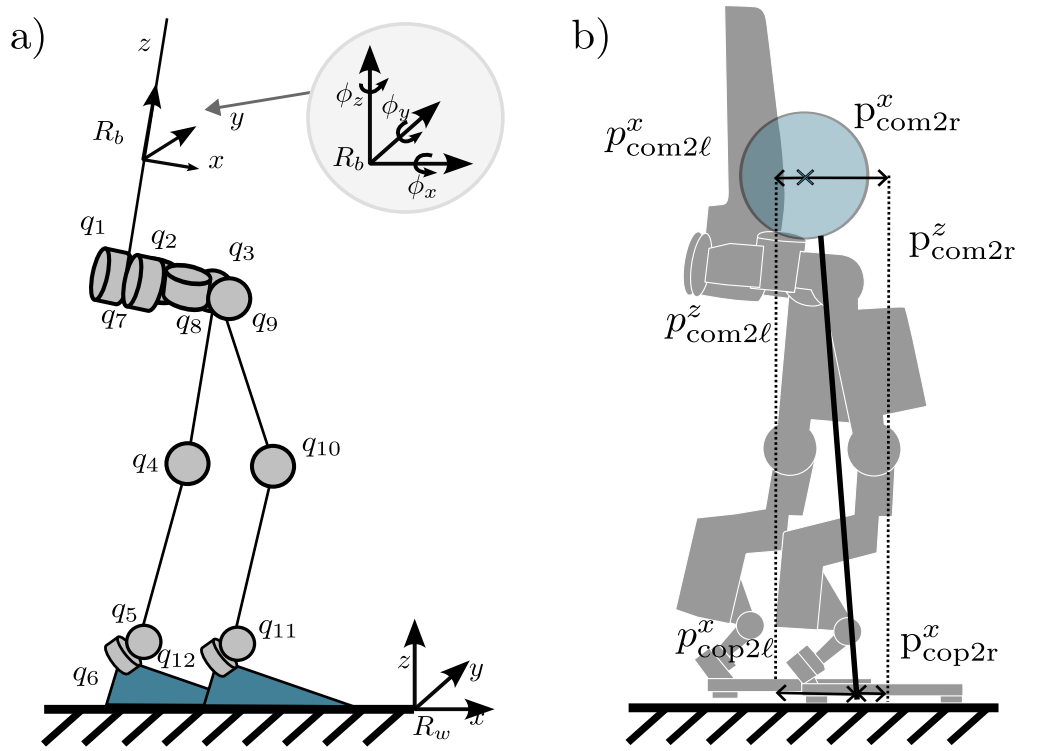


Figure 5.10: System representation of Atalante exoskeleton for Hankel Matrix construction. a) Generalized coordinates for the lower-body exoskeleton Atalante. b) The input and output variables in the x-direction to be used for the Hankel matrix construction.

to [214, 219] to obtain the data-driven model as

$$\boldsymbol{\eta}_N = Y_f \gamma = \underbrace{Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{\dagger}}_{\mathcal{G}} \begin{bmatrix} \boldsymbol{\mu}_{T_{\text{ini}}} \\ \boldsymbol{\eta}_{T_{\text{ini}}} \\ \boldsymbol{\mu}_N \end{bmatrix}, \tag{5.13}$$

where $(\cdot)^{\dagger}$ represents the pseudo inverse and $\mathcal{G}$ denotes the data-driven state transition matrix over N-steps.

Data-Driven Motion Planner

In this subsection, we presents a layered framework utilizing data-driven predictive control (DDPC) for trajectory planning and control of the lower-body exoskeleton Atalante as illustrated in Fig. 5.9. The primary contributions of this study are threefold. First, we propose a data-driven dynamic model employing behavioral systems theory with time-domain trajectories of the center of mass (CoM) and center

of pressure (CoP), inspired by the LIP model for bipedal locomotion. Second, we design a trajectory planner based on convex DDPC with the proposed data-driven model at the higher level of the proposed layered framework. At the lower level of the framework, the controller incorporates inverse kinematics and passivity-based controllers to translate the planned trajectory from DDPC into the full-order model of the lower-body exoskeleton (Fig. 5.10a). Third, we conduct experimental evaluations of the proposed data-driven layered framework through numerical simulations and hardware demonstrations. Comparative analysis in simulations demonstrates that the proposed data-driven layered framework effectively stabilizes bipedal gaits at higher speeds, in contrast to the traditional model predictive control (MPC)-based planner for the LIP model. Furthermore, hardware experiments with different kinematics and payloads showcase the ability of this framework to account for user variability as well as the robustness under deviations from nominal data-driven model.

This subsection aims to present the proposed DDPC-based trajectory planner at the high level of the control scheme for bipedal locomotion (see Fig. 5.9).

Construction of the Data-Driven Model

Assuming reasonable behavior for the actuated coordinates, the difficulty and complexity of the bipedal locomotion usually lie in the control and planning for the weakly actuated or underactuated Centroidal states. In this context, designing the CoM trajectory encapsulated all the requisite DoFs' information, although their individual dynamics are not described explicitly. Hence instead of using the full-order states x for trajectory planning, we are focusing on the Centroidal states to construct the Hankel matrices. Specifically, we draw inspiration from the LIP model that considers the CoM and CoP of the robot. We consider a local representation of these states with respect to either the left or right foot frame as $\mathrm{p}^{x,y}_{\mathrm{cop2}\ell/\mathrm{r}} := \mathrm{p}^{x,y}_{\mathrm{cop}} - \mathrm{p}^{x,y}_{\ell/\mathrm{r}}$ and $\mathrm{p}^{x,y,z}_{\mathrm{com2}\ell/\mathrm{r}} := \mathrm{p}^{x,y,z}_{\mathrm{com}} - \mathrm{p}^{x,y,z}_{\ell/\mathrm{r}}$ as shown in Fig. 5.10b.

An intuitive choice would be using the CoM and CoP trajectories in the stance-foot frame. However, this choice would create discontinuity during domain switches, posing challenges for planning through impact. To prevent the state space from monotonically increasing during forward walking and to maintain continuity in the trajectory planning, we adopt a redundant representation where the trajectory of the CoM and CoP with respect to both stance and swing feet are being considered. Hence, the input $\mu \in \mathbb{R}^4$ and output $\eta \in \mathbb{R}^6$ of the date-driven model are chosen as

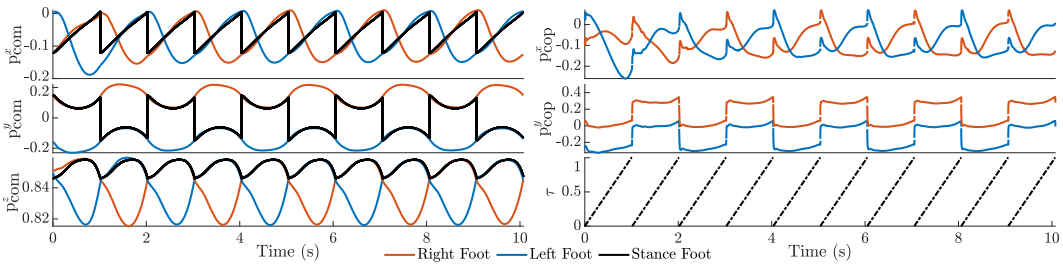


Figure 5.11: One set of the planned CoM and CoP trajectories from the DDPC planner and the tracked trajectory in simulation in right foot frame, the left foot frame trajectories, and the Stance foot. The stance foot frame trajectory in black is generated from DDPC. The corresponding phase variables are plotted in the dashed line.

follows:

$$\eta := \mathrm{col}(\mathrm{p}^{x,y,z}_{\mathrm{com2}\ell},\ \mathrm{p}^{x,y,z}_{\mathrm{com2r}}) \qquad \mu := \mathrm{col}(\mathrm{p}^{x,y}_{\mathrm{cop2}\ell},\ \mathrm{p}^{x,y}_{\mathrm{cop2r}}).$$

Notably, with this redundant formulation, we also provide trajectories in the following domain throughout the prediction horizon for the low-level controller to track in case of unexpected early or late impact. This is especially advantageous in scenarios involving uncertain impact timings or unexpected communication delays, significantly enhancing the framework's practical applicability. We do not include velocity terms in our representation as the velocity and acceleration information is implicitly captured in the position trajectory in the Hankel matrix. We remark that the proposed data-driven model, inspired by the LIP model, captures more comprehensive information about the system with implicit consideration of swing foot trajectory and the effect of the low-level whole-body controller on the system dynamics. A comparative analysis of the performance between these two models and their corresponding planner architectures will be presented in Section 5.2.

DDPC Algorithm for Trajectory Optimization

We are now positioned to present the DDPC-based trajectory planner for optimizing the I-O (i.e., CoP and CoM) trajectories. The real-time DDPC planner is formulated

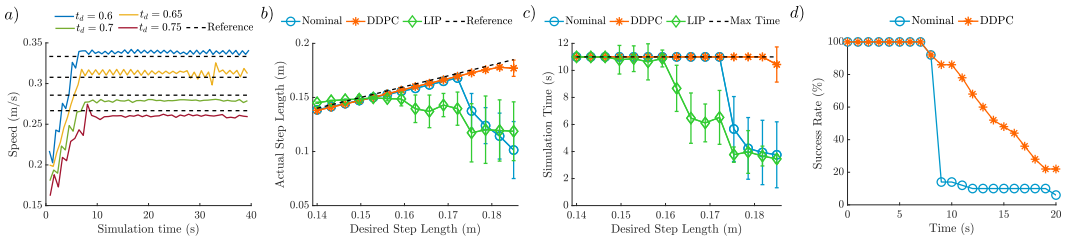


Figure 5.12: Simulation comparison over nominal indicated by blue circles, DDPC indicated by orange stars, and MPC indicated by green diamonds. a) DDPC planner planning trajectories for increasing desired speed, capped at maximum step length 0.2 m at different step duration t_d. b) tracking performance over different desired step length with the same step duration. The dashed line indicate the ideal performance. Error bar indicates the standard deviation over 50 models. c) Simulation time before robot falling for the tracking performance comparison. Error bar indicates the standard deviation over 50 models. The maximum simulation time is 11 s, indicated by the horizontal dash line. d) Comparison of Nominal and DDPC under time-varying perturbation applied on the negative x direction.

as the following strictly convex quadratic program (QP)

$$\begin{aligned} \min_{(\boldsymbol{\mu}_N, \boldsymbol{\eta}_N)} \quad & \sum_{k=0}^{N-1} \left(\|\eta_k - r_k^\eta\|_Q^2 + \|\mu_k - r_k^\mu\|_R^2 \right) \\ \text{subject to} \quad & \boldsymbol{\eta}_N = \mathcal{G} \begin{bmatrix} \boldsymbol{\mu}_{T_{\text{ini}}} \\ \boldsymbol{\eta}_{T_{\text{ini}}} \\ \boldsymbol{\mu}_N \end{bmatrix} \quad \text{(Data-Driven Model)} \\ & \mu_k \in \mathcal{U}, \quad \eta_k \in \mathcal{P}, \quad k = 0, \ldots, N-1, \end{aligned} \tag{5.14}$$

where $\boldsymbol{r}^\eta := \text{col}(r_0^\eta, \cdots, r_{N-1}^\eta)$ represents a reference output (i.e., CoM) trajectory, $\boldsymbol{r}^\mu := \text{col}(r_0^\mu, \cdots, r_{N-1}^\mu)$ denotes a reference input (i.e., CoP) trajectory, and $(\boldsymbol{\mu}_{T_{\text{ini}}}, \boldsymbol{\eta}_{T_{\text{ini}}})$ represents the pair of past actual I-O trajectories (i.e., feedback to the DDPC). In addition, $\mathcal{U} \subseteq \mathbb{R}^4$ and $\mathcal{P} \subseteq \mathbb{R}^6$ denotes the input and output feasibility sets, respectively. Finally, $Q \in \mathbb{R}^{6\times6}$ and $R \in \mathbb{R}^{4\times4}$ are chosen as positive definite weighting matrices.

Layered Control Framework for Atalante

In this section, we introduce the remaining components and details for practically implementing the entire layered control framework used to realize locomotion on the lower-body exoskeleton Atalante (see Fig. 5.8). The device weighs 82 kg and has adjustable thigh and shin length to account for the varying heights of users. Depending on the physical parameters of the users, the largest kinematically feasible

step length in x direction that the device is capable to achieve for flat-footed walking is less than 0.2 m.

Trajectory Planner with DDPC

We generate a reference trajectory with the open-source toolbox FROST [41] using the full-order system to account for kinematics and dynamics feasibility. This trajectory is described by 7-th order Bézier polynomials with the coefficient matrix of $\alpha = [\alpha_{\text{com}}, \alpha_{\phi}, \alpha_{\text{sw}}]$, where α_{com} describes the CoM trajectory, α_{ϕ} described the pelvis orientation and α_{sw} described the swing foot position and orientation. The Bézier polynomials are evaluated based on a time-based phase variable. Specifically, considering a desired step duration t_d and the initial time at the beginning of the domain t_0, we can calculate the phase variable $\tau_k = \frac{t_k - t_0}{t_{\text{d}}}$ at time point t_k. The reference for the CoM trajectory is then determined by $r_k^{\eta} = \text{col}(\text{p}_{\text{com}}^{x,y,z}(\alpha, \tau_k) - p_{\ell}^{x,y,z}(\alpha, \tau_k),\ \text{p}_{\text{com}}^{x,y,z}(\alpha, \tau_k) - \text{p}_{\text{r}}^{x,y,z}(\alpha, \tau_k))$. In addition, the reference CoP trajectory is generated via $r_k^{\mu} = \text{col}(-\text{p}_{\text{sw}}^{x,y}(\alpha, \tau_k), 0, 0)$ for the right stance and $r_k^{u} = \text{col}(0, 0, -\text{p}_{\text{sw}}^{x,y}(\alpha, \tau_k))$ for the left stance.

Without explicit guidelines to construct the Hankel matrix for nonlinear systems, we empirically determine the hyperparameters of the DDPC algorithm via a grid search over the space of the discrete-time interval between nearby points $\delta_t \in [0.01, 0.03]$, the trajectory length of $T \in [50, 600]$, the initial trajectory length of $T_{\text{ini}} \in [5, 50]$, and the control horizon of $N \in [10, 300]$. This search space is constructed considering the computation speed, low-level controller frequency, noisy level of the data, and the amount of data required. We choose $T = 400$, $T_{\text{ini}} = 10$, $N = 20$, and $\delta_t = 0.02$ with a selection criteria on the accuracy of the least-square approximation over some unseen trajectory. In total, 8 s of data are used for the Hankel matrix construction, and a trajectory for 0.4 s is planned. This means our DDPC-based trajectory planner primarily acts as a short-term regulator to stabilize the system. We remark that even though $\text{p}_{\text{com}}^{z}$ is planned, it is not being used by the low-level controller but only used as part of the states to determine system dynamics. The trajectories are planned at 100 Hz, which is faster than the interval δ_t specified. However, since the reference generation is based on the current domain and phase variable, the trajectory planner still has the ability to regulate the system behavior during this interval till the next time sample at which $\boldsymbol{\mu}_{T_{\text{ini}}}$ and $\boldsymbol{\eta}_{T_{\text{ini}}}$ are updated.

Output Synthesis

The desired walking behavior is encoded by the task space output $\mathbf{y} = \mathbf{y}^{\text{act}} - \mathbf{y}^{\text{des}}$, where $\mathbf{y}^{\text{act}} \in \mathbb{R}^{12}$ and $\mathbf{y}^{\text{des}} \in \mathbb{R}^{12}$ are the actual and desired outputs, respectively. In particular, we choose the following outputs for the system:

$$\mathbf{y}^{\text{act}} = \begin{bmatrix} \mathrm{p}_{\text{com2st}}^{x,y,z}(q) & \phi_{\text{pelv}}^{x,y,z}(q) & \mathrm{p}_{\text{sw}}^{x,y,z}(q) & \phi_{\text{sw}}^{x,y,z}(q) \end{bmatrix}$$
$$\mathbf{y}^{\text{des}} = \begin{bmatrix} \mathrm{p}_{\text{com2st}}^{x,y,z} & \phi_{\text{pelv}}^{x,y,z}(\alpha) & \mathrm{p}_{\text{sw}}^{x,y,z}(\alpha, \lambda^{x,y}) & \phi_{\text{sw}}^{x,y,z}(\alpha) \end{bmatrix},$$

where the desired COM position $\mathrm{p}_{\text{com2st}}^{x,y}$ is generated by the high-level DDPC planner, and the other desired components are taken as Bézier polynomials with the coefficient matrix of α and the step length of $\lambda^{x,y}$. More specifically, the coefficients of pelvis orientation ϕ_{pelv} and swing foot orientation ϕ_{sw}, and z-height of CoM and swing foot trajectory are fixed and from the aforementioned reference trajectory. The swing foot x, y trajectories are determined by Bézier polynomials connecting the swing foot position at the beginning of the domain (i.e., post-impact state) and the desire foot targets, i.e., $\mathrm{p}_{\text{sw}}^{x,y}(\tau) = (1 - \beta(\tau))\, \mathrm{p}_{\text{sw}}(q^+) + \beta(\tau)\, \lambda^{x,y}$, where β is a phase-based weighting function.

Low-Level Feedback Controller

Our low-level controller, implemented in C++, receives trajectories for the CoM position target from the DDPC-based trajectory planner. Given that the controller operates at 1 kHz, much faster than the planner's replan frequency and the discrete-time interval, CoM target positions and velocities for each control tick is obtained via linear interpolation. Subsequently, we employ a Newton-Raphson numerical inverse kinematics (IK) algorithm to calculate the desired joint targets, which are then tracked using a passivity-based control method [207]. The kinematics and dynamics evaluation is performed with Pinocchio [220]. To account for uncertainty in state estimation and impact time, we switch to the next stance domain only when the swing foot ground reaction force exceeds some pre-defined threshold instead of time-based switching.

Experimental Validations of DDPC

In this subsection, we present the numerical simulation and hardware experiment results for our proposed framework. The experiment video could be found in [221].

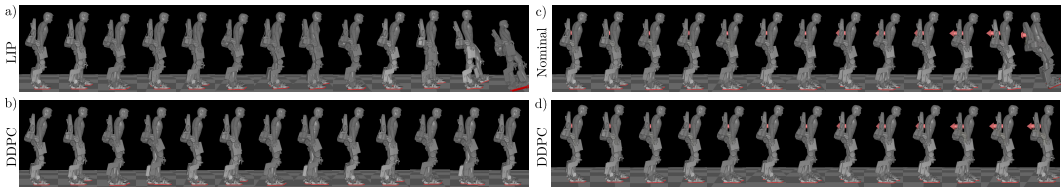


Figure 5.13: Gait tiles for simulation. Simulations with a) LIP-based MPC and b) DDPC controllers for walking at the speed of 0.16 (m/s), c) Nominal trajectory and d) DDPC under time-varying perturbations.

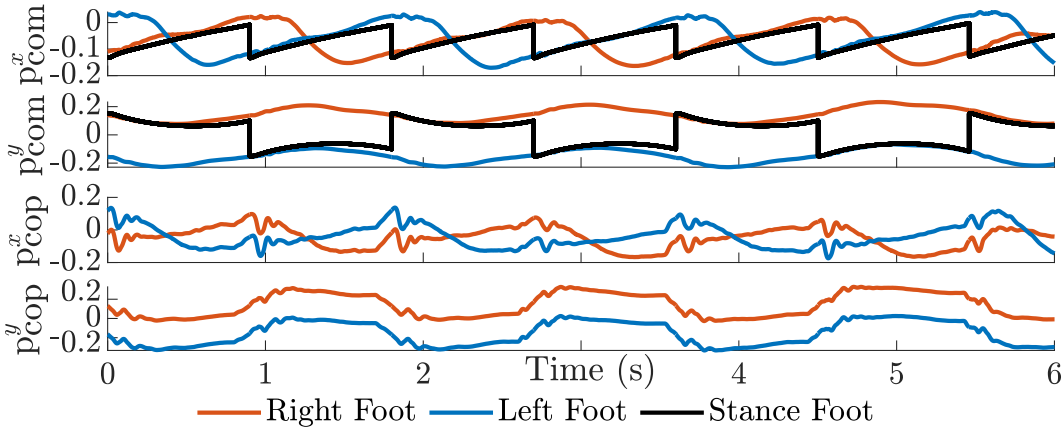


Figure 5.14: Desired trajectories from the DDPC-based trajectory planner (black solid line) and actual evolving CoM/CoP states from the hardware experiment.

Simulation Results

We validate the effectiveness of our framework with numerical simulations in MuJoCo [206]. To account for uncertainty in mass distribution estimation for human users, we generated 50 randomized models with identical user total mass by varying the CoM offset and inertia properties, resulting in a p_{com} ranging from $[-0.122, -0.106]$ m in nominal standing configuration. The low level controller is using the information from the same nominal model constructed with parameters from [15].

Tracking Performance: We compare tracking performance over these randomly generated models. For each model, simulation data with different desired step lengths ranging from 0.1 to 0.15 m were collected to construct the $\mathcal{G}$ matrix. An example trajectory generated by the DDPC planner and the corresponding actual CoM and CoP states are shown in Fig. 5.11. Moreover, DDPC is able to achieve stable walking in various speed as described in Fig. 5.12a. Additionally, we implemented an MPC planner based on the LIP dynamics for comparative analysis. This MPC shares a similar problem structure to that described in (5.14), but it substitutes the Hankel matrix trajectory constraints with those of the LIP dynamics. Considering

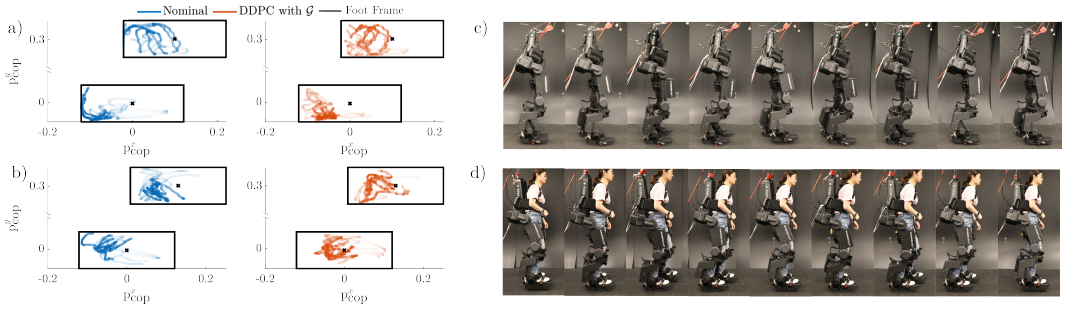


Figure 5.15: The system evolution with the data-driven layered framework is shown in orange, and the system evolution with nominal trajectory is shown in blue. a) Experiment result for exoskeleton carrying 20 kg of payload. CoP position in the foot frame for 5 left stance foot steps and 5 right stance foot steps. b) CoP position for exoskeleton with user inside. c) Gait tiles with the DDPC planner for the 20 kg payload experiment. d) Gait tiles with the DDPC planner for the experiment with user.

the kinematic limits and feasibility concerns, we opted for a 0.8 m CoM z-position in the LIP model (as opposed to 0.85 m) and set $\delta_t = 0.01$ s to discretize the LIP dynamics. The time horizon for the MPC is also chosen as $N_{\text{LIP}} = 300$.

The DDPC planner's tracking performance was evaluated against that of the MPC and a nominal trajectory for desired walking speeds between 0.14 and 0.19 m/s, with a constant step duration of 1 s. As shown in Fig. 5.12b, the DDPC planner reliably achieves close tracking of the desired step length. As the desired speed increases by increasing the desired step length, the validity of LIP dynamics starts to degrade. In particular, the LIP MPC's performance begins to decline at speeds above 0.16 m/s, and the nominal trajectory cannot be appropriately tracked beyond 0.17 m/s, as described in Fig. 5.12b. Furthermore, the DDPC planner maintains the system's stability for a longer duration than the other controllers as demonstrated in Fig. 5.12c. Here, we define a robot maintaining CoM above a certain threshold throughout the maximum simulation time as a success. It is a failure if QP or IK becomes infeasible or CoM falls below a specified threshold. An example of failure case is shown in Fig. 5.13a for MPC.

Since the DDPC-generated trajectory is based on the feasible trajectory used to construct $\mathcal{G}$, it tends to run into kinematics issues less frequently than physics-based reduced-order model like LIP, where no full-order model system information is exposed to the planner.

Adaptation to Perturbation: We further evaluate the DDPC's robustness against time-varying external disturbances (see Fig. 5.13c and 5.13d). Specifically, we

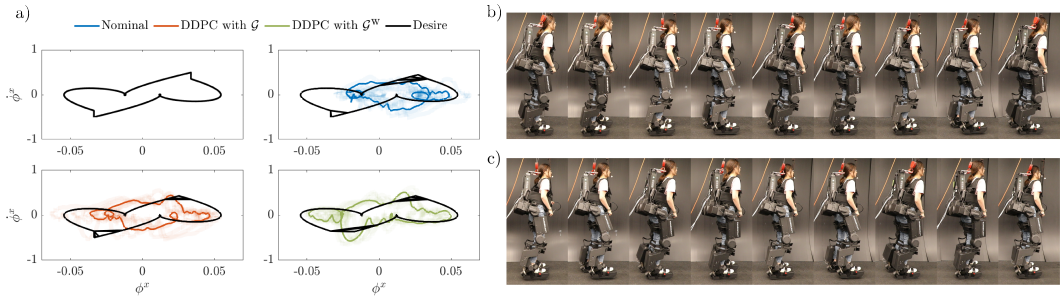


Figure 5.16: Hardware experiment with additional weight. a) Phase portrait over pelvis roll ϕ_x and $\dot{\phi}_x$ for hardware experiment with user wearing additional weight for nominal, DDPC with $\mathcal{G}$, DDPC with $\mathcal{G}^w$ b) Gait tiles for experiment with DDPC controller with $\mathcal{G}$ with additional weight c) Gait tiles for experiment with DDPC controller with $\mathcal{G}^w$ with additional weight

continuously apply an external perturbation force to the torso's negative x direction. This direction is chosen because the device is more sensitive towards perturbation applied along this axis with its fairly negative nominal CoM position. The initial perturbation force was set at 5 N with a discrete increase of 3 N every 3 s. To adapt to this time-varying perturbation, the Hankel matrix is updated online. The choice of the Hankel matrix size requires a balance between satisfying persistently exciting requirements and using obsolete data. The simulation's first 5 s were used for data collection, which was a smaller trajectory length $T = 250$, and $\mathcal{G}$ is reconstructed every 1.5 s. The least-square approximation step is completed within $70 - 100$ ms. We ran the same setup for the aforementioned 50 randomly generated models and compared it with the nominal trajectory at 0.13 m/s desired speed. The DDPC has a higher success rate, defined by the percentage of the models being upright, compared to the nominal trajectory (see Fig. 5.12d).

Hardware Results

The same C++ low-level controller used in simulation, with additional code to interface with Wandercraft API, is run directly on the Atalante onboard computer (i5-4300U CPU @1.90GHz with 8GB RAM at 1kHz). The DDPC planner is run on an external PC (i7-8700K CPU @3.70GHz) communicating with the onboard computer via a UDP network. To account for package delay, a segment of the desired trajectory is sent, and the low-level controller finds the closest discrete target point given the current phase variable. As a result, the low-level controller receives planned trajectory with the upcoming time stamps and could handle the delay caused by planning computation time and UDP communication. An example planned trajectory

of CoM and CoP under this setup is shown in Fig. 5.14.

Different User Settings: We conducted two sets of experiments to test the framework with different payloads and kinematics as this platform is designed to be used by different users. The data collection part for the hardware experiment is similar to that of the simulation setup for tracking performance comparison with gaits at different step lengths. The first set of experiments is conducted with the exoskeleton carrying 20 kg of payload with the link lengths of a subject of 1.74 m. DDPC planner is able to regulate the CoP position to a more centered position as depicted in Fig. 5.15a and 5.15b. A different set of hyperparameters with $T = 800$, $T_{\text{ini}} = 20$, $N = 100$, and $\delta_t = 0.015$ is used. The second experiment is conducted with a user of height 1.63 m and 52 kg with the same hyperparameter to construct the Hankel Matrix as in tracking performance case. The gait tiles for the experiment with different payloads are shown in Fig. 5.15c and 5.15d.

Uncertain User Mass: To investigate the effect of the planner's capability to handle the uncertain mass of the user or in case of the user wanting to carry additional payload, we also conducted a set of experiments with the user carrying an additional 14 kg of weight (see Fig. 5.16). We test both DDPCs with $\mathcal{G}$ constructed from data without carrying the additional weight and with $\mathcal{G}^{\text{w}}$ which is constructed with trials where the user is carrying additional weight. From the phase portrait described in Fig. 5.16a, we could see that the DDPC with $\mathcal{G}^{\text{w}}$ resembles a desired limit cycle compared to the DDPC with $\mathcal{G}$ and the nominal controller. The gait tiles for the experiments with $\mathcal{G}$ and $\mathcal{G}^{\text{w}}$ are shown in Fig. 5.16b and 5.16c, respectively. We also evaluate the cumulative tracking error over $t_{\text{total}} = 10$ s for output other than the CoM position and yaw related, denoted by $\mathbf{y}_{\text{part}}$, as the CoM positions are different across controllers and the yaw related output is not well approximated due to IMU drifting via evaluating $e = \int_0^{t_{\text{total}}} \|\mathbf{y}_{\text{part}}^{\text{act}}(t) - \mathbf{y}_{\text{part}}^{\text{des}}(t)\|^2 dt$. This value for the DDPC with $\mathcal{G}$, $\mathcal{G}^{\text{w}}$, and nominal is 0.4627, 0.4319, and 0.5181, respectively.

DDPC Summary

We successfully demonstrated the DDPC framework's application on lower-body exoskeleton, both in simulations and on hardware. Through detailed simulation analyses, the DDPC framework proved its effectiveness in stabilizing the system, surpassing traditional physics-based template models such as LIP, particularly at increased desired speeds. The framework's robustness was also validated on hardware, showcasing its ability to accommodate model discrepancies beyond the initial model

used for data collection. Furthermore, we introduced a time-varying perturbation in the simulation while updating the transition matrix online. Although further research is necessary to refine the online update process of the Hankel matrices systematically, these results underscore the framework's capacity to adapt to changing environments. However, it is crucial to note that the DDPC planner's performance is intrinsically linked to the trajectory used to construct the Hankel matrices. It can only enhance and build upon the capabilities of the nominal controller used for data collection. Moreover, considering the inherent limitations in the actuation of the CoM horizontal position, future studies will explore how to extend the proposed framework to incorporate foot placement and step timing planning to enhance stability and performance (as shown in section 5.3. Additionally, further investigation into incorporating more information on user movement would be beneficial.

5.3 Hybrid Data-Driven Predictive Control

While DDPC offers a powerful framework for trajectory generation from data, its formulation is restricted to continuous dynamics. Bipedal locomotion, however, inherently couples continuous swing-phase motions with discrete step-to-step transitions that govern stability and foot placement. For exoskeletons—where range of motion is limited—robustness depends on carefully coordinating these two levels, unlike robots with greater kinematic freedom. This motivates extending DDPC beyond purely continuous trajectory generation to explicitly incorporate step-to-step reasoning, which we term Hybrid Data-Driven Predictive Control (HDDPC).

Achieving robust exoskeleton walking requires generating stable periodic motions while simultaneously replanning foot placement online to handle disturbances and constrained environments. A common approach is to decouple footstep planning (discrete dynamics) from motion synthesis (continuous swing-phase dynamics), often relying on heuristics with simplified models [222]. Reduced-order models have been effective for capturing step-to-step (S2S) dynamics, enabling agile and robust bipedal locomotion [34]. However, their validity typically assumes lightweight legs and can break down in more dynamic or constrained settings. This motivates approaches that jointly optimize both contact schedules and continuous motion planning [223], or learn step transitions in a data-driven way [224]. What remains missing is a unified framework that integrates S2S footstep planning with continuous trajectory generation in a fully data-driven manner.

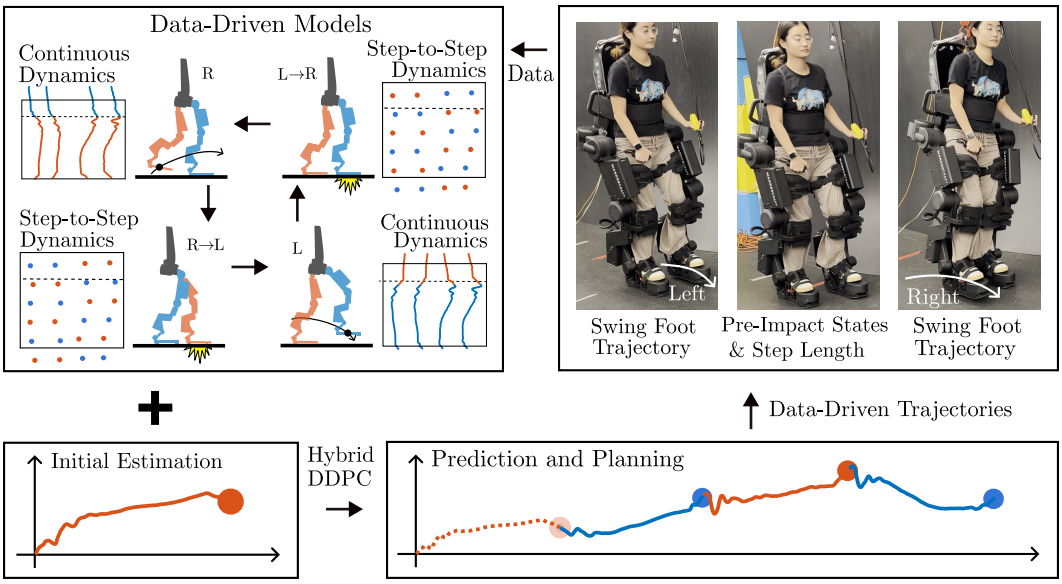


Figure 5.17: Overview of Hybrid DDPC.

In this section, we introduce HDDPC (Fig. 5.17), a framework that simultaneously plans both contact schedules and continuous-domain trajectories. Our method leverages a data-driven reduced-order model based on Hankel matrices, extending DDPC to incorporate discrete S2S dynamics. This unified approach integrates both discrete and continuous aspects of locomotion in a data-driven manner, enabling adaptation that supports robust and dynamic gait generation (Fig. 5.18). We demonstrate HDDPC on the lower-body exoskeleton Atalante, in both simulation and hardware, showing that it produces stable walking across speeds and effectively rejects disturbances. These results highlight how a hybrid data-driven formulation can enable the synthesis of robust and reactive walking gaits.

Hybrid Data-driven Model

We introduce a hybrid data-driven model for full-order system dynamics (Fig. 5.19), consisting of two components: a Hankel matrix $\mathscr{H}(\cdot)$ for continuous dynamics and a second one $\mathscr{H}^{\text{S2S}}(\cdot)$ for discrete step-to-step dynamics.

Data-driven Model with Step-to-step Dynamics

Inspired by the input-output structure of the H-LIP model in Section 2.1, we define the input $\mu_{\text{S2S}} \in \mathbb{R}^3$ and the output $\eta_{\text{S2S}} \in \mathbb{R}^2$ of the S2S date-driven model as

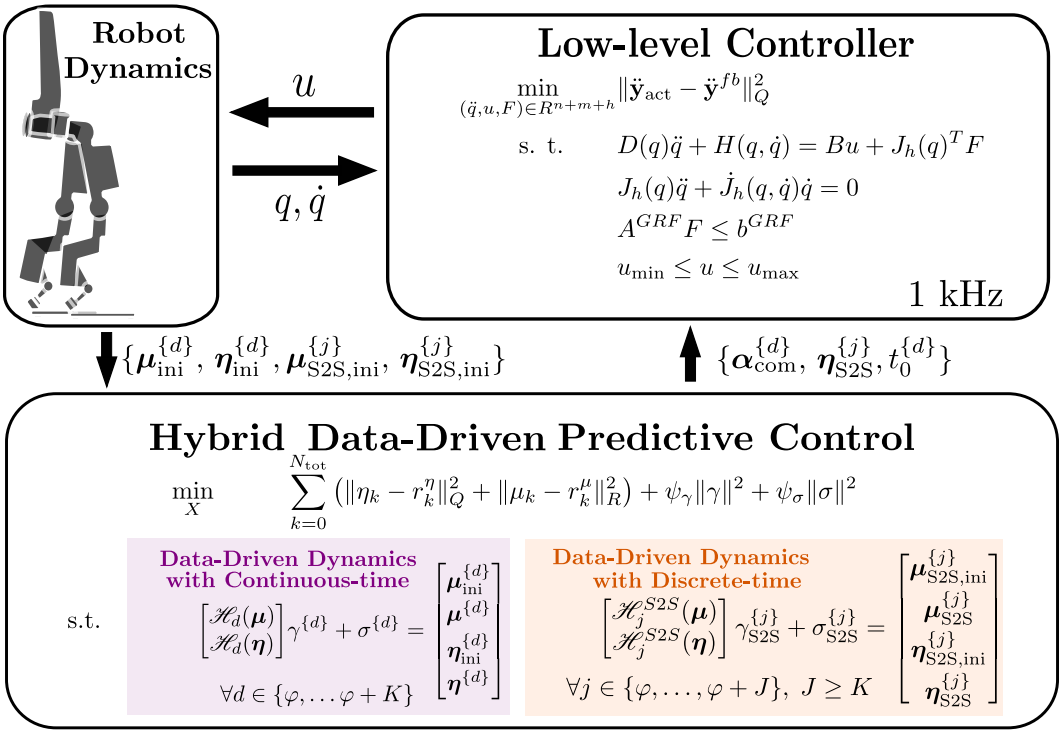


Figure 5.18: Control Overview for Hybrid Data-Driven Predictive Control (HDDPC). We utlilize a layered architecture with HDDPC planner and low-level controller.

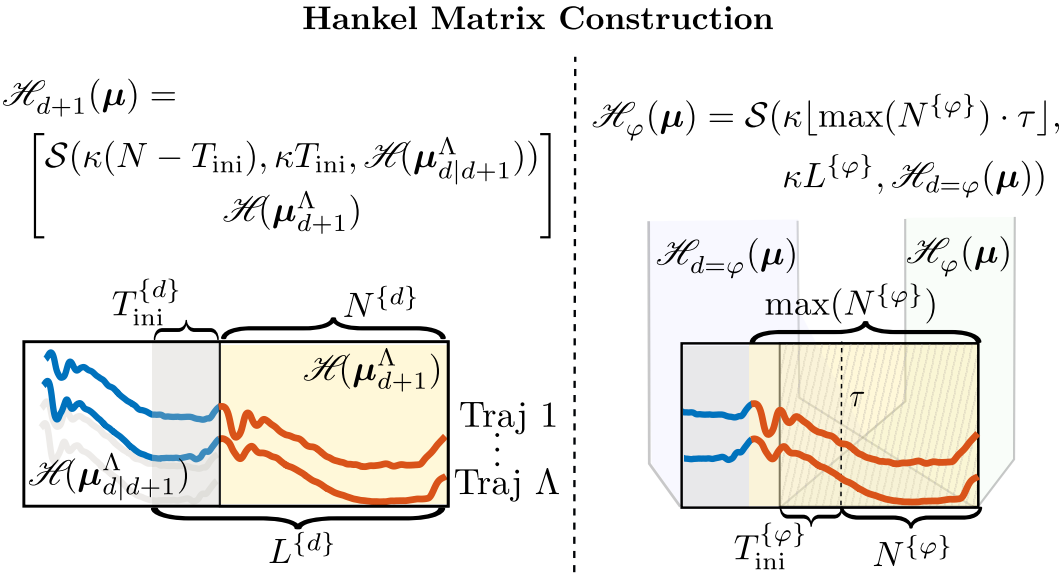


Figure 5.19: An illustration of Trajectory Hankel matrix construction for different domains.

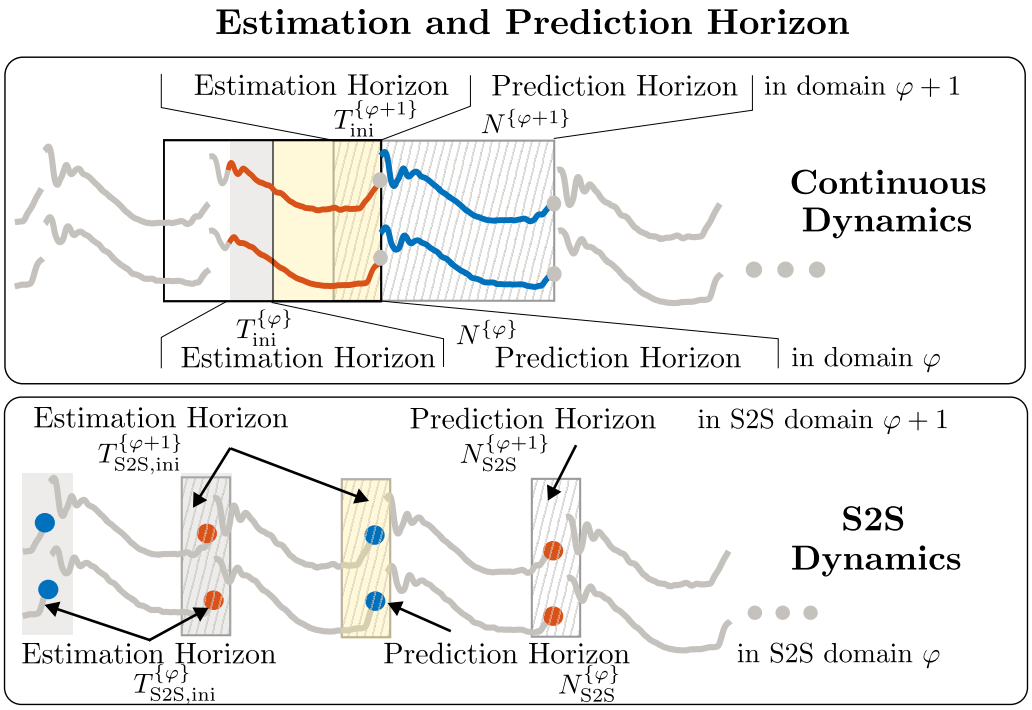


Figure 5.20: An illustration of the estimation and prediction horizon of HDDPC.

follows:

$$\boldsymbol{\mu}_{\text{S2S}} := \text{col}(\lambda^{x,y}, T_{\text{step}}) \qquad \boldsymbol{\eta}_{\text{S2S}} := (\text{p}_{\text{com}}^{x,y})^{-},$$

where $\lambda^{x,y}$ is the foot placement for the next step and T_{step} is the desired step duration, and $(\text{p}_{\text{com}}^{x,y})^{-}$ are the pre-impact CoM states. By using this I-O structure, the data-driven model captures the essential S2S dynamics that govern transitions in walking, allowing us to approximate the full system's behavior through a step-wise dynamics evolution. We denote the S2S Trajectory Hankel matrix constructed using a dataset $\mu_{\text{S2S}}^{(i)}$ that describe step-to-step transitions from domain $\mathcal{D}^d$ to $\mathcal{D}^{d+1}$ as $\mathscr{H}_d^{\text{S2S}}(\boldsymbol{\mu}_{\text{S2S}}^{\Lambda})$, where $d \in \mathbb{Z}_{\geq 0}$.

Data-driven Model with Continuous Dynamics

Drawing the inspiration from LIP model introduced in Section 2.1, the input $\mu \in \mathbb{R}^2$ and the output $\eta \in \mathbb{R}^2$ are chosen as follows:

$$\mu := \text{p}_{\text{cop}}^{x,y} \qquad \eta := \text{p}_{\text{com}}^{x,y}.$$

When considering multiple continuous domains, each domain is represented by its own Trajectory Hankel matrix, capturing the system dynamics specific to that

walking phase. Since we focus exclusively on position outputs, there are no discrete jumps in the CoM trajectory across domains in the global coordinate frame. However, each trajectory is expressed relative to the corresponding stance foot, resulting in a periodic oscillations in the x-direction and a sign flip in the y-direction as stance foot changes. To account for transitions between domains, we apply a frame transformation $\Phi(\cdot,\cdot)$ to express CoM trajectory relative to the stance foot frame of the next domain. Specifically, when transitioning from domain $\mathcal{D}^d$ to $\mathcal{D}^{d+1}$, the transformation is given by $\boldsymbol{\mu}_{d|d+1} = \Phi(\boldsymbol{\mu}_d, \lambda_d^{x,y})$, where $\lambda_d^{x,y}$ is the foot placement during the transition, measured relative to the stance frame in domain $\mathcal{D}^d$. Here $\boldsymbol{\mu}_d$ represents the CoM trajectory in domain $\mathcal{D}^d$ relative to its stance foot frame, while $\boldsymbol{\mu}_{d|d+1}$ denotes the same trajectory but relative to the stance foot frame in $\mathcal{D}^{d+1}$. For a trajectory dataset $\boldsymbol{\mu}^\Lambda$ containing multiple domains, we can construct the Trajectory Hankel matrix $\mathscr{H}_{d+1}(\boldsymbol{\mu}) \in \mathbb{R}^{\kappa L \times \Lambda}$ that describe the dynamics for domain $\mathcal{D}^{d+1}$ as follows:

$$\mathscr{H}_{d+1}(\boldsymbol{\mu}) = \begin{bmatrix} \mathcal{S}(\kappa(N - T_{\text{ini}}), \kappa T_{\text{ini}}, \mathscr{H}(\boldsymbol{\mu}^\Lambda_{d|d+1})) \\ \mathscr{H}(\boldsymbol{\mu}^\Lambda_{d+1}) \end{bmatrix}, \tag{5.15}$$

where $\boldsymbol{\mu}_d^\Lambda$ denotes the trajectory data set captured in domain d and $\mathscr{H}(\boldsymbol{\mu}_d^\Lambda), \mathscr{H}(\boldsymbol{\mu}_{d|d+1}^\Lambda) \in \mathbb{R}^{\kappa N \times \Lambda}$ and to streamline the partitioning, we define a selection operator $\mathcal{S}(\cdot,\cdot,\cdot)$ as follows $U_p = \mathcal{S}(0, \kappa T_{\text{ini}}, \mathscr{H}(\boldsymbol{\mu}^\Lambda)),\ U_f = \mathcal{S}(\kappa T_{\text{ini}}, \kappa N, \mathscr{H}(\boldsymbol{\mu}^\Lambda))$. The first entry indicates the starting point of the selection, the second entry indicates the size of row block being selected. We can partition $\mathscr{H}(\boldsymbol{\eta}^\Lambda)$ in a similar fashion with νT_{ini} and νN. Since all trajectories are transformed with respect to the stance foot frame in domain $\mathcal{D}^{d+1}$, continuity in the transition between adjacent domains is thereby guaranteed in Trajectory Hankel matrix construction. Similarly, $\mathscr{H}_{d+1}(\boldsymbol{\eta})$ can be constructed as illustrated in (5.15).

Hybrid Data-Driven Predictive Control

Next, we present a planning problem over continuous trajectory for $K+1$ domains indexed by d, where $d \in \mathbb{Z}_{\geq 0}$ and J step-wise dynamics indexed by j, where j should be the corresponding transition between d-th and $d+1$-th domain. Each domain has its own continuous Trajectory Hankel matrices and each step-wise transition is associated with its corresponding S2S Trajectory Hankel matrices.

Prediction Horizon and Estimation Horizon The total planning horizon over the $K+1$ domains is defined to be $N_{\text{tot}} = \sum_{d=\varphi}^{\varphi+K} N^{\{d\}}$, where $N^{\{d\}}$ represents the prediction horizon for d-th domain. When planning over $K+1$ domains, for

all the upcoming domains (i.e., $d > \varphi$), this prediction horizon is fixed (i.e., $L^{\{d\}} = T_{\text{ini}}^{\{d\}} + N^{\{d\}},\ \forall d > \varphi$), which is the maximum possible prediction horizon determined by the construction. But for current domain $d = \varphi$, we are shrinking the prediction horizon as we move along the domain.

Data-driven Constraint To construct the data-driven constraints similar to (5.10), we select the appropriate portion of the Trajectory Hankel matrices and compose it to have the structure in (5.15). Analogous to (5.11), T_{ini} is for determining the initial condition estimation. However, since for current domain the prediction horizon is changing, we have to partition the Trajectory Hankel matrix differently. Specifically, we need to select the appropriate $\kappa(T_{\text{ini}} + N^{\{\varphi\}})$ rows from $\mathscr{H}_{\varphi}(\boldsymbol{\mu})$, starting from $\kappa\lfloor \max(N^{\{\varphi\}}) \cdot \tau \rfloor$-th row, where $\tau \in [0, 1]$ indicates a phasing variable evolve from 0 to 1 on each step. Additionally, $\max(N^{\{\varphi\}})$ denotes the maximum prediction horizon for the current domain φ, which happens when $\tau = 0$.

With a slight abuse of notation, we define $\mathscr{H}_{\varphi}(\boldsymbol{\mu})$ to be

$$\mathscr{H}_{\varphi}(\boldsymbol{\mu}) = \mathcal{S}(\kappa\lfloor \max(N^{\{\varphi\}}) \cdot \tau \rfloor, \kappa L^{\{\varphi\}}, \mathscr{H}_{d=\varphi}(\boldsymbol{\mu})),$$

where $\mathscr{H}_{d=\varphi}(\boldsymbol{\mu})$ indicates the Trajectory Hankel matrix constructed analogous to (5.15) when $d = \varphi$, while $\mathscr{H}_{\varphi}(\boldsymbol{\mu})$ indicates the extracted portion from $\mathscr{H}_{d=\varphi}(\boldsymbol{\mu})$ when τ evolves during the gait in current domain φ.

Since we are dealing with nonlinear systems, we added slack variable $\sigma^{\{d\}}$ to account for noise, unknown requirement for persistently exciting order and ensure numerical feasibility:

$$\begin{bmatrix} \mathscr{H}_d(\boldsymbol{\mu}) \\ \mathscr{H}_d(\boldsymbol{\eta}) \end{bmatrix} \gamma^{\{d\}} + \sigma^{\{d\}} = \begin{bmatrix} \boldsymbol{\mu}_{\text{ini}}^{\{d\}} \\ \boldsymbol{\mu}^{\{d\}} \\ \boldsymbol{\eta}_{\text{ini}}^{\{d\}} \\ \boldsymbol{\eta}^{\{d\}} \end{bmatrix}, \tag{5.16}$$

where $\boldsymbol{\mu}_{\text{ini}}^{\{d\}}$ and $\boldsymbol{\eta}_{\text{ini}}^{\{d\}}$ are the T_{ini} data points from the past trajectory data. The past trajectory data is either in the current domain only, or the portion of the previous domain can be collected in the past trajectory data set as the estimation horizon ,T_{ini}, is fixed. Notably, the portion of the past trajectory data is subject to the frame transformation, $\Phi(\cdot, \cdot)$, in the case that the data from previous domain is employed in $\boldsymbol{\mu}_{\text{ini}}^{\{d\}}$ and $\boldsymbol{\eta}_{\text{ini}}^{\{d\}}$ construction.

We further note that the trajectory planning is over both discrete points $\{\boldsymbol{\mu}^{\{d\}}, \boldsymbol{\eta}^{\{d\}}\}$ and Bezier coefficient $\boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{d\}}$ so that Bézier function, $\text{Bez}(t_k^{\{d\}}, \boldsymbol{\alpha}^{\{d\}}, T^{\{d\}})$, and its

derivative dBez$(t_k^{\{d\}}, \boldsymbol{\alpha}^{\{d\}}, T^{\{d\}})$, where $t_k^{\{d\}} = T^{\{d\}} \cdot \tau_k^{\{d\}}$ and $\tau_k^{\{d\}} = \{0, \delta_\tau^{\{d\}}, \dots, 1\}$, is used to obtain a smooth trajectory for low level controller. Note that δ_τ is a sampling period that help us keep the same shape of Trajectory Hankel matrix across datasets with different duration for the same phase.

Now we introduce our **Hybrid Data-Driven Predictive Controller (HDDPC)**:

$$
\begin{aligned}
\min_{X} \quad & \sum_{k=0}^{N_{\text{tot}}} \left(\|\eta_k - r_k^\eta\|_Q^2 + \|\mu_k - r_k^\mu\|_R^2 \right) + \psi_\gamma \|\gamma\|^2 + \psi_\sigma \|\sigma\|^2 \\
\text{s.t} \quad & \begin{bmatrix} \mathscr{H}_d(\boldsymbol{\mu}) \\ \mathscr{H}_d(\boldsymbol{\eta}) \end{bmatrix} \gamma^{\{d\}} + \sigma^{\{d\}} = \begin{bmatrix} \boldsymbol{\mu}_{\text{ini}}^{\{d\}} \\ \boldsymbol{\mu}^{\{d\}} \\ \boldsymbol{\eta}_{\text{ini}}^{\{d\}} \\ \boldsymbol{\eta}^{\{d\}} \end{bmatrix} \\
& \forall d \in \{\varphi, \dots \varphi + K\} \\
& \begin{bmatrix} \mathscr{H}_j^{S2S}(\boldsymbol{\mu}) \\ \mathscr{H}_j^{S2S}(\boldsymbol{\eta}) \end{bmatrix} \gamma_{\text{S2S}}^{\{j\}} + \sigma_{\text{S2S}}^{\{j\}} = \begin{bmatrix} \boldsymbol{\mu}_{\text{S2S,ini}}^{\{j\}} \\ \boldsymbol{\mu}_{\text{S2S}}^{\{j\}} \\ \boldsymbol{\eta}_{\text{S2S,ini}}^{\{j\}} \\ \boldsymbol{\eta}_{\text{S2S}}^{\{j\}} \end{bmatrix} \\
& \forall j \in \{\varphi, \dots, \varphi + K, \dots, \varphi + J\},\ J \geq K \\
& \mu_k^{\{d\}} \in \text{Support Polygon} \\
& \eta_k^{\{d\}} = \text{Bez}(t_k^{\{d\}}, \boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{d\}}, T_{\text{step}}^{\{d\}}) \\
& \eta_{\text{S2S}}^{\{j\}} = \text{Bez}(T_{\text{step}}^{\{j\}}, \boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{j\}}, T_{\text{step}}^{\{j\}}) \\
& \text{v}_{\text{com}}(t_k^{\{d\}}) = \text{dBez}(t_k^{\{d\}}, \boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{d\}}, T_{\text{step}}^{\{d\}}) \in [\text{v}_{\text{com}}^{\text{min}}, \text{v}_{\text{com}}^{\text{max}}] \\
& \text{p}_{\text{com}}(t_0) = \text{Bez}(t_0, \boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{\varphi\}}, T_{\text{step}}^{\{\varphi\}} - t_0) \\
& \text{v}_{\text{com}}(t_0) = \text{dBez}(t_0, \boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{\varphi\}}, T_{\text{step}}^{\{\varphi\}} - t_0),
\end{aligned}
\tag{5.17}
$$

where Q and R are positive definite matrices, ψ_γ and ψ_σ are positive weighting factors, and the decision variables $X^{\{d\}}$, $X_{\text{S2S}}^{\{j\}}$, and X are defined as

$$
\begin{aligned}
X^{\{d\}} &= \text{col}(\boldsymbol{\alpha}_{\text{com}^{x,y}}^{\{d\}}, \boldsymbol{\mu}^{\{d\}}, \boldsymbol{\eta}^{\{d\}}, \boldsymbol{\gamma}^{\{d\}}, \boldsymbol{\sigma}^{\{d\}}) \\
X_{\text{S2S}}^{\{j\}} &= \text{col}(\boldsymbol{\mu}_{\text{S2S}}^{\{j\}}, \boldsymbol{\eta}_{\text{S2S}}^{\{j\}}, \boldsymbol{\gamma}_{\text{S2S}}^{\{j\}}, \boldsymbol{\sigma}_{\text{S2S}}^{\{j\}}) \\
X &= \text{col}(X^{\{\varphi\}}, \dots, X^{\{\varphi+K\}}, X_{\text{S2S}}^{\{\varphi\}}, \dots, X_{\text{S2S}}^{\{\varphi+J\}}),
\end{aligned}
$$

respectively, and the reference trajectory for μ and η are denoted by r^μ and r^η, respectively.

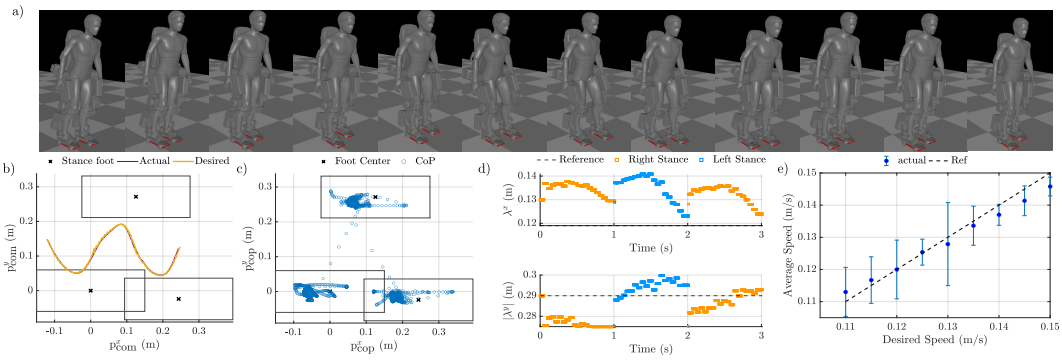


Figure 5.21: Trajectory and tracking results from HDDPC Planner a) Gait tiles for the resulting trajectory b) Desired CoM trajectory from HDDPC planner and actual evolving CoM trajectory in simulation in global coordinate c) actual CoP in simulation d) planned foot placement location for three steps. e) the tracking performance for the HDDPC planner under different desired speed vs. actual realized average speed.

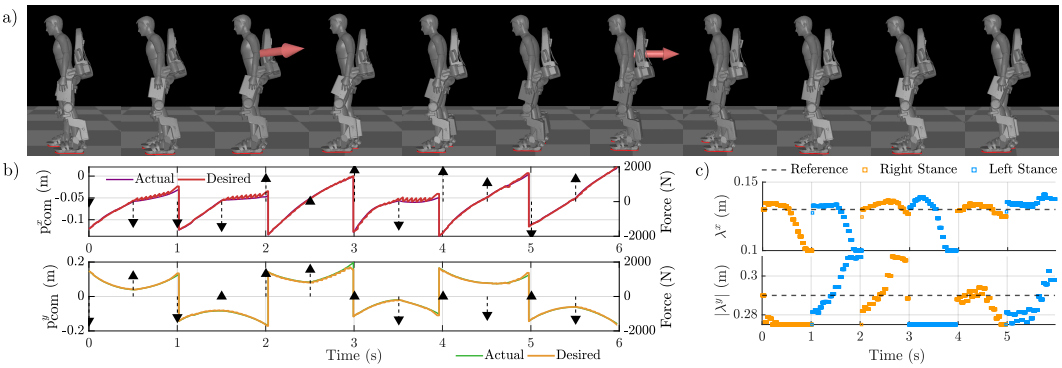


Figure 5.22: Recovery performance of HDDPC under random perturbations. a) Gait tiles of simulated perturbation recovery b) CoM trajectory under random perturbation force. The time, direction, and magnitude of the perturbation is represented by the black arrows. The perturbation force is applied as a 10 ms impulse with magnitude range between 1400-2000 N c) The corresponding step location planned by the planner. The desired step size is indicated by the dashed line.

Layered Data-Driven Control Framework

In this subsection, we introduce the remaining components and details for practically implementing the layered control framework (Fig. 5.18) used to realize locomotion on the lower-body exoskeleton Atalante. We use a version of the model with the user height of 1.65 m and a total weight of 136 kg, which includes both the user's weight and the device.

Trajectory Hankel Matrix and Planner Parameters: Trajectory Hankel matrices are constructed from data collected over five gait cycles (10 steps), with step lengths

varying between 0.11 m and 0.15 m and step durations ranging from 0.9 s to 1.1 s. For the continuous dynamics Hankel matrix, data is sampled at 50 evenly spaced points across the gait cycle (i.e., sampling period δ_τ is 0.02 for normalized step time). We set the estimation horizon that is used to determine system evolution for the trajectory Hankel matrix as $T_{\text{ini}} = 4$. The same data sequence is used to construct the S2S trajectory Hankel matrix. Specifically, we maintain one Hankel matrix for left stance, one for right stance, and two S2S matrices representing transitions from right-to-left (R2L) and left-to-right (L2R) dynamics. In total, we plan for trajectory over three domains (e.g., a Left-Right-Left sequence). In order to ensure a faster planning rate, we select a different δ_τ for the second and third domain ($\delta_\tau = 0.08$) to reduce the number of decision variables required in (5.17).

The planning problem is solved using IPOPT [225] with its C++ interface and HSL MA97 solver at 20-40 Hz depending on the planning horizon and application scenario. We terminate the planner at a pre-specified maximum wall clock time and employ the feasible trajectories only (i.e., with constraint satification under tolerance). The planned trajectory, specifically $\{\boldsymbol{\alpha}_{\text{com}}^{\{d\}}, \boldsymbol{\eta}_{\text{S2S}}^{\{j\}}, t_0^{\{d\}}\}$ is sent to low-level controller. When the planner is deployed, we do not update the step duration for the current domain. When an impact occurs, the step duration planned for the next domain will be used to evaluate the phasing variable.

Output Synthesis: The desired walking behavior is encoded by the task space output $\mathbf{y} = \mathbf{y}_{\text{act}} - \mathbf{y}_{\text{des}}$, where $\mathbf{y}^{\text{act}} \in \mathbb{R}^{12}$ and $\mathbf{y}^{\text{des}} \in \mathbb{R}^{12}$ and are chosen to be the following

$$\mathbf{y}^{\text{act}} = \begin{bmatrix} \mathrm{p}_{\text{com2st}}^{x,y,z}(q) & \phi_{\text{pelv}}^{x,y,z}(q) & \mathrm{p}_{\text{sw}}^{x,y,z}(q) & \phi_{\text{sw}}^{x,y,z}(q) \end{bmatrix}$$

$$\mathbf{y}^{\text{des}} = \begin{bmatrix} \mathrm{p}_{\text{com2st}}^{x,y}(\boldsymbol{\alpha}_{\text{com}^{x,y}}) & \mathrm{p}_{\text{com2st}}^{z}(\boldsymbol{\alpha}) & \\ \phi_{\text{pelv}}^{x,y,z}(\boldsymbol{\alpha}) & \mathrm{p}_{\text{sw}}^{x,y,z}(\boldsymbol{\alpha}, \lambda^{x,y}) & \phi_{\text{sw}}^{x,y,z}(\boldsymbol{\alpha}) \end{bmatrix}.$$

The desired CoM position, foot placement, and step duration that determine the phasing variable is generated from the hybrid DDPC planner, and the other desired components are taken as Bézier polynomials with the coefficient matrix of $\boldsymbol{\alpha}$. More specifically, the coefficients of pelvis orientation ϕ_{pelv}, swing foot orientation ϕ_{sw}, z-height of CoM, and swing foot trajectory are fixed. The swing foot x, y trajectories are determined by Bézier polynomials connecting the swing foot position at the beginning of the domain (i.e., post-impact state) and the desire foot targets, i.e., $\mathrm{p}_{\text{sw}}^{x,y}(\tau) = (1 - \beta(\tau))\, \mathrm{p}_{\text{sw}}(q^+) + \beta(\tau)\, \lambda^{x,y}$, where β is a phase-based weighting function.

Whole-body Controller: We apply a task-space based QP controller solved via OSQP

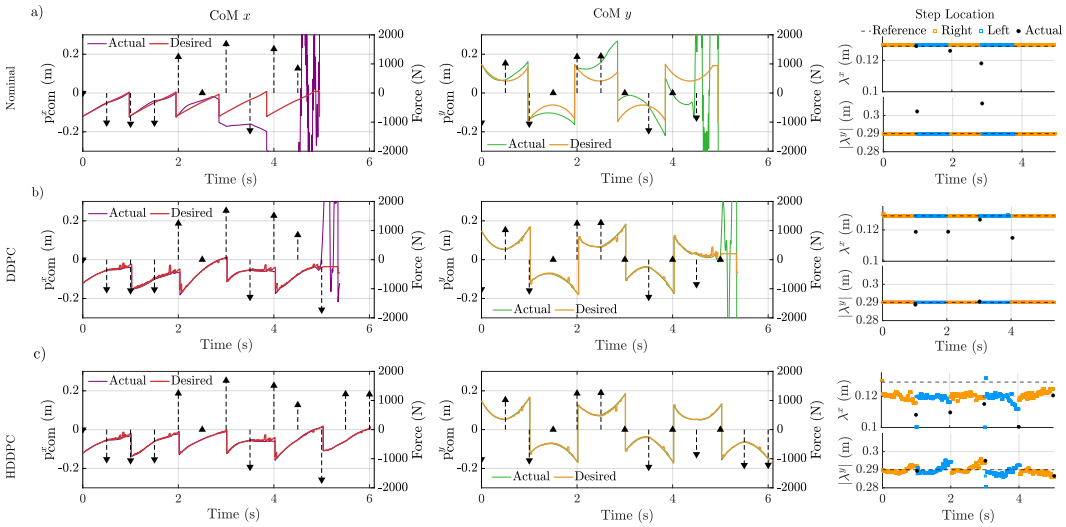


Figure 5.23: Perturbation Recovery Comparison: a) Nominal: The controller follows a fixed reference trajectory with a predetermined step size and step duration. b) DDPC: Functionally equivalent to HDDPC but with a fixed contact schedule. The upper and lower bounds of the step size and the step duration are constrained to match those used in the nominal reference trajectory. c) HDDPC: The proposed control framework.

[226] with a maximum iteration of 200 running at 1kHz, formulated as following

$$\begin{aligned} \min_{(\ddot{q},u,F)\in R^{n+m+h}} \quad & \|\ddot{\mathbf{y}}_{\text{act}} - \ddot{\mathbf{y}}^{fb}\|^2_Q && (5.18)\\ \text{subject to} \quad & (2.5), (2.6) && \text{(dynamics)}\\ & A^{GRF}F \leq b^{GRF} && \text{(contact)}\\ & u_{\min} \leq u \leq u_{\max} && \text{(torque limit)} \end{aligned}$$

where $\ddot{\mathbf{y}}^{fb} = -\mathcal{K}_p\mathbf{y} - \mathcal{K}_d\dot{\mathbf{y}}$, A^{GRF} describes friction cone constraint and zmp constraint for each contact frame.

Experimental Validations

In this section, we present the numerical simulation in MuJoCo [206] and hardware experiment results to validate the effectiveness of our proposed framework. The experiment video can be found in [227].

Tracking Performance (Simulation): We first evaluate the planner's capability to realize stable walking. An example gait tiles from the MuJoCo simulation is shown in Fig. 5.21a. An example planned and realized CoM trajectory, actual CoP position is shown in Figs. 5.21b and 5.21c, respectively. The planned step location are shown

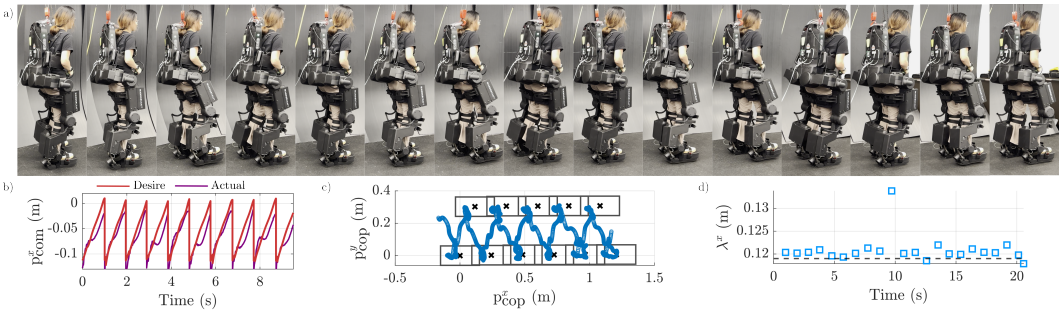


Figure 5.24: HDDPC hardware results. a) Gait tiles of walking on hardware together with b) CoM trajectory, c) CoP trajectory d) Planned foot placement.

in Fig. 5.21d over three steps. Additionally, we assess the tracking performance of the hybrid DDPC controller under different forward velocities, with the mean and standard deviation of the realized speed over a 10-second window plotted in Fig. 5.21e. As the desired speed increases, tracking error also increases, but the realized walking remains stable.

Perturbation Recovery (Simulation): We further evaluate the framework's robustness by introducing external perturbations (see Fig. 5.22a). Impulses were applied to the torso at intervals of 0.5 s, lasting for 10 ms, with random perturbation directions selected from 45-degree intervals in the horizontal plane (i.e., 0°, 45°, etc.) and varying maximum magnitudes ($1400 - 2000N$). Figure 5.22b illustrates the continuous CoM trajectory, highlighting the moments when perturbations occurred. The top-down view of the step pattern in Fig. 5.22c shows the controller's adjustments in response to each disturbance.

We compared our proposed framework with two baselines. The first is a nominal approach that tracks a fixed reference trajectory with a predetermined step size and step duration. While a desired step duration is set, transitions to the next domain are enforced based on a combination of a minimum phasing variable threshold and impact events detected by the force sensor, a mechanism shared across all three comparison cases discussed here. Additionally, we include another baseline which we referred to as DDPC. We argue that this is conceptually similar to our previous DDPC results [228] but implemented differently. This is essentially an HDDPC variant with a fixed contact schedule, where the lower and upper bounds are set to match those of the nominal approach. This setup essentially allows for replanning only over the CoM trajectory. Both DDPC and HDDPC use the same set of hyperparameters but with the only difference being the restricted contact schedule bound.

We applied the same perturbation sequence by fixing the random seed number. The desired step duration was 1 s, but the nominal controller impacted earlier and failed to maintain the desired step size very well. As shown in Fig. 5.23a, the robot began accumulating significant tracking errors around 3 s and failed shortly thereafter. In contrast, DDPC (Fig. 5.23b) maintained system stability until 5 s before eventually failing, while demonstrating better adherence to the desired step duration and step size compared to the nominal approach. Meanwhile, HDDPC (Fig. 5.23c) successfully rejected all disturbances.

Hardware Results: In hardware experiments (see Fig. 5.24a), we first collected the data set from the exo when subject is in it. During the data collection, we varied the foot step length for each gait to enable the foot step adjustment capability in HDDPC. Based on the collected data set, HDDPC planner was deployed to synthesize CoM motion and foot placement in the x direction for the exoskeleton, running on an external PC (Intel i9-14900K CPU), which communicated with exo through a UDP network. The HDDPC planner performed trajectory replanning of the desired CoM trajectory and foot placement at the beginning of each domain. While the model parameters in the low-level controller did not incorporate detailed subject-specific information, the HDDPC implementation with the collected data set successfully demonstrated stable locomotion on exo with human subject without compromising stability. The detailed tracking performance is illustrated in Fig. 5.24. The evolution of CoM trajectories effectively tracked the desired CoM trajectories (see. Fig. 5.24b), while the CoP was successfully regulated within each stance foot as described in Fig. 5.24c. Figure 5.24d further illustrates that the HDDPC actively modulated foot step length to track the desired trajectories while maintaining locomotion stability. Within this capacity of HDDPC framework, the exo demonstrated the stable bipedal locomotion.

HDDPC Summary

We presented a novel HDDPC framework that integrates contact scheduling with continuous domain trajectory planning for lower-body exoskeletons. Through both simulation and hardware experiments, the HDDPC framework demonstrated its capability to achieve stable and adaptive walking. In simulations, the integration of S2S dynamics and continuous domain trajectory enhanced the exoskeleton's reactive capabilities, enabling effective recovery from external disturbances. The hardware results confirmed the effectiveness of the HDDPC controller. Future work will focus

on running the planner faster online, enhancing the framework's ability to adapt to time-varying perturbations by updating the Hankel matrix online, and demonstrating robustness across subject-to-subject variability. Additionally, we aim to extend application to more complex scenarios, such as stair climbing and other challenging setups in dynamic environments.

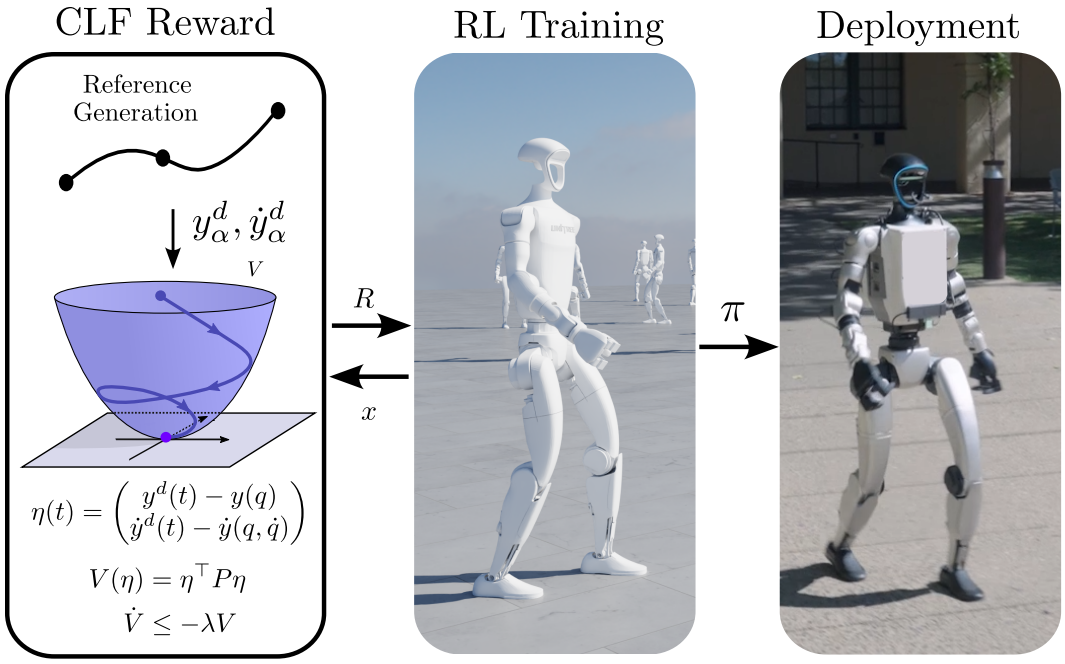


Figure 5.25: Overview of our approach. A reference generator produces target trajectories, which are used to construct a CLF-based reward. An RL policy is trained in simulation with this reward and deployed on a real humanoid robot.

5.4 CLF-RL: Control Lyapunov Guided Reinforcement Learning

So far, robustness has been pursued through offline analysis of trajectories and online replanning via predictive control, both of which depend strongly on explicit system models. A complementary strategy is to leverage reinforcement learning (RL), which can generalize across disturbances and modeling errors by training policies in diverse simulated environments.

With recent advances in computation and GPU-parallel physics simulators, RL has emerged as a practical option for locomotion control: policies can be trained offline in massively parallel simulations and then executed on hardware with lightweight inference at runtime [7, 8, 9, 10, 11]. However, applying RL to bipedal locomotion remains challenging due to its reliance on heuristic reward design, which is tedious to construct, sensitive to tuning, and, if poorly shaped, can lead to unstable gaits, prolonged training, and poor sim-to-real transfer. One way to mitigate these issues is to embed model-based structure into the reward, using reference trajectories to guide policy learning. This approach provides the policy with physically meaningful targets while reducing reliance on ad hoc reward shaping. This idea has been explored

using both reduced order models [229, 230] and full-order models within the HZD framework [231]. [232] proposes a different way to incorporate a LIP controller into RL by generating desired footstep locations during training and using it to provide feedback control for rough terrain.

While prior model-guided approaches improve structure during training, they typically reward only the instantaneous tracking error, treating two states with the same error magnitude equally—even if one is actively converging toward the reference and the other is diverging. This can cause policies to undervalue transient corrections. To address these limitations, we propose a reward shaping framework built on control Lyapunov functions (CLFs), a fundamental tool in nonlinear control theory for generating certifiably stable controllers [233, 234, 235]. CLFs have previously been used in bipedal locomotion, often in conjunction with HZD, to formally guarantee the stability of periodic gaits [43] and have also been integrated into learning-based methods in other domains [236, 237]. For instance, [238] introduces a reward-reshaping method that incorporates a candidate CLF for fine-tuning policies using minimal hardware data.

Our approach embeds both the CLF and its associated decrease condition directly into the RL reward, where the decrease condition encodes a desired minimal convergence rate, providing meaningful intermediate rewards whenever the error is being reduced, similar to potential-based reward shaping, which has been shown to be more robust to scaling and hence simplify tuning [239]. This yields a reward component that is principled, easy to integrate, and less sensitive to manual reward balancing than conventional tracking formulations.

Building on these properties, we propose a structured reward shaping approach that combines model-based reference planning with CLF-inspired objectives to guide policy learning toward stable and robust behaviors (Fig. 5.25). We use two complementary planners for generating velocity-conditioned reference trajectories: (1) a reduced-order linear inverted pendulum (LIP) model for online generation, and (2) a precomputed hybrid zero dynamics (HZD) gait library from full-body offline trajectory optimization. Both produce periodic orbits but differ in fidelity and real-time suitability. From these trajectories, we construct a CLF $V = \eta^\top P \eta$, where η denotes the output tracking error. Once a reference trajectory is available, implementing the CLF decrease condition requires minimal effort: the tuning process is straightforward, involving far fewer reward terms than conventional rewards and resembling gain tuning in a tracking controller. The CLF and its associated stability

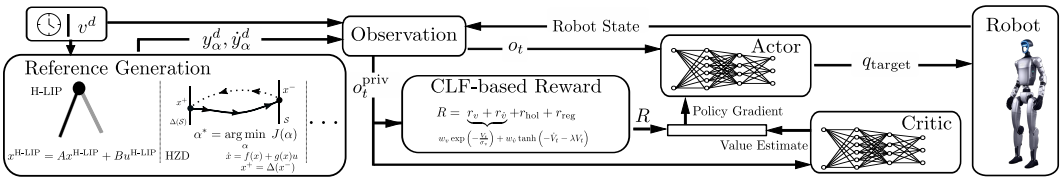


Figure 5.26: Overview of the proposed CLF-guided reinforcement learning framework. A desired velocity v^d is passed to a reference generator (e.g., H-LIP or HZD) to produce targets $y_\alpha^d, \dot{y}_\alpha^d$. These, along with the robot state and privileged variables o_t^{priv}, are used to compute a CLF-based reward. The actor-critic policy is trained with this reward and outputs joint targets q_{target} for the robot.

condition is embedded into the RL reward to shape the policy to promote stable behaviors during training. We validate our approach through both simulation and hardware experiments on the Unitree G1 humanoid robot. Results demonstrate that CLF-based reward shaping improves tracking performance at high velocities, reduces variance under randomized model perturbations, and enhances robustness compared to standard RL baselines—offering a promising path toward more robust and theoretically grounded locomotion learning.

5.5 Reference-Guided Reward Shaping

We consider the reinforcement learning (RL) problem of learning a locomotion policy $\pi_\theta(a_t|o_t)$ that maps proprioceptive observations o_t to actions a_t. The policy is trained using Proximal Policy Optimization (PPO) [240], a widely used actor-critic method that optimizes a clipped surrogate objective with entropy regularization. Instead of relying solely on sparse or handcrafted rewards, we embed model-based prior knowledge into the reward through a control Lyapunov function (CLF), promoting stability with respect to a reference trajectory. The overall framework, including reference generation and CLF-based reward shaping, is illustrated in Fig. 4.12.

Reference Trajectory Generation

Our approach assumes access to a nominal reference trajectory that encodes a periodic or quasi-periodic walking motion given a desired velocity. In this work, we focus on two sources of reference trajectories: the H-LIP model and full-order HZD optimization. However, the framework is agnostic to the origin of the trajectory. The details of the HlLIP generation has been introduced in Ch. 2. To obtain HZD

trajectories, we use the generic form specified in earlier section (2.26). In our implementation, τ is time-based, however, a common alternative is a state-based phase variable, such as the horizontal hip displacement, which ensures monotonicity. This setup guarantees that the Bézier curve is evaluated at the correct phase value during optimization. The optimized trajectory specifies desired joint or end-effector positions and velocities as functions of phase over a single step with one stance leg. Once a single step is solved, the solution is symmetrically remapped to produce a full gait cycle with alternating stance legs, yielding reference trajectories for the complete walking motion.

CLF-Based Reward Terms

We denote the output error between the current state and reference trajectory as

$$\eta(t) = \begin{pmatrix} y^d(t) - y(q) \\ \dot{y}^d(t) - \dot{y}(q, \dot{q}) \end{pmatrix}$$

. Based on this tracking error, we construct a control Lyapunov function as:

$$V(\eta) = \eta^\top P \eta,$$

. This formulation implicitly assumes a double integrator reference model for the output dynamics, allowing us to define a Lyapunov function without explicitly computing Lie derivatives $L_f V$ or $L_g V$ for the full-order system. The matrix $P \succ 0$ is the unique solution to the Continuous-Time Algebraic Riccati Equation (CARE) corresponding to this linearized model.

To avoid explicitly computing $\dot{\eta}$ and $\dot{V}$, we approximate the Lyapunov derivative using finite differences:

$$\dot{V}_t \approx \frac{V_{t+1} - V_t}{\Delta t}, \quad \text{with} \quad V_t = V\big(\eta(t)\big).$$

We define a CLF tracking reward:

$$r_v = w_v \exp\left(-\frac{V_t}{\sigma_v}\right) \qquad \text{(CLF Tracking)}$$

where $\sigma_v = \mu_{\max}(P)\, \eta_{\max}^2$, with $\mu_{\max}(P)$ denoting the maximum eigenvalue of the CLF matrix P, $\eta_{\max}$ representing an empirical bound on the tracking error.

We consider two versions of the CLF decay condition: a smooth formulation using the hyperbolic tangent, and a clipped version:

$$r_{\dot{v}}^{\tanh} = w_{\dot{v}} \tanh\left(-\left(\dot{V}_t + \lambda V_t\right)\right) \qquad \text{(CLF Decay: Tanh)}$$

$$r_{\dot{v}}^{\text{clip}} = w_{\dot{v}} \cdot \max\left(\min\left(\frac{\dot{V}_t + \lambda V_t}{\sigma_{\dot{v}}}, 1\right), 0\right), \qquad \text{(CLF Decay: Clipped)}$$

where $\sigma_{\dot{v}} = 2||P||\eta_{\max}\dot{\eta}_{\max} + \lambda\mu_{\max}\eta_{\max}^2$ and $\lambda > 0$ specifying the desired CLF decay rate. The tanh-based reward provides smooth gradients and naturally saturates large violations, but can overly dominate the overall reward depending on the relative weight between r_v and $r_{\dot{v}}$. In contrast, the clipped version is less sensitive to weighting but requires careful normalization to ensure meaningful reward shaping, especially under significant domain randomization.

Other Reward Terms

In addition to the CLF terms, we incorporate several auxiliary rewards commonly used in RL for legged system:

Stance Foot Holonomic Reward: To enforce stance-foot holonomic constraints, we define a combined reward:

$$r_{\text{hol}} = w_{\text{hpos}} \exp\left(-\frac{\|p_{\text{st}} - p_{\text{st}}^0\|}{\sigma_p}\right) + w_{\text{hvel}} \exp\left(-\frac{\|v_{\text{st}}\|}{\sigma_v}\right),$$

where p_{st} and p_{st}^0 are the current and the initial stance foot positions when it enter the domain, v_{st} is the stance foot velocity, and $w_{\text{hpos}}, w_{\text{hvel}}$ are weighting terms.

Regularization Term: To discourage excessive control effort, abrupt actions, and joint limit violations, we include a regularization reward:

$$\begin{aligned} r_{\text{reg}} = & -w_u\|u_t\|^2 - w_{\Delta a}\|a_t - a_{t-1}\|^2 \\ & - w_{q_{\text{limit}}} \left\|\max\left(0, q_{\min} - q_t\right) + \max\left(0, q_t - q_{\max}\right)\right\|_1 . \end{aligned}$$

where each term is weighted by its corresponding coefficient to balance control effort, motion smoothness, and adherence to joint limits.

Total Reward

The final reward used for policy training is the weighted sum of all components:

$$R = r_v + r_{\dot{v}} + r_{\text{hol}} + r_{\text{reg}},$$

This reward structure encodes both task performance and physical feasibility, allowing the policy to balance tracking, stability, and regularization objectives. By combining model-based references (e.g., from reduced-order planning or offline trajectory optimization) with CLF-inspired reward shaping and constraint-aware regularization, our framework embeds control-theoretic structure into the reinforcement learning process. This guides training toward stable, robust, and physically plausible locomotion, while preserving the flexibility and adaptability of model-free RL.

5.6 CLF-RL Implementation

We demonstrate our framework on the 29-DoF Unitree G1 humanoid robot, using 21 actuated degrees of freedom for motion planning. The hand and wrist joints are fixed during training and controlled with zero desired position and velocity during deployment.

Reference Generation: We use the same set of end-effector and joint positions for both HZD- and H-LIP-based trajectories. The full reference output $y_\alpha^d(t) \in \mathbb{R}^{n_y}$, with $n_y = 21$ is structured as follows:

$$y_\alpha^d = \begin{pmatrix} p_{\text{com}}^d & \phi_{\text{pelvis}}^d & p_{\text{sw}}^d & \theta_{\text{sw}}^d & q_{\text{shoulder}}^d & q_{\text{elbow}}^d \end{pmatrix}^\top,$$

where the terms represent the desired CoM position, pelvis orientation, swing foot position and orientation, and arm joint angles. To account for turning motions during training, we heuristically adjust the reference trajectory based on the commanded angular velocity. Specifically, we modify the yaw orientation of all end-effector frames to align with the integrated yaw derived from the commanded angular velocity over time.

The HZD optimization is computed offline using IPOPT [241] and Casadi [242]. Hard constraints are used to enforce dynamics, periodicity, virtual constraints, and step length among others. The optimization problem is formulated as a multiple-shooting problem over a single swing phase.

Policy Structure: The policy receives a combination of proprioceptive and task-relevant inputs, including angular velocity, projected gravity, commanded linear and angular velocities, relative joint positions and velocities, the previous action, and a phase-based time encoding using $\sin(2\pi t/t_{\text{period}})$ and $\cos(2\pi t/t_{\text{period}})$. The stepping period t_{period} is set to $0.8\,\text{s}$, corresponding to a full gait cycle (i.e., two

steps). The policy outputs joint position commands relative to a default symmetric standing pose at 50 Hz. This default pose is fixed across all policies we trained.

Both the actor and critic networks use a fully connected feedforward architecture with hidden layer dimensions of [512, 256, 128], using the ELU (Exponential Linear Unit) activation function at each layer. To facilitate learning, the critic (value network) receives additional privileged information o_t^{priv} not accessible to the actor, such as stance and swing foot linear and angular velocities, reference trajectory positions and velocities, and binary contact state indicators. These inputs help stabilize value estimation and improve training efficiency.

Training Procedure: We use *IsaacLab*, a GPU-accelerated simulation framework built on top of NVIDIA Isaac Sim, with its development initiated from the *Orbit* framework [243]. For policy training, we use the PPO implementation from Robotics System Lab RL library [29].

All training use the same set of terrain settings, robot models, and domain randomization parameters. To improve policy robustness and facilitate transfer to hardware, we apply domain randomization to physical properties such as link masses, static and dynamic friction coefficients, and the center of mass position. Additionally, external perturbations are introduced by applying randomized velocity impulses in the x and y directions to the base link at fixed time intervals.

Training is conducted over a range of commanded velocities, with linear velocity $v_x \in [-0.75, 0.75]$ and yaw rate $\omega_z \in [-0.5, 0.5]$. Given a commanded velocity, v^d, the reference trajectory is either retrieved from the precomputed gait library by selecting the closest matching forward velocity in the case of HZD, or generated online using the analytic form of the H-LIP gait library, which provides closed-form solutions across the velocity space.

The baseline policy is trained with a manually designed reward composed of several heuristic terms. These include tracking of linear x, y and angular yaw velocity, while penalizing vertical velocity, angular velocity about the x and y axes, joint accelerations, joint velocities, joint torques, and abrupt changes in actions. The reward also discourages deviations from an upright base orientation and enforces joint limit compliance. Additional terms encourage behaviors such as staying upright (“stay alive"), avoiding foot slip during contact, maintaining a nominal hip posture and torso height, ensuring adequate foot clearance, and regulating contact scheduling. Furthermore, the policy is penalized for deviations of the arm and torso joints from

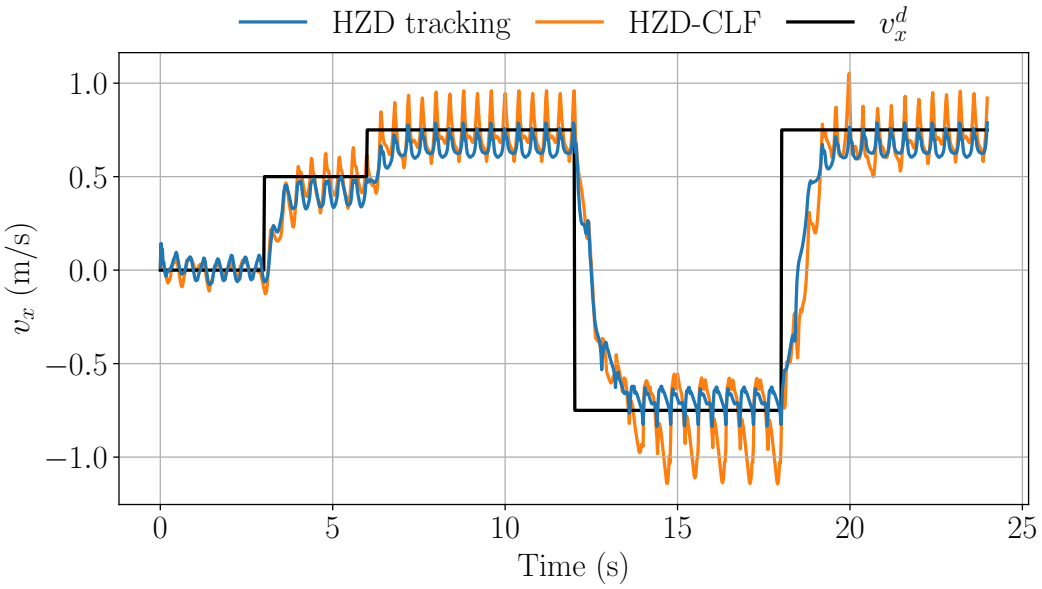


Figure 5.27: Tracking performance comparison between a policy trained using only the reference tracking reward (r_v) and one trained with both reference tracking and CLF decrease condition rewards (r_v and $r_{\dot{v}}$). Overall, the HZD-CLF policy achieves better tracking performance.

predefined reference configurations.

Table 5.1: Reward weight coefficients used during training for both HZD-CLF and LIP-CLF.

Reward Term	**Weight**
Torque penalty w_τ	1×10^{-5}
Action-rate penalty $w_{\Delta a}$	1×10^{-3}
Joint limit penalty $w_{q_{\text{limit}}}$	1.0
CLF tracking reward w_v	10.0
CLF decay penalty $w_{\dot{v}}$	2.0
Holonomic position reward w_{hpos}	4.0
Holonomic velocity reward w_{hvel}	2.0

5.7 Results

We evaluate three policy variants: a baseline trained with hand-tuned rewards, a H-LIP-based CLF shaped policy (LIP-CLF), and an HZD-based CLF-shaped policy (HZD-CLF). All policies are validated in both sim-to-sim transfer using MuJoCo [206] and on hardware (Fig. 5.30). The experiment video can be found [244]. The

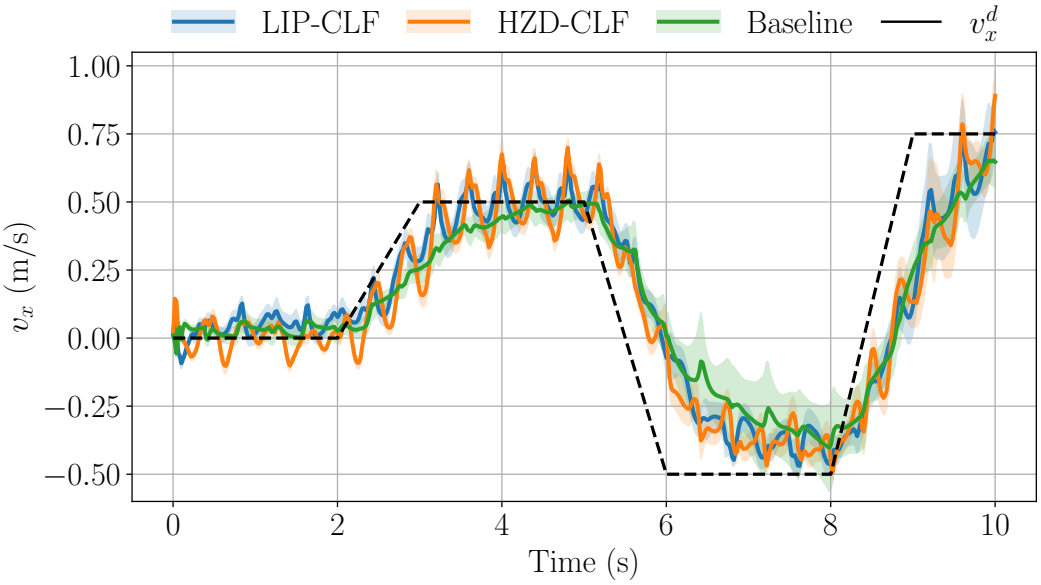


Figure 5.28: Tracking performance with torso mass randomly displaced within a box of size $\pm[0.05$ (x), 0.05 (y), 0.01 (z)]m around the nominal location. Fifty displacements are uniformly sampled, and the resulting mean and standard deviation of performance are plotted. The CLF-RL policies demonstrate lower variability, indicating improved consistency and robustness across different mass configurations.

policy runs onboard on an additional laptop mounted to the front of the torso during deployment, which also added additional mass (0.616 kg) to the robot.

CLF Decay Condition (Sim): To evaluate whether reference tracking alone is sufficient, or if enforcing the CLF decay condition provides additional benefit, we conduct an ablation study. Fig. 5.27 compares two policies: one trained with only the tracking reward (r_v), and another with both the tracking and CLF decay rewards (r_v and $r_{\dot{v}}$). Both policies track the commanded velocity v_x^d reasonably well across varying commands. However, the HZD-CLF policy shows improved steady-state tracking, maintaining average velocity closer to the desired value in forward and backward walking phases. While the overall transient behavior remains similar, the inclusion of the CLF decay condition reduces residual tracking error and promotes more consistent steady-state performance.

Mass Location Randomization (Sim): To assess robustness to modeling mismatches, we introduce random perturbations to the torso mass location to simulate sim-to-real discrepancies in inertial properties. Specifically, the torso center of mass is uniformly displaced within a box of size $\pm[0.05$ m $(x), 0.05$ m $(y), 0.01$ m $(z)]$. For each policy, we sample 50 randomized configurations and evaluate the velocity tracking

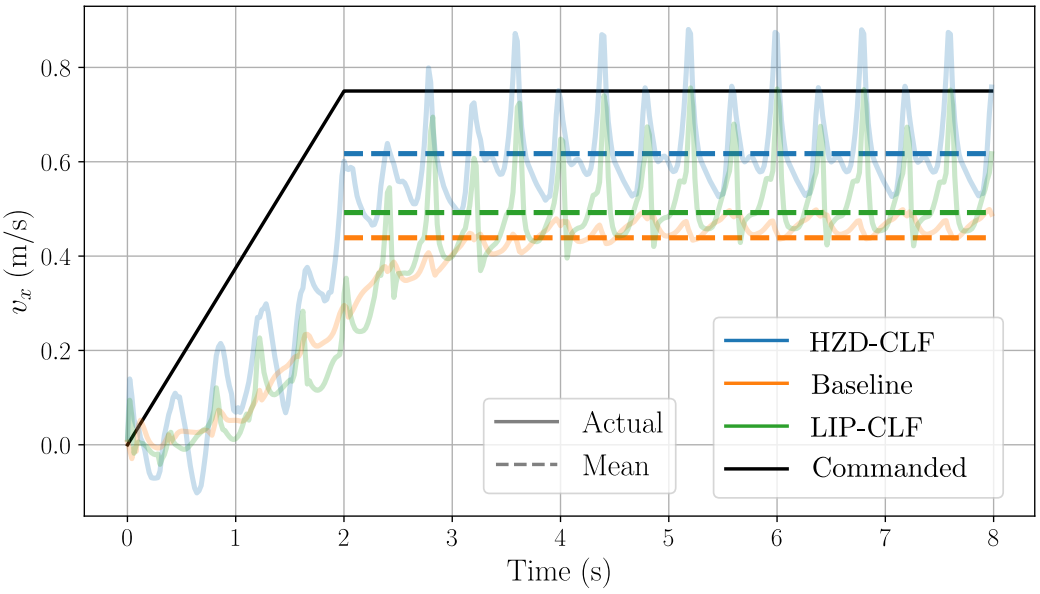


Figure 5.29: Robustness testing in simulation with different policies: HZD-CLF, LIP-CLF and a baseline RL policy are compared with an additional 8kg mass added to the torso. A two second ramp up to the maximum trained velocity is commanded. The steady-state mean of the velocities is plotted in a dashed line. The CLF-shaped policies show superior robustness to the baseline.

performance across episodes, reporting the mean and standard deviation. As shown in Fig. 5.28, both CLF-RL policies exhibit significantly lower performance variance compared to the baseline. This indicates improved robustness and consistency under structural uncertainties.

Added Mass Perturbation (Sim): To further evaluate robustness, we introduce an 8 kg payload to the torso and compare tracking performance across the three policy variants under a velocity ramp command. As shown in Fig. 5.29, the CLF-based RL policies maintain more accurate velocity tracking, demonstrating improved resilience to payload-induced dynamics changes.

Indoor Experiments (Hardware): We deploy all policies on the Unitree G1 robot in a controlled indoor environment. Fig. 5.30 shows all three policies operating successfully on hardware. To quantify velocity tracking performance and robustness, we use a motion capture system to record global position and orientation data. This data is used solely for evaluation and is not fed into the controller. To remain consistent with simulation analysis, we track the pelvis frame and compute local-frame velocity by finite-differencing the position data (recorded at 240 Hz) and aligning it with the robot's yaw orientation. Fig. 5.31 demonstrates the improved

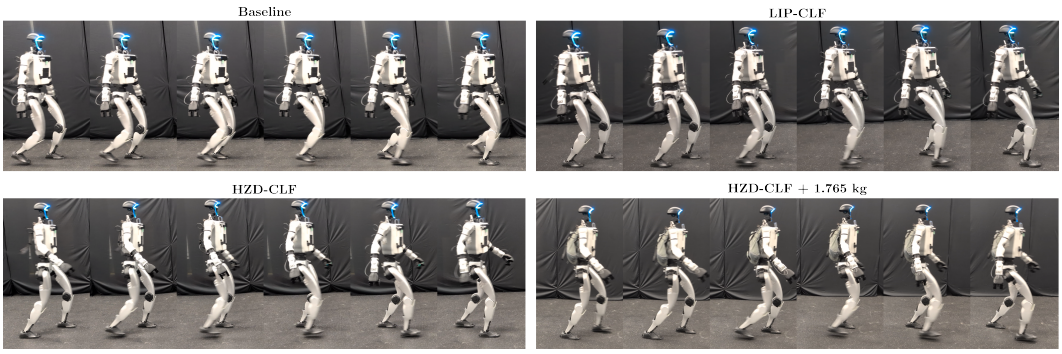


Figure 5.30: Snapshots of the three policies throughout a stride on the Unitree G1 robot. These images depict the walking motion in steady state walking with a commanded velocity of $v_x^d = 0.75$ m/s.

robustness of the HZD-CLF policy relative to the baseline. To test this robustness, we attach a backpack to the robot and load it with either 1.765 kg or 3.55 kg of additional mass. The HZD-CLF policy maintains consistent tracking performance across all conditions, whereas the baseline policy exhibits significant degradation. In the heavier case, the baseline policy drifts off the walking course and fails to complete the test. These hardware results corroborate the simulation findings in Fig. 5.28, which show that CLF-based policies exhibit lower performance variance under structural perturbations.

Outdoor Experiments (Hardware): To evaluate the HZD-CLF policy under real-world conditions, we deploy the robot outdoors across a variety of environments. As shown in Fig. 5.32, the test route includes diverse flat-ground surfaces such as concrete and tiled walkways, as well as mild slopes like ADA-compliant ramps. The robot traverses all terrain types within a single continuous 0.25-mile walking trial without any failures. The distance was determined by experimental design considerations, rather than reflecting any limitation of the policy.

Summary

We presented CLF-RL, a structured reward shaping framework that integrates reference trajectory tracking with control Lyapunov functions (CLFs). By embedding CLF-based objectives directly into the reinforcement learning reward, our approach replaces heuristic reward design with a theoretically grounded stability metric. We showed that incorporating the CLF decay condition improves tracking performance over baseline tracking-only rewards. CLF-based policies exhibit lower variance across a range of perturbations in both simulation and hardware, demonstrating

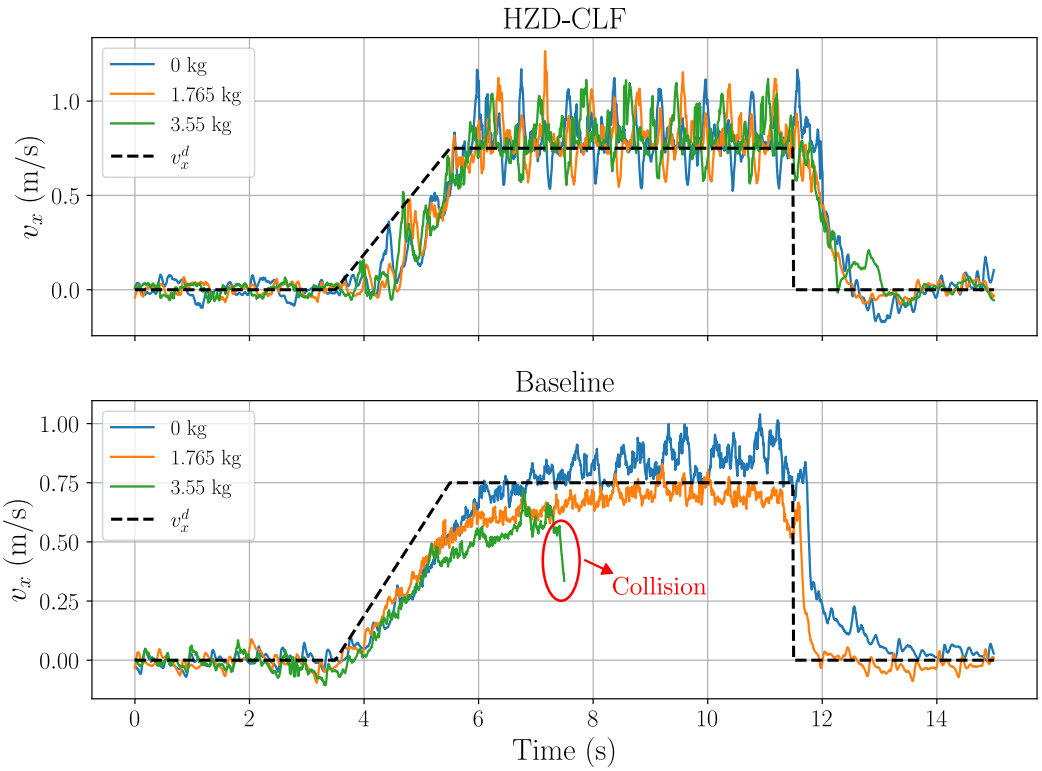


Figure 5.31: Quantitative hardware testing shows the difference in robustness of the baseline and HZD-CLF policies. Additional mass is added to a backpack on the back and the velocity tracking is compared. We can see that the HZD-CLF policy has effectively no change in performance with the additional mass. For the heavier mass on the baseline, the policy drifted so much as to exit the walking course and collide with a table, as indicated in the plot.

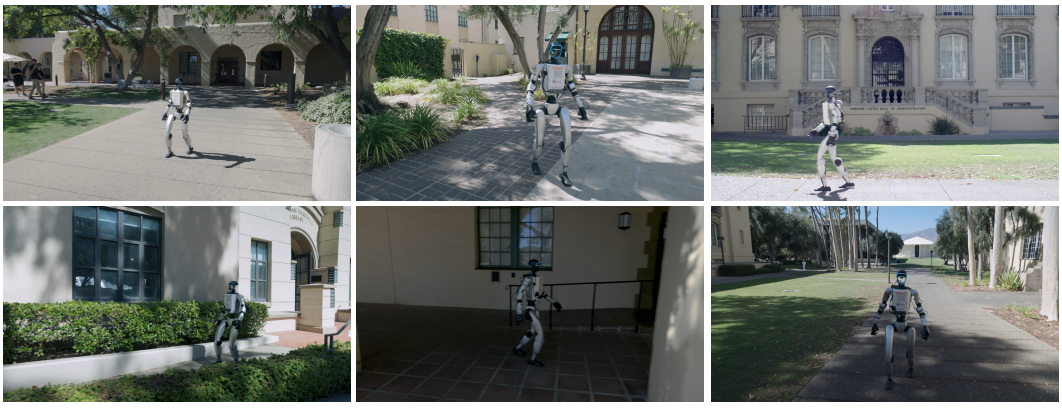


Figure 5.32: Demonstration of extensive outdoor testing of the HZD-CLF policy shows its ability to handle diverse flat-ground surfaces, including various tile and concrete types, as well as mild uphill and downhill slopes.

increased reliability.

Our framework is modular and extensible. We demonstrated its compatibility with both reduced-order (H-LIP) and full-order (HZD) reference generators, though other trajectory planners could be incorporated. While our training only used steady-state reference motions, the learned policies generalize well to transient motions—highlighting the framework's flexibility. Finally, we validated the real-world robustness of the approach through extended outdoor trials, showing consistent and stable walking performance across varied flat-ground terrains.

Chapter 6

CONCLUSION

This dissertation has addressed the fundamental challenge of enabling robust, effective, and personalized locomotion in robotic assistive devices and humanoid robots. The central contributions were organized around two complementary objectives: user alignment and robustness. Together, these objectives capture the dual requirements of human-centered locomotion: controllers must adapt to the variability of individual users while remaining reliable under the uncertainty of real-world environments.

To advance user alignment, the work developed preference-based learning formulations that characterize subjective trade-offs while discouraging unsafe or undesirable behaviors, and incorporated biomechanical principles into trajectory generation to yield anthropomorphic, dynamically feasible gaits. To advance robustness, the work proposed methods at multiple levels of the control hierarchy, from robustness metrics for trajectory design, to data-driven predictive replanning, to reinforcement learning policies shaped by control-inspired rewards.

Although presented separately, these two objectives are inseparable in practice: controllers that align with human preferences must also be robust to variability, and robustness is meaningful only when grounded in user needs. The unifying thread across this dissertation is the integration of model-based structure with data-driven adaptability, showing that neither alone is sufficient, but together they enable locomotion strategies that are principled, flexible, and human-centered.

Several promising directions build naturally on this foundation. Extending beyond periodic walking to running, stair climbing, and terrain adaptation would test the generality of these methods. More tightly coupling preference learning with robustness—for example through preference–robust optimization—could yield controllers that are simultaneously safe, adaptive, and customizable. Finally, scaling to collaborative multi-user or multi-robot settings would open opportunities to study locomotion under multi-objective and multi-agent conditions.

In summary, this dissertation shows that progress in bipedal robotics arises not from model-based or data-driven methods alone, but from their principled integration into controllers that are adaptable, reliable, and user-centered.

BIBLIOGRAPHY

[1] Boston Dynamics. *"What's new, ATLAS?"* https://youtu.be/fRj34o4hN4I.

[2] Unitree Robotics. *"Unitree Strikes Double Gold on Day One"* https://www.youtube.com/watch?v=p8f9axifPQY.

[3] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation". In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 1. IEEE. 2001, pp. 239–246.

[4] Eric R. Westervelt, Jessy W. Grizzle, and Daniel E. Koditschek. "Hybrid zero dynamics of planar biped walkers." In: *IEEE transactions on Automatic Control* 48.1 (2003), pp. 42–56.

[5] Jacob P. Reher, Ayonga Hereid, Shishir Kolathaya, Christian M Hubicki, and Aaron 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*. Springer. 2020, pp. 400–415.

[6] Ruben Grandia, Farbod Farshidian, René Ranftl, and Marco Hutter. "Feedback MPC for torque-controlled legged robots." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 4730–4737.

[7] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. "Learning quadrupedal locomotion over challenging terrain." In: *Science Robotics* 5.47 (2020), eabc5986.

[8] Zhaoming Xie, Patrick Clary, Jeremy Dao, Pedro Morais, Jonanthan Hurst, and Michiel Panne. "Learning locomotion skills for cassie: Iterative design and sim-to-real." In: *Conference on Robot Learning*. PMLR. 2020, pp. 317–329.

[9] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. *Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning*. en. arXiv:2109.11978 [cs]. Aug. 2022. URL: http://arxiv.org/abs/2109.11978 (visited on 05/16/2024).

[10] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. *Humanoid Parkour Learning*. en. arXiv:2406.10759 [cs]. Sept. 2024. URL: http://arxiv.org/abs/2406.10759 (visited on 11/16/2024).

[11] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. *Reinforcement Learning for Versatile, Dynamic, and Robust Bipedal Locomotion Control*. en. arXiv:2401.16889 [cs, eess]. Jan. 2024. URL: http://arxiv.org/abs/2401.16889 (visited on 05/17/2024).

[12] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. "Real-world humanoid locomotion with reinforcement learning". In: *Science Robotics* 9.89 (2024), eadi9579.

[13] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. "π0: A Vision-Language-Action Flow Model for General Robot Control". In: *arXiv preprint arXiv:2410.08584* (2024). Physical Intelligence.

[14] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control". In: *arXiv preprint arXiv:2307.15818* (2023).

[15] David A. Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[16] Thomas Gurriet, Maegan Tucker, Alexis Duburcq, Guilhem Boeris, and Aaron D. Ames. "Towards variable assistance for lower body exoskeletons." In: *IEEE Robotics and Automation Letters* 5.1 (2019), pp. 266–273. URL: http://dx.doi.org/10.1109/LRA.2019.2955946.

[17] Ryan J. Farris, Hugo A. Quintero, and Michael Goldfarb. "Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals." In: *Transactions on Neural Systems and Rehabilitation Engineering* 19.6 (2011), pp. 652–659.

[18] Sai K. Banala, Seok Hun Kim, Sunil K. Agrawal, and John P. Scholz. "Robot assisted gait training with active leg exoskeleton (ALEX)." In: *Transactions on Neural Systems and Rehabilitation Engineering* 17.1 (2008), pp. 2–8.

[19] Edwin H.F. van Asseldonk and Herman van der Kooij. “Robot-aided gait training with LOPES.” In: *Neurorehabilitation Technology*. Springer, 2012, pp. 379–396.

[20] Giulia Barbareschi, Rosie Richards, Matt Thornton, Tom Carlson, and Catherine Holloway. “Statically vs dynamically balanced gait: Analysis of a robotic exoskeleton compared with a human.” In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2015, pp. 6728–6731.

[21] Aaron D. Ames. “Human-inspired control of bipedal walking robots.” In: *IEEE Transactions on Automatic Control* 59.5 (2014), pp. 1115–1130.

[22] Maegan Tucker, Ellen Novoseller, Claudia Kann, Yanan Sui, Yisong Yue, Joel W Burdick, and Aaron D Ames. “Preference-based learning for exoskeleton gait optimization.” In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 2351–2357.

[23] Maegan Tucker, Myra Cheng, Ellen Novoseller, Yisong Yue, Joel Burdick, and Aaron D. Ames. “Human Preference-Based Learning for High-dimensional Optimization of Exoskeleton Walking Gaits.” In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.

[24] Myunghee Kim, Ye Ding, Philippe Malcolm, Jozefien Speeckaert, Christoper J. Siviy, Conor J. Walsh, and Scott Kuindersma. “Human-in-the-loop Bayesian optimization of wearable device parameters.” In: *PloS one* 12.9 (2017), e0184054.

[25] Maegan Tucker, Noel Csomay-Shanklin, Wen-Loong Ma, and Aaron D Ames. “Preference-based learning for user-guided hzd gait generation on bipedal walking robots.” In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 2804–2810.

[26] Noel Csomay-Shanklin, Maegan Tucker, Min Dai, Jenna Reher, and Aaron D. Ames. “Learning controller gains on bipedal walking robots via user preferences.” In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 10405–10411.

[27] Remco I. Leine and Henk Nijmeijer. *Dynamics and bifurcations of non-smooth mechanical systems*. Vol. 18. Springer Science & Business Media, 2013.

[28] Min Dai, Xiaobin Xiong, Jaemin Lee, and Aaron D. Ames. “Data-driven adaptation for robust bipedal locomotion with step-to-step dynamics.” In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 8574–8581.

[29] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. “Learning to walk in minutes using massively parallel deep reinforcement learning”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 91–100.

[30] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. "Reinforcement learning for robust parameterized locomotion control of bipedal robots." In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 2811–2817.

[31] Benjamin Morris and Jessy W. Grizzle. "A restricted Poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots." In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 4199–4206.

[32] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. "Biped walking pattern generation by using preview control of zero-moment point." In: *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*. Vol. 2. IEEE. 2003, pp. 1620–1626.

[33] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. "Capture point: A step toward humanoid push recovery." In: *2006 6th IEEE-RAS international conference on humanoid robots*. Ieee. 2006, pp. 200–207.

[34] Xiaobin Xiong and Aaron Ames. "3-d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization." In: *IEEE Transactions on Robotics* 38.4 (2022), pp. 2405–2425.

[35] Lawrence Perko. *Differential Equations and Dynamical Systems*. Vol. 7. Springer Science & Business Media, 2013.

[36] Alberto Isidori, S. Shankar Sastry, Petar V. Kototovic, and Christopher I. Byrnes. "Singularly perturbed zero dynamics of nonlinear systems." In: *Transactions on Automatic Control* 37.10 (1992), pp. 1625–1631.

[37] `https://pomax.github.io/bezierinfo/`. 2023.

[38] David F. Rogers and James Alan Adams. *Mathematical elements for computer graphics*. McGraw-Hill, Inc., 1989.

[39] Ayonga Hereid, Shishir Kolathaya, Mikhail S. Jones, Johnathan Van Why, Jonathan W. Hurst, and Aaron D. Ames. "Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control". In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*. 2014, pp. 263–272.

[40] Jenna Reher and Aaron D. Ames. "Dynamic walking: Toward agile and efficient bipedal robots." In: *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), pp. 535–572.

[41] Ayonga Hereid and Aaron D. Ames. "Frost*: Fast robot optimization and simulation toolkit." In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 719–726.

[42] Kejun Li, Zachary Olkin, Yisong Yue, and Aaron D. Ames. “CLF-RL: Control Lyapunov Function Guided Reinforcement Learning.” In: *arXiv preprint arXiv:2508.09354* (2025).

[43] Aaron D. Ames, Kevin Galloway, Koushil Sreenath, and Jessy W. Grizzle. “Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics”. en. In: *IEEE Transactions on Automatic Control* 59.4 (Apr. 2014), pp. 876–891. ISSN: 0018-9286, 1558-2523. DOI: 10.1109/TAC.2014.2299335. URL: http://ieeexplore.ieee.org/document/6709752/ (visited on 08/04/2025).

[44] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. “Control barrier function based quadratic programs for safety critical systems.” In: *Transactions on Automatic Control* 62.8 (2017), pp. 3861–3876.

[45] Quan Nguyen and Koushil Sreenath. “Exponential control barrier functions for enforcing high relative-degree safety-critical constraints.” In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 322–328.

[46] Patrick L. Jacobs and Mark S. Nash. “Exercise recommendations for individuals with spinal cord injury.” In: *Sports Medicine* 34.11 (2004), pp. 727–751.

[47] Kathleen A. Martin Ginis, Amy E. Latimer, Kyle McKechnie, David S. Ditor, Neil McCartney, Audrey L. Hicks, Joanne Bugaresti, and B. Catharine Craven. “Using exercise to enhance subjective well-being among people with spinal cord injury: The mediating influences of stress and pain.” In: *Rehabilitation Psychology* 48.3 (2003), p. 157.

[48] Jamie Wolff, Claire Parker, Jaimie Borisoff, W. Ben Mortenson, and Johanne Mattie. “A survey of stakeholder perspectives on exoskeleton technology.” In: *Journal of Neuroengineering and Rehabilitation* 11.1 (2014), p. 169.

[49] Paula Kocina. “Body composition of spinal cord injured adults.” In: *Sports Medicine* 23.1 (1997), pp. 48–60.

[50] Gülçin Demirel, HuÈrriyet Yilmaz, Nurdan Paker, and Selma OÈnel. “Osteoporosis after spinal cord injury.” In: *Spinal Cord* 36.12 (1998), pp. 822–825.

[51] Nebahat Sezer, Selami Akkuş, and Fatma Gülçin Uğurlu. “Chronic complications of spinal cord injury.” In: *World Journal of Orthopedics* 6.1 (2015), p. 24.

[52] Martha Freeman Somers. *Spinal cord injury: Functional rehabilitation.* Prentice Hall, 2001.

[53] Samar Hamid and Ray Hayek. “Role of electrical stimulation for rehabilitation and regeneration after spinal cord injury: An overview.” In: *European Spine Journal* 17.9 (2008), pp. 1256–1269.

[54] Susan Harkema, Yury Gerasimenko, Jonathan Hodes, Joel W. Burdick, Claudia Angeli, Yangsheng Chen, Christie Ferreira, Andrea Willhite, Enrico Rejc, Robert G. Grossman, et al. "Effect of epidural stimulation of the lumbosacral spinal cord on voluntary movement, standing, and assisted stepping after motor complete paraplegia: A case study." In: *The Lancet* 377.9781 (2011), pp. 1938–1947.

[55] Carsten Bach Baunsgaard, Ulla Vig Nissen, Anne Katrin Brust, Angela Frotzler, Cornelia Ribeill, Yorck-Bernhard Kalke, Natacha León, Belén Gómez, Kersti Samuelsson, Wolfram Antepohl, et al. "Gait training after spinal cord injury: Safety, feasibility and gait function following 8 weeks of training with the exoskeletons from Ekso Bionics." In: *Spinal Cord* 56.2 (2018), pp. 106–116.

[56] Atif S. Khan, Donna C. Livingstone, Caitlin L. Hurd, Jennifer Duchcherer, John E. Misiaszek, Monica A. Gorassini, Patricia J. Manns, and Jaynie F. Yang. "Retraining walking over ground in a powered exoskeleton after spinal cord injury: A prospective cohort study to examine functional gains and neuroplasticity." In: *Journal of Neuroengineering and Rehabilitation* 16.1 (2019), pp. 1–17.

[57] Rosanne B. van Dijsseldonk, Hennie Rijken, Ilse J.W. van Nes, Henk van de Meent, and Noël L.W. Keijsers. "Predictors of exoskeleton motor learning in spinal cord injured patients." In: *Disability and Rehabilitation* 43.14 (2021), pp. 1982–1988.

[58] Omar Harib, Ayonga Hereid, Ayush Agrawal, Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, M. Eva Mungai, Mattieu Masselin, Aaron D. Ames, et al. "Feedback control of an exoskeleton for paraplegics: Toward robustly stable, hands-free dynamic walking." In: *IEEE Control Systems Magazine* 38.6 (2018), pp. 61–87.

[59] Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, Ayonga Hereid, Omar Harib, Matthieu Masselin, Jessy Grizzle, and Aaron D Ames. "Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons." In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2804–2811.

[60] Ayush Agrawal, Omar Harib, Ayonga Hereid, Sylvain Finet, Matthieu Masselin, Laurent Praly, Aaron D Ames, Koushil Sreenath, and Jessy W Grizzle. "First steps towards translating HZD control of bipedal robots to decentralized control of exoskeletons." In: *IEEE Access* 5 (2017), pp. 9919–9934.

[61] Jacques Kerdraon, Jean Gabriel Previnaire, Maegan Tucker, Pauline Coignard, Willy Allegre, Emmanuel Knappen, and Aaron Ames. "Evaluation of safety and performance of the self balancing walking system Atalante in patients with complete motor spinal cord injury." In: *Spinal Cord Series and Cases* 7.1 (2021), p. 71.

[62] Xinyu Wu, Du-Xin Liu, Ming Liu, Chunjie Chen, and Huiwen Guo. "Individualized gait pattern generation for sharing lower limb exoskeleton robot." In: *IEEE Transactions on Automation Science and Engineering* 15.4 (2018), pp. 1459–1470.

[63] Shixin Ren, Weiqun Wang, Zeng-Guang Hou, Badong Chen, Xu Liang, Jiaxing Wang, and Liang Peng. "Personalized gait trajectory generation based on anthropometric features using Random Forest." In: *Journal of Ambient Intelligence and Humanized Computing* (2019), pp. 1–12.

[64] Juanjuan Zhang, Pieter Fiers, Kirby A. Witte, Rachel W. Jackson, Katherine L. Poggensee, Christopher G. Atkeson, and Steven H. Collins. "Human-in-the-loop optimization of exoskeleton assistance during walking." In: *Science* 356.6344 (2017), pp. 1280–1284.

[65] Nitish Thatte, Helei Duan, and Hartmut Geyer. "A method for online optimization of lower limb assistive devices with high dimensional parameter spaces." In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 5380–5385.

[66] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. "Accurately interpreting clickthrough data as implicit feedback." In: *SIGIR*. Vol. 5. 2005, pp. 154–161.

[67] Huihua Zhao, Ayonga Hereid, Wen-loong Ma, and Aaron D. Ames. "Multi-contact bipedal robotic locomotion." In: *Robotica* 35.5 (2017), pp. 1072–1106.

[68] Huihua Zhao, Eric Ambrose, and Aaron D. Ames. "Preliminary results on energy efficient 3D prosthetic walking with a powered compliant transfemoral prosthesis." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1140–1147.

[69] Hong Han, Wei Wang, Fengchao Zhang, Xin Li, Jianyu Chen, Jianda Han, and Juanjuan Zhang. "Selection of Muscle-Activity-Based Cost Function in Human-in-the-Loop Optimization of Multi-Gait Ankle Exoskeleton Assistance." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), pp. 944–952.

[70] Samuel K Au, Paolo Bonato, and Hugh Herr. "An EMG-position controlled system for an active ankle-foot prosthesis: An initial experimental study." In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE. 2005, pp. 375–379.

[71] Carl D. Hoover and Kevin B. Fite. "A configuration dependent muscle model for the myoelectric control of a transfemoral prosthesis." In: *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE. 2011, pp. 1–6.

[72] Sai-Kit Wu, Garrett Waycaster, and Xiangrong Shen. "Electromyography-based control of active above-knee prostheses." In: *Control Engineering Practice* 19.8 (2011), pp. 875–882.

[73] Jing Wang, Oliver A. Kannape, and Hugh M. Herr. "Proportional EMG control of ankle plantar flexion in a powered transtibial prosthesis." In: *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE. 2013, pp. 1–5.

[74] Andrea Cimolato, Giovanni Milandri, Leonardo S. Mattos, Elena De Momi, Matteo Laffranchi, and Lorenzo De Michieli. "Hybrid Machine Learning-Neuromusculoskeletal Modeling for Control of Lower Limb Prosthetics." In: *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE. 2020, pp. 557–563.

[75] Robert L. Waters and Sara Mulroy. "The energy expenditure of normal and pathologic gait." In: *Gait & Posture* 9.3 (1999), pp. 207–231.

[76] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. "Positive force feedback in bouncing gaits?" In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1529 (2003), pp. 2173–2183.

[77] Hartmut Geyer and Hugh Herr. "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities." In: *IEEE Transactions on neural systems and rehabilitation engineering* 18.3 (2010), pp. 263–273.

[78] Jessy W Grizzle, Christine Chevallereau, Ryan W Sinnet, and Aaron D Ames. "Models, feedback control, and open problems of 3D bipedal robotic walking". In: *Automatica* 50.8 (2014), pp. 1955–1988.

[79] Aaron D Ames, Ramanarayan Vasudevan, and Ruzena Bajcsy. "Human-data based cost of bipedal robotic walking." In: *Proceedings of the 14th international conference on Hybrid systems: computation and control*. 2011, pp. 153–162.

[80] Jonathan Camargo, Aditya Ramanathan, Will Flanagan, and Aaron Young. "A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions". In: *Journal of Biomechanics* 119 (2021), p. 110320.

[81] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1447–1454.

[82] Maegan Tucker, Myra Cheng, Ellen Novoseller, Richard Cheng, Yisong Yue, Joel W. Burdick, and Aaron D. Ames. "Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits." In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2020, pp. 3423–3430.

[83] *Supplementary video for "Natural Multicontact walking for Robotic Assistive Devices via Musculoskeletal Models and Hybrid Zero Dynamics."* https://www.youtube.com/watch?v=gOhZlTypNIs.

[84] Seungmoon Song and Hartmut Geyer. "Generalization of a muscle-reflex control model to 3D walking." In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2013, pp. 7463–7466.

[85] Brian S. Armour, Elizabeth A. Courtney-Long, Michael H. Fox, Heidi Fredine, and Anthony Cahill. "Prevalence and Causes of Paralysis—United States, 2013". In: *American Journal of Public Health* 106.10 (2016), pp. 1855–1857.

[86] Kejun Li, Maegan Tucker, Erdem Bıyık, Ellen Novoseller, Joel W Burdick, Yanan Sui, Dorsa Sadigh, Yisong Yue, and Aaron D Ames. "ROIAL: Region of Interest Active Learning for Characterizing Exoskeleton Gait Preference Landscapes". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 3212–3218. URL: http://dx.doi.org/10.1109/ICRA48506.2021.9560840.

[87] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. "Safe exploration for optimization with Gaussian processes." In: *International Conference on Machine Learning*. 2015.

[88] Jens Schreiter, Duy Nguyen-Tuong, Mona Eberts, Bastian Bischoff, Heiner Markert, and Marc Toussaint. "Safe exploration for active learning with Gaussian processes." In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2015, pp. 133–149.

[89] Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. "Safe controller optimization for quadrotors with Gaussian processes." In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016.

[90] Yanan Sui, Joel Burdick, Yisong Yue, et al. "Stagewise safe Bayesian optimization with Gaussian processes". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4781–4789.

[91] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. "Bayesian active learning for classification and preference learning." In: *arXiv preprint arXiv:1112.5745* (2011).

[92] Erdem Bıyık, Nicolas Huynh, Mykel J. Kochenderfer, and Dorsa Sadigh. "Active Preference-Based Gaussian Process Regression for Reward Learning." In: *Proceedings of Robotics: Science and Systems (RSS)*. 2020.

[93] Erdem Bıyık, Malayandi Palan, Nicholas C. Landolfi, Dylan P. Losey, and Dorsa Sadigh. "Asking easy questions: A user-friendly approach to active reward learning." In: *arXiv preprint arXiv:1910.04365* (2019).

[94] Z. Xu, K. Kersting, and T. Joachims. "Fast Active Exploration for Link-Based Preference Learning using Gaussian Processes." In: *European Conference on Machine Learning*. 2010, pp. 499–514.

[95] Johannes Fürnkranz and Eyke Hüllermeier. "Preference learning and ranking by pairwise comparison." In: *Preference Learning*. Springer, 2010, pp. 65–82.

[96] Nils Wilde, Dana Kulic, and Stephen L. Smith. "Active Preference Learning using Maximum Regret." In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.

[97] Li Qian, Jinyang Gao, and HV Jagadish. "Learning user preferences by adaptive pairwise comparison". In: *Proceedings of the VLDB Endowment* 8.11 (2015), pp. 1322–1333.

[98] Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. "Multi-dueling bandits with dependent arms." In: *arXiv preprint arXiv:1705.00253* (2017).

[99] Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. "Preference-based online learning with dueling bandits: A survey". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 278–385.

[100] Wei Chu and Zoubin Ghahramani. "Gaussian processes for ordinal regression." In: *Journal of machine learning research* 6.Jul (2005), pp. 1019–1041.

[101] *Repository for ROIAL.* `https://github.com/kli58/ROIAL`.

[102] Wei Chu and Zoubin Ghahramani. "Preference learning with Gaussian processes." In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005, pp. 137–144.

[103] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[104] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. "High dimensional Bayesian optimisation and bandits via additive models." In: *International Conference on Machine Learning*. 2015, pp. 295–304.

[105] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas, et al. "Bayesian Optimization in High Dimensions via Random Embeddings." In: *IJCAI*. 2013, pp. 1778–1784.

[106] Nir Ailon. "An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity." In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 137–164.

[107] *Supplementary video for "ROIAL: Region of Interest Active Learning for Characterizing Exoskeleton Gait Preference Landscapes."* `https://www.youtube.com/watch?v=04lMJmKmZrQ`.

[108] Ayonga Hereid, Christian M. Hubicki, Eric A. Cousineau, and Aaron D. Ames. "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization." In: *IEEE Transactions on Robotics* 34.2 (2018), pp. 370–387.

[109] *Supplementary Website*. `https://sites.google.com/view/roial-icra2021`.

[110] Maegan Tucker, Kejun Li, Yisong Yue, and Aaron D. Ames. "Polar: Preference Optimization and Learning Algorithms for robotics." In: *arXiv preprint arXiv:2208.04404* (2022).

[111] Pannaga Shivaswamy and Thorsten Joachims. "Coactive learning." In: *Journal of Artificial Intelligence Research* 53 (2015), pp. 1–40.

[112] Pannaga Shivaswamy and Thorsten Joachims. "Online structured prediction via coactive learning." In: *International Conference on Machine Learning (ICML)*. 2012, pp. 59–66.

[113] Steven A. Wolfman, Tessa Lau, Pedro Domingos, and Daniel S. Weld. "Mixed initiative interfaces for learning tasks: SMARTedit talks back." In: *International Conference on Intelligent User Interfaces*. ACM. 2001, pp. 167–174.

[114] James C. Lester, Brian A. Stone, and Gary D. Stelling. "Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments." In: *User Modeling and User-Adapted Interaction* 9.1-2 (1999), pp. 1–44.

[115] William R. Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples." In: *Biometrika* 25.3/4 (1933), pp. 285–294.

[116] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten Rijke. "Relative upper confidence bound for the k-armed dueling bandit problem." In: *International Conference on Machine Learning (ICML)*. PMLR. 2014, pp. 10–18.

[117] Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. "Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces." In: *International Conference on Machine Learning (ICML)*. PMLR. 2019, pp. 3429–3438.

[118] Timothy T. Roberts, Garrett R. Leonard, and Daniel J. Cepela. "Classifications in brief: American spinal injury association (ASIA) impairment scale". In: *Clinical Orthopaedics and Related Research* 475.5 (2017), p. 1499.

[119] Andrew Singletary, Karl Klingebiel, Joseph Bourne, Andrew Browning, Phil Tokumaru, and Aaron Ames. "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance." In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 8129–8136.

[120] Anil Alan, Andrew J Taylor, Chaozhe R. He, Gábor Orosz, and Aaron D. Ames. "Safe controller synthesis with tunable input-to-state safe control barrier functions." In: *IEEE Control Systems Letters* 6 (2021), pp. 908–913.

[121] Wen-Loong Ma, Shishir Kolathaya, Eric R. Ambrose, Christian M. Hubicki, and Aaron D. Ames. "Bipedal robotic running with DURUS-2D: Bridging the gap between theory and experiment". In: *Proceedings of the 20th international conference on hybrid systems: computation and control*. 2017, pp. 265–274.

[122] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. "Control barrier function based quadratic programs with application to adaptive cruise control". In: *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 6271–6278.

[123] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. "Control barrier functions: Theory and applications." In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 3420–3431.

[124] Kerianne L. Hobbs, Mark L. Mote, Matthew C. L. Abate, Samuel D. Coogan, and Eric M. Feron. "Run Time Assurance for Safety-Critical Systems: An Introduction to Safety Filtering Approaches for Complex Control Systems." In: *IEEE Control Systems Magazine* 43.2 (2023), pp. 28–65.

[125] Li Wang, Evangelos A. Theodorou, and Magnu Egerstedt. "Safe learning of quadrotor dynamics using barrier certificates." In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2460–2465.

[126] Andrew J. Taylor and Aaron D. Ames. "Adaptive Safety with Control Barrier Functions." In: *American Control Conference (ACC)*. IEEE. 2020, pp. 1399–1405.

[127] Fernando Castaneda, Jason J. Choi, Bike Zhang, Claire J. Tomlin, and Koushil Sreenath. "Gaussian Process-based Min-norm Stabilizing Controller for Control-Affine Systems with Uncertain Input Effects." In: *arXiv preprint arXiv:2011.07183* (2020).

[128] Andrew J. Taylor, Andrew Singletary, Yisong Yue, and Aaron D. Ames. "Learning for Safety-Critical Control with Control Barrier Functions." In: *Proceedings of Machine Learning Research (PMLR)* 120 (2020), pp. 708–717.

[129] Mrdjan Jankovic. "Robust control barrier functions for constrained stabilization of nonlinear systems." In: *Automatica* 96 (2018), pp. 359–367.

[130] Shishir Kolathaya, Jacob Reher, and Aaron D. Ames. "Input to state stability of bipedal walking robots: Application to durus." In: *arXiv preprint arXiv:1801.00618* (2018).

[131] Andrew Clark. "Control barrier functions for complete and incomplete information stochastic systems." In: *American Control Conference (ACC)*. IEEE. 2019, pp. 2928–2935.

[132] Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. "Verification and control for finite-time safety of stochastic systems via barrier functions." In: *2019 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2019, pp. 712–717.

[133] Jason J. Choi, Donggun Lee, Koushil Sreenath, Claire J. Tomlin, and Sylvia L. Herbert. "Robust Control Barrier-Value Functions for Safety-Critical Control." In: *arXiv preprint arXiv:2104.02808* (2021).

[134] Rin Takano and Masaki Yamakita. "Robust Constrained Stabilization Control Using Control Lyapunov and Control Barrier Function in the Presence of Measurement Noises." In: *Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 300–305.

[135] Sarah Dean, Andrew Taylor, Ryan Cosner, Benjamin Recht, and Aaron Ames. "Guaranteeing Safety of Learned Perception Modules via Measurement-Robust Control Barrier Functions." In: *Conference on Robotics Learning (CoRL)*. 2020.

[136] Ryan K. Cosner, Andrew W. Singletary, Andrew J. Taylor, Tamas G. Molnar, Katherine L. Bouman, and Aaron D. Ames. "Measurement-robust control barrier functions: Certainty in safety with uncertainty in state." In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 6286–6291.

[137] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. "Active preference-based learning of reward functions." In: *Robotics: Science and Systems*. 2017.

[138] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. "Boolean composability of constraints and control synthesis for multi-robot systems via nonsmooth control barrier functions." In: *Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 897–902.

[139] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[140] Alexander Domahidi, Eric Chu, and Stephen Boyd. "ECOS: An SOCP solver for embedded systems". In: *European Control Conference (ECC)*. IEEE. 2013, pp. 3071–3076.

[141] *Supplementary Video for "Safety-aware Preference-based Learning for Safety-critical Control."* `https://youtu.be/QEuwRDTG7TE`. 2022.

[142] Jonas uchli, Mrinal Kalakrishnan, Michael Mistry, Peter Pastor, and Stefan Schaal. "Compliant quadruped locomotion over rough terrain." In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 814–820.

[143] Eric Brochu, Tyson Brochu, and Nando De Freitas. "A Bayesian interactive optimization approach to procedural animation design." In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2010, pp. 103–112.

[144] Jens Brehm Bagger Nielsen, Jakob Nielsen, and Jan Larsen. "Perception-Based Personalization of Hearing Aids Using Gaussian Processes and Active Learning." In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1 (2015), pp. 162–173.

[145] Kush Bhatia, Ashwin Pananjady, Peter Bartlett, Anca Dragan, and Martin J. Wainwright. "Preference learning along multiple criteria: A game-theoretic perspective." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7413–7424.

[146] Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. "Dueling posterior sampling for preference-based reinforcement learning." In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 1029–1038.

[147] Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Javier González, Pablo García Moreno, and Aki Vehtari. "Preferential batch Bayesian optimization." In: *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing*. IEEE. 2021, pp. 1–6.

[148] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. "The k-armed dueling bandits problem." In: *Journal of Computer and System Sciences* 78.5 (2012), pp. 1538–1556.

[149] Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. "A survey of preference-based reinforcement learning methods." In: *Journal of Machine Learning Research* 18.136 (2017), pp. 1–46.

[150] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D. Lawrence. "Preferential Bayesian optimization." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1282–1291.

[151] Raul Astudillo, Zhiyuan Jerry Lin, Eytan Bakshy, and Peter Frazier. "qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization." In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 1093–1114.

[152] Marco Maccarini, Filippo Pura, Dario Piga, Loris Roveda, Lorenzo Mantovani, and Francesco Braghin. "Preference-Based Optimization of a Human-Robot Collaborative Controller." In: *IFAC-PapersOnLine* 55.38 (2022), pp. 7–12.

[153] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. "Direct Preference Optimization: Your Language Model is Secretly a Reward Model." In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[154] Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. "Beyond one-preference-for-all: Multi-objective direct preference optimization." In: *arXiv preprint arXiv:2310.03708* (2023).

[155] Kaisa Miettinen. "Nonlinear Multiobjective Optimization." In: International Series in Operations Research & Management Science. Springer, 1999.

[156] R. Timothy Marler and Jasbir S. Arora. "Survey of multi-objective optimization methods for engineering." In: *Structural and Multidisciplinary Optimization* 26 (2004), pp. 369–395.

[157] Kalyanmoy Deb. "Multi-objective optimization." In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2013, pp. 403–449.

[158] Nazan Khan, David E. Goldberg, and Martin Pelikan. "Multi-objective Bayesian optimization algorithm." In: *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. 2002, pp. 684–684.

[159] Joshua Knowles. "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems." In: *IEEE Transactions on Evolutionary Computation* 10.1 (2006), pp. 50–66.

[160] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. "Max-value entropy search for multi-objective Bayesian optimization." In: *Advances in Neural Information Processing Systems* 32 (2019).

[161] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. "A flexible framework for multi-objective Bayesian optimization using random scalarizations." In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 766–776.

[162] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. "Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9851–9864.

[163] Jürgen Branke and Kalyanmoy Deb. "Integrating user preferences into evolutionary multi-objective optimization." In: *Knowledge Incorporation in Evolutionary Computation*. Springer, 2005, pp. 461–477.

[164] Handing Wang, Markus Olhofer, and Yaochu Jin. "A mini-review on preference modeling and articulation in multi-objective optimization: Current status and challenges." In: *Complex & Intelligent Systems* 3 (2017), pp. 233–245.

[165] Jussi Hakanen and Joshua D. Knowles. "On using decision maker preferences with ParEGO." In: *Evolutionary Multi-Criterion Optimization*. Springer. 2017, pp. 282–297.

[166] Zhiyuan Jerry Lin, Raul Astudillo, Peter Frazier, and Eytan Bakshy. "Preference exploration for efficient Bayesian optimization with multiple outcomes." In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 4235–4258.

[167] Ralph L. Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value trade-offs.* Cambridge university press, 1993.

[168] Jean-Charles Pomerol and Sergio Barba-Romero. *Multicriterion decision in management: Principles and practice.* Vol. 25. Springer Science & Business Media, 2000.

[169] H. Peyton Young. "A note on preference aggregation." In: *Econometrica: Journal of the Econometric Society* (1974), pp. 1129–1131.

[170] James S. Dyer and Rakesh K. Sarin. "Group preference aggregation rules based on strength of preference." In: *Management Science* 25.9 (1979), pp. 822–832.

[171] Jacob P. Baskin and Shriram Krishnamurthi. "Preference aggregation in group recommender systems for committee decision-making." In: *Proceedings of the Third ACM Conference on Recommender Systems*. 2009, pp. 337–340.

[172] Quoc Phong Nguyen, Sebastian Tay, Bryan Kian Hsiang Low, and Patrick Jaillet. "Top-k ranking Bayesian optimization." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 9135–9143.

[173] Raul Astudillo, Kejun Li, Maegan Tucker, Chu Xin Cheng, Aaron D. Ames, and Yisong Yue. "Preferential Multi-objective Bayesian Optimization." In: *Transactions on Machine Learning Research* (2024).
K.L. URL: https://openreview.net/pdf?id=mjsoESaWDH.

[174] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabas Poczos. "Parallelised Bayesian Optimisation via Thompson Sampling." In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, 2018, pp. 133–142.

[175] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. "BoTorch: A framework for efficient Monte-Carlo Bayesian optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 21524–21538.

[176] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. "Performance assessment of multiobjective optimizers: An analysis and review". In: *IEEE Transactions on Evolutionary Computation* 7.2 (2003), pp. 117–132.

[177] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in Neural Information Processing Systems* 20 (2007).

[178] Shion Takeno, Masahiro Nomura, and Masayuki Karasuyama. "Towards Practical Preferential Bayesian Optimization with Skew Gaussian Processes". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 33516–33533.

[179] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. "Scalable Test Problems for Evolutionary Multiobjective Optimization". In: *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 105–145. DOI: 10.1007/1-84628-137-7_6.

[180] Ryoji Tanabe and Hisao Ishibuchi. "An easy-to-use real-world multi-objective optimization problem suite". In: *Applied Soft Computing* 89 (2020), p. 106078.

[181] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL". In: *The International Journal of Robotics Research* 30.9 (2011), pp. 1170–1193.

[182] Nicolaus A Radford, Philip Strawser, Kimberly Hambuchen, Joshua S Mehling, William K Verdeyen, A Stuart Donnan, James Holley, Jairo Sanchez, Vienny Nguyen, Lyndon Bridgwater, et al. "Valkyrie: Nasa's first bipedal humanoid robot". In: *Journal of Field Robotics* 32.3 (2015), pp. 397–419.

[183] Christian Hubicki, Jesse Grimes, Mikhail Jones, Daniel Renjewski, Alexander Spröwitz, Andy Abate, and Jonathan Hurst. "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot". In: *The International Journal of Robotics Research* 35.12 (2016), pp. 1497–1521.

[184] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". In: *Autonomous robots* 40 (2016), pp. 429–455.

[185] Xingye Da, Ross Hartley, and Jessy W. Grizzle. "Supervised learning for stabilizing underactuated bipedal robot locomotion, with outdoor experiments on the wave field." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3476–3483.

[186] Xiaobin Xiong and Aaron D. Ames. "Dynamic and versatile humanoid walking via embedding 3D actuated slip model with hybrid lip based stepping." In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6286–6293.

[187] Victor C. Paredes and Ayonga Hereid. "Resolved motion control for 3D underactuated bipedal walking using linear inverted pendulum dynamics and neural adaptation." In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 6761–6767.

[188] Devin Crowley, Jeremy Dao, Helei Duan, Kevin Green, Jonathan Hurst, and Alan Fern. "Optimizing Bipedal Locomotion for The 100m Dash With Comparison to Human Running". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 12205–12211.

[189] Siyuan Feng, X. Xinjilefu, Christopher G. Atkeson, and Joohyung Kim. "Robust dynamic walking using online foot step optimization." In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2016, pp. 5373–5378.

[190] Gabriel García, Robert Griffin, and Jerry Pratt. "MPC-based locomotion control of bipedal robots with line-feet contact using centroidal dynamics." In: *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2021, pp. 276–282.

[191] Nathan J. Kong, J. Joe Payne, James Zhu, and Aaron M. Johnson. "Saltation Matrices: The Essential Tool for Linearizing Hybrid Dynamical Systems." In: *arXiv preprint arXiv:2306.06862* (2023).

[192] James Zhu, Nathan J. Kong, George Council, and Aaron M. Johnson. "Hybrid event shaping to stabilize periodic hybrid orbits." In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 01–07.

[193] Maegan Tucker, Noel Csomay-Shanklin, and Aaron D. Ames. "Robust bipedal locomotion: Leveraging saltation matrices for gait optimization." In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 12218–12225.

[194] Hongkai Dai and Russ Tedrake. "Optimizing robust limit cycles for legged locomotion on unknown terrain." In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 1207–1213.

[195] Ian R. Manchester, Mark M. Tobenkin, Michael Levashov, and Russ Tedrake. "Regions of attraction for hybrid limit cycles of walking robots." In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 5801–5806.

[196] Maegan Tucker and Aaron D. Ames. "An input-to-state stability perspective on robust locomotion." In: *IEEE Control Systems Letters* (2023).

[197] Pranav A. Bhounsule, Myunghee Kim, and Adel Alaeddini. "Approximation of the step-to-step dynamics enables computationally efficient and fast optimal control of legged robots." In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 83990. American Society of Mechanical Engineers. 2020, V010T10A050.

[198] Ayush Agrawal and Koushil Sreenath. "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." In: *Robotics: Science and Systems*. Vol. 13. Cambridge, MA, USA. 2017, pp. 1–10.

[199] Mohamadreza Ahmadi, Andrew Singletary, Joel W. Burdick, and Aaron D. Ames. "Safe policy synthesis in multi-agent POMDPs via discrete-time barrier functions." In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 4797–4803.

[200] Ryan K. Cosner, Preston Culbertson, Andrew J. Taylor, and Aaron D. Ames. "Robust Safety under Stochastic Uncertainty with Discrete-Time Control Barrier Functions." In: *arXiv preprint arXiv:2302.07469* (2023).

[201] Aaron D. Ames. "First steps toward automatically generating bipedal robotic walking from human data." In: *Robot Motion and Control 2011*. Springer, 2012, pp. 89–116.

[202] Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. "Whole-body motion planning with centroidal dynamics and full kinematics." In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2014, pp. 295–302.

[203] Ye Zhao, Benito R. Fernandez, and Luis Sentis. "Robust optimal planning and control of non-periodic bipedal locomotion with a centroidal momentum model." In: *The International Journal of Robotics Research* 36.11 (2017), pp. 1211–1242.

[204] Jessy W. Grizzle, Christine Chevallereau, and Ching-Long Shih. "HZD-based control of a five-link underactuated 3D bipedal robot." In: *2008 47th IEEE Conference on Decision and Control*. IEEE. 2008, pp. 5206–5213.

[205] *Supplementary Video for "Synthesizing Robust Walking Gaits via Discrete-Time Barrier Functions with Application to Multi-Contact Exoskeleton Locomotion."* `https://youtu.be/6aXsBKMxDH0`.

[206] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control." In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 5026–5033.

[207] Yukai Gong and Jessy W. Grizzle. "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion." In: *Journal of Dynamic Systems, Measurement, and Control* 144.12 (2022), p. 121006.

[208] Kejun Li, Maegan Tucker, Rachel Gehlhar, Yisong Yue, and Aaron D. Ames. "Natural Multicontact walking for Robotic Assistive Devices via Musculoskeletal Models and Hybrid Zero Dynamics." In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4283–4290. URL: http://dx.doi.org/10.1109/LRA.2022.3149568.

[209] Juraj Kabzan, Lukas Hewing, Alexander Liniger, and Melanie N. Zeilinger. "Learning-based model predictive control for autonomous racing." In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3363–3370.

[210] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. "Neural lander: Stable drone landing control using learned dynamics." In: *2019 international conference on robotics and automation (icra)*. IEEE. 2019, pp. 9784–9790.

[211] Xiaobin Xiong, Yuxiao Chen, and Aaron D. Ames. "Robust disturbance rejection for robotic bipedal walking: System-level-synthesis with step-to-step dynamics approximation." In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 697–704.

[212] Ivan Markovsky, Jan C. Willems, Sabine Van Huffel, and Bart De Moor. *Exact and approximate modeling of linear systems: A behavioral approach*. SIAM, 2006.

[213] Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC." In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 307–312.

[214] Randall T. Fawcett, Kereshmeh Afsari, Aaron D. Ames, and Kaveh Akbari Hamed. "Toward a data-driven template model for quadrupedal locomotion." In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 7636–7643.

[215] Randall T. Fawcett, Leila Amanzadeh, Jeeseop Kim, Aaron D. Ames, and Kaveh Akbari Hamed. "Distributed data-driven predictive control for multi-agent collaborative legged locomotion." In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 9924–9930.

[216] Osman Ulkir, Gazi Akgun, Ahad Nasab, and Erkan Kaplanoglu. "Data-Driven Predictive Control of a Pneumatic Ankle Foot Orthosis." In: *Advances in Electrical & Computer Engineering* 21.1 (2021).

[217] Julian Berberich, Johannes Köhler, Matthias A. Müller, and Frank Allgöwer. "Data-driven model predictive control with stability and robustness guarantees." In: *IEEE Transactions on Automatic Control* 66.4 (2020), pp. 1702–1717.

[218] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L. M. De Moor. "A note on persistency of excitation." In: *Systems & Control Letters* 54.4 (2005), pp. 325–329.

[219] Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control for grid-connected power converters". In: *IEEE Conference on Decision and Control*. 2019, pp. 8130–8135.

[220] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. "The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives." In: *IEEE International Symposium on System Integrations (SII)*. 2019.

[221] *Supplementary Video for "Data-driven Predictive Control for Robust Exoskeleton Locomotion."* https://www.youtube.com/watch?v=rgs36YXRb4I,

[222] Benjamin J. Stephens and Christopher G. Atkeson. "Push recovery by stepping for humanoid robots with force controlled joints." In: *2010 10th IEEE-RAS International conference on humanoid robots*. IEEE. 2010, pp. 52–59.

[223] Alexander W. Winkler, C. Dario Bellicoso, Marco Hutter, and Jonas Buchli. "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization." In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1560–1567.

[224] Tim Seyde, Jan Carius, Ruben Grandia, Farbod Farshidian, and Marco Hutter. "Locomotion planning through a hybrid bayesian trajectory optimization". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5544–5550.

[225] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." In: *Mathematical programming* 106 (2006), pp. 25–57.

[226] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. "OSQP: an operator splitting solver for quadratic programs." In: *Mathematical Programming Computation* 12.4 (2020), pp. 637–672. DOI: 10.1007/s12532-020-00179-2. URL: https://doi.org/10.1007/s12532-020-00179-2.

[227] *Supplementary Video for "Hybrid Data-Driven Predictive Control for Robust and Reactive Exoskeleton Locomotion Synthesis."* https://www.youtube.com/watch?v=0mVjbQzUGQ0,

[228] Kejun Li, Jeeseop Kim, Xiaobin Xiong, Kaveh Akbari Hamed, Yisong Yue, and Aaron D Ames. "Data-driven Predictive Control for Robust Exoskeleton Locomotion." In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 162–169. URL: https://doi.org/10.1109/IROS58592.2024.10802759.

[229] Ryan Batke, Fangzhou Yu, Jeremy Dao, Jonathan Hurst, Ross L. Hatton, Alan Fern, and Kevin Green. *Optimizing Bipedal Maneuvers of Single Rigid-Body Models for Reinforcement Learning*. en. arXiv:2207.04163 [cs]. July 2022. DOI: 10.48550/arXiv.2207.04163. URL: http://arxiv.org/abs/2207.04163 (visited on 07/30/2025).

[230] Kevin Green, Yesh Godse, Jeremy Dao, Ross L. Hatton, Alan Fern, and Jonathan Hurst. *Learning Spring Mass Locomotion: Guiding Policies with a Reduced-Order Model*. en. arXiv:2010.11234 [cs]. Mar. 2021. DOI: 10.48550/arXiv.2010.11234. URL: http://arxiv.org/abs/2010.11234 (visited on 05/06/2025).

[231] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. "Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots." en. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE, May 2021, pp. 2811–2817. ISBN: 978-1-7281-9077-8. DOI: 10.1109/ICRA48506.2021.9560769. URL: https://ieeexplore.ieee.org/document/9560769/ (visited on 05/13/2025).

[232] Ho Jae Lee, Seungwoo Hong, and Sangbae Kim. "Integrating Model-Based Footstep Planning with Model-Free Reinforcement Learning for Dynamic Legged Locomotion." en. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 11248–11255. ISBN: 979-8-3503-7770-5. DOI: 10.1109/IROS58592.2024.10801468. URL: https://ieeexplore.ieee.org/document/10801468/ (visited on 05/06/2025).

[233] Eduardo D. Sontag. "A Lyapunov-Like Characterization of Asymptotic Controllability." en. In: *SIAM Journal on Control and Optimization* 21.3 (May 1983), pp. 462–471. ISSN: 0363-0129, 1095-7138. DOI: 10.1137/0321028. URL: http://epubs.siam.org/doi/10.1137/0321028 (visited on 08/04/2025).

[234] Eduardo Sontag. *A 'universal' contruction of Artstein's theorem on nonlinear stabilization.*

[235] Zvi Artstein. "Stabilization with relaxed controls". en. In: *Nonlinear Analysis: Theory, Methods & Applications* 7.11 (Jan. 1983), pp. 1163–1173. ISSN: 0362546X. DOI: 10.1016/0362-546X(83)90049-4. URL: https://linkinghub.elsevier.com/retrieve/pii/0362546X83900494 (visited on 08/04/2025).

[236] Tyler Westenbroek, Ayush Agrawal, Fernando Castañeda, S. Shankar Sastry, and Koushil Sreenath. "Combining Model-Based Design and Model-Free Policy Optimization to Learn Safe, Stabilizing Controllers." en. In: *IFAC-PapersOnLine* 54.5 (2021), pp. 19–24. ISSN: 24058963. DOI: 10.1016/j.ifacol.2021.08.468. URL: https://linkinghub.elsevier.com/retrieve/pii/S240589632101243X (visited on 08/07/2025).

[237] Jason Choi, Fernando Castañeda, Claire J. Tomlin, and Koushil Sreenath. *Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions*. en. arXiv:2004.07584 [eess]. June 2020. DOI: 10.48550/arXiv.2004.07584. URL: http://arxiv.org/abs/2004.07584 (visited on 08/07/2025).

[238] Tyler Westenbroek, Fernando Castaneda, Ayush Agrawal, Shankar Sastry, and Koushil Sreenath. *Lyapunov Design for Robust and Efficient Robotic Reinforcement Learning*. en. arXiv:2208.06721 [cs]. Nov. 2022. DOI: 10.48550/arXiv.2208.06721. URL: http://arxiv.org/abs/2208.06721 (visited on 08/07/2025).

[239] Se Hwan Jeon, Steve Heim, Charles Khazoom, and Sangbae Kim. "Benchmarking potential based rewards for learning humanoid locomotion." In: *arXiv preprint arXiv:2307.10142* (2023).

[240] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." In: *arXiv preprint arXiv:1707.06347* (2017).

[241] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. ISSN: 0025-5610, 1436-4646. DOI: 10.1007/s10107-004-0559-y. URL: http://link.springer.com/10.1007/s10107-004-0559-y (visited on 08/27/2023).

[242] Joel Andersson, Joris Gillis, Greg Horn, James Rawlings, and Moritz Diehl. "CasADi - A software frameowrk for nonlinear optimization and optimal control." In: *Mathematical Programming Computation* 11 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4.

[243] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments." In: *IEEE Robotics and Automation Letters* 8.6 (2023), pp. 3740–3747. DOI: 10.1109/LRA.2023.3270034.

[244] *Supplementary video for "CLF-RL: Control Lyapunov Function Guided Reinforcement Learning."'* https://youtu.be/f8iuwgCZs3A.