Enabling Robust and User-Customized Bipedal Locomotion on Lower-Body Assistive Devices via Hybrid System Theory and Preference-Based Learning

Thesis by
Maegan Tucker

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy in Mechanical Engineering

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended May 12, 2023

Maegan Tucker
ORCID: 0000-0001-7363-6809

ACKNOWLEDGEMENTS

To everyone who I have met throughout my time at Caltech, thank you for making these last six years so wonderful. Importantly, to my fellow AMBER Lab members, you're all amazing and I can't wait to see what you accomplish, both personally and professionally. I would also like to thank the entire Wandercraft team for providing me with guidance and technical support throughout my PhD.

Also of particular note are the four advisory figures who have shaped my academic journey: Aaron Ames, Yisong Yue, Joel Burdick, and Richard Murray. First, Dr. Ames, you have given me every opportunity I have ever asked for and more. I sincerely thank you for your time and effort in shaping who I am as a researcher. Second, Dr. Yisong Yue, you have taught me so much about the field of machine learning and how to navigate the research world. I will forever be grateful for the guidance you have given me. Third, Dr. Joel Burdick, the AMBER lab students are so fortunate to have you as our academic uncle. You are a constant source of robotics wisdom, and I look forward to continuing to learn from you and your robotics empire. And lastly, Dr. Richard Murray, Caltech is lucky to have you as a pillar of the controls community. I so enjoyed being your TA and witnessing first-hand how you inspire young researchers.

Finally, I would like to dedicate this thesis to my family: to my mom, Anita Carson, who has always been a shining example of what it means to be a strong woman in academia; to my dad, Jeff Tucker, who instilled in me my love of engineering through countless visits to the Science Museum; to my sister, Audrey Tucker, who never ceases to make me laugh; and to my soon-to-be husband, Kyle Ahrens, who has patiently supported me throughout these past 10 years—from moving across the country with me to exploring local cuisine, from consoling me after paper rejections to celebrating with me after every academic milestone, and most importantly, for not being scared away many years ago by an ambitious (and sometimes feisty) wanna-be academic.

ABSTRACT

Practical robotic assistive devices have the potential to transform many aspects of our society, from enabling locomotive autonomy to facilitating rehabilitation. However, as is typically the case when having autonomous systems interact closely with humans, one must simultaneously solve multiple grand challenges. My work focuses specifically on 1) leveraging hybrid system theory to achieve stable and robust walking that generalizes well across various human models and environmental conditions, and 2) developing an online learning strategy to customize the experimental walking for individual user comfort. The presented methodology is grounded in realizing lower-body exoskeleton locomotion for subjects with motor complete paraplegia, with extensions to other robotic applications. The contributions are broken down as follows.

First, by leveraging tools from nonlinear control theory, I propose techniques for systematically addressing locomotive robustness. These techniques include: using saltation matrices to generate robust gaits with experimental demonstrations on the Atalante lower-body exoskeleton; and developing an input-to-state stability perspective to certify robustness to uncertain impact events. Importantly, these methods aim to better understand the mathematical conditions underlying robust locomotion—a necessary step towards realizing safe locomotion across varying human models and environmental conditions. Second, I develop a preference-based learning framework to explicitly optimize user comfort during exoskeleton locomotion (achieved using the aforementioned nonlinear control methodology) by learning directly from subjective feedback. This framework is implemented in real-world settings, including the clinical realization of user-preferred locomotion for two subjects with motor complete paraplegia. Third, the extensibility of this framework is demonstrated through three general robotic applications: tuning constraints of the gait generation optimization problem with demonstrations on a planar biped; tuning Lyapunov-based controller gains on a 3D biped; and tuning control barrier function parameters for performant yet safe exploration on a quadrupedal platform. Lastly, I discuss other relevant clinical considerations for lower-body assistive devices including how exoskeleton locomotion influences metabolic cost of transport, the study of latent factors underlying user-preferred walking, and embedding musculoskeletal models directly in the gait generation process.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] 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*, pages 1020–1033. PMLR, 2022. URL https://proceedings.mlr.press/v168/cosner22a.html.
M.T. developed the preference-based learning framework, conducted the learning simulations, and participated in the writing.

[2] 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)*, pages 10405–10411. IEEE, 2022. URL http://dx.doi.org/10.1109/ICRA46639.2022.9811541.
M.T. participated in the project conception, the writing, the experimental implementation, and developed the preference-based learning framework used in this work.

[3] Preston Culbertson, Ryan K. Cosner, Maegan Tucker, and Aaron D. Ames. Input-to-state stability in probability. *In Review*, 2023. URL https://arxiv.org/pdf/2304.14578.pdf.
M.T. participated in the project conception, the writing of the manuscript, and implemented the example of ISSp applied to the seven-link walker.

[4] Rachel Gehlhar, Maegan Tucker, Aaron J. Young, and Aaron D. Ames. A review of current state-of-the-art control methods for lower-limb powered prostheses. *Annual Reviews in Control*, 2023. URL https://doi.org/10.1016/j.arcontrol.2023.03.003.
M.T. wrote this review article in collaboration with Rachel Gehlhar and under the guidance of Dr. Aaron Ames and Dr. Aaron Young.

[5] Thomas Gurriet, Maegan Tucker, Alexis Duburcq, Guilhem Boeris, and Aaron D. Ames. Towards variable assistance for lower body exoskeletons. *IEEE Robotics and Automation Letters*, 5(1):266–273, 2019. URL http://dx.doi.org/10.1109/LRA.2019.2955946.
M.T. participated in the experiments, the writing, the experimental data analysis, and conducted the metabolic expenditure (indirect calorimetry) analysis.

[6] Jacques Kerdraon, Jean Gabriel Previnaire, Maegan Tucker, Pauline Coignard, Willy Allegre, Emmanuel Knappen, and Aaron D. Ames. Evaluation of safety and performance of the self balancing walking system Ata-

lante in patients with complete motor spinal cord injury. *Spinal cord series and cases*, 7(1):71, 2021. URL http://dx.doi.org/10.1038/s41394-021-00432-3.
M.T. participated in the data analysis and the writing of the manuscript.

[7] 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)*, pages 3212–3218. IEEE, 2021. URL http://dx.doi.org/10.1109/ICRA48506.2021.9560840.
M.T. participated in the project conception, the development of the learning framework, the writing of the manuscript, and led the experimental implementation.

[8] 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. *IEEE Robotics and Automation Letters*, 7(2):4283–4290, 2022. URL http://dx.doi.org/10.1109/LRA.2022.3149568.
M.T. participated in the project conception, the experimental implementation of the work, and the writing.

[9] Maegan Tucker and Aaron D. Ames. An input-to-state stability perspective on robust locomotion. *In Review*, 2023. URL https://arxiv.org/pdf/2303.10231.pdf.
M.T. conceived the project, participated in the development of the mathematical framework proposed in this work, the writing of the manuscript, and implemented the algorithms in simulation.

[10] 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 (IROS)*, pages 3423–3430. IEEE, 2020. URL http://dx.doi.org/10.1109/IROS45743.2020.9341416.
M.T. participated in the project conception and the development of the mathematical methods and algorithms used in this work, led the experimental implementation of the work, participated in the experiments, and participated in the writing.

[11] 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)*, pages 2351–2357. IEEE, 2020. URL http://dx.doi.org/10.1109/ICRA40945.2020.9196661.

M.T. participated in the project conception and the development of the learning framework, led the experimental implementation, participated in the experiments, and participated in the writing.

[12] 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. URL http://dx.doi.org/10.1109/ICRA48506.2021.9561515.
M.T. developed the learning algorithm, participated in the project conception and the writing, and led the experimental implementation.

[13] 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. URL https://arxiv.org/pdf/2209.10452.pdf.
M.T. participated in the project conception, implemented the mathematical methods and algorithms used in this work, led the experimental implementation, and participated in the writing of the manuscript.

RELEVANT PUBLISHED VIDEO CONTENT

[1] Supplemental Video for "Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits." https://youtu.be/c6a0kXMyML0, 2020.

[2] Supplemental Video for "Learning controller gains on bipedal walking robots via user preferences." https://youtu.be/jMX5a_6Xcuw, 2022.

[3] Supplemental Video for "Preference-based learning for exoskeleton gait optimization." https://youtu.be/-27sHXsvONE, 2020.

[4] Supplemental Video for "Preference-based learning for user-guided HZD gait generation on bipedal walking robots." https://youtu.be/rLJ-m65F6C4, 2021.

[5] Supplemental Video for "Robust bipedal locomotion: Leveraging saltation matrices for gait optimization." https://youtu.be/BZu-9UStG2E, 2023.

[6] Supplemental Video for "ROIAL: Region of interest active learning for characterizing exoskeleton gait preference landscapes." https://www.youtube.com/watch?v=04lMJmKmZrQ, 2021.

[7] Supplementary Video for "Natural multicontact walking for robotic assistive devices via musculoskeletal models and hybrid zero dynamics." https://www.youtube.com/watch?v=gOhZlTypNIs, 2022.

[8] Supplementary Video for "Safety-aware preference-based learning for safety-critical control." https://youtu.be/QEuwRDTG7TE, 2022.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Achieving stable, robust, and natural robotic bipedal locomotion in the real world is a challenging endeavor [160], especially when interacting closely with humans. This application is particularly relevant for the field of lower-body assistive devices such as powered prostheses [69, 209] and lower-limb exoskeletons [19, 114]. Specifically, one of the primary challenges when considering robotic assistive devices is that the system must be able to generalize robustly across a diverse set of human models [69], yet also be customized to individual users to maximize potential clinical benefits [175]. These considerations are exceedingly important for translating assistive devices to commercial settings.

Overall, my work approaches the challenges of generalization and customization by developing systematic methods for improving locomotive robustness via hybrid system theory and enabling fast user-customization via preference-based learning. Importantly, in the quest for a full-stack solution, my work integrates learning and control to take advantage of theoretical guarantees, informed by knowledge of the underlying dynamics, as well as a human's natural ability to judge *good* experimental behavior. Also, to ensure that the framework is practical, several real-world experiments are conducted, with the experiments involving humans being conducted across various subjects.

Concretely, the contributions of this thesis are four-fold: Chapter 2 presents an approach for systematically synthesizing robust walking gaits and proposes a theoretical perspective towards certifying locomotive robustness; Chapter 3 introduces a novel preference-based learning framework that leverages three forms of subjective human feedback to both optimize and characterize human preference with various demonstrations towards optimizing user comfort during exoskeleton locomotion; Chapter 4 extends the proposed learning framework towards three additional robotic applications to demonstrate its extensibility; and Chapter 5 presents three additional studies towards better understanding the clinical implications of lower-body assisted locomotion.

While the thesis is aimed generally at systematically realizing bipedal locomotion in the real world, the proposed methodologies are grounded in achieving

locomotive autonomy on the Atalante lower-body exoskeleton for subjects with complete motor paraplegia [114] (illustrated in Figure 1.1); this application is particularly important as lower-body exoskeletons have the potential of restoring autonomy to millions of individuals with ambulatory disabilities, thereby improving quality of life. Additional details and motivation for research involving lower-body exoskeleton control are provided later. Extensions of the presented work are also demonstrated for additional robotic applications. In general, by addressing locomotive robustness through hybrid system theory and user-customization through preference-based learning, my research works towards accelerating clinical implementation of robotic assistive technologies and enabling the use of these devices in everyday life.

Motivation for Research on Lower-Body Exoskeletons

While there is extensive literature on the biomechanics surrounding non-disabled human locomotion [225], it is poorly understood how to translate the ideas of natural and efficient walking to robotic platforms, especially in the context of robotic assistive devices which necessitate cooperation with human users. Of particular interest in this work are powered lower-body exoskeletons, which promise to augment our physical capabilities and restore mobility [85, 233]. For people with complete motor paraplegia, the benefits extend much further, including pressure relief, increased circulation, improved bone density, improved bowel and bladder function, and other general benefits associated with standing and walking [72, 105, 227].

Specifically, individuals who experience a spinal cord injury (SCI) often have a diminished ability to perform and benefit from regular exercise, which over time causes physical deconditioning [105]. In turn, this can lead to multisystem medical complications including respiratory, cardiovascular, urinary, and bowel complications, as well as spasticity, pressure ulcurs, osteoporosis, and worsened bone density [36, 62, 119, 144, 169]. Consequently, SCI rehabilitation is often aimed at facilitating physical exercise in a clinical setting [145], with two of the most common strategies being clinician-led mobility exercises [148, 177] and electrical stimulation [83, 86]. Throughout this thesis, specific interest is placed in the advancement of lower-body exoskeletons for full weight-bearing locomotion since it is a widely unexplored technology that has several promising health benefits associated with standing and walking [72, 105, 227]. It is even hypothesized that regular use of a lower-body exoskeleton for gait

training can promote changes in the nervous or muscular systems, and lead to activation of the neuromuscular system below the level of lesion [28].

Despite the clear motivation, existing lower-body exoskeletons for people with complete motor paraplegia have limited capabilities due to the challenge in tackling the issue of locomotive stability. Specifically, existing lower-body exoskeletons are either only capable of walking at slow speeds [25], or are reliant on either forearm crutches [27, 66] or overhead weight support [24, 30, 210] to maintain stability. These modifications result in locomotion that is either restricted to a clinical setting or that prevents proper arm motions during walking.

Even with the use of forearm-crutches, an intensive training period [115] with as many as 24 training sessions is often required to achieve independent locomotion [211]. During this training phase exoskeleton users learn how to shift their body weight to maintain stability during the walking. Instead, recent work leverages techniques from the bipedal robotics community [8] to achieve dynamic crutch-less exoskeleton walking [3, 79, 85], which reduces stress placed on the upper body and allows for proper arm motions during walking. The first clinical evaluation of this crutch-less exoskeleton locomotion reported only 6-12 training sessions required for most subjects to achieve independent locomotion [114]. Therefore, this thesis is dedicated to developing systematic methods of achieving robust and user-customized dynamically stable walking in an attempt to bring the benefits of crutch-less exoskeleton locomotion out of clinical settings and into the real world.

The Atalante Lower-Body Exoskeleton

The exoskeleton used in this work, Atalante, was developed by the company Wandercraft. This novel lower-body exoskeleton has 12 actuated joints as shown in Figure 1.1. Each leg of the exoskeleton consists of three actuated joints controlling the spherical motion of the hip, a single actuated joint for the flexion/extension motion of the knee, and two actuated joints for the hinge motion (inversion/eversion, dorsiflexion/plantarflexion) at the ankle. The joints controlling the motion of the hip and the knee are each actuated by brushless DC motors. The ankle joints have a more complex actuation mechanism that provides rotation in the sagittal plane and about the Henke axis. The position and velocity of each actuated joint is measured using a digital encoder.

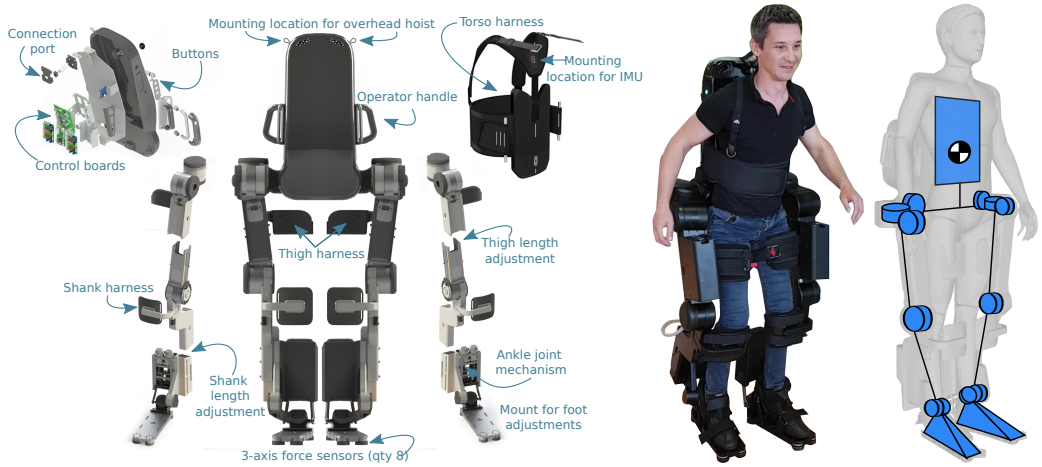


Figure 1.1: The Atalante lower-body exoskeleton designed by Wandercraft: (a) A breakdown of the Atalante exoskeleton components including patient-harnesses and electronics; (b) A patient inside the exoskeleton; and (c) A depiction of the locations of the 12 actuated joints.

Additionally, the exoskeleton has six Inertial Measurement Units (IMUs) that are positioned on the torso, the pelvis, the left/right shank, and the left/right foot. These IMUs are used to provide additional information about the attitude of the robot with respect to the world. To detect ground contact, four single-axis force sensors are attached to the bottom of each foot. All of the actuator and sensors are controlled by an embedded computer unit running a real-time operating system.

Other components of the exoskeleton indicated in Figure 1.1 include secure loops for mounting the exoskeleton to an overhead hoist, buttons to change the operating mode of the exoskeleton, a connection port to connect the exoskeleton to a computer, handles on either side of the exoskeleton for the operator to assist the exoskeleton if needed, thigh and shank harnesses to secure a patient to the exoskeleton, thigh and shank length adjustments to change the dimensions of the exoskeleton to match that of a patient, and a torso harness that a patient wears to secure their torso to the exoskeleton.

Thesis Overview

First, Chapter 2 begins by introducing the Hybrid Zero Dynamics method of gait generation and its application to realizing dynamically stable crutch-less exoskeleton locomotion on the Atalante exoskeleton. Then, two theoretical

tools are proposed for addressing the robustness of the resulting locomotion to model and environmental uncertainties. The first methods introduces a framework for improving the robustness of generated gaits by minimizing the induced matrix norm of the saltation matrix by including it in the objective function of the gait generation optimization problem. The second method offers an input-to-state stability perspective on locomotion with uncertain terrain, thereby providing mathematical certificates of forward invariance for nominal walking gaits to given distributions of ground heights.

Following, Chapter 3 presents the details of the proposed preference-based learning framework, which systematically optimizes exoskeleton walking for user comfort while characterizing the underlying reward function. This approach is advantageous compared to traditional numerical learning techniques since it only relies on subjective user feedback, which has been found to be more reliable than numerical mechanisms and does not require explicitly defining a reward function. The framework also addresses several challenges associated with human-in-the-loop learning, including learning from limited data due to time-intensive human subject experiments, ensuring user comfort and safety, accounting for noisy feedback, and exploring the large search space of parameterized walking gaits. The end-to-end solution is experimentally deployed on the Atalante lower-body exoskeleton to realize user-preferred exoskeleton locomotion for two subjects with complete motor paraplegia, illustrating the success of the approach towards addressing subjective human metrics such as user comfort.

In Chapter 4, the preference-based learning framework is extended to three additional robotic applications: identifying constraint bounds of the gait generation optimization problem to realize stable, efficient, and natural bipedal locomotion on the planar biped AMBER-3M; identifying controller gains for stable and robust walking on AMBER-3M and the 3D biped Cassie; and tuning parameters of a control barrier function for both performant and safe navigation of the Unitree quadrupedal platform A1. In general, these applications demonstrate that the proposed methodology is a powerful tool for leveraging a human's natural ability to judge *performant* robotic behavior.

Finally, in Chapter 5, three additional clinical considerations for lower-body assistive devices are explored. These considerations include: the effect of variable assistance of the Atalante exoskeleton on human users' metabolic cost of

transport; the study of potential latent factors underlying user-preferred exoskeleton locomotion; and the directly enforcement of musculoskeletal model constraints in the gait generation procedure to experimentally realize both stable and natural robotic-assisted locomotion on a dual-actuated prosthesis. Ultimately, all three studies aim to better understand the criteria underlying clinically beneficial lower-body assisted locomotion.

Related Work

Here I will briefly underscore how the thesis contributes to existing work.

Achieving Robust Bipedal Locomotion. A common approach towards achieving robust bipedal locomotion is to develop controllers and online planning methods that are capable of reacting to changes in the environment [82, 151]. While these methods successfully yield robust locomotion, I choose to focus on improving the robustness of the nominally generated reference gait since the existing work has shown that optimizing the robustness of nominal trajectories improves overall performance regardless of the chosen method of online stabilization [60, 141], and that online planning strategies can have unpredictable behavior without the use of a reference trajectory [49, 231].

Existing work towards robust gait synthesis includes evaluating metrics of robustness in the gait generation process such as the infinite-horizon cost-to-go resulting from early and late impact events [60]. However, these existing methods can be computationally expensive and do not scale easily to high-dimensional systems. Thus, in my work, I develop a method of generating robust limit cycles in a way that is scalable to high-dimensional systems (such as the 18 degree-of-freedom (DOF) Atalante exoskeleton shown in Fig. 1.1).

Specifically, my work towards robust gait synthesis leverages saltation matrices. These matrices, originally used in the field of non-smooth analysis, have been receiving growing attention and have been recently demonstrated towards state estimation for hybrid systems [121, 152] and hybrid event shaping [240]. Inspired by this recent research, our work similarly utilizes saltation matrices to generate stable periodic walking gaits that lead to robust behavior in the real world, even for high-dimensional systems; specifically, we propose including the induced norm of the extended saltation matrix in the HZD optimization problem.

Theoretical Certificates of Locomotive Robustness. One of the most common methods for analyzing the stability of nominal limit cycles is the method of Poincaré sections for systems with impulse effects [77]. However, in practice it can be computationally expensive to evaluate the Poincaré section for high-dimensional systems. Moreover, the Poincaré method of analysis is only informative for impact events infinitesimally close to the fixed point of the limit cycle. Thus, this method is not informative about real-world behavior since the timing of impact events for real-world systems is noisy.

Other existing work towards mathematically describing robust locomotion includes the development of numerical metrics such as the gait sensitivity norm [95] and the transverse linearization [137]. However, these tools do not provide theoretical certificates. As a step in this direction, input-to-state stability (ISS) [179] has been effectively leveraged in the context of robotic walking and running for uncertain dynamics [120, 134], but framing the robustness of walking gaits to uncertain environments remains an open problem.

Thus, the work presented in this thesis formulates a notion of robust walking that quantifies the gap between stability and robustness mathematically. The main result is the formulation of robust Lyapunov functions that certify the robustness of periodic orbits to given disturbances in the environment. These results are also extended to handle stochastic terrain.

User-Customization. Existing work towards user-customization in the context of lower-body assistive devices includes optimizing numerical factors such as body parameters and targeted walking speeds [162, 230], minimizing known numerical objective function terms such as metabolic cost [117, 236], and optimizing human preference during prosthesis locomotion [200] and spinal cord stimulation [183, 184]. While all of these existing techniques help to inform our approach towards lower-body exoskeleton user-customization, we will briefly discuss here why they are not immediately applicable.

First, most existing techniques towards optimizing numerical factors cannot be translated to subjective evaluations such as human preference. Moreover, it is important to note that several of these numerical factors are not appropriate for evaluating exoskeleton locomotion for patients with complete motor paraplegia. For example, while it is hypothesized that the metabolic cost of transport is a major factor underlying non-assisted walking preferences [31],

this metric is not appropriate for exoskeleton users since metabolic cost of transport remains constant unless variable assistance methods [80] are introduced as discussed in Section 5.

Most similar to the learning framework presented later in Chapter 3 is [200], which is also aimed at optimizing parameters of a lower-limb device for user comfort using a preference-based human-in-the-loop framework. However, our work differs from [200] in that we model the underlying utility function as a Gaussian process that is informed by up to three subjective feedback mechanisms. In comparison, [200] defined each action's objective value using the Copeland Score: the number of other actions that each action was preferred to on average. There is also existing work on characterizing the objective function underlying non-assisted human walking [31], but unfortunately, this understanding cannot be applied to exoskeleton locomotion since the addition of the exoskeleton alters the users perception of safety and exertion.

Lastly, the preference-based learning framework presented in Chapter 3 builds upon two existing algorithms. CoSpar and LineCoSpar build upon the Self-Sparring algorithm [185], a Bayesian dueling bandits approach that enjoys both competitive theoretical convergence guarantees and empirical performance. The dueling bandits setting [234] is an extension of the bandit setting in which only the comparative preference between two sampled actions is known. In comparison, the bandits setting is one in which the reward of the sampled action is known, along with that of the previously sampled actions. Self-Sparring learns a Bayesian posterior over each action's utility to the user and leverages Thompson sampling to draw multiple samples from the model's posterior to "duel" or "spar" via preference elicitation. To improve the sample efficiency of the algorithm, we modify it to include coactive feedback and ordinal labels.

The final algorithm presented in Chapter 3, ROIAL, builds upon the sampling strategy proposed in [34], in which preference queries that are easy for users to answer accurately are prioritized [33, 34, 98]. Our framework further adapts this acquisition function to only consider actions within a *region of interest* (ROI) in order to avoid sampling actions with low utilities. In the exoskeleton setting, the ROI is defined as all actions that do not make the user feel unsafe or uncomfortable. Notably, the ROIAL algorithm is the first to tackle such a region of interest active learning task.

Chapter 2

REALIZING STABLE AND ROBUST LOCOMOTION

Achieving stable and robust locomotion on legged systems is a challenging control task due to underactuation, power limitations, and ground impacts [160]. Two main approaches that have proven successful towards mitigating these challenges in the real world include: 1) generating stable reference trajectories [47, 149, 150, 159] and modifying these behaviors online using regulators (such as modifying the swing foot location based on lateral velocity [156, 160]); and 2) determining the desired behavior of the robot in real time using online planning via model predictive control [68, 75, 116, 176] or reinforcement learning [65, 129, 166, 172].

In this work, we utilize the Hybrid Zero Dynamics method of gait generation—a mathematical framework that has been repeatedly demonstrated towards experimentally realizing dynamically stable walking gaits on bipedal robots, encoded by stable periodic orbits. The HZD method [77, 220] guarantees that if there exists an exponentially stable periodic orbit on the zero dynamics submanifold of the full hybrid model, termed the *zero dynamics surface*, then the corresponding periodic orbit on the full model of an underactuated system is exponentially stabilizable [14]. In the case of full-actuation (as is the case for the exoskeleton Atalante with flat-foot walking), an extension of the HZD method, termed the *partial hybrid zero dynamics (PHZD)*, can be used that provides several advantages. The PHZD method has also been experimentally demonstrated on numerous robotic platforms [13, 90, 133, 158], including prostheses [238].

However, it is important to note that even though the HZD and PHZD methods have provable guarantees of stability, real-world disturbances and modeling errors void these guarantees and interfere with the resulting locomotive stability. Moreover, when a human is introduced into the system, the accuracy of the human-robot model is relatively low. This leads to the need for either extensive user tuning [208] or subject training [114] for stable locomotion in real world applications. For example, a clinical study that investigated the HZD method applied to the Atalante exoskeleton for 12 subjects with motor

complete paraplegia found that only 63.6% of the subjects who participated 12 training sessions were able to successfully completed the 10-meter Walk Test at Session 12 [114]. This 10-meter walk test consisted of 10 meters of exoskeleton walking without any assistance from clinicians or external devices. While this percentage is low, it is important to note that all 12 of the subjects were able to walk in the exoskeleton for at least 10 meters at some point throughout the 12 clinical sessions but with varying levels of assistance from clinicians.

Thus, a critical step towards improving locomotion in cases when a human is part of the system, such as the case with the Atalante exoskeleton, is to better define the theoretical conditions underlying provably robust and stable locomotion in the presence of real-world disturbances and uncertainties. Once we better understand these notions, we can develop systematic methods of achieving robust and stable locomotion on a variety of bipedal platforms. In this chapter, I will first present the preliminaries on the HZD and PHZD methods. Then, I will present two approaches towards better understanding and improving the robustness of nominal periodic orbits. Mainly, I consider robustness in terms of uncertain impact events, caused by either uncertain terrain or model uncertainties.

2.1 Preliminaries on the Hybrid Zero Dynamics Method

In this section, the fundamental concepts associated with the HZD method of gait generation are outlined [220], along with the details of extending to the PHZD method in the case of the Atalante exoskeleton.

Discrete Structure. As walking is comprised of alternating sequences of continuous dynamics followed by intermittent discrete impact events, it is naturally modeled as a hybrid system [11, 12, 78, 131, 132, 220, 221]. Moreover, since steady state bipedal walking is naturally periodic, it can be represented using a temporal ordering of events (changes in contact) [100]. This temporal ordering is modeled as a hybrid system in which the domain graph is a *directed cycle*; a cyclic oriented graph with a set of nodes and a set of edges. The discrete structure can be simplified even further by assuming flat-foot symmetric walking, which reduces the system to a single continuous domain and a single discrete impact phase. Thus, the directed graph is reduced to one node (the swing phase) and one edge (swing foot ground impact). In the following discussion we will only consider this simplified directed graph, but

it is important to note that the presented methodology can be extended to more complex walking behaviors with diverse collections of contact sequences, including natural heel-toe roll during walking [159, 174, 237]. This extension only requires the introduction of a directed graph describing how the continuous and discrete domains are related. In Chapter 5 a multi-domain HZD framework will be leveraged to realize multi-contact walking on the AMRPO3 prosthesis.

Hybrid Systems. Consider a hybrid control system with states $x \in \mathcal{X} \subset \mathbb{R}^n$ and a control input $u \in \mathcal{U} \subset \mathbb{R}^m$. Given a continuously differentiable function[1] $h : \mathcal{X} \to \mathbb{R}$, let $\mathcal{D} \subset \mathcal{X}$ denote the admissible domain on which the continuous-time dynamics evolve and $\mathcal{S} \subset \mathcal{D}$ denote the *guard* (also commonly called the *switching surface*), defined as:

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

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

For states $x^- \in \mathcal{S}$, a discrete impact map $\Delta : \mathcal{S} \to \mathcal{D}$, termed the *reset map* is applied. Thus, the complete hybrid system can be modeled as:

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

where (2.3) and (2.4) denote the continuous-time and discrete-time dynamics, respectively. Note that the continuous-time dynamics are assumed to be control affine with continuously differentiable functions $f : \mathbb{R}^n \to \mathbb{R}^n$ (termed the drift vector) and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ (termed the actuation matrix). It is assumed (as is typical) that all quantities in $\mathcal{HC}$ are locally Lipshitz continuous, e.g., the impact map Δ is locally Lipschitz. This follows from the assumption of perfectly plastic impacts [73]. Importantly, note that for impact maps based on rigid-body contacts [99], the impact map does not depend on the ground height.

Generalized Coordinates. For a robotic system with n degrees of freedom, the system state is described using configuration coordinates $q \in \mathcal{Q} \subset \mathbb{R}^n$, with the full system state $x = (q^T, \dot{q}^T)^T \in \mathcal{X} \subset \mathsf{T}\mathcal{Q}$. Moreover, for a robotic system modeled as a *floating-base* model, the configuration coordinates are be

[1] Note that h must be selected such that it does not lie within the null space of the actuation matrix, i.e., $L_g h(x) \neq 0$

further broken down as $q = (p, \phi, q_b) \in \mathcal{Q} \subset \mathbb{R}^n$. Here, $(p, \phi) \in SE(3)$ denotes the Cartesian position $p \in \mathbb{R}^3$, and orientation $\phi \in SO(3)$, of the floating-base frame relative to the world frame for a robot navigating the 3D world. Alternatively, for a planar robot, the floating-base frame would instead live in $SE(2)$ with $p \in \mathbb{R}^2$ and $\phi \in SO(2)$. The remaining coordinates describe the local coordinates of the robot, $q_b \in \mathcal{Q}_b \subset \mathbb{R}^{n_b}$. Commonly, the local coordinates represent joint angles of the robotic. It is also possible to model the robot using a *pinned model* whereby the holonomic constraints enforcing contact with the ground are imposed at the contact locations [221].

Hybrid Zero Dynamics and Virtual Constraints. The dynamics of walking can be separated into those that can be controlled using actuation, and those that are uncontrollable – termed the zero dynamics. The main idea of the HZD method is that exponentially stable periodic orbits of the zero dynamics correspond to exponentially stabilizable orbits of the full hybrid system model [220]. This lower-dimensional surface, termed the zero dynamics surface [104], follows from classic results in nonlinear control which state that, with the proper choice of coordinates, the ordinary differential equations that model the natural continuous dynamics of a system can be represented on the zero dynamics surface. When this approach is extended to systems with discrete impacts, the surface is termed a *hybrid zero dynamics* surface.

Consider the *zero dynamics surface*:

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

where $y_\alpha : \mathcal{X} \to \mathbb{R}^m$ denotes the set of *outputs* (also called *virtual constraints*). When enforced, the virtual constraints regulate the behavior of the system to some desired motion. Importantly, the choice of virtual constraints y_α also determines the shape of the manifold $\mathcal{Z}_\alpha$, i.e., the underactuated coordinates can be indirectly shaped through the virtual constraints. Specifically, the virtual constraints are defined as:

$$y_\alpha(x) = y^a(x) - y^d_\alpha(\tau(x)), \tag{2.5}$$

with $y^a : \mathcal{X} \to \mathbb{R}^m$ denoting the actual measured outputs of the system and $y^d_\alpha : \mathbb{R} \to \mathbb{R}^m$ denoting the desired outputs parameterized by some function $\tau(x) : \mathcal{X} \to \mathbb{R}$ that is monotonically increasing across the limit cycle. The desired outputs are enforced on the system by driving $y_\alpha \to 0$ exponentially through

the use of a stabilizing controller $u^*(x)$, e.g., given by feedback linearization or control Lyapunov functions [14, 16, 221]. The end result is the *closed-loop dynamics*: $\dot{x} = f_{\text{cl}}(x) = f(x) + g(x)u^*(x)$.

Note that we intentionally use the subscript α to denote the specific choice of desired outputs (and thus also the corresponding zero dynamics surface) since a popular and convenient approach towards designing y_α^d is through the use of Bézier polynomials parameterized by the set of Bézier coefficients $\alpha \in \mathbb{R}^{n_o \times B+1}$ with $n_0 \in \mathbb{N}$ denoting the total number of outputs. Note that the number of outputs depends on both the degrees of freedom of the system and the holonomic constraints defined in each domain of the hybrid system. Briefly, the total number of outputs and holonomic constraints must be less than or equal to the total degrees of freedom of the system. The number of outputs must also not exceed the degrees of actuation.

The desired behavior for each output (indexed by $v \in [1, \dots, n_o]$) is described by an B^{th} order Bézier polynomial:

$$b(\tau, \alpha_v) \triangleq \sum_{k=0}^{B} \alpha_v[k] \frac{B!}{k!(B-k!)} \tau^k(x)(1-\tau(x))^{B-k}, \tag{2.6}$$

where $\alpha_v = [\alpha_v[0], \dots, \alpha_v[B]] \in \mathbb{R}^{B+1}$ denotes the set of Bézier control points. Advantages of Bézier polynomials include their ability to describe smooth motions using a finite set of control points, the property that the trajectories are bounded by the control points, and the explicit representation of their derivatives [1, 165]. Furthermore, τ is typically defined as:

$$\tau(x) \triangleq \frac{\theta(x) - \theta^+}{\theta^- - \theta^+}, \tag{2.7}$$

with $\theta : \mathcal{X} \to \mathbb{R}$ being a monotonically increasing phase variable (for example, θ is later selected to be the linearized horizontal hip position), and $\theta^+ \triangleq \theta(x^+)$ and $\theta^- \triangleq \theta(x^-)$ respectively denoting the phase variable evaluated at the beginning and end of the limit cycle. Using this definition, the function τ is restricted to the range $[0, 1]$, i.e., $\tau : \mathcal{X} \to [0, 1]$.

Hybrid Invariance Condition. To guarantee stability of a hybrid system, the virtual constraints must be specifically chosen such that when enforced, discrete impact events do not cause the system to be thrown off the zero dynamics surface. In other words, the system must be impact invariant. This condition

is encoded by the following hybrid invariance condition (also known as the *HZD condition*):

$$\Delta(\mathcal{S} \cap \mathcal{Z}_\alpha) \subset \mathcal{Z}_\alpha. \tag{2.8}$$

Trajectory Optimization. The final step of the HZD method is to select α such that when the virtual constraints are driven to zero ($y_\alpha \to 0$), the closed-loop dynamics are stable, the system is hybrid invariant, and the physical constraints are satisfied. To obtain α, we use a direct collocation based optimization algorithm, FROST [87], which has been previously utilized for efficient gait generation of walking [160], running [134], and quadrupedal locomotion [136]. Direct collocation is an implicit Runge–Kutta method to approximate the numerical solution of certain dynamical systems, namely differential-algebraic equations and partial differential equations. The trajectory optimization problem is stated as:

HZD Optimization:

$$\begin{aligned} \{\alpha^*, X^*\} = \underset{\alpha, X}{\text{argmin}}\ & \Phi(X) && (2.9)\\ \text{s.t.}\quad & \dot{x} = f_{cl}(x) && \text{(Closed-loop Dynamics)}\\ & \Delta(\mathcal{S} \cap \mathcal{Z}_\alpha) \subset \mathcal{Z}_\alpha && \text{(HZD Condition)}\\ & X_{\min} \preceq X \preceq X_{\max} && \text{(Decision Variables)}\\ & c_{\min} \preceq c(X) \preceq c_{\max} && \text{(Physical Constraints)} \end{aligned}$$

---

where $X = (x_0, ..., x_N, T) \in \mathbb{R}^{n_d}$ is the collection of $n_d \in \mathbb{N}$ decision variables with $x_i \in \mathcal{X}$ denoting the i^{th} discretization of the state and $T \in \mathbb{R}_{>0}$ denoting the limit cycle duration, $\Phi : \mathbb{R}^{n_d} \to \mathbb{R}_{\geq 0}$ is the cost function, and $c : \mathbb{R}^{n_d} \to \mathbb{R}^{n_p}$ is the set of $n_p \in \mathbb{N}$ physical constraints. These physical constraints are included in every gait generation framework to encode the physical laws of real-word, such as the friction cone condition, workspace limit, and motor capacity [159]. The end result of the optimization problem is a stable periodic solution to the walking dynamics that is parameterized by some static set of Bézier coefficients $\alpha^* \in \mathbb{R}^{n_o \times B+1}$.

In practice, the optimization problem can be solved using any nonlinear programming (NLP) solver such as IPOPT [214]. Additionally, a convenient way to formulate the optimization problem is using the MATLAB toolkit FROST [87]. In general, FROST uses a direct collocation-based gait optimization to

design periodic gaits for a hybrid system model. Other methods of solving the nonlinear optimization problem also exist, such as direct shooting methods [89]. However, direct collocation was found to be the fastest and most computationally inexpensive way of solving optimization problem [91].

Extension to the Partial Hybrid Zero Dynamics. As mentioned earlier, in the case of full-actuation, the *Partial Hybrid Zero Dynamics* (PHZD) method [8] is leveraged. This approach utilizes a specific choice of nonlinear control to drive the system to a *partial* zero dynamics surface that is both attractive and forward invariant. The impact invariance of this surface results in a theoretical guarantee that a stable periodic orbit *must* exist on the reduced-order PHZD surface and therefore the full-order dynamics (i.e., there *must* exist a stable walking gait). The PHZD method also enables the regulation of forward progression via the velocity regulating outputs. This specific choice of nonlinear control will be outlined next for the Atalante exoskeleton.

The PHZD Method applied to the Atalante exoskeleton

The procedure for obtaining stable locomotion on the Atalante exoskeleton is illustrated in Figure 2.1. As discussed earlier, a unique aspect of achieving stable locomotion on the Atalante exoskeleton is that the model of the human-exoskeleton system changes for each individual user. Thus, we will first discuss how we obtain this patient-specific model. Following, we will discuss details of the PHZD method applied to the Atalante exoskeleton, and how the obtained periodic orbits are enforced on the physical robot.

Patient-Exoskeleton Model Generation. A model of each patient is generated to account for each person's unique physical characteristics. Key measurements of each patient are made: height, mass, thigh length, and shank length. The thigh length is approximated by the measurement between the gluteus maximus and the patella when the patient is in a seated position. The shank length is approximated as the measurement between the femoral condyles and the ground when the patient is in a seated position. The measured thigh length and shank length are used to adjust the leg lengths of the exoskeleton to match that of the patient. The patient model is then created as follows. First, the total height of the patient is used to extrapolate the length of each segment of the patient model. The segments were chosen to be: head, arms and trunk (HAT); Pelvis; Left Thigh; Left Shank; Left Foot; Right Thigh; Right Shank;

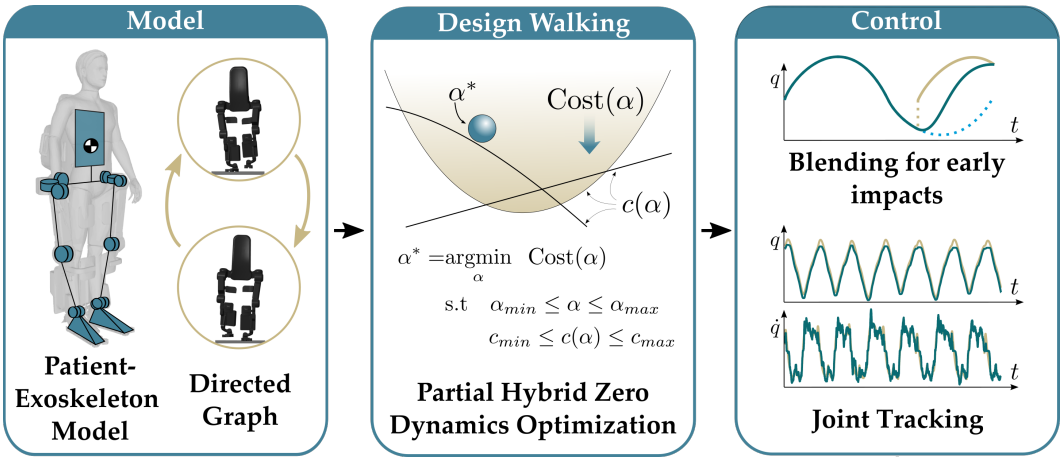


Figure 2.1: The diagram illustrates the procedure for achieving dynamically stable crutch-less exoskeleton locomotion using the Partial Hybrid Zero Dynamics (PHZD) framework. First, a model of the fused patient-exoskeleton system is constructed, along with a directed graph describing the hybrid system. Then, a nominal walking gait is generated using a nonlinear optimization problem. Lastly, the desired behavior is enforced on the physical robot using joint-level tracking with blending to remove discontinuities due to early impact events.

Right Foot. These extrapolations were first derived by Drillis and Contini [64]. Using these segment lengths, the center of mass (COM) and inertia for each body segment are then calculated using anthropometric data [225]. The inertia and COM of each segment is given with respect to the proximal end of that segment. The inertia and COM of each segment are then combined with those of the corresponding segments of the rigid body exoskeleton model to form the patient-exoskeleton model.

This combined human-exoskeleton model can be mathematically represented as a rigid-body system with configuration coordinates $q = (p, \phi, q_b) \in \mathcal{Q} \subset \mathbb{R}^{18}$. Note that for Atalante, $q \subset \mathbb{R}^{18}$ since the system has 18 degrees of freedom, and $q_b \subset \mathbb{R}^{12}$ denoting the 12 actuated joint angles. The system is fully actuated when one foot is flatly in contact with the ground, as enforced by holonomic constraints [78].

Constructing the Hybrid Control System. Given the mass and inertia properties of each rigid link in the patient-exoskeleton model, the continuous dy-

namics can be expressed as solution to the Euler-Lagrange equations:

$$\dot{x} = \underbrace{\begin{bmatrix} \dot{q} \\ -D(q)^{-1}H(q,\dot{q}) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ D(q)^{-1}B \end{bmatrix}}_{g(x)} u. \tag{2.10}$$

where $D : \mathcal{Q} \to \mathbb{R}^{18\times 18}$ is the mass-inertia matrix, $H : \mathsf{T}\mathcal{Q} \to \mathbb{R}^{18}$ contains the Coriolis and gravity terms, $B \in \mathbb{R}^{18\times 12}$ is the actuation matrix, and $u \in \mathbb{R}^{12}$ is the control input.

Next, we define the guard function for the guard defined in (2.2) as $h(x) \triangleq p^z_{sw}(x)$, with $p^z_{sw} : \mathcal{X} \to \mathbb{R}$ denoting the vertical position of the swing foot. Thus, the guard can be interpreted as the set of states when the swing foot strikes the ground with a negative velocity. As the system flows into the guard, $n_c \in \mathbb{N}$ holonomic constraints $\eta_{st} : \mathcal{Q} \to \mathbb{R}^{n_c}$ are enforced in the continuous domain succeeding the impact event. For 3D flat-foot exoskeleton walking, there are six individual holonomic constraints ($n_c = 6$) which are described as:

$$\eta_{st}(q) = \begin{bmatrix} p_{st}(q) \\ \phi_{st}(q) \end{bmatrix} = \text{constant}, \tag{2.11}$$

where $p_{st}(q) \in \mathbb{R}^3$ denotes the euclidean position of the stance foot and $\phi_{st}(q) \in SO(3)$ denotes the orientation.

As the system flows into the guard, it also undergoes a discrete jump in the state that can be expressed using the momentum transfer equation [99]:

$$D(q^-)(\dot{q}^+ - \dot{q}^-) = J_c(q^-)^\top \delta F \tag{2.12}$$

where $(q^{-T}, \dot{q}^{-T})^T \in \mathsf{T}\mathcal{Q}$ and $(q^{+T}, \dot{q}^{+T})^T \in \mathsf{T}\mathcal{Q}$, respectively, represent the state just before and after impact, $\delta F \in \mathbb{R}^{n_c}$ is the impulse force of the impact event, and $J_c : \mathcal{Q} \to \mathbb{R}^{n_c\times 18}$ is the Jacobian of the holonomic constraints.

Enforcing the holonomic constraints through impacts can equivalently be represented as $J_c(q^-)\dot{q}^+ = 0$. With this, we can rewrite the impact equation [73] as:

$$\begin{bmatrix} D(q^-) & -J_c(q^-)^\top \\ J_c(q^-) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \delta F \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix}. \tag{2.13}$$

Noting that the configuration is continuous through impact, solving for $\dot{q}^+$ in (2.13) allows us to explicitly define the reset map, i.e., the map describing the

discrete event at footstrike [78], as:

$$x^+ = \Delta(x^-) \triangleq \begin{bmatrix} Rq^- \\ R(-D^{-1}J_c^\top (J_c D^{-1} J_c^\top)^{-1} J_c + I)\dot{q}^- \end{bmatrix} \tag{2.14}$$

where the dependence of D and J_c on q^- is suppressed and $R \in \mathbb{R}^{18\times 18}$ denotes the relabeling matrix which is used to maintain state consistency between domains. Explicitly, for symmetric walking on the Atalante exoskeleton, the relabeling matrix is:

$$R \triangleq \left[\begin{array}{cc|cc} \begin{matrix} 1 & & & & & \\ & -1 & & & & \\ & & 1 & & & \\ & & & -1 & & \\ & & & & 1 & \\ & & & & & -1 \end{matrix} & & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{6\times 12}$}} \\ & & & & & \\ \hline & & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{6\times 6}$}} & \begin{matrix} -1 & & & & & \\ & -1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & -1 \end{matrix} & \\ \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}_{12\times 6}$}} & & & & \\ \hline \multicolumn{2}{c|}{} & \begin{matrix} -1 & & & & & \\ & -1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & -1 \end{matrix} & \multirow{2}{*}{$\mathbf{0}_{6\times 6}$} \\ \multicolumn{2}{c|}{} & & & & \end{array}\right], \tag{2.15}$$

in the case where the local coordinates q_b are defined using the following joint ordering (abbreviations are provided in the caption of Figure 2.2): LFH, LTH, LSH, LSK, LSA, LHA, RFH, RTH, RSH, RFK, RSA, RHA. Note here that the non-negative entries only apply to joints in the sagittal plane.

We are now fully equipped to mathematically describe the hybrid system $\mathcal{HC}$ as defined in (2.3) and (2.4) using the continuous-time dynamics provided in (2.10) and the impact map provided in (2.14).

Selecting Virtual Constraints. As mentioned, since the Atalante exoskeleton is fully actuated, it leverages an extension of the HZD method called the PHZD method. Thus, as is typical for the PHZD method, the virtual constraints are defined using a specific form that includes both relative degree 1 and relative degree 2 outputs:

$$y_\alpha(x) \triangleq \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} y_1^a(q,\dot{q}) - v_d \\ y_2^a(q) - y_2^d(\tau(x),\alpha) \end{bmatrix} \in \mathbb{R}^m \tag{2.16}$$

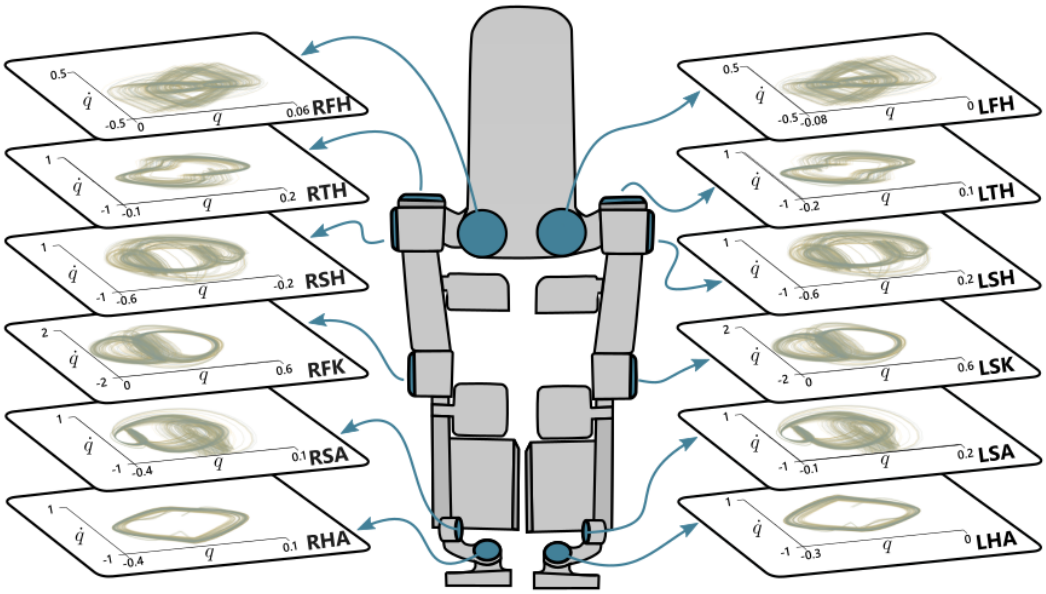


Figure 2.2: An example gait for the Atalante exoskeleton. The gait is encoded via periodic joint-level trajectories, plotted here as phase portraits. Each nominal gait consists of 12 trajectories, one for each actuated joint: left frontal hip (LFH), left transverse hip (LTH), left sagittal hip (LSH), left sagittal knee (LSK), left sagittal ankle (LSA), left ankle aligned with the Henke axis (LHA). The corresponding joints are also labeled for the right leg as: RFH, RTH, RSH, RSK, RSA, and RHA.

where $y_1(x)$ regulates the forward velocity and $y_2(x)$ regulates the positions of the robot to move in a synchronous fashion. More specifically, $y_1^a(q, \dot{q})$ is the forward hip velocity, while $y_2^a(q)$ consists of the angular positions of the exoskeleton joints and torso as listed in [3]. The output y_1^a is driven to a constant desired velocity v_d, while $y_2^a(q)$ is driven to a vector of desired trajectories, $y_2^d(\tau(x), \alpha)$. The trajectories $y_2^d : [0, 1] \times \mathbb{R}^{12\times M+1} \to \mathbb{R}^{12}$ are represented using a Bézier polynomial as defined in (2.6) with coefficients $\alpha \in \mathbb{R}^{12\times M+1}$:

$$y_2^d(\tau(x), \alpha) \triangleq [b(\tau(x), \alpha_1), \ldots, b(\tau(x), \alpha_{12})]^T,$$

with the state-based timing variable as defined in (2.7) with the phase variable selected to be $\theta(x) \triangleq \delta p_{hip}(x)$. Here, $\delta p_{hip} : \mathcal{X} \to \mathbb{R}$ returns the horizontal hip position relative to the stance foot position in the sagittal plane.

Driving the Virtual Constraints to Zero.. The continuous control objective for the system is formulated as driving the virtual constraints to zero ($y_\alpha(x) \to 0$).

This is typically accomplished using a feedback linearizing controller [12]. This control law, under the conditions given in [78], drives the virtual constraints to zero exponentially. Concretely, the virtual constraints are first differentiated along solutions of the continuous dynamics until the control input appears, as explained in [9]. First, noting that $p_{hip}(x)$ can be computed only using the configuration coordinates, it will be written as $p_{hip}(q)$ for the following derivation. Also, noting that $y_1^a(q,\dot{q}) = \frac{\partial \delta p_{hip}(q)}{\partial q}\dot{q}$, differentiating y_1 yields:

$$\dot{y}_1(x) = \underbrace{\frac{\partial}{\partial q}\Big(\frac{\partial \delta p_{hip}(q)}{\partial q}\dot{q}\Big)\dot{q} + \frac{\partial \delta p_{hip}(q)}{\partial q}\Big[-D^{-1}(q)H(q,\dot{q})\Big]}_{L_f y_1(x)} + \underbrace{\frac{\partial \delta p_{hip}(q)}{\partial q}D^{-1}(q)B}_{L_g y_1(x)}\, u \tag{2.17}$$

where $L_f y_1(x)$ and $L_g y_1(x)$ denote Lie derivatives of y_1 with respect to the vector field (f,g). Differentiating $y_2(x)$ (twice) until the control input appears results in:

$$\ddot{y}_2(x) = \underbrace{\frac{\partial}{\partial q}\Big(\frac{\partial y_2(x)}{\partial q}\dot{q}\Big)\dot{q} + \frac{\partial y_2(x)}{\partial q}\Big[-D^{-1}(q)H(q,\dot{q})\Big]}_{L_f^2 y_2(x)} + \underbrace{\frac{\partial y_2(x)}{\partial q}D^{-1}(q)B}_{L_g L_f y_2(x)}\, u \tag{2.18}$$

Combining these expressions yields:

$$\begin{bmatrix}\dot{y}_1(x)\\ \ddot{y}_2(x)\end{bmatrix} = \underbrace{\begin{bmatrix}L_f y_1(x)\\ L_f^2 y_2(x)\end{bmatrix}}_{L_f y_\alpha(x)} + \underbrace{\begin{bmatrix}L_g y_1(x)\\ L_g L_f y_2(x)\end{bmatrix}}_{A(q,\dot{q})} u. \tag{2.19}$$

Thus, to drive y_α to zero, consider the nonlinear feedback linearizing controller:

$$u^*(x) = A(x)^{-1}\left(-L_f y_\alpha(x) + \begin{bmatrix}-\frac{1}{\varepsilon}y_1(x)\\ -\frac{K_p}{\varepsilon^2}y_2(x) + \frac{K_d}{\varepsilon}\dot{y}_2(x)\end{bmatrix}\right). \tag{2.20}$$

Applying this controller to the system yields the output dynamics:

$$\underbrace{\begin{bmatrix}\dot{y}_1(x)\\ \dot{y}_2(x)\\ \ddot{y}_2(x)\end{bmatrix}}_{\dot{\eta}_\alpha} = \underbrace{\begin{bmatrix}-\frac{1}{\varepsilon} & 0 & 0\\ 0 & 0 & I\\ 0 & -\frac{K_p}{\varepsilon^2} & -\frac{K_d}{\varepsilon}\end{bmatrix}}_{A_{cl}} \underbrace{\begin{bmatrix}y_1(x)\\ y_2(x)\\ \dot{y}_2(x)\end{bmatrix}}_{\eta_\alpha}, \tag{2.21}$$

which is a stable linear system. Therefore, achieves the desired objective of driving y_1, y_2 and $\dot{y}_2$ to zero exponentially, i.e., the virtual constraints are driven to zero. As a result, the (continuous) dynamics of the system are driven to the *partial hybrid zero dynamics* surface defined by:

$$\mathcal{PZ}_\alpha = \{x \in \mathcal{D} \mid y_2(x) = 0, \dot{y}_2(x) = 0\}. \tag{2.22}$$

Additionally, this surface is rendered forward invariant by the nonlinear controller.

Finding an attractive and forward invariant periodic orbit. In the case of exoskeleton walking, a stable periodic solution is a trajectory that yields continuous walking in the presence of swing foot impacts. Similar to the HZD condition, the orbit is impact invariant if it stays on the partial zero dynamics surface $\mathcal{PZ}_\alpha$ after an impact event, i.e.:

$$\Delta(\mathcal{S} \cap \mathcal{PZ}_\alpha) \subset \mathcal{PZ}_\alpha. \tag{2.23}$$

To solve for the Bézier coefficients α^* that yield an impact invariant periodic orbit, we formulate an optimization problem as in (2.9) but with the replacement of the PHZD condition (2.23) and the closed-loop dynamics taken to be $\dot{\eta}_\alpha = A_{cl}\eta_\alpha$ (2.21). Additionally, the cost function $\Phi(X)$ is typically taken to be the mechanical cost of transport [159]. Lastly, examples of physical constraints $c(X)$ for the Atalante exoskeleton are provided in Appendix III of [3] and additional details on the formulation of the optimization problem are provided in [79, 91].

Trajectory tracking on the Atalante exoskeleton. In practice, the walking behavior as encoded by the generated trajectories (such as the ones illustrated in Figure 2.2) is enforced on the exoskeleton using the following control structure. At the low level, PD control of the joints is implemented to track the desired joint angles and velocities as obtained from the generated gaits. This low-level control is responsible for translating these desired joint angles and velocities into motor torque commands. The high-level controller is responsible for adjusting the joint targets based on key features such as torso angle and foot contact, e.g., if the foot strikes early or late, the current configuration is smoothly transitioned into the target configuration of the next step. Finally, the lowest level of control uses a feedback control loop to generate current

commands at the joint level based on the motor torque commands. These current commands are sent to the motor controllers that handles the low level control of the motors.

2.2 Improving Robustness via Saltation Matrices

The ability to generate robust walking gaits on bipedal robots is key to their successful realization on hardware. To this end, the first part of my thesis extends the method of (Partial) Hybrid Zero Dynamics (HZD)—which traditionally only accounts for locomotive stability via periodicity constraints under perfect impact events—through the inclusion of the saltation matrix with a view toward synthesizing robust walking gaits. The proposed approach is motivated by previous research, which has shown that optimizing the robustness of nominal trajectories improves overall performance regardless of the chosen method of online stabilization [60, 141], and that online planning strategies can have unpredictable behavior without the use of a reference trajectory [49, 231].

As mentioned in Chapter 1, existing work towards generating robust limit cycles [60] does not scale easily to high-dimensional systems. Thus, a particular goal of the work presented in this section is to develop a method that is scalable to high-dimensional systems. Briefly, by jointly minimizing the norm of the extended saltation matrix and the torque of the robot directly in the gait generation process, we demonstrate that the synthesized gaits are more robust than gaits generated with either term alone. These results are shown in simulation and on hardware for the 7-DOF AMBER-3M planar biped and the 18-DOF Atalante lower-body exoskeleton (both with and without a human subject). The end result is experimental validation that combining saltation matrices with HZD methods produces more robust bipedal walking in practice.

Preliminaries on Saltation Matrices

The saltation matrix is a standard tool used in the field of non-smooth analysis that describes a systems sensitivity to discontinuities (otherwise called 'saltations' or 'jumps') [125]. Typically, the saltation matrix $S(t_i, x(t_i)) \in \mathbb{R}^{2n\times 2n}$ is defined at time $t_i \in \mathbb{R}^+$ for states in the guard $x(t_i) \in \mathcal{S}$, by the relationship:

$$\delta x(t_{i+1}) = S(t_i, x(t_i))\delta x(t_i), \tag{2.24}$$

where $x(t_{i+1}) \in \Delta(\mathcal{S})$ denotes the post-transition state. Since we are interested in evaluating the saltation matrix for pre-computed gaits with a known impact

time $T \in \mathbb{R}_{>0}$, we will specifically derive the saltation matrix $S \triangleq S(T, x^-)$ for the relationship:

$$\delta x^+ = S\delta x^-, \tag{2.25}$$

where $\delta x^-, \delta x^+ \in \mathbb{R}^{2n}$ define the variation in the pre- and post- impact states, respectively, at the time of impact T. A visualization of these variations are provided in Figure 2.3. Explicitly, these state variations are defined as:

$$\delta x^+ \triangleq \tilde{x}^+ - x^+, \quad \delta x^- \triangleq \tilde{x}^- - x^-, \tag{2.26}$$

where $\tilde{x}^-, \tilde{x}^+ \in \mathbb{R}^{2n}$ denote the perturbed states. Let $F^-, F^+ \in \mathbb{R}^{2n}$, respectively, denote the vector fields of the linearized closed-loop dynamics evaluated at the pre-impact state x^- and post-impact state x^+, defined as:

$$F^- \triangleq f_{cl}(x^-), \; F^+ \triangleq f_{cl}(x^+). \tag{2.27}$$

As explained in [125], the derivation of the saltation matrix approximates the perturbed states by flowing the system along F^-, F^+ for some duration of time $\delta t \in \mathbb{R}$. Specifically, without loss of generality, the post-impact state can be represented as:

$$x^+ = \Delta(x^-) + F^+\delta t, \tag{2.28}$$

and the perturbed states as:

$$\tilde{x}^+ \triangleq \Delta(\tilde{x}^-), \tag{2.29}$$

$$\tilde{x}^- \triangleq x^- + \delta x^- + F^-\delta t. \tag{2.30}$$

By substituting these approximations into (2.26) and taking the first-order Taylor series expansion, we obtain:

$$\delta x^+ = \Delta(x^- + \delta x^- + F^-\delta t) - \Delta(x^-) - F^+\delta t, \tag{2.31}$$

$$\approx J_\Delta \delta x^- + J_\Delta F^-\delta t - F^+\delta t. \tag{2.32}$$

Here, $J_\Delta \triangleq \frac{\partial}{\partial x}\Delta(x^-) \in \mathbb{R}^{2n\times 2n}$ denotes the Jacobian of the reset map evaluated at the pre-impact state x^-. Note that because this work exclusively utilizes the reset map shown in (2.14), we continue the derivation using the assumption that the reset map has no dependence on time, i.e., $\frac{\partial \Delta}{\partial t} = \mathbf{0}_{2n\times 2n}$.

Lastly, we can represent δt in terms of the pre-impact state error δx^- by observing the perturbed guard condition. For generality, we will denote the guard

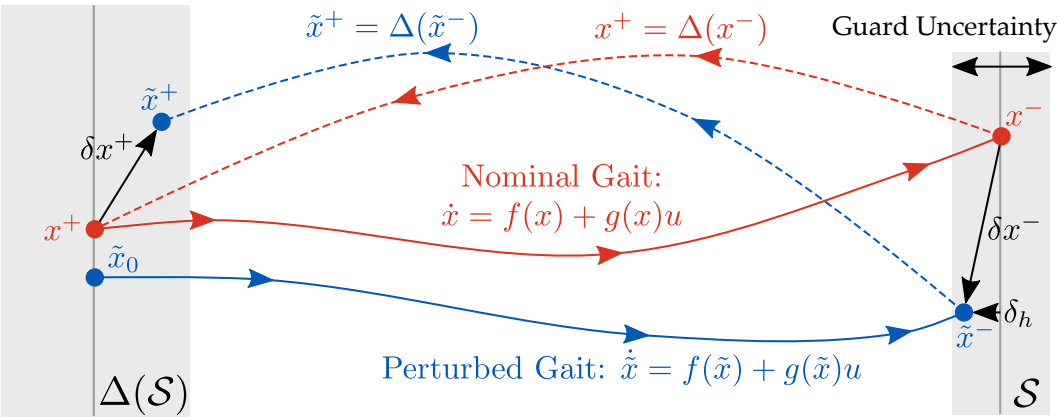


Figure 2.3: Illustration of a perturbed flow (blue) with uncertain guard conditions (illustrated by the grey region) compared to a nominal periodic orbit (red) which assumes a known guard (illustrated by the black vertical line). The perturbed initial condition $\tilde{x}_0 \in \Delta(\mathcal{S})$ results in pre-impact state error δx^- and post-impact state error δx^+. In general, saltation matrices capture the relationship between these errors: $\delta x^+ = S\delta x^-$.

condition as $h : \mathcal{X} \to \mathbb{R}$, but note that as mentioned before, we specifically define $h(x) \triangleq p^z_{sw}(x)$. Applying a first-order Taylor series expansion to the perturbed guard condition, we obtain:

$$0 = h(\tilde{x}^-), \tag{2.33}$$

$$\approx \underbrace{h(x^-)}_{=0} + J_h^\top(\delta x^- + F^-\delta t). \tag{2.34}$$

Here, $J_h \triangleq \frac{\partial}{\partial x}h(x^-) \in \mathbb{R}^{2n}$ denotes the Jacobian of the guard condition with respect to x^-. Similar to before, we assume that the guard condition has no dependence on time, i.e., $\frac{\partial h}{\partial t} = 0$; such reset maps are termed *autonomous switching boundary functions*. For information on deriving the saltation matrix for non-autonomous switching boundaries (i.e., $h(x,t)$) refer to [125]. Through the manipulation of (2.34), we arrive at our expression of δt in terms of δx^-:

$$\delta t = \frac{-J_h^\top \delta x^-}{J_h^\top F^-}. \tag{2.35}$$

Using this relationship, we can substitute (2.35) into (2.32) to obtain the "traditional saltation matrix," $S \in \mathbb{R}^{2n\times 2n}$:

$$\delta x^+ \triangleq \underbrace{\left(J_\Delta + \frac{(F^+ - J_\Delta F^-)J_h^\top}{J_h^\top F^-},\right)}_{S}\delta x^-. \tag{2.36}$$

Accounting for Guard Uncertainty. Recently, Payne et al. [152] extended the traditional saltation matrix to also account for guard uncertainty. This is accomplished by adapting (2.33) to also account for perturbations in the guard location along the normal direction, denoted as $\delta_h \in \mathbb{R}$ (shown in Figure 2.3):

$$\delta_h = h(\tilde{x}^-) \tag{2.37}$$

$$\delta_h \approx J_h^\top (\delta x^- + F^- \delta t) \tag{2.38}$$

$$\delta t = \frac{-J_h^\top \delta x^- + \delta_h}{J_h^\top F^-}. \tag{2.39}$$

Then, substituting (2.39) into (2.32), we arrive at the expression:

$$\delta x^+ = S\delta x^- + \underbrace{\left(\frac{J_\Delta F^- - F^+}{J_h^\top F^-}\right)}_{S_g} \delta_h, \tag{2.40}$$

where S is the same as in (2.36), and $S_g \in \mathbb{R}^{2n\times 1}$ is termed the *guard saltation matrix*. Together, these matrices can be combined together to obtain the *extended saltation matrix*, $S_e \in \mathbb{R}^{2n+1\times 2n+1}$, which is defined by the expression:

$$\begin{bmatrix} \delta x^+ \\ \delta_h \end{bmatrix} = \underbrace{\begin{bmatrix} S & S_g \\ 0 & 1 \end{bmatrix}}_{S_e} \begin{bmatrix} \delta x^- \\ \delta_h \end{bmatrix}. \tag{2.41}$$

Robust Gait Generation

Now that we have presented the preliminaries on saltation matrices, we will discuss how and why we incorporate the saltation matrix evaluation in the HZD gait generation framework. First, to address the question of *why* evaluating the saltation matrix improves the robustness of generated gaits, we refer to the field of contraction theory. As first noted by Lohmiller in 1988, discussing stability alone does not capture the behavior of a system relative to a nominal motion [130]. Instead, Lohmiller proposed a new field of analysis – contraction analysis – which explores how trajectories evolve relative to nearby trajectories. Specifically, a system is defined as *contractive* if all trajectories converge to some nominal trajectory. In this work, we leverage the notion of contractivity to define *robust* walking behaviors as those that are more contractive. In other words, when disturbed, a robust gait will converge to a nontrivial periodic orbit faster than a non-robust gait.

Moreover, Burden et al. [40] recently leveraged saltation matrices to induce contractivity of a hybrid system through discrete events. Motivated by this, we propose optimizing the contractivity of discrete events by minimizing the induced matrix norm of the extended saltation matrix; the induced matrix norm is equivalent to the largest singular value of the extended saltation matrix (i.e., $\|S_e\|_2 \triangleq \sqrt{\lambda_{\max}(S_e^\top S_e)} \in \mathbb{R}$). Lastly, since the HZD gait generation framework relies heavily on the cost function, we propose adding the induced norm of the extended saltation matrix directly to the cost.

To investigate the influence of including the saltation matrix in the optimization problem, the remainder of this section compares gaits generated with different weightings of the commonly used cost function torque-squared and the induced matrix norm of the extended saltation matrix:

$$\Phi(X) = w_1\|U\|_2^2 + w_2\|S_e\|_2^2. \tag{2.42}$$

Here, $w_1, w_2 \in \mathbb{R}_{\geq 0}$ denote constant weighting terms, $U \in \mathbb{R}^{m\times N}$ denotes the vectorized torques throughout the nominal gait (assuming a decision variable with $N \in \mathbb{R}$ discretizations) and $S_e \in \mathbb{R}^{2n+1\times 2n+1}$ denotes the extended saltation matrix evaluated at the pre-impact state of the generated nominal gait. The extended saltation matrix is again computed as in (2.41), with the traditional saltation matrix and guard saltation matrix explicitly computed as:

$$S = J_\Delta + \frac{(\dot{x}^+ - J_\Delta \dot{x}^-)J_h^\top}{J_h^\top \dot{x}^-},\ S_g = \frac{J_\Delta \dot{x}^- - \dot{x}^+}{J_h^\top \dot{x}^-}. \tag{2.43}$$

To preview the experimental results, it was found that increasing w_2 in (2.42) relative to w_1 improves robustness of the generated gaits, but also results in increased torque (this is illustrated in Figure 2.4). This trade-off between performance (characterized by successful implementation on hardware) and robustness (characterized by a systems ability to return to nominal periodic orbits in the presence of disturbances) is further explored in the experimental results.

Lastly, we would like to note that computing the Jacobian of the reset map, J_Δ, can be computationally expensive for high-dimensional systems because the reset map (2.14) requires several matrix inversions. Thus, for implementation purposes, we numerically approximate the Jacobian of the reset map. However, future work could more efficiently obtain these terms using autodiff or other tools for computing efficient analytical derivatives [173].

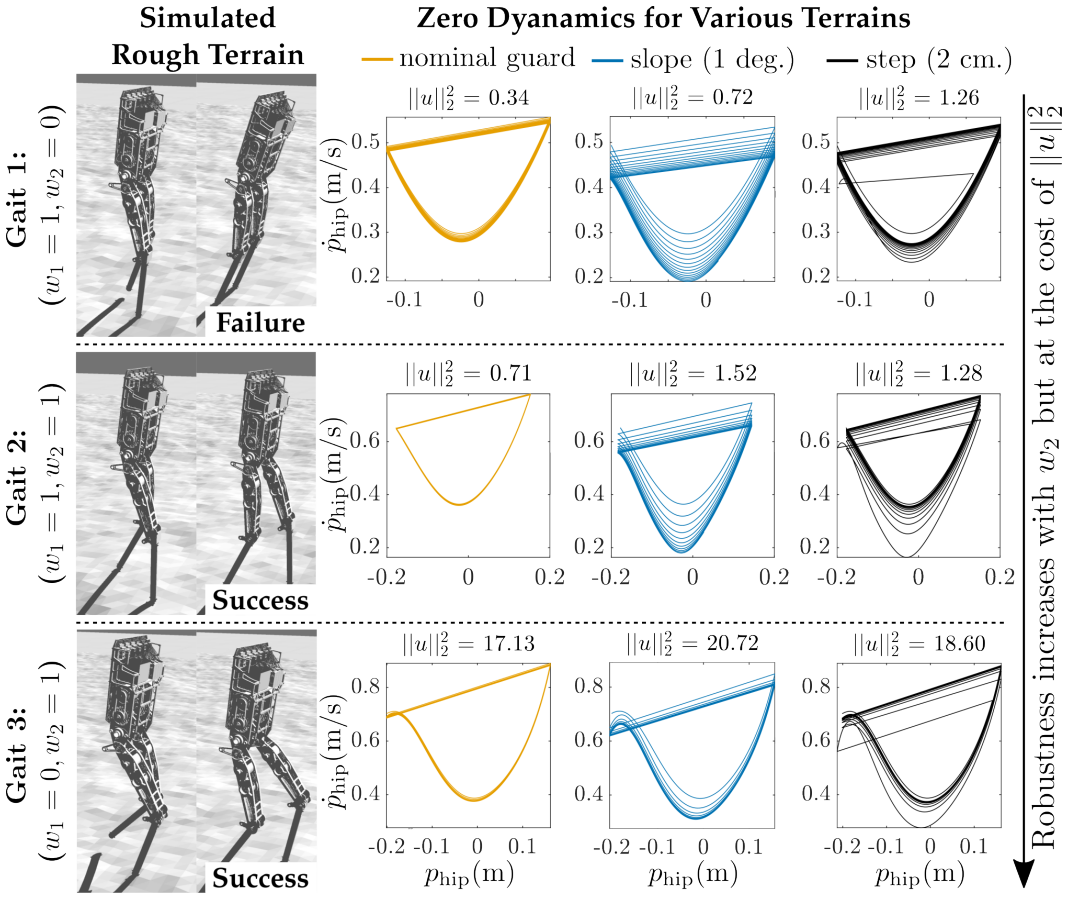


Figure 2.4: This figure illustrates the simulated behavior of three gaits for AMBER-3M, where each gait was generated with different weighting terms w_1 and w_2. The behavior is illustrated using the zero dynamics coordinates of the linearized hip position and velocity ($p_{\text{hip}}, \dot{p}_{\text{hip}} \in \mathbb{R}$), across three different environment conditions: flat ground as captured by the nominal guard (left); 1 degree slope (middle); and 2cm step height (right). The phase portraits show that the robustness of the walking behavior increases as w_2 increases relative to w_1, but the gait with $w_1 = 0$ results in significantly increased torque.

Experimental Results

We demonstrate the application of saltation matrices towards robust gait generation on two robotic platforms: the AMBER-3M planar biped, and the Atalante exoskeleton. As illustrated in Figure 2.4, three gaits were generated for each robotic platform: 1) a nominal gait with the cost function equal to torque squared; 2) a gait with the cost function being a weighting of both torque squared and the induced matrix norm of the extended saltation matrix at impact; and 3) a gait with the cost function only including the saltation matrix. For AMBER-3M, the three compared gaits were generated with weight

values of[2]: $w_1 = 1, w_2 = 0$; $w_1 = 1, w_2 = 1$; $w_1 = 0, w_2 = 1$. For Atalante, the weights were selected as: $w_1 = 1e^{-6}, w_2 = 0$; $w_1 = 1e^{-6}, w_2 = 1e^{6}$; $w_1 = 0, w_2 = 1e^{6}$. Note that for each robotic platform, the remaining constraints and bounds of the HZD gait generation framework were held constant; only the weights w_1 and w_2 varied. All gaits were generated using the FROST toolbox [87]. The presented experimental results are best demonstrated via the supplemental video [192].

AMBER-3M Simulation Results. The AMBER-3M planar biped[3] is a custom planar robot [7] with four motorized joints (left hip, left knee, right hip, right knee). The measured joint positions are denoted as $q^a \in \mathbb{R}^4$ and are selected as the outputs of the generated gaits, i.e., $y^a(q) \triangleq q^a \in \mathbb{R}^4$. The phasing variable $\tau(x) \in \mathbb{R}$ is selected to be the linearized forward hip position, and the desired outputs are described using a 5^{th}-order Bézier polynomial ($\alpha \in \mathbb{R}^{4\times 6}$). The generated joint-level trajectories are enforced using PD control.

The three aforementioned gaits were first demonstrated in a planar RaiSim [101] simulation environment with randomly generated terrain. As shown in Figure 2.4, both gaits generated with the inclusion of the saltation matrix ($w_2 = 1$) were able to walk on rough terrain, while the gait generated with only torque-squared ($w_2 = 0$) failed. The figure also compares three environmental guard conditions: flat ground (as captured by the nominal guard condition), a 1deg slope, and a 2cm step. As illustrated by the phase portraits of the zero dynamics (selected as linearized forward hip position and velocity), the gaits with $w_2 = 1$ again show improved robustness. However, it is interesting to note that the gait with $w_1 = 0$ suffers from significantly increased torque while the gait with both terms ($w_1, w_2 = 1$) only has a moderate increase in required torque.

AMBER-3M Experimental Results. Once demonstrated in simulation, the three gaits were also demonstrated on hardware, as shown in Figure 2.5. The generated joint-level trajectories are enforced on AMBER-3M using an off-board joint-level PD controller that computes desired torques and sends them to the on-board motor controllers via UDP communication. The motor driver

[2]The purpose of scaling w_1 and w_2 is to ensure that the platform-specific torque-squared term $\|U\|_2^2$ has a similar magnitude to that of $\|S_e\|_2^2$, thus preventing either term from heavily dominating the NLP objective.

[3]In this work, we specifically utilize the point-foot configuration of AMBER-3M, termed AMBER3M-PF.

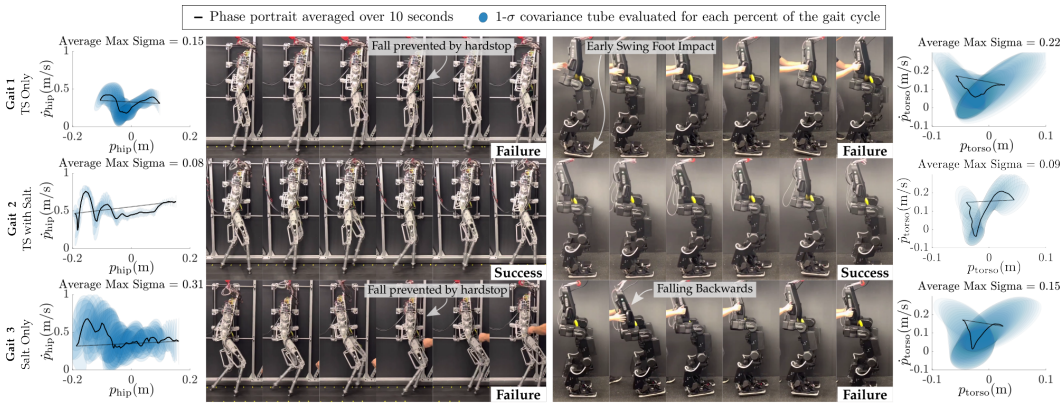


Figure 2.5: Gait tiles demonstrating the experimental performance of all three gaits on the AMBER-3M planar biped (left), and the empty Atalante exoskeleton (right). For both platforms, only the gait generated with both torque and the saltation matrix in the cost function ($w_1, w_2 > 0$) was able to sustain stable locomotion. The experimental data is also visualized via phase diagrams of the linearized hip position and velocity ($p_{\text{hip}}, \dot{p}_{\text{hip}} \in \mathbb{R}$) for AMBER-3M and the forward position and velocity of the floating-base frame ($p_{\text{torso}}, \dot{p}_{\text{torso}} \in \mathbb{R}$) relative to the stance foot for the exoskeleton. The black line shows the average zero dynamics across a single step, with the blue region illustrating the 1-sigma tube.

communication and control logic run at approximately 1kHz. As shown in Figure 2.5, the gait generated using only torque-squared ($w_2 = 0$), and the gait generated using only the extended saltation matrix ($w_1 = 0$) were unstable, while the gait generated with the inclusion of both torque-squared and the extended saltation matrix ($w_1, w_2 > 0$) in the cost function was independently stable.

To further demonstrate the performance of the gait generated with $w_1, w_2 > 0$, several robustness tests were also performed, as shown in Figure 2.6. For these experiments, random wooden objects were placed on the treadmill in front of AMBER-3M. These experiments highlight the robustness of the gait generated with the inclusion of both torque-squared and the extended saltation matrix in the cost function.

Atalante Simulation Results. The second set of experiments demonstrated the framework on the Atalante exoskeleton, with sets of three gaits generated for the exoskeleton with various human-subject models, as well as for the empty exoskeleton. As before, all constraints and bounds of the gait generation framework were held constant except for w_1 and w_2, and the generated joint-level

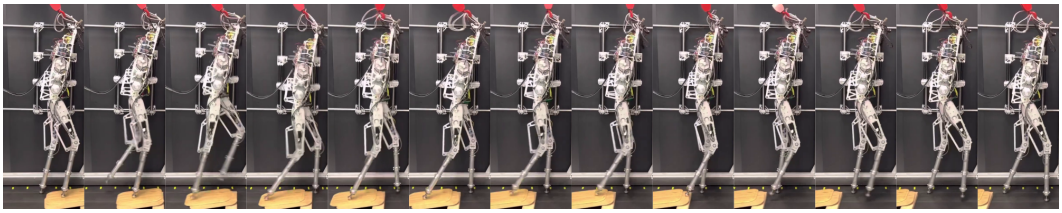


Figure 2.6: Gait tiles demonstrating robustness of the AMBER-3M gait generated with both the saltation matrix and torque in the objective function ($w_1, w_2 > 0$).

trajectories were enforced using PD control. As with AMBER, the outputs are again selected as the positions of the motorized joints, i.e., $y^a(q) \triangleq q^a \in \mathbb{R}^{12}$. The gaits are described using 7^{th}-order Bézier polynomials ($\alpha \in \mathbb{R}^{12\times 8}$) and are parameterized using time as the phasing variable. It is important to note that by using time as a phase-variable, the theoretical guarantees of stability are no longer valid. For this reason, many results using periodic orbits on 3D robots also incorporate regulators to stabilize the walking [159]. However, in this work, since we are interested in how *robust* the periodic gaits are independent of regulators, no additional regulators were used (aside from desired output filtering to prevent discontinuities caused by early impacts). The generated joint-level trajectories are enforced on the Atalante exoskeleton using an on-board PD controller which sends current commands to the low-level motor drivers.

First, sets of three gaits were generated and deployed for six human models in a 3D simulation environment. As with prior work [79, 80, 206, 207], a human-exoskeleton model is synthesized by collecting the human's height, mass, thigh length, and shank length and using this information to approximate the remaining human segment inertia and remaining segment lengths based on anthropomorphic models from [225]. The simulation results, illustrated in Figure 2.7, found that the gaits with $w_2 > 0$ resulted in more stable steps being taken before the exoskeleton fell (characterized by the COM vertical height falling below 0.4 meters). However, the gait generated with only the saltation matrix resulted in significantly increased torque. In contrast, using both the saltation matrix and torque-squared in the cost function resulted in increased robustness with only a small increase in torque.

Atalante Experimental Results. Once demonstrated in simulation, a set of three gaits was also generated for the empty Atalante exoskeleton and de-

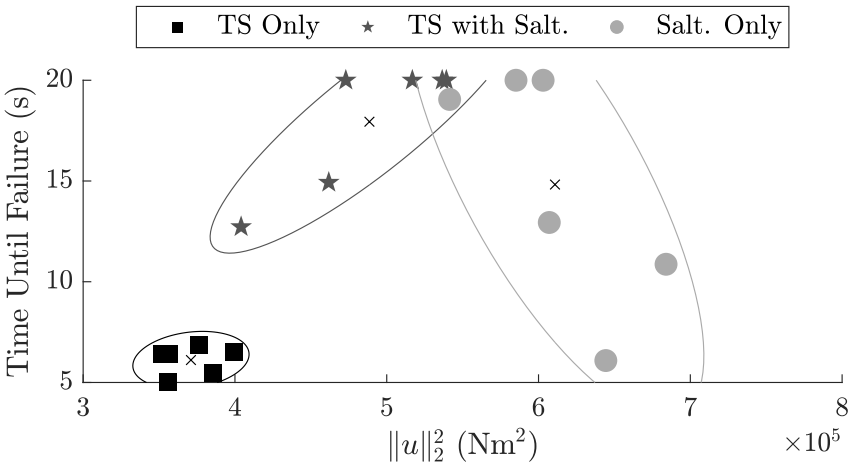


Figure 2.7: Simulation results on the Atalante exoskeleton for 6 random exoskeleton subject models. Each marker indicates an individual simulation with the corresponding time until failure (defined as the COM height falling below 0.4 meters) and nominal torque-squared evaluation. Each simulation was limited to 20 seconds total. The results for each gait condition are highlighted by ellipses constructed using 2-sigma fits to the data.

ployed on hardware. The motivation for conducting experiments with the empty exoskeleton is to isolate the effects of the generated gaits independent of the human-subject's motion inside of the exoskeleton. The performance of the generated gaits was evaluated by whether or not the exoskeleton could locomote without operator interference for 3 meters. As shown in Figure 2.5, only the gait generated with both torque-squared and the saltation matrix ($w_1, w_2 > 0$) was able to successfully and independently walk for the full 3-meter test. Lastly, a set of gaits was also generated and deployed on the Atalante exoskeleton for a human subject. Again, only the gait generated with the inclusion of both cost function terms ($w_1, w_2 > 0$) was able to successfully complete the 3-meter walk test. These results are shown in Figure 2.6 and in the supplemental video [192].

2.3 Input to State Stability Perspective on Uncertain Terrain

While adding the saltation matrix to the gait synthesis optimization problem successfully demonstrated improved locomotive robustness in real-world experiments, there is still a lack of theoretic tools to formally asses the robustness of the resulting behavior, i.e., characterizing the domain on which behaviors are stable. Similarly, as discussed in Chapter 1, while there exist various metrics for the robustness of limit cycles [95, 137], these metrics do not

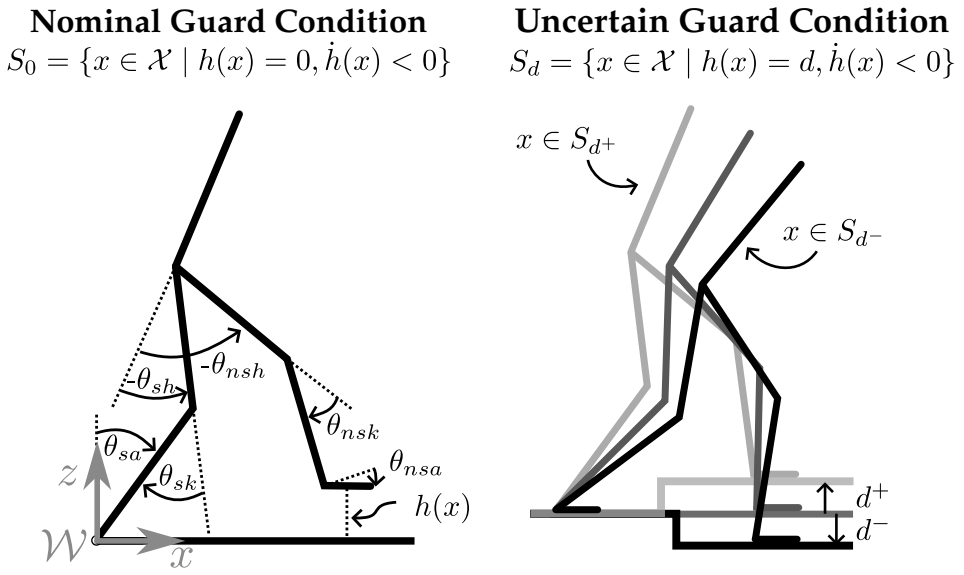


Figure 2.8: A depiction of (left) the configuration coordinates for a seven-link walker and (right) the uncertain guard condition.

provide theoretical certificates of robustness. Towards certifying robustness, input-to-state stability (ISS) [179] has been effectively leveraged in the context of robotic walking and running for uncertain dynamics [120, 134], but has yet to be framed in the context of uncertain terrain.

Thus, the second contribution of this chapter is a notion of robust walking that quantifies the gap between stability and robustness mathematically. Explicitly, the goal of this work is to define what it means for a periodic orbit to be certifiably robust to uncertain terrain as illustrated in Figure 2.8. Towards this, we define the δ-robustness of periodic orbits in hybrid systems by leveraging input-to-state stability of the Poincaré return map. Specifically, δ-robustness is defined as the maximum disturbance in the guard condition (commonly selected to be the ground height) that can be accommodated while remaining stable to a neighborhood. The main result is the formulation of robust Lyapunov functions that certify the robustness of periodic orbits to disturbances in the environment. The leads to an algorithm for certifying the δ-robustness of walking gaits, as demonstrated in simulation with a seven-link bipedal robot walking on uneven terrain.

Preliminaries on Poincaré return maps

Given a locally Lipschitz feedback controller $u^* : \mathcal{D} \to \mathbb{R}^m$, such as a feedback linearizing controller, the result of applying this to the hybrid control system

is a hybrid system:

$$\mathcal{H} = \begin{cases} \dot{x} = f_{\text{cl}}(x) \triangleq f(x) + g(x)u^*(x) & x \in \mathcal{D} \setminus \mathcal{S}, \quad (2.44) \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S}. \quad (2.45) \end{cases}$$

The local Lipschitz continuity of the continuous dynamics (2.44) implies that solutions exist and are unique locally. We will use the flow notation for these solutions, $\varphi_t(x_0)$, which is the solution to the continuous dynamics at time $t \in \mathbb{R}_{\geq 0}$ with initial condition $x_0 \in \mathcal{D}$. Under the assumption of non-Zenoness, the flow of the hybrid system is given by:

$$\varphi_t(x_0) = \varphi_{t-\tau_k}(x_k^+), \qquad t \in [\tau_k, \tau_{k+1}), \tag{2.46}$$

where τ_k are the "impact" times and x_k^+ the post-impact states, determined by the consistency conditions:

$$x_k^+ = \Delta(x_k^-), \qquad x_k^- = \varphi_{\tau_k - \tau_{k-1}}(x_{k-1}^+) \in \mathcal{S}, \tag{2.47}$$

for $k \geq 1$, with $\tau_0 = 0$ and $x_0 \in \mathcal{D}$ the initial condition. When $x_0 \in \mathcal{S}$ one trivially takes $x_1^- = x_0$ and $\tau_1 = \tau_0$.

The flow $\varphi_t(x_0)$ of (2.44) is periodic with period $T \in \mathbb{R}_{\geq 0}$ if there exists a point $x^* \in \mathcal{S}$ satisfying $\varphi_T(\Delta(x^*)) = x^*$. The periodic orbit associated with this periodic flow is denoted:

$$\mathcal{O} \triangleq \{\varphi_t(\Delta(x^*)) \in \mathcal{D} \mid 0 \leq t \leq T_I(x^*) = T\}, \tag{2.48}$$

with $T_I : \widetilde{\mathcal{S}} \to \mathbb{R}$ being the time-to-impact function:

$$T_I(x) = \inf\{t \geq 0 \mid \varphi_t(\Delta(x)) \in \mathcal{S}\}. \tag{2.49}$$

As proven in Lemma 3 of [77], the time-to-impact function is continuous at points $x \in \widetilde{\mathcal{S}}$ satisfying the conditions $\widetilde{\mathcal{S}} \triangleq \{x \in \mathcal{S} \mid 0 < T_I(x) < \infty\}$. Thus, T_I is well-defined for $\widetilde{\mathcal{S}}$. The periodic orbit, $\mathcal{O}$, is exponentially stable if it is exponentially stable as a set: for $x_0 \in \mathcal{D}$:

$$\|\varphi_t(x_0)\|_{\mathcal{O}} \leq Me^{-\mathfrak{a}t}\|x_0\|_{\mathcal{O}}, \tag{2.50}$$

where $\|x\|_{\mathcal{O}} = \inf_{y \in \mathcal{O}} \|x - y\|$ is the set distance.

The exponential stability of this periodic orbit $\mathcal{O}$ can be analyzed via the Poincaré map. In particular, $\mathcal{S}$ is a Poincaré section (and well-defined as

such due to the assumption that $\dot{h}(x) < 0$), and associated with this Poincaré section is the Poincaré map $P : \widetilde{\mathcal{S}} \to \mathcal{S}$ defined as:

$$P(x^-) \triangleq \varphi_{T_I(x^-)}\left(\Delta(x^-)\right). \tag{2.51}$$

The Poincaré map describes the evolution of the hybrid system as a discrete-time system:

$$x^-_{k+1} = P(x^-_k),\ k = 0, 1, \dots, \tag{2.52}$$

wherein x^-_k is just given as in (2.47). In [143] (see also [146], Theorem 2.1), it was proven that a periodic orbit $\mathcal{O}$ is exponentially stable if and only if $x^* \in \mathcal{O} \cap \mathcal{S}$ is an exponentially stable fixed point of the discrete-time system (2.52). This is summarized in the following:

Theorem 1 ([143])**.** *A periodic orbit $\mathcal{O}$ is exponentially stable if and only if for the corresponding fixed point $P(x^*) = x^* \in \mathcal{S}$, there exist $M > 0$, $\mathfrak{a} \in (0, 1)$, and some $\delta > 0$ such that:*

$$\forall\ x \in B_\delta(x^*) \cap \widetilde{\mathcal{S}} \quad \Longrightarrow \quad \|P^i(x) - P(x^*)\| \leq M\mathfrak{a}^i\|x - x^*\|,$$

with $P^i(x)$ denoting the Poincaré map applied $i \in \mathbb{N}_{\geq 0} = \{0, 1, \dots, n, \dots\}$ times.

The Extended Poincaré Map. It is important to note that the Poincaré map can be extended such that it is defined on a neighborhood[4] of the fixed point x^*. In particular, the time-to-impact function exists as a result of the implicit function theorem applied to the implicit function (of time) $h(\varphi_t(\Delta(x)))$ which therefore satisfies: $h(\varphi_T(\Delta(x^*))) = 0$, and $\dot{h}(\varphi_T(\Delta(x^*))) < 0$, for $x^* \in \mathcal{O} \cap \mathcal{S}$. Thus, there exists an explicit function $T_e : B_\rho(x^*) \subset \mathcal{D} \to \mathbb{R}$, for some $\rho > 0$, termed the *extended time-to-impact function* satisfying:

$$h(\varphi_{T_e(x)}(\Delta(x))) = 0, \qquad \forall\ x \in B_\rho(x^*). \tag{2.53}$$

It follows that T_I in (2.49) is just $T_I = T_e|_S$, wherein the Poincaré map is given by considering only $x \in B_\rho(x^*) \cap \mathcal{S}$. Yet, this restriction is not necessary—which leads to the notion of the extending the Poincaré map so its domain

[4] We assume throughout that for all $\rho > 0$ of interest, the domain $\mathcal{D}$ of the continuous dynamics is appropriately chosen so that $B_\rho(x^*) \subset \mathcal{D}$.

of definition is $B_\rho(x^*)$ (see [153]). In particular, define the *extended Poincaré map*:

$$\begin{aligned} P_0 : B_\rho(x^*) \subset \mathcal{D} &\to \mathcal{S} \\ x &\mapsto \varphi_{T_e(x)}(\Delta(x)). \end{aligned} \tag{2.54}$$

The importance of extending the domain of definition of the Poincaré map will be seen in the context of set theoretic notions of robust walking.

An ISS Perspective on Walking: δ-Robustness

We will now provide the key formulation of robustness considered in this section—that of δ-robustness. The core concept behind this definition is stability in and of itself is not a sufficiently rich concept to capture robustness, since it is purely local. Thus, we define a notion of robustness leveraging the extended Poincaré map and input-to-state stability, wherein the inputs are the disturbances associated with uncertain guard conditions.

Practically, the stability of periodic orbits can be analyzed by evaluating the eigenvalues of the Poincaré return map linearized around the fixed point. Specifically, if the magnitude of the eigenvalues of $DP(x^*) = \frac{\partial P}{\partial x}(x^*)$ is less than one (i.e., $\max |\lambda(DP(x^*))| < 1$), then the fixed point is stable [143, 153]. Since it is often difficult to compute the Poincaré map analytically, it is commonly numerically approximated. Each row of the Jacobian is successively computed by applying small perturbations to each state and forward simulating one step to obtain $P(x^* + \delta)$. This implicitly implies that the Poincaré map is robust to δ perturbations—as long as δ is sufficiently small. Yet, this small amount of robustness inherent in stable gaits is often confounded with the eigenvalues themselves. That is, it is sometimes assumed that the magnitude of the eigenvalues of the Poincaré map say something deeper about the broader robustness of the periodic orbit to perturbations. This is not the case, as the following example illustrates.

Example 1. Consider a seven-link bipedal robot as shown in Figure 2.8. To illustrate how the eigenvalues associated with the linearization fail to tell the whole story, we will consider the robustness of two gaits to differing ground height conditions. As illustrated in Figure 2.9, the classic Poincaré analysis does not accurately reflect the robustness of periodic orbits to local disturbances in the guard condition. That is, the gait with the smaller maximum eigenvalue (magnitude) is more fragile to changing ground heights.

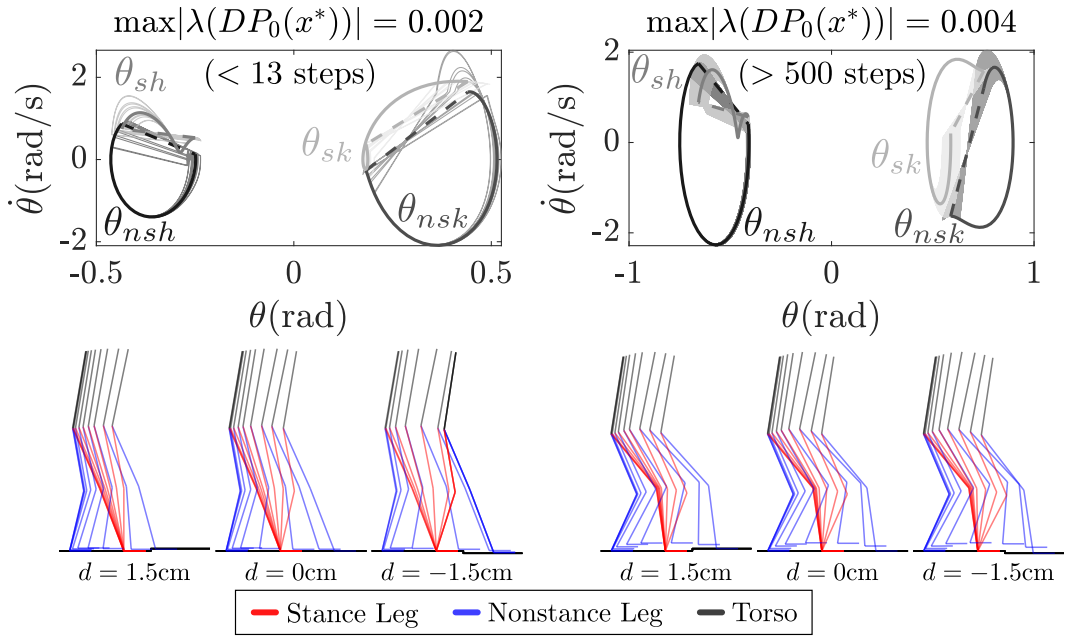


Figure 2.9: The phase portraits at the top of the figure illustrate the walking for uncertain guard conditions $\mathcal{S}_{d_k}$ with $d_k \sim \text{Uniform}(-\delta, \delta)$ (in this example, $\delta = 1.5\text{cm}$) for $k = 500$ steps. Visualizations of the walking gaits for three step conditions are provided at the bottom. The results demonstrate that a periodic orbit with $\max|\lambda(DP_0(x^*))| < 1$ (on the left) is not robust to variations in the guard condition (the orbit diverged after only 13 steps), while a periodic orbit with a larger $|\lambda|$ (on the right) is comparatively more robust. This motivates the need for an ISS perspective.

Uncertain Guard Conditions. To formulate a notion of robustness of the extended Poincaré map, uncertain guard conditions are considered—this, for example, captures uncertain ground height for walking robots. Specifically, we consider the general guard (as defined in [82]):

$$\mathcal{S}_d = \{x \in \mathcal{X} \mid h(x) = d,\ \dot{h}(x) < 0\}, \tag{2.55}$$

with $d \in \mathbb{D}$ and $\mathbb{D} \triangleq [d^-, d^+] \subset \mathbb{R}$ for some $d^- < 0 < d^+$. Using this general guard definition, the previous guard (2.2) is now denoted as $\mathcal{S}_0$. Under the assumption that $\mathcal{S}_d \subset \mathcal{D}$ for all $d \in \mathbb{D}$, we have a corresponding hybrid system:

$$\mathcal{H}_d = \begin{cases} \dot{x} = f_{\text{cl}}(x) \triangleq f(x) + g(x)k(x) & x \in \mathcal{D} \setminus \mathcal{S}_d, \quad (2.56) \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S}_d. \quad (2.57) \end{cases}$$

Consider the extended time-to-impact function $T_e : B_\rho(x^*) \to \mathbb{R}$ defined implicitly in (2.53). This function can be further extended (as a partial function)

to account for varying guards: $T_e : B_\rho(x^*) \times \mathbb{D} \rightharpoonup \mathbb{R}$:

$$T_e(x_0, d) \triangleq \inf\{t \geq 0 \mid \varphi_t(\Delta(x_0)) \in \mathcal{S}_d\}. \tag{2.58}$$

Importantly, this is a partial function because (by the implicit function theorem) it is only well-defined for $d = 0$ and by continuity sufficiently small d^- and d^+. Using this extended time-to-impact function, we can redefine the extended Poincaré map as a partial function: $P_d : B_\rho(x^*) \rightharpoonup \mathcal{S}_d$:

$$P_d(x^-) \triangleq \varphi_{T_e(x^-,d)}(\Delta(x^-)). \tag{2.59}$$

This allows us to frame walking with uncertain guards as a discrete-time control system.

Connections with Input-to-State Stability. It is important to note that we can view (2.59) as a dynamical system evolving with an "input" given by the guard height: $d = h(x)$. In particular, this leads to the discrete-time dynamical system:

$$x_{k+1} = \mathcal{P}(x_k, d_k) \triangleq P_{d_k}(x_k), \tag{2.60}$$

for some sequence of $d_k \in [d^-, d^+] \subset \mathbb{R}$, $k \in \mathbb{N}_{\geq 0}$, determining the guard height specific to step $k \in \mathbb{N}_{\geq 0}$ such that $x_{k+1} \in \mathcal{S}_{d_k}$. The result is a partial function:

$$\mathcal{P} : B_\rho(x^*) \times [d^-, d^+] \rightharpoonup \mathcal{S}_{[d^-,d^+]} := \bigcup_{d \in [d^-,d^+]} \mathcal{S}_d,$$

wherein we assume that $B_\rho(x^*) \subset \mathcal{S}_{[d^-,d^+]}$ (or a smaller ρ is chosen so that this holds). The partial function nature of $\mathcal{P}$ implies that solutions may not exist for all time, i.e., the solution x_k might leave the ball $B_\rho(x^*)$ on which $\mathcal{P}$ is well-defined.

Given the discrete-time system (2.60), and the fact that we view the input d as a disturbance, there are obvious connections with input-to-state stability [108]. Input-to-State Stability (ISS) is fundamental in mathematically quantifying how stability degrades in the presence of bounded disturbances. If a system is ISS, its trajectories will remain bounded, and will converge to a neighborhood of an equilibrium of the undisturbed system. This graceful degradation of stability in the presence of disturbances describes a variety of real-world control implementations.

In our setting, the discrete-time system $x_{k+1} = \mathcal{P}(x_k, d_k)$ (with d_k viewed as an input) is *input-to-state stable (ISS)* if:

$$\|x_k - x^*\| \leq \beta(\|x_0 - x^*\|, k) + \gamma(\|d\|_\infty), \tag{2.61}$$

for $k \in \mathbb{N}_{\geq 0}$, β a class $\mathcal{KL}$ function, and γ a class $\mathcal{K}$ function. Note that here $\|d\|_\infty = \max\{-d^-, d^+\}$ since $d : \mathbb{N}_{\geq 0} \to [d^-, d^+]$ is scalar valued and takes values in an interval. Also note that, in the context of locomotion, we are especially interested in exponential stability. To certify exponential ISS, the class $\mathcal{KL}$ function becomes: $\beta(r, k) = M\mathfrak{a}^k r$ for $M > 0$ and $\mathfrak{a} \in (0, 1)$. The end result is the exponential ISS (E-ISS) condition:

$$\|x_k - x^*\| \leq M\mathfrak{a}^k\|x_0 - x^*\| + \gamma(\max\{-d^-, d^+\}). \tag{2.62}$$

This allows us to formulate a notion of robustness.

<u>δ-Robustness.</u> We now have the necessary components to present the key concept of this section: δ-robustness. The goal in formulating this notion of robustness is to find a single scalar constant, $\delta \geq 0$, that characterizes the robustness of a periodic orbit $\mathcal{O}$ in the context of uncertain guard height. In this context, we wish to leverage (2.62)—yet the class $\mathcal{K}$ function γ gives a degree of freedom that is undesirable in designing a metric for robustness. This observation leads to:

Definition 1. The periodic orbit $\mathcal{O}$ is **δ-robust** for a given $\delta > 0$ if for the discrete-time dynamical system in (2.60) with $d^- = -\delta$ and $d^+ = \delta$, that is:

$$\begin{aligned} \mathcal{P} :& B_\rho(x^*) \times [-\delta, \delta] \to \mathcal{S}_{[-\delta,\delta]} \\ & x_{k+1} = \mathcal{P}(x_k, d_k), \qquad d_k \in [-\delta, \delta], \end{aligned} \tag{2.63}$$

there exists a forward invariant set $W \subset B_\rho(x^*)$ and for all $x_0 \in W$:

$$\|x_k - x^*\| \leq M\mathfrak{a}^k\|x_0 - x^*\| + \gamma\delta, \qquad \forall k \in \mathbb{N}_{\geq 0}, \tag{2.64}$$

for some $\gamma > 0$, $M > 0$, and $\mathfrak{a} \in (0, 1)$. The periodic orbit is **robust** if it is δ-robust for some $\delta > 0$, and the largest scalar $\overline{\delta}$ such that $\mathcal{O}$ is $\overline{\delta}$-robust is the **robustness** of $\mathcal{O}$.

This seemingly simple definition encodes a surprising amount of information. First, the forward invariance of $W \subset B_\rho(x^*)$ implies that $\mathcal{P} : B_\rho(x^*) \times [-\delta, \delta] \to$

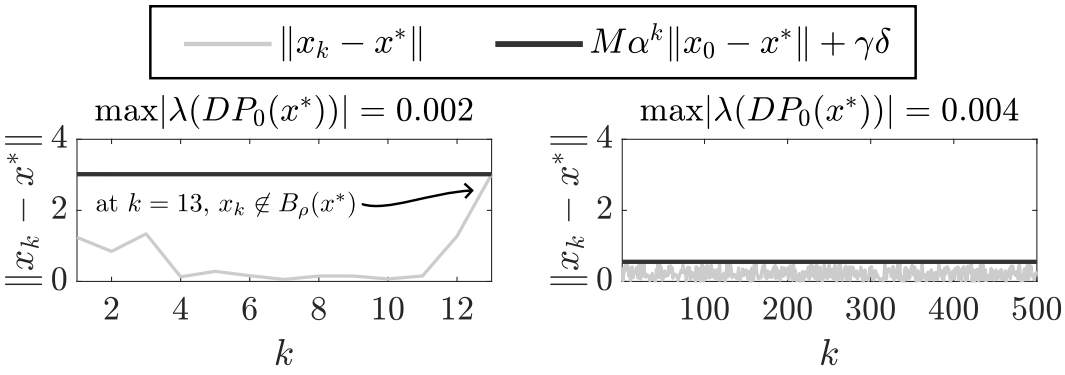


Figure 2.10: On the left, the non-robust periodic orbit (as illustrated on the left of Figure 2.9) does not satisfy the conditions for δ-robustness for $\delta = 0.015$ (specifically, there does not exist a forward invariant set W). In comparison, the robust orbit (as illustrated on the right of Figure 2.9) satisfies the definition of δ-robustness with $\gamma = 36.8$ and $\delta = 0.015$m.

$\mathcal{S}_{[-\delta,\delta]}$ is a function (rather than a partial function) when restricted to the set W. Additionally, the actual δ-robustness condition (2.64) is an ISS condition, albeit slightly stronger to remove the dependence on the class $\mathcal{K}$ function and replace this with the constant γ. Even so, the connections with ISS are important since the associated machinery can be leveraged.

To provide an example of how ISS can inform our thinking on δ-robustness, consider the case when $\mathcal{O}$ is exponentially stable, i.e., $x_{k+1} = \mathcal{P}(x_k, 0)$ has an exponentially stable fixed point: $x^* = P_0(x^*)$, i.e., the 0-input system is exponentially stable. There are no guarantees that $\mathcal{O}$ is thus δ-robust (see [108] where a counter example shows that stability does not imply ISS). That is, stability does not imply robustness.

Example 2. Returning to the example of the seven-link walker, we can heuristically calculate the δ-robustness associated with the two gaits. Specifically, Figure 2.10 illustrates the ISS-perspective of δ-robustness for the orbits first illustrated in Figure 2.9. As shown, the orbit that was robust in Figure 2.9 satisfies the condition that $W \subset B_\rho(x^*)$ is forward invariant ($\delta = 1.5$cm in this example), and $\|x_k - x^*\|$ remains bounded for $\gamma = 36.8$. Comparatively, the orbit that was not robust in Figure 2.9 experienced a pre-impact state that was outside of $B_\rho(x^*)$ and therefore W was not forward invariant.

Lyapunov Conditions for δ-robustness

In this section we present the main theoretic result: Lyapunov conditions for the δ-robustness of periodic orbits. These conditions, and constructions, follow naturally from the ISS perspective employed in defining δ-robustness. But care is needed given the complexity of the Poincaré map. Importantly, these conditions will lead to an approach for the verification of δ-robustness, as presented in the next section.

Definition 2. Consider the discrete-time dynamical system in (2.63). A function $V : B_\rho(x^*) \to \mathbb{R}_{\geq 0}$, for $B_\rho(x^*)$ as in (2.54), is a **robust Lyapunov function** if:

$$k_1\|x - x^*\|^c \leq V(x) \leq k_2\|x - x^*\|^c \tag{2.65}$$

$$\|x - x^*\| \geq \chi d \quad \Longrightarrow \tag{2.66}$$

$$\Delta V(x,d) \triangleq V(\mathcal{P}(x,d)) - V(x) \leq -k_3\|x - x^*\|^c$$

for $\chi, k_1, k_2, k_3, c > 0$ and all $x \in B_\rho(x^*)$.

Remark 1. Note that the condition (2.66) can be equivalently restated as:

$$V(\mathcal{P}(x,d)) - V(x) \leq -k_4\|x - x^*\|^c + \frac{1}{2}\sigma|d|^c, \tag{2.67}$$

where $\sigma > 0$. In particular, the corresponding quantities are related via: $k_3 = \frac{1}{2}k_4$ and $\chi = k_4^{-\frac{1}{c}}\sigma^{\frac{1}{c}}$.

Main result. We can now state the main result of the section. To do so, recall that a *Lyapunov sublevel set* is given by:

$$\Omega_r = \{x \in \mathbb{R}^n \mid V(x) \leq r\}. \tag{2.68}$$

This will be essential in establishing:

Theorem 2. *Consider the discrete-time dynamical system $x_{k+1} = \mathcal{P}(x_k, d_k)$ in* (2.63) *with associated periodic orbit $\mathcal{O}$. If there exists a robust Lyapunov function, $V : B_\rho(x^*) \to \mathbb{R}_{\geq 0}$, and:*

$$\delta < \delta_{\max} \triangleq \left(\frac{k_1}{\chi^c k_2}\right)^{\frac{1}{c}} \rho, \tag{2.69}$$

then the periodic orbit $\mathcal{O}$ is δ-robust with:

$$W = \Omega_{r(\delta)}, \quad \text{for} \quad r(\delta) \triangleq k_2(\chi\delta)^c \tag{2.70}$$

$$\gamma = \left(\frac{k_2}{k_1}\right)^{\frac{1}{c}} \chi, \quad M = \left(\frac{k_2}{k_1}\right)^{\frac{1}{c}}, \quad \mathfrak{a} = \left(1 - \frac{k_3}{k_2}\right)^{\frac{1}{c}}.$$

This theorem is, overall, a variation on Lemma 3.5 in [108]. The proof here follows a similar overall arc, although there are key differences made necessary by the fact that $\mathcal{P}$ is only a partial function. This motivates the first Lemma.

Lemma 1. *The function $\mathcal{P} : B_\rho(x^*) \times [-\delta, \delta] \to \mathcal{S}_{[-\delta,\delta]}$ given in (2.63) is well-defined for all $x \in B_\rho(x^*)$, i.e., for all $x \in B_\rho(x^*)$, $\mathcal{P}(x, d)$ exists and satisfies $\mathcal{P}(x, d) \in \mathcal{S}_{[-\delta,\delta]}$.*

Proof. By the construction of the extended Poincaré map, P_0 is well-defined on $B_\rho(x^*)$, i.e., for all $x \in B_\rho(x^*)$ it follows that $\mathcal{P}(x, 0) \in \mathcal{S}_0$, i.e., $h(\varphi_{T_e(x,0)}(\Delta(x))) = 0$. Therefore:

$$h(\varphi_t(\Delta(x))) = \int_{T_e(x,0)}^{t} \dot{h}(\varphi_\tau(\Delta(x)))d\tau.$$

But $\dot{h}(x) < 0$ for all $x \in \mathcal{S}_{[-\delta,\delta]}$ by definition. Therefore, on the closed set defined by $-\delta \leq h(x) \leq \delta$, $\dot{h}$ takes a minimum and maximum value: $\underline{h} < \overline{h} < 0$. This implies that:

$$\underline{h}(t - T_e(x, 0)) \leq h(\varphi_t(\Delta(x))) \leq \overline{h}(t - T_e(x, 0)).$$

Thus, there exists a t (possibly negative) such that $h(\varphi_t(\Delta(x))) = d$. This $t = T_e(x, d)$. □

Since $\mathcal{P}$ is well-defined, we can now find a set such that $x_{k+1} = \mathcal{P}(x_k, d_k)$ is defined for all k, i.e., a forward invariant set contained in $B_\rho(x^*)$, using Lyapunov sublevel sets.

Lemma 2. *If $\delta < \delta_{\max}$, with $\delta_{\max}$ in (2.69), then for $r(\delta) \triangleq k_2(\chi\delta)^c$ it follows that:*

$$B_{\chi\delta}(x^*) \subset \Omega_{r(\delta)} \subset B_\rho(x^*).$$

Moreover, the set $\Omega_{r(\delta)}$ is forward invariant.

Proof. For $x \in B_{\chi\delta}(x^*)$:

$$\|x - x^*\| < \chi\delta \;\Rightarrow\; V(x) \leq k_2\|x - x^*\|^c < k_2(\chi\delta)^c = r(\delta)$$

and therefore $B_{\chi\delta}(x^*) \subset \Omega_{r(\delta)}$. Now if $r(\delta) < k_1\rho^c$ (which is equivalent to the condition (2.69)) it follows that:

$$V(x) \leq r(\delta) \quad \Rightarrow \quad k_1\|x - x^*\|^c \leq V(x) \leq r(\delta) < k_1\rho^c$$

and therefore $\Omega_{r(\delta)} \subset B_\rho(x^*)$. Finally, since for $\delta < \delta_{\max}$ we have $B_{\chi\delta}(x^*) \subset \Omega_{r(\delta)}$, it follows that on the boundary of $\Omega_{r(\delta)}$, namely $\partial\Omega_{r(\delta)}$, condition (2.66) is active and therefore: $\Delta V(x,d) < 0$. The forward invariance of $\Omega_{r(\delta)}$ follows. □

Lemma 2 gives an upper bound on the δ-robustness of a given periodic orbit $\mathcal{O}$, namely $\delta_{\max}$, based upon the domain of definition of $\mathcal{P}$. It also establishes the forward invariance of $\Omega_{r(\delta)}$. Leveraging this, we can prove the main result.

Proof of Theorem 2. Let $x_0 \in \Omega_{r(\delta)}$, wherein the forward invariance of $\Omega_{r(\delta)}$ (Lemma 2) implies $x_k \in \Omega_{r(\delta)} \subset B_\rho(x^*)$ for all $k \in \mathbb{N}_{\geq 0}$. Thus both $\mathcal{P}$ and V are well-defined. We consider two cases: $x_0 \notin B_{\chi\delta}(x^*)$ and $x_0 \in B_{\chi\delta}(x^*)$.

$\underline{\|x_0 - x^*\| \geq \chi\delta}$: In this case the implication (2.66) is active:

$$\Delta V \leq -\frac{k_3}{k_2} V \quad \implies \quad V(x_k) \leq \left(1 - \frac{k_3}{k_2}\right)^k V(x_0)$$

where the implication follows from applying the inequality on the right recursively (see also the comparison lemma [109]). Therefore, using the inequalities in (2.65) we have:

$$\|x_k - x^*\| \leq \underbrace{\left(\frac{k_2}{k_1}\right)^{\frac{1}{c}}}_{M} \underbrace{\left(1 - \frac{k_3}{k_2}\right)^{\frac{k}{c}}}_{\mathfrak{a}^k} \|x_0 - x^*\|. \tag{2.71}$$

Finally, note that $k_3/k_2 < 1$ as otherwise $V(x_k)$ would be negative for $k = 1$ which is impossible. Therefore, $\mathfrak{a} < 1$.

$\underline{\|x_0 - x^*\| < \chi\delta}$: While the implication in (2.66) no longer holds, we still have $x_k \in \Omega_{r(\delta)}$. As a result:

$$\begin{aligned} k_1\|x_k - x^*\|^c \leq V(x_k) \leq r(\delta) =& k_2(\chi\delta)^c \\ \implies \quad \|x_k - x^*\| \leq& \underbrace{\left(\frac{k_2}{k_1}\right)^{\frac{1}{c}} \chi}_{\gamma} \delta. \end{aligned} \tag{2.72}$$

Therefore, for M, $\mathfrak{a}$ in (2.71) and γ in (2.72) we have:

$$\begin{aligned} \|x_k - x^*\| &\leq \max\{M\mathfrak{a}^k\|x_0 - x^*\|, \gamma\delta\} \\ &\leq M\mathfrak{a}^k\|x_0 - x^*\| + \gamma\delta \end{aligned}$$

as desired, i.e., δ-robustness is established with $W = \Omega_{r(\delta)}$ the required forward invariant set. □

Algorithmic Verification of δ-Robustness

This section seeks to answer the question: How do we check δ-robustness of a given periodic orbit $\mathcal{O}$? To answer this question we derive conditions for δ-robustness through the use of robust Lyapunov functions (as introduced in the previous section). In particular, we will synthesize an optimization framework for verifying δ-robustness.

Problem Setup. Assume the existence of a stable periodic orbit $\mathcal{O}$ and so $x_{k+1} = \mathcal{P}(x_k, 0)$ has an exponentially stable fixed point x^*. For simplicity we will take $x^* = 0$ (achieved via the simple coordinate transformation $x \mapsto x - x^*$). As a result, the linearization:

$$x_{k+1} = \mathbf{A}x_k \triangleq D\mathcal{P}(0,0)x_k$$

is exponentially stable. Next, the Lyapunov matrix $\mathbf{P} = \mathbf{P}^T > 0$ is obtained by solving the discrete-time Lyapuov equation:

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

for $\mathbf{Q} = \mathbf{Q}^T > 0$. The end result is that the discrete-time Lyapunov function $V(x) = x^T\mathbf{P}x$ satisfies:

$$\lambda_{\min}(\mathbf{P})\|x\|^2 \leq V(x) \leq \lambda_{\max}(\mathbf{P})\|x\|^2 \tag{2.73}$$

$$V(Ax) - V(x) \leq -\lambda_{\min}(\mathbf{Q})\|x\|^2 \tag{2.74}$$

and thereby establishes exponential stability of the linear system (and the nonlinear system locally). Unlike stability, it is not guaranteed that this Lyapunov function can be used to establish robustness. Yet we will use it as a "guess" for a robust Lyapunov function in order to develop an algorithm to establish the robustness of a given gait $\mathcal{O}$.

Optimization Problem. Recall that the invariant set used to establish δ robustness was defined in Lemma 2, namely $\Omega_{r(\delta)}$. In this case:

$$\Omega_{r(\delta)} = \{x \in \mathbb{R}^n | V(x) = x^T\mathbf{P}x \leq r(\delta) \triangleq k_2(\chi\delta)^c\}. \tag{2.75}$$

Per the proof of Lemma 2 we therefore have:

$$B_{r_1}(0) \subset \Omega_{r(\delta)} \subset B_{r_2}(0), \tag{2.76}$$

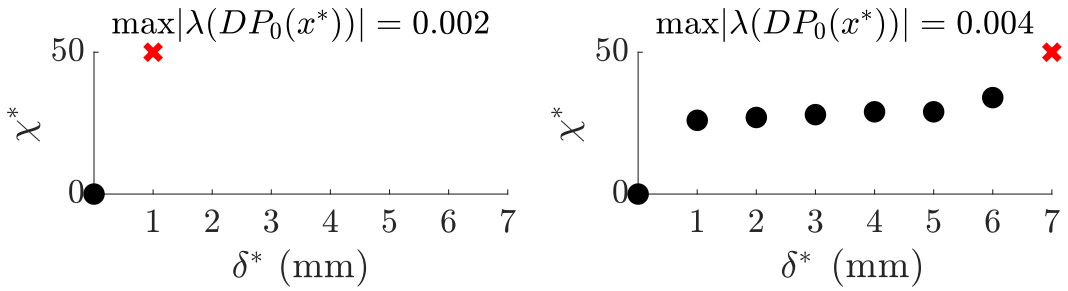


Figure 2.11: Results of the algorithmic approach to Opt. (2.78) for the gaits shown in Figure 2.9 with the maximum allowable χ set to 50. As shown, the gaits were determined to be δ-robust for $\delta^* = 0$ and $\delta^* = 6$mm, respectively.

with:

$$r_1 \triangleq \chi\delta, \qquad r_2 \triangleq \left(\frac{\lambda_{\max}(\mathbf{P})}{\lambda_{\min}(\mathbf{P})}\right)^{\frac{1}{2}} \chi\delta. \tag{2.77}$$

Then with the goal of finding the largest $\delta^* > 0$ such that $\mathcal{O}$ is δ robust, we formulate the following optimization problem:

$$\begin{aligned}(\delta^*, \chi^*) = \underset{\delta,\chi>0}{\text{argmax}}\ & \delta \\ \text{s.t. } & V(\mathcal{P}(x,d)) - V(x) \leq -k\|x\|^2 \\ & \forall\ r_1 < \|x\| < r_2, \quad \forall\ d \in [-\delta, \delta],\end{aligned} \tag{2.78}$$

where $k \in (0,1)$ is a user-defined variable, and we take $\mathbf{Q} = I$ (wherein $\lambda_{\min}(\mathbf{Q}) = 1$) to remove decision variables. This optimization problem can be solved algorithmically by slowly increasing χ for each candidate δ and checking the Lyapunov condition in (2.78) for random samples $x \in B_{r_1}(0)$.

We demonstrate this algorithmic approach for each of the two gaits illustrated in Figure 2.9 with the results[5] provided in Figure 2.11. As expected, the second gait illustrated in Figure 2.9 and Figure 2.10 was verified to be δ-robust, with $\delta^* = 6$mm. A visualization of the Lyapunov condition for 100 random samples of $x \in B_{r_1}(0)$ is provided in Figure 2.12 with the corresponding ISS bound illustrated in Figure 2.13.

Overall, the work presented in this section contributes a novel notion of robustness, *δ-robustness*, from the perspective of input-to-state stability. This

[5]The implementation of the algorithm, as well as its application towards evaluating the δ-robustness of bipedal walking gaits, is provided in the repository: https://github.com/maegant/deltaRobustness.git

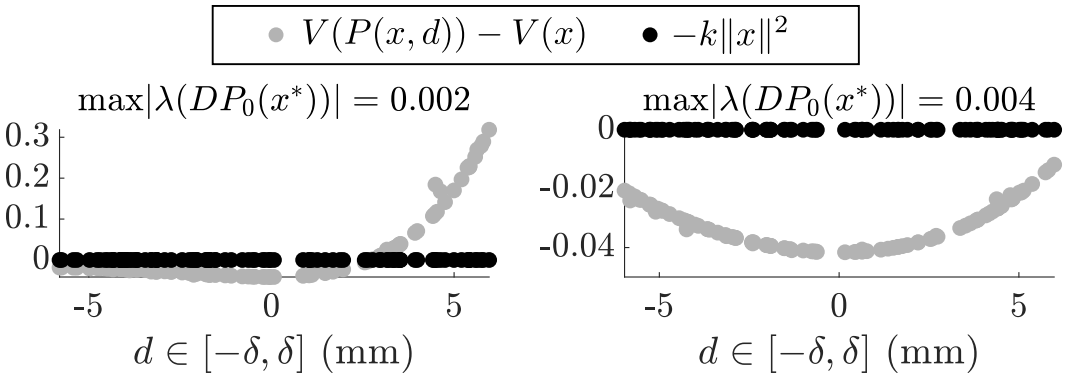


Figure 2.12: Illustration of the Lyapunov condition in (2.78) for 100 random samples $x \in B_{r_1}(0)$ with $d \sim \text{Uniform}(-6\text{mm}, 6\text{mm})$. As shown, the Lyapunov condition is satisfied for the gait identified as being δ-robust for $\delta = 6$mm with $\chi = 34$ (the corresponding ISS bound is illustrated in Figure 2.13).

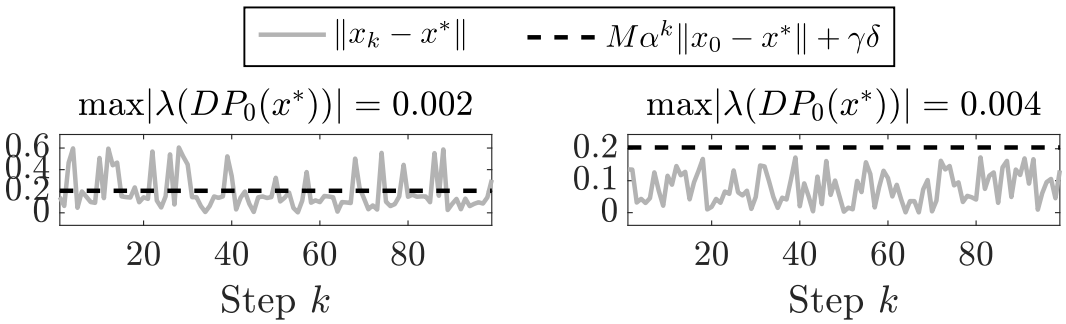


Figure 2.13: Verification of δ-robustness for $\delta^* = 6$mm and $\chi^* = 34$ (selected based on the algorithm results shown in Figure 2.11). As shown in the figure, Gait 1 was not δ-robust while Gait 2 was δ-robust with M, γ, and $\mathfrak{a}$ defined using the relationships derived in Theorem 2 and $V(x) = x^T\mathbf{P}x$.

definition differs from classic notions of stability by instead quantifying the magnitude of perturbations a periodic orbit can withstand while remaining stable. Robust Lyapunov conditions were also derived to certify δ-robustness for a nominal periodic orbit. Future work includes directly evaluating δ-robustness in the gait generation process to systematically generate periodic orbits that are robust to uncertain terrain. Additionally, in the next section we will discuss how the discrete-time Lyapunov condition can be translated to a stochastic condition in order to obtain more realistic (albeit probabilistic) estimates of the δ-robustness.

2.4 Extension to Input-to-State Stability in Probability

Despite the utility of input-to-state stability, this property requires the disturbance to be bounded and provides invariance and stability guarantees only

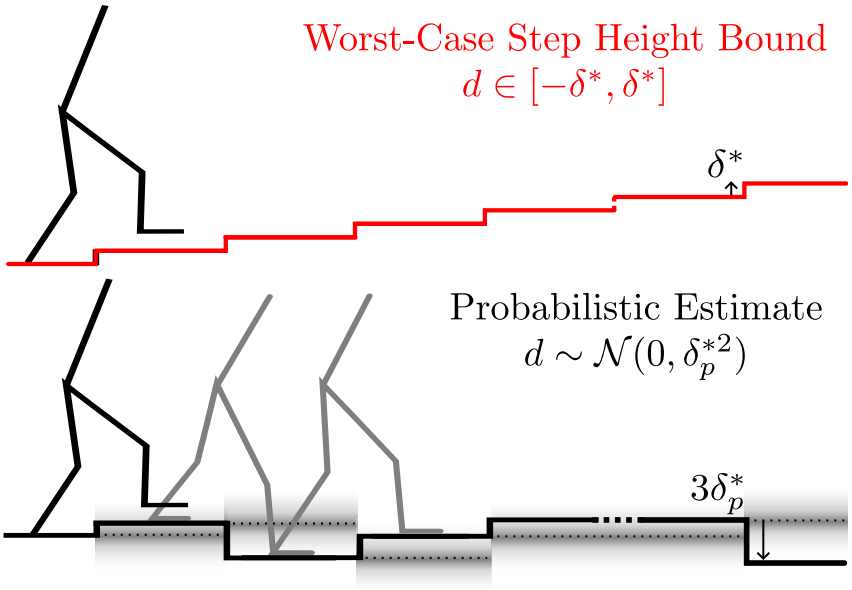


Figure 2.14: Input-to-state stability in probability (ISSp) generalizes input-to-state stability (ISS) to systems with unbounded disturbances. When applied to a seven-link walker traversing stochastic terrain, the presented work finds that the ISSp approach yields more reasonable estimates for the tolerable set of step heights. (Top): ISS-based guarantees must hold for any (bounded) disturbance signal; even for worst-case terrain (e.g., stairs) the walker must be able to remain stable. (Bottom): ISSp reasons instead about how systems behave over finite horizons. While the stochastic step heights (shown as gradients) can sometimes be large, their distribution is concentrated near zero, and thus the walker has a high probability of remaining upright.

with respect to this worst-case bound, as illustrated in Figure 2.14. Thus, the final thesis contribution in this chapter is the concept of *ISS in probability* (ISSp), which generalizes ISS to discrete-time systems subject to unbounded stochastic disturbances. Using tools from martingale theory, we provide Lyapunov conditions for a system to be exponentially ISSp, and connect ISSp to stochastic stability conditions found in literature. We exemplify the utility of this method through its application to the seven link walker confronted with step heights sampled from a truncated Gaussian distribution.

Preliminaries and Notation

Consider a general discrete-time autonomous system,

$$x_{k+1} = \mathbf{f}(x_k, \mathbf{d}_k) \tag{2.79}$$

with $k \in \mathbb{N}_{\geq 0}$, state $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$, equilibrium point $(x^*, \mathbf{d}^*) = (\mathbf{0}, \mathbf{0})$, random disturbance $\mathbf{d}_k \in \mathbb{R}^d$, and continuous dynamics $\mathbf{f} : \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}^n$. We assume each disturbance $\mathbf{d}_k \overset{\text{i.i.d.}}{\sim} \mathfrak{D}$ from some disturbance distribution $\mathfrak{D}$.

Definition 3 (Input-to-State Stability [108])**.** The system (2.79) is input-to-state stable (ISS) if there exist functions[6] $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$ such that, for each deterministic disturbance input $\mathbf{d}_k \in \mathbb{R}^m$ and each $x_0 \in \mathbb{R}^n$, it holds that

$$\|x_k\| \leq \beta(\|x_0\|, k) + \gamma\left(\sup_{k \in \mathbb{N}_{\geq 0}} \|\mathbf{d}_k\|\right) \tag{ISS}$$

for each $k \in \mathbb{N}_{\geq}0$ and some $p \geq 1$.

Intuitively, the bound on the state trajectory is a function of a sequence which converges to zero in time, $\beta(\|x_0\|, k)$ and a term which grows with respect to the disturbance bound, $\gamma\left(\sup_{k \in \mathbb{N}_{\geq 0}} \|\mathbf{d}_k\|\right)$. If $\|\mathbf{d}_k\| = 0$ for all k, then ISS systems are asymptotically stable. Note that a similar inequality regarding an essentially bounded disturbance distribution $\mathfrak{D}$:

$$\|x_k\| \leq \beta(\|x_0\|, k) + \gamma\left(\text{ess sup}\|\mathfrak{D}\|\right), \tag{2.80}$$

can be employed to achieve ISS almost surely, where ess sup is the essential supremum of the distribution $\mathfrak{D}$, also written as the L^∞-norm of $\mathfrak{D}$.

The general form of ISS-Lyapunov functions will now be presented. Note that in the preceeding section, the robust Lyapunov function was restricted to using constant constants $k_1, k_2, k_3 > 0$, while here the more general case with class $\mathcal{K}$ functions is presented.

Definition 4 (ISS-Lyapunov Function [108])**.** A continuous function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is an ISS Lyapunov function for (2.79) if there exist $\kappa_1, \kappa_2, \kappa_3 \in \mathcal{K}_\infty$ and $\kappa_4 \in \mathcal{K}$ such that:

$$\kappa_1(\|x\|) \leq V(x) \leq \kappa_2(\|x\|) \tag{2.81}$$

$$V(\mathbf{f}(x, \mathbf{d})) - V(x) \leq -\kappa_3(\|x\|) + \kappa_4(\|\mathbf{d}\|) \tag{2.82}$$

[6] A continuous function $\gamma : [0, a) \to [0, \infty)$ for $a > 0$ is said to belong to class $\mathcal{K}$ ($\gamma \in \mathcal{K}$) if it is strictly monotonically increasing and $\gamma(0) = 0$. If additionally $a = \infty$ and $\gamma(r) \to \infty$ as $r \to \infty$ then γ belongs to $\mathcal{K}_\infty$. A continuous function $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ is said to belong to class $\mathcal{KL}$ if for each fixed $s \geq 0$ the function $\beta(\cdot, s)$ is class $\mathcal{K}$ and for each $r \geq 0$ the function $\beta(r, \cdot)$ is decreasing and $\beta(r, s) \to 0$ as $s \to \infty$.

for all $x \in \mathbb{R}^n$ and all $\mathbf{d} \in \mathbb{R}^d$. V is exponential-ISS (E-ISS) Lyapunov function if there exist constants $a, b, c > 0$ and $\mathfrak{a} \in (0, 1)$ such that $\kappa_1(r) = ar^c, \kappa_2(r) = br^c, \kappa_3(r) = \mathfrak{a}r^c$.

The existence of an ISS-Lyapunov function is used to verify that the system is ISS via Lemma 3.5 in [108] (restated below).

Theorem 3 ([108])**.** *If there exists an ISS Lyapunov Function for system* (2.79)*, then system* (2.79) *is ISS.*

Next, a brief discussion of random variables, martingales, and other tools are presented at a level necessary to communicate these concepts clearly and accessibly. These tools will be used to generalize ISS to the case of unbounded, stochastic disturbances. Readers are referred to [76] for a precise measure-theoretic presentation of these ideas.

Consider disturbance signals which are sequences of random variables. A continuous random variable $\mathbf{y}$ sampled from a distribution $\mathcal{Y}$ (denoted $\mathbf{y} \sim \mathcal{Y}$) is a quantity that takes on values in $\mathbb{R}^y$ according to a probability density $p_{\mathbf{y}}(\mathbf{y}) \geq 0$, with $\mathbb{P}\{\mathbf{y} \in A\} \triangleq \int_A p_{\mathbf{y}}(\boldsymbol{\upsilon})d\boldsymbol{\upsilon}$. By definition $\int_{\mathbb{R}^y} p_{\mathbf{y}}(\boldsymbol{\upsilon})d\boldsymbol{\upsilon} = 1$ and the expectation of a random variable is given by $\mathbb{E}[\mathbf{y}] \triangleq \int_{\mathbb{R}^y} \boldsymbol{\upsilon} p_{\mathbf{y}}(\boldsymbol{\upsilon})d\boldsymbol{\upsilon}$.

Definition 5 (L^p Space [222])**.** A random variable $\mathbf{y} \sim \mathcal{Y}$ belongs to L^p (denoted as $\mathbf{y} \in L^p$), for $p > 0$, if

$$\|\mathbf{y}\|_{L^p} \triangleq \mathbb{E}\left[\|x\|^p\right]^{\frac{1}{p}} < \infty. \tag{2.83}$$

We call $\|\cdot\|_{L^p}$ the p-norm of a random variable, which is finite for any random variable in L^p. Intuitively, for $0 < p \leq q$, $L^q \subseteq L^p$ [222, Thm. 8.2] since random variables in L^q have tails that decay faster than those in L^p; additionally, L^∞ is the smallest L^p space and only contains random variables that are essentially bounded. Note that any norm $\|\cdot\|$ appearing without a subscript defines a typical norm on $\mathbb{R}^n$.

We can also reason about a random variable's conditional probability, i.e., its distribution given that another random variable has taken on a particular value. For two random variables X, Y the density of X given $Y = y$ is:

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x, y)}{p_Y(y)}, \tag{2.84}$$

where $p_{X,Y}(x, y)$ is the joint probability density of X, Y. The conditional expectation of X given $Y = y$ is $\mathbb{E}[X \mid Y]$.

The key tool used to reason about Lypaunov functions for our probabilisitc notion of ISS is a *nonnegative supermartingale*, a specific type of expectation-governed random process:

Definition 6. Let x_k be a sequence of random variables that take values in $\mathbb{R}^n$, $W : \mathcal{X} \times \mathbb{N}_{\geq 0} \to \mathbb{R}$, and suppose that $W(x_k, k) \in L^1$ for $k \in \mathbb{N}_{\geq 0}$. The process $W_k \triangleq W(x_k, k)$ is a supermartingale if:

$$\mathbb{E}[W_{k+1} \mid x_{0:k}] \leq W_k \text{ almost surely for all } k \in \mathbb{N}_{\geq 0}, \tag{2.85}$$

where $x_{0:k}$ indicates the random variables $\{x_0, x_1, \ldots, x_k\}$. If, additionally, $W_k \geq 0$ for all $k \in \mathbb{N}_{\geq 0}$, W_k is a nonnegative supermartingale. If the process is non-decreasing in expectation, the process W_k is a submartingale. If the inequality (2.85) holds with equality, the process W_k is a martingale.

An important result from martingale theory that we will use is *Ville's inequality*, which bounds the probability that a nonnegative supermartingale rises above a certain value:

Theorem 4 (Ville's Inequality [212])**.** *Let W_k be a nonnegative supermartingale. Then for all $\lambda \in \mathbb{R}_{>0}$,*

$$\mathbb{P}\left\{\sup_{k \in \mathbb{N}} W_k > \lambda\right\} \leq \frac{\mathbb{E}[W_0]}{\lambda}. \tag{2.86}$$

Intuitively, Ville's inequality can be compared with Markov's inequality for nonnegative random variables; since the process W_k is nonincreasing in expectation, Ville's inequality allows us to reason about the probability the process instead reaches some value above λ.

Stability of Stochastic Discrete-Time Systems

Traditional notions of stability may not necessarily apply to stochastic systems. For example, asymptotic stability to a point or forward invariance of a bounded set may be impossible in the presence of unbounded, stochastic disturbances. Thus more nuanced notions of stability are required [123]. In this section we provide an abridged discussion of three existing stability notions for stochastic

systems: recurrence, boundedness of trajectories in probability, and input-to-state stability for distributions with bounded support.

Recurrence. Recurrence is an important notion of stability used in the analysis of Markov chains. A bounded set $A \subset \mathcal{X}$ is recurrent if trajectories enter A in finite time and visit A infinitely often with probability 1 for all initial conditions $x_0 \in \mathcal{X}$.

Definition 7 (Recurrence)**.** For some bounded set $A \subset \mathcal{X}$ let the hitting time $\tau_A(x) \triangleq \inf\{k \in \mathbb{N}_{\geq 0} \text{ s.t. } x_k \in A,\ x_0 = x\}$. A set A is recurrent if for every $x \in \mathcal{X}$, $\mathbb{P}\{\tau_A(x) < \infty\} = 1$. We say a system (2.79) is recurrent if there exists a recurrent set A.

Recurrence relates to the notion of stability for deterministic systems where trajectories remain within a set for all time, a property which is guaranteed for ISS systems. We refer the reader to [140] for a more thorough treatment of Markov chain stablity, recurrence, and ergodic theory.

Boundedness in Probability. Another notion of stability for stochastic systems is the probability that the state remains in a bounded region. Since it is often impossible to keep trajectories of (2.79) bounded for all time [181], it is common to discuss these probabilities over some finite horizon $k \in \{0, \dots, K\}$ for some $K \in \mathbb{N}_{\geq 0}$.

Definition 8 (Bounded in Probability)**.** The system (2.79) is bounded in probability for some $K \in \mathbb{N}_{\geq 0}$ if there exists an $M > 0$ and $\varepsilon \in (0, 1)$ such that:

$$\mathbb{P}\left\{\max_{k \leq K} \|x_k\| \leq M\right\} \geq 1 - \varepsilon. \tag{2.87}$$

This notion of stability is central to Harold Kushner's work on on stochastic stability [122] which is drawn on for the later proposed definition of ISSp, and which formed the basis for recent martingale-based approaches to finite-time stability [181] and safety [57, 167, 168] for systems with unbounded uncertainty. This relates directly to the forward invariant region guaranteed to exist around the equilibrium point of ISS systems.

ISS for Bounded Disturbance Distributions. If the disturbance distribution $\mathfrak{D}$ for system (2.79) is only supported on a bounded set, then the essential

supremum $\|\mathfrak{D}\|_{L^\infty}$ is well defined; thus if a system satisfies the ISS condition, it is said to be stable in the ISS sense. Several authors have worked to extend ISS to the setting of unbounded stochastic disturbances. [204] proposed an ISS condition for continuous-time systems with unbounded disturbances, but required the disturbance magnitude to be upper bounded by a class-$\mathcal{K}$ of the state norm (thus, the disturbance vanishes at the equilibrium, a common but restrictive assumption). [139, 197] also study stochastic variants of ISS, but only require that the (ISS) condition hold for the expected trajectory (which does not guarantee boundedness of any trajectories).

Input-to-State Stability for Unbounded Random Disturbances

In this section, we seek to generalize the notion of input-to-state stability to systems that are subject to unbounded random disturbances. Specifically, two issues arise when the support of $\mathfrak{D}$ is unbounded: (i) the essential supremum $\|\mathfrak{D}\|_{L^\infty}$ does not exist, rendering the ISS condition inapplicable, and (ii) the probability that x_k remains in any bounded set for all $k \in \mathbb{N}$ is generally zero.

This second point is somewhat non-intuitive; however, consider a system with additive Gaussian noise, $x_{k+1} = f(x_k) + \mathbf{d}_k$, with $\mathbf{d}_k \sim \text{Normal}(\mu, \Sigma)$. Then, since the tails of $\mathbf{d}$ are unbounded, for any $B > 0$, $\mathbb{P}\{\|\mathbf{d}\| > 2B\} = \varepsilon > 0$. This means, with probability ε, $\|\mathbf{f}(x, \mathbf{d})\| \geq \|\mathbf{d}\| - \|\mathbf{f}(x)\| > 2B - B = B$.[7] Thus, for any $K \in \mathbb{N}_{\geq 0}$,

$$\mathbb{P}\{\|x_k\| < B,\ \forall k \leq K\} \leq \mathbb{P}\{\|\mathbf{d}_k\| \leq 2B,\ \forall k \leq K\} \tag{2.88}$$

$$= (1 - \varepsilon)^K, \tag{2.89}$$

since all $\mathbf{d}_k$ are independent. Thus, as $K \to \infty$, the probability of the state remaining bounded goes to zero. Therefore, when generalizing ISS to the case of unbounded disturbances, we should expect weaker guarantees than those provided by the typical condition (ISS). With this in mind, we now define the main result of this section, *Input-to-State Stability in Probability (ISSp)*, which is well-defined for systems subject to unbounded noise.

Definition 9 (Input-to-State Stable in Probability)**.** The system (2.79) is *input-to-state stable in probability (ISSp)* with repect to L^p if, for any $\varepsilon \in (0, 1)$, $K \in \mathbb{N}_{\geq 0}$ and distribution $\mathfrak{D}$ such that $\|\mathfrak{D}\|_{L^p}$, there exist functions $\beta \in \mathcal{KL}$,

[7] We must have $\|\mathbf{f}(x_k)\| \leq B$ for $\|x_k\| \leq B$; otherwise deterministic trajectories starting at x_k would leave the set in one step.

and $\gamma \in \mathcal{K}$ such that:

$$\mathbb{P}\Big\{ \|x_k\| \leq \beta(\|x_0\|, k) + \gamma\big(\|\mathfrak{D}\|_{L^p} \big), \forall k \leq K \Big\} \geq 1 - \varepsilon. \tag{2.90}$$

If this holds for $\beta(\|x_0\|, k) = M\mathfrak{a}^k \|x\|$, for $M > 0, \mathfrak{a} \in (0,1)$, the system is exponentially input-to-state stable in probability (ISSp).

ISSp is a generalization of ISS to systems with (unbounded) stochastic disturbances. Intuitively, a system is ISSp if, for any disturbance in L^p, the ISS condition (ISS) (with the L^∞ norm relaxed to the L^p) holds over a finite horizon with a probability arbitrarily close to 1. As with ISS, we now relate ISSp to Lyapunov functions which can be used to verify this property.

Definition 10 (ISSp Lyapunov Function)**.** A continuous function $V : \mathbb{R}^n \to \mathbb{R}_{\geq}0$ is an *ISSp Lyapunov Function* for the system (2.79) if there exist functions $\kappa_1, \kappa_2, \kappa_3 \in \mathcal{K}_\infty$ and $\kappa_4 \in \mathcal{K}$ such that,

$$\kappa_1(\|x\|) \leq V(x) \leq \kappa_2(\|x\|) \tag{2.91}$$

$$\mathbb{E}[V(\mathbf{f}(x, \mathbf{d}) - V(x)] \leq -\kappa_3(V(x)) + \kappa_4(\|\mathfrak{D}\|_{L^p}) \tag{2.92}$$

for all $x \in \mathcal{X}$ and $\|\mathfrak{D}\|_{L^p} < \infty$. If there exist constants $a, b, c > 0$ and $\mathfrak{a} \in (0,1)$ such that $\kappa_1(r) = ar^c, \kappa_2(r) = br^c$, and $\kappa_3(r) = \mathfrak{a}r$, then V is an *Exponential ISSp (E-ISSp) Lyapunov Function.*

Remark 2. As in the typical ISS definition (ISS), since $\max\{a, b\} \leq a + b \leq \max\{2a, 2b\}$, for suitable choices of β, γ, the ISSp condition (2.90) is equivalent to:

$$\mathbb{P}\Big\{ \|x_k\| \leq \max\left\{\beta(\|x_0\|, k), \gamma\big(\|\mathfrak{D}\|_{L^p}\big)\right\}, \ \forall k \leq K \Big\} \geq 1 - \varepsilon. \tag{2.93}$$

In this work, for simplicity of exposition, we will consider exponential ISSp. Note that the results presented apply in the more general case, but the proofs become more complex.

Lyapunov Conditions for E-ISSp

As for ISS, there exist Lyapunov conditions for E-ISSp. To this end, we will use tools from martingale theory (in particular, Ville's inequality) to demonstrate that the existence of a Lyapunov function satisfying a drift condition in expectation implies a system is E-ISSp.

Theorem 5. *If there exists an E-ISSp Lyapunov function for system* (2.79)*, then system* (2.79) *is E-ISSp.*

Proof. We begin by constructing a nonnegative supermartingale $W(x_k, k)$ via a time-varying, affine transform of the Lyapunov function $V(x_k)$. Rearranging the Lyapunov drift condition (2.92) with $\kappa_3(r) = \mathfrak{a}r$ for some $\mathfrak{a} \in (0, 1)$, we can see that $V(x_k)$ almost resembles a supermartingale[8],

$$\mathbb{E}\left[V(x_{k+1}) \mid x_k\right] \leq (1 - \mathfrak{a})V(x_k) + \varphi \tag{2.94}$$

where we define $\varphi \triangleq \kappa_4(\|\mathfrak{D}\|_{L^p})$. However, this is not exactly a supermartingale due to the $(1 - \mathfrak{a})$ scaling and the additive constant φ.

Thus, for a particular horizon $K \in \mathbb{N}_{\geq 0}$, we construct $W(x_k, k)$ by undoing this scaling and translation. Letting $W_k \triangleq W(x_k, k)$ for simplicity, this construction is:

$$W_k = \underbrace{\theta^k V(x_k)}_{\text{rescale}} \underbrace{- \varphi \sum_{i=1}^{k} \theta^i}_{\text{translate}} + \underbrace{\varphi \sum_{i=1}^{K} \theta^i}_{\text{ensure } W_k \geq 0}, \tag{2.95}$$

with $\theta \triangleq \frac{1}{1-\mathfrak{a}}$ and the constant term $\varphi \sum_{i=1}^{K} \theta^i$ added to ensure $W_k \geq 0$.

Next we now show W_k is a nonnegative supermartingale. We have $W_k \geq 0$ for any $x_k \in \mathcal{X}$, since $V(x_k) \geq 0$ by definition, and $\theta, \varphi \geq 0$. Further, we have:

$$\mathbb{E}\big[W_{k+1} \mid x_k\big] = \mathbb{E}\left[\theta^{k+1} V(x_{k+1}) + \varphi \sum_{i=k+2}^{K} \theta^i\right] \tag{2.96}$$

$$\leq \theta^{k+1}\left((1 - \mathfrak{a})V(x_k) + \varphi\right) + \varphi \sum_{i=k+2}^{K} \theta^i \tag{2.97}$$

$$= \theta^k V(x_k) + \varphi \sum_{i=k+1}^{K} \theta^i = W_k, \tag{2.98}$$

where the inequality (2.97) follows from the drift condition (2.92) and (2.98) uses the fact that $\theta = \frac{1}{1-\mathfrak{a}}$.

Since W_k is a nonnegative supermartingale, we can apply Ville's inequality (2.86) to bound the probability W_k that remains below any $\lambda > 0$ for all $k \leq K$. Specifically,

$$\mathbb{P}\Big\{W(x_k) \leq \lambda,\ \forall k \leq K\Big\} \geq 1 - \frac{W(x_0)}{\lambda}. \tag{2.99}$$

[8]Note that $\mathbb{E}\left[V(x_{k+1}) \mid x_k\right] = \mathbb{E}\left[V(x_{k+1}) \mid x_{0:k}\right]$ since system (2.79) is Markovian.

We also note that, using the geometric series identity $\sum_{i=1}^{k} \theta^{i-1} = \frac{\theta^k - 1}{\theta - 1}$, we can write W_k as:

$$W_k = \theta^k V(x_k) + \frac{\theta\varphi(\theta^K - \theta^k)}{\theta - 1}. \tag{2.100}$$

Examining the structure of W_k, if for all $k \leq K$ we have $W_k \leq \lambda$, rearranging (2.100) results in:

$$\begin{aligned} V(x_k) &\leq \left(\lambda - \frac{\theta^{K+1}\varphi}{\theta - 1}\right)\theta^{-k} + \frac{\theta}{\theta - 1}\varphi && (2.101)\\ &\leq (\lambda - \varphi)\theta^{-k} + \frac{\theta}{\theta - 1}\varphi && (2.102)\\ &\triangleq \left(M\left\|x_0\right\|^c + \eta\varphi\right)\theta^{-k} + \frac{\theta}{\theta - 1}\varphi && (2.103)\\ &\leq M\left\|x_0\right\|^c \theta^{-k} + \eta\varphi + \frac{\theta}{\theta - 1}\varphi. && (2.104) \end{aligned}$$

Inequalities (2.102) and (2.104) follows from $\theta > 1$ and $\varphi \geq 0$. Equality (2.103) follows from choosing $\lambda = M\left\|x_0\right\|^c + (1 + \eta)\,\varphi$ for some $M,\ \eta \geq 0$ and $c > 0$.

Further, using the lower bound (2.91) on $V(x_k)$ and the definition of θ, we can write:

$$a\left\|x_k\right\|^c \leq M\left\|x_0\right\|^c (1 - \mathfrak{a})^k + \left(\eta + \frac{1}{\mathfrak{a}}\right)\varphi \tag{2.105}$$

for some $a > 0$ which, rearranging, and raising both sides to the power of $\frac{1}{c}$ (which preserves order since $c > 0$), yields:

$$\begin{aligned} \left\|x_k\right\| &\leq \left(\frac{M}{a}\left\|x_0\right\|^c (1 - \mathfrak{a})^k + \frac{(\eta + \frac{1}{\mathfrak{a}})}{a}\varphi\right)^{\frac{1}{c}} && (2.106)\\ &\leq \left(\frac{M}{a}\right)^{\frac{1}{c}} \zeta\left\|x_0\right\| (1 - \mathfrak{a})^{\frac{k}{c}} + \zeta\left(\frac{(\eta + \frac{1}{\mathfrak{a}})\varphi}{a}\right)^{\frac{1}{c}} && (2.107)\\ &\triangleq \tilde{M}\tilde{\mathfrak{a}}^k\left\|x_0\right\| + \gamma_\eta(\|\mathfrak{D}\|_{L^p}), && (2.108) \end{aligned}$$

for $\tilde{M} > 0$, $\tilde{\mathfrak{a}} \in (0, 1)$, some $\zeta > 0$ as needed, and $\gamma_\eta(r) \triangleq \zeta\left(\frac{\eta + \frac{1}{\mathfrak{a}}}{a}\right)\kappa_4(r)^{\frac{1}{c}}$ which is a class $\mathcal{K}$ function for all $\eta \geq 0$.

Thus, we now must ensure there exists a suitable choice of M, η such that the probability that this bound holds for all $k \in \{1, \ldots, K\}$ is greater than $1 - \varepsilon$. By Ville's inequality,

$$\begin{aligned} &\mathbb{P}\left\{\left\|x_k\right\| \leq \tilde{M}\tilde{\mathfrak{a}}^k\left\|x_0\right\| + \gamma_\eta(\|\mathfrak{D}\|_{L^p}),\ \forall k \leq K\right\}\\ &\quad\geq 1 - \frac{W_0}{\lambda} = 1 - \frac{V(x_0) + \frac{\varphi}{\mathfrak{a}}((1 - \mathfrak{a})^{-K} - 1)}{M\left\|x_0\right\|^c + (1 + \eta)\varphi}, \end{aligned} \tag{2.109}$$

with $W_0 \geq 0$ by definition. Thus, as long as $\|x_0\|^c$, $\|\mathfrak{D}\|_{L^p}$ are not both zero, we can choose M, η large enough to have

$$\mathbb{P}\Big\{ \|x_k\| \leq \tilde{M}\tilde{\mathfrak{a}}^k \|x_0\| + \gamma_\eta(\|\mathfrak{D}\|_{L^p}),\ k \leq K \Big\} \geq 1 - \varepsilon, \tag{2.110}$$

for any $\varepsilon \in (0,1)$, so the system must be E-ISSp.

□

Remark 3. The variables $M \geq 0$ and $\eta \geq 0$ are free parameters which can be varied to analyze the probability of convergence and boundedness, respectively. Also, we note that the bound in (2.110) may be very weak; stronger bounds can be achieved by removing the bounding steps in (2.102) and (2.104), but clarity was chosen over tightness for this proof.

Connections to Other Stability Notions

Here we discuss connections between ISS, ISSp, and other notions of stability for stochastic systems.

Corollary 1. *If the system* (2.79) *is ISS, then it is ISSp with respect to* L^∞.

Proof. By definition, if a system is ISS, then for all $k \in \mathbb{N}_{\geq 0}$, there exist $\beta \in \mathcal{KL}, \gamma \in \mathcal{K}$ such that:

$$\|x_k\| \leq \beta(\|x_0\|, k) + \gamma(\vartheta) \tag{2.111}$$

for all $\vartheta \geq \sup_{k \in \mathbb{N}_{\geq 0}} \|\mathbf{d}_k\|$.

Thus, since the L^∞-norm (equivalently, the essential supremum) is finite for all random variables in L^∞, for any $\mathbf{d} \sim \mathfrak{D}$ with $\mathbf{d} \in L^\infty$, we have:

$$\mathbb{P}\Big\{ \|x_k\| \leq \beta(\|x_0\|, k) + \gamma(\|\mathfrak{D}\|_{L^\infty}),\ \forall k \in \mathbb{N} \Big\} = 1.$$

Thus trivially, for any $K \in \mathbb{N}_{\geq 0}$, $\varepsilon \in (0,1)$, we have $\beta \in \mathcal{KL}, \gamma \in \mathcal{K}$ such that (2.90) holds for all distributions $\mathfrak{D}$ with $\mathbf{d} \in L^\infty$. Therefore, the system is ISSp w.r.t. L^∞. □

Corollary 1 provides a clear connection between ISS and ISSp: if a system is ISS, it by definition is ISSp for disturbances in L^∞. Next, we discuss the relationship between ISS and ISSp w.r.t. L^2 which is a much larger class of unbounded random variables.

Corollary 2. *If the system* (2.79) *is additive with respect to its disturbance and admits a twice-continuously differentiable, convex E-ISS Lyapunov function* $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *such that* $\sup_{x \in \mathcal{X}} \|\nabla^2 V(x)\|_2 \leq \lambda_{max}$ *for some* $\lambda_{max} \geq 0$*, then it is E-ISSp for* $\mathbf{d} \in L_p$ *with* $\mathbb{E}[\mathbf{d}] = 0$ *for* $p \geq 2$*.*

Proof. The dynamics are additive with respect to the disturbance so system (2.79) can be rewritten as:

$$x_k = \mathbf{f}(x_k, \mathbf{d}) \triangleq \widehat{\mathbf{f}}(x_k) + \mathbf{d}. \tag{2.112}$$

The function V is a E-ISS Lyapunov function for (2.112) so it satisfies:

$$V\left(\widehat{\mathbf{f}}(x) + \mathbf{d}\right) - V(x) \leq -\mathfrak{a} V(x) + \kappa_4(\|\mathfrak{D}\|_{L^\infty}) \tag{2.113}$$

for all $x \in \mathcal{X}$, some $\mathfrak{a} \in (0, 1)$, $\sigma \in \mathcal{K}$, and any $\mathbf{d} \in L^\infty$.

The expected value of the left side of this inequality is:

$$\begin{aligned}
&\mathbb{E}\left[V\left(\widehat{\mathbf{f}}(x) + \mathbf{d}\right) - V(x)\right] = \mathbb{E}\left[\mathbf{V}\left(\widehat{\mathbf{f}}(x) + \mathbf{d}\right)\right] - V(x) \\
&\leq V\left(\widehat{\mathbf{f}}(x) + \mathbb{E}[\mathbf{d}]\right) - V(x) + \frac{\lambda_{\max}}{2}\mathrm{tr}(\mathrm{cov}(\mathbf{d})) && (2.114) \\
&= V\left(\widehat{\mathbf{f}}(x) + 0\right) - V(x) + \frac{\lambda_{\max}}{2}\mathrm{tr}(\mathrm{cov}(\mathbf{d})) && (2.115) \\
&\leq -\mathfrak{a} V(x) + \frac{\lambda_{\max}}{2}\mathrm{tr}(\mathrm{cov}(\mathbf{d})), && (2.116)
\end{aligned}$$

where (2.114) accounts for Jensen's inequality as in [57, Lemma 1], (2.115) is due to the assumption that the $\mathbb{E}[\mathbf{d}] = 0$, and (2.116) is an application of the E-ISS bound (2.113).

Since bounded covariance implies boundedness in L^2, if $\mathbf{d} \in L^2$ then V is an E-ISSp Lyapunov function for (2.112). Furthermore, since $L^2 \supseteq L_p$ for $p \geq 2$, V is an E-ISSp Lyapunov function for $\mathbf{d} \in L^p$ for all $p \geq 2$. □

Next, we discuss the relationship between ISSp and trajectories that are bounded in probability.

Corollary 3. *If system* (2.79) *is ISSp w.r.t.* L^p*, then for any* $\mathbf{d} \in L^p$*, the system's trajectories are bounded in probability.*

Proof. If the system is ISSp w.r.t. L^p, then for any $K \in \mathbb{N}_{\geq 0}$ and $\varepsilon \in (0, 1)$, there exist $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$ such that:

$$\mathbb{P}\left\{\|x_k\| \leq \beta(\|x_0\|, k) + \gamma\left(\|\mathfrak{D}\|_{L^p}\right),\ \forall k \leq K\right\} \geq 1 - \varepsilon. \tag{2.117}$$

Then, since β is decreasing in k, we have that for $B_0 \triangleq \beta(\|x_0\|, 0) + \gamma(\|\mathfrak{D}\|_{L^p})$:

$$\mathbb{P}\{\|x_k\| \leq B_0,\ \forall k \leq K\} \geq 1 - \varepsilon. \tag{2.118}$$

Thus the system trajectories are bounded in probability. □

Like with traditional ISS, the ISSp condition 2.90 is equivalent to system trajectories remaining in a ball whose radius scales with the initial condition and the norm of the disturbance. Thus, if a system is ISSp, its trajectories (over a finite horizon) must be bounded in probability.

Finally, we look to discuss the relationship between ISSp and recurrence. To do this we, first restate an important result from drift analysis (see [126] for a detailed survey).

Theorem 6 (Variable Drift [126])**.** *Suppose there exists some function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$, with γ-sublevel set $V_\gamma \triangleq \{x \in \mathcal{X} \mid V(x) \leq \gamma\}$ such that for all $x \in \mathcal{X} \setminus V_\gamma$:*

$$\mathbb{E}[V(\mathbf{f}(x, \mathbf{d})) - V(x)] \leq -h(V(x)), \tag{2.119}$$

for some increasing function $h : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$. Then, for any trajectory with initial state x_0, the hitting time $\tau_\gamma(x_0) = \inf\{k \mid V(x_k) \leq \gamma\}$ is bounded in expectation by:

$$\mathbb{E}[\tau_\gamma(x_0)] \leq \frac{\gamma}{h(\gamma)} + \int_\gamma^{V(x_0)} \frac{1}{h(\sigma)} d\sigma. \tag{2.120}$$

Using this result, we can show that, if a system admits an E-ISSp Lyapunov function, then any Lyapunov sublevel set (above a particular value) must be recurrent.

Theorem 7. *If there exists an E-ISSp Lyapunov function w.r.t. L^p for system* (2.79)*, then* (2.79) *is recurrent.*

Proof. Suppose there exists an E-ISSp Lyapunov function V for the system (2.79). Then, for $h(V(x_k)) = \mathfrak{a}V(x_k) - \varphi$, with $\varphi \triangleq \kappa_4(\|\mathfrak{D}\|_{L^p})$, we have:

$$\mathbb{E}[V(x_{k+1}) - V(x_k) \mid x_k] \leq -h(V(x_k)). \tag{2.121}$$

For any $\gamma > \frac{\varphi}{\mathfrak{a}}$, $h(V(x)) > 0$ for all $x \in \mathcal{X} \setminus V_\gamma$; thus our system meets the variable drift condition (2.119).

Thus, consider some trajectory with an initial state $x_0 \in \mathcal{X} \setminus V_\gamma$. Then, by Theorem 6, we have

$$\mathbb{E}\left[\tau_\gamma(x_0)\right] \leq \frac{\gamma}{\mathfrak{a}\gamma - \varphi} + \int_\gamma^{V(x_0)} \frac{1}{\mathfrak{a}\sigma - \varphi} d\sigma \tag{2.122}$$

$$\leq \frac{\gamma}{\mathfrak{a}\gamma - \varphi} + \frac{1}{\mathfrak{a}} \log\left(\frac{\mathfrak{a}V(x_0) - \varphi}{\mathfrak{a}\gamma - \varphi}\right) < \infty. \tag{2.123}$$

Since $\mathbb{E}\left[\tau_\gamma(x_0)\right] < \infty$, we must have $\mathbb{P}\{\tau_\gamma(x_0) < \infty\} = 1$. Thus, for any $\gamma > \frac{\mathfrak{a}}{\varphi}$, the sublevel set V_γ is recurrent. Since κ_1 is radially unbounded, V_γ must be bounded for all $\gamma \geq 0$, thus the system is recurrent. □

Application to Walking

Finally, to obtain more reasonable estimates of the maximum step heights a given periodic gait can withstand, we will consider step heights drawn from some distribution $d_k \sim \mathfrak{D}$ and apply the ISSp methodology. Specifically, we take $\mathfrak{D} \triangleq \text{Normal}(0, \delta_p^2)$ such that $\delta_p > 0$ now represents the standard deviation of the distribution. Note that due to the partial nature of the $\mathcal{P}$, we will truncate $\mathfrak{D}$ at $3\delta_p$ to ensure that there exists some δ_p such that $\mathbb{E}[\Delta V] < +\infty$, we will denote this truncated Gaussian as $\text{Normal}(\cdot, \cdot, \pm a)$ where a denotes the truncation interval [76].

Consider the seven-link walker as shown in Figure 2.8 and Figure 2.14. As detailed in Section 2.3, walking can be distilled down to the discrete-time dynamical system described by the Poincaré return map:

$$\mathcal{P} : B_\rho(x^*) \times [d^-, d^+] \rightharpoonup \mathcal{S}_{[d^-,d^+]} := \bigcup_{d \in [d^-,d^+]} \mathcal{S}_d,$$

$$x_{k+1} = \mathcal{P}(x_k, d_k), \qquad d_k \in [d^-, d^+], \tag{2.124}$$

for some sequence of step heights $d_k \in [d^-, d^+] \subset \mathbb{R}$, and $k \in \mathbb{N}_{\geq 0}$.

Also recall that in Section 2.3, a candidate robust Lyapunov function was synthesized by approximating the exponentially stable discrete-time system using the linearization of the Poincaré return map for $d_k = 0$:

$$x_{k+1} = \mathbf{A}x_k := D\mathcal{P}(0,0)x_k. \tag{2.125}$$

Then, the Lyapunov matrix $\mathbf{P} = \mathbf{P}^T > 0$ was obtained by solving the discrete-time Lyapuov equation ($\mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P} = -\mathbf{Q}$) for $\mathbf{Q} = \mathbf{Q}^T > 0$ which provides a discrete-time Lyapunov function $V(x) = x^T\mathbf{P}x$.

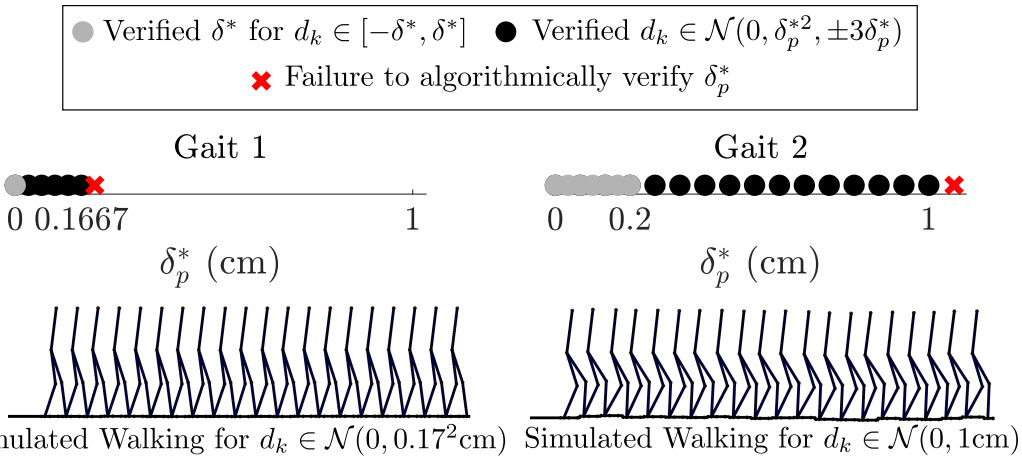


Figure 2.15: Algorithmic results of stochastic ISSp compared to deterministic ISSp. As shown, the stochastic ISSp condition yields much more realistic predictions of the tolerable step heights for two gaits (the same gaits as those compared in Figure 2.12).

Consider the ISSp Lyapunov condition from Def. (10):

$$\mathbb{E}[V(\mathcal{P}(x,d)) - V(x) \mid x] \leq -\tilde{\mathfrak{a}} V(x) + \tilde{\sigma}, \tag{2.126}$$

with $\tilde{\mathfrak{a}} = 2k\lambda_{\max}(\mathbf{P})$, and $\tilde{\sigma} = k(\chi\delta_p)^c$, where $k \in (0,1)$ is a user-defined variable dictating the convergence of the Lyapunov condition. Note that this condition can be equivalently expressed in the form:

$$\begin{aligned} &\|x - x^*\| \geq \chi\delta_p \quad \Longrightarrow \\ &\quad \mathbb{E}[V(\mathcal{P}(x,d)) - V(x) \mid x] \leq -k\|x - x^*\|^c. \end{aligned} \tag{2.127}$$

This ISSp Lyapunov function can be utilized with the algorithmic approach introduced earlier for (2.78) to solve the following optimization problem:

$$\begin{aligned} (\delta_p^*, \chi_p^*) = \underset{\delta_p, \chi_p > 0}{\text{argmax}}\ & \delta_p \\ \text{s.t. } & \mathbb{E}_{d\sim\mathfrak{D}}[V(\mathcal{P}(x,d)) - V(x) \mid x] \leq -k\|x\|^2 \\ & \forall\ \|x\| = \chi_p\delta_p, \quad d \sim \text{Normal}(0, \delta_p^2, \pm 3\delta_p), \end{aligned} \tag{2.128}$$

As shown in Figure 2.15, the algorithmic approach to the updated optimization (2.128) results in more reasonable estimates of the maximum tolerable step height for each of the two gaits considered in [205].

Probabilistic Guarantees for ISSp. While relaxing the Lyapunov condition to the one in (2.78) yields more realistic estimates of δ^*, this relaxed condition

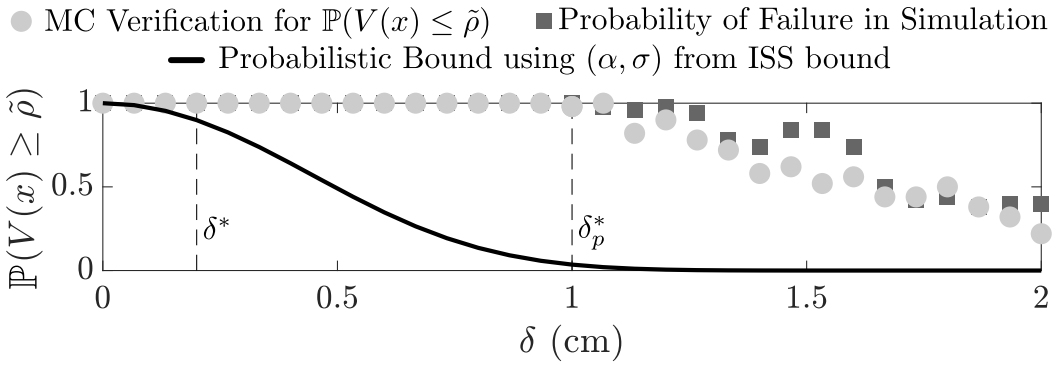


Figure 2.16: The probabilistic bound, evaluated for Gait 2, quickly decreases as δ increases. However, the simulation results show that the walking is able to remain periodic for all values determined to be δ-robust in Opt. 2.78, highlighting that ISSp yields more reasonable estimates of δ compared to the strict ISS condition presented in Equation 2.66.

no longer satisfies the assumptions needed to be provably ISS. Instead, we can use probabilistic bounds to assert that the system is ISSp.

To do this, we first need to approximate a reasonable estimate of the Lyapunov level set that bounds the evolution of the system after K steps. Rearranging the Lyapunov condition (2.126) for the largest χ_p^* and δ_p^* identified by Opt. (2.78), and using the fact that the forward invariant set is defined in Theorem 2 as the set such that $V(x) \leq \kappa_2(\chi\delta)^c$, we obtain the Lyapunov bound:

$$V(x_K) \leq \tilde{\rho} \triangleq (1-\mathfrak{a})^K \lambda_{\max}(\mathbf{P})(\chi_p^*\delta_p^*)^2 + k(\chi_p^*\delta_p^*)^2.$$

The probabilistic bound associated with remaining within this Lyapunov level set can be obtained from Kushner [122]. Importantly, when your Lyapunov level set $\tilde{\rho}$ is less than $\tilde{\sigma}/\mathfrak{a}$, the bound is extremely conservative. Following Kushner, it is possible to find a better choice of W_k that yields a better probability bound. Specifically, one can use the bound:

$$\begin{aligned} &\mathbb{P}(V(x_K) \leq \tilde{\rho}),\ \forall k \leq K\} \\ &\quad \geq \begin{cases} \frac{\tilde{\rho}-V(x_0)}{\tilde{\rho}} \left(\frac{\tilde{\rho}-\tilde{\sigma}}{\tilde{\rho}}\right)^K, & \tilde{\rho} \geq \frac{\tilde{\sigma}}{\mathfrak{a}} \\ 1 - \frac{V(x_0)(1-\mathfrak{a})^K + \tilde{\sigma}\sum_{i=1}^{K}(1-\mathfrak{a})^{i-1}}{\tilde{\rho}}, & \text{otherwise} \end{cases} \end{aligned} \tag{2.129}$$

with $\tilde{\sigma}$ determined for each evaluated δ_p using a sampling technique such as Monte Carlo sampling.

In Figure 2.16, the probabilistic bound is illustrated for the values (χ_p^*, δ_p^*) obtained using (2.78) for Gait 2. To verify the probabilistic bound, Monte

Carlo sampling was implemented to estimate the true probability that the system remains within $\tilde{\rho}$ after $K = 10$ steps. We simulate the system for this horizon and report both the fraction of trajectories remaining stable, as well as the fraction of trajectories remaining in the Lyapunov sublevel set $\tilde{\rho}$.

2.5 Summary

In summary, while the HZD method of gait generation produces nominal reference trajectories that enjoy mathematical guarantees of stability, these certificates are only valid when the robot has a pre-impact state infinitesimally close to the fixed point of the generated limit cycle. Thus, to improve the robustness of nominal gaits, impact uncertainty must be accounted for. This thesis presented two mathematical approaches towards addressing impact uncertainty. First, by maximizing the induced matrix norm of the saltation matrix directly in the gait generation optimization problem, it was found that the generated gaits led to more robust locomotion on both the Atalante exoskeleton and the AMBER-3M planar biped. Second, a new definition of locomotive robustness was proposed – δ-robustness – which leverages input-to-state stability for disturbances to the guard condition (taken to be ground height). Using this new definition of robustness, Lyapunov conditions can be synthesized to certify robustness of nominal gaits to bounded disturbances. This definition was also extended to *input-to-state stability in probability* (ISSp). This stochastic notion of robustness was shown to yield more realistic estimates of robustness.

Chapter 3

USER-CUSTOMIZATION VIA SUBJECTIVE FEEDBACK

As discussed in the preceding chapter, the partial hybrid zero dynamics (PHZD) method of gait generation has been successfully demonstrated towards realizing dynamically stable crutch-less exoskeleton locomotion on the Atalante exoskeleton. While this method, originally designed for bipedal robots [3, 79, 85], generates stable bipedal locomotion, there is no existing framework for optimizing the walking for user comfort; yet, user comfort should be a critical objective of gait optimization for exoskeleton walking. Notably, there exist methods capable of generating human-like walking gaits for bipedal robots [8], but it is unlikely that these methods fulfill the preferences of individuals using robotic assistance.

With the goal of user comfort in mind, the scientific question becomes "how can we systematically realize exoskeleton walking that optimizes user comfort?" This question can be further distilled into short-term and long-term objectives. The short-term objective, which we term *preference optimization*, is to directly optimize exoskeleton walking for each individualized user comfort via human-in-the-loop learning. This provides an immediate method of obtaining optimal exoskeleton walking for an individual user. While this method is ideal for gait personalization, it requires a series of human-in-the-loop exoskeleton trials for each user. This motivates the long-term objective, which we term *preference characterization*. This second objective aims to better understand the relationship between exoskeleton gait features and user comfort by learning the entire preference landscape of an exoskeleton user through human-in-the-loop, while avoiding gaits that make the user feel unsafe or uncomfortable. The main benefit of preference characterization is that it would provide a mechanism to understand user preference landscapes such that in the future, with enough data, a generalized model of user comfort could be obtained and leveraged to predict which gait features optimize user comfort. This generalized model would reduce the need for extensive human-in-the-loop learning.

Major challenges of accomplishing both objectives include: 1) ensuring safety

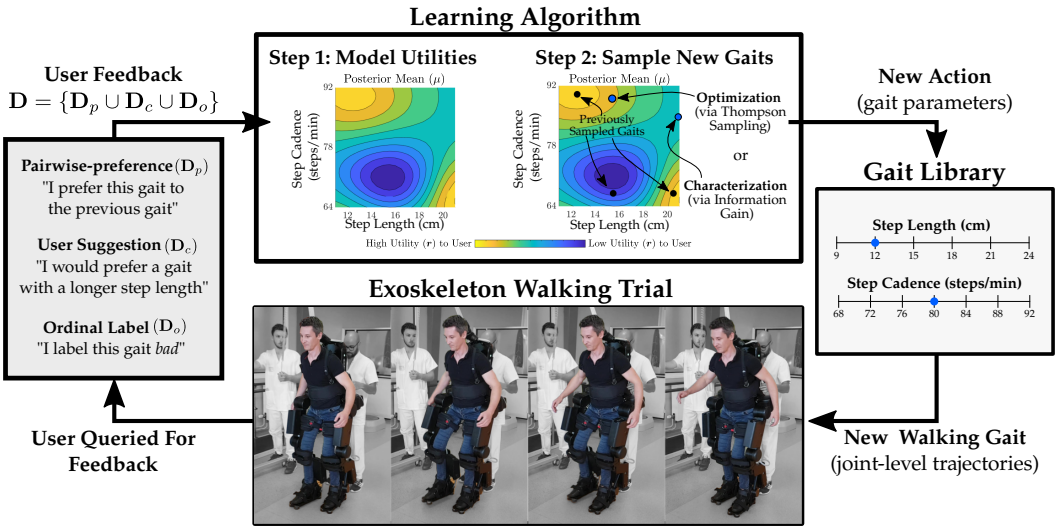


Figure 3.1: Illustration of the preference-based learning framework applied to exoskeleton gait optimization and characterization. The learning framework consists of four main components: 1) collecting subjective user feedback from exoskeleton users; 2) using the collected feedback to model the underlying 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.

and comfort of the human user; 2) the time-intensive nature of human subject trials; 3) obtaining reliable user feedback; and 4) exploring the vast space of possible exoskeleton walking behaviors. Addressing these challenges, this chapter develops a human-in-the-loop preference-based learning framework (illustrated in Figure 3.1) that offers a promising alternative to pure numeric optimization by instead optimizing and characterizing user comfort. The overall benefits of the proposed learning framework include: 1) ensuring the user's comfort and safety via preference-based learning and the partial hybrid zero dynamics method, 2) achieving sample-efficient data collection via Gaussian process modeling, 3) obtaining feedback by asking users for qualitative information, which humans can provide more reliably than numerical scores, and 4) exploring high-dimensional spaces of possible exoskeleton walking behaviors.

As illustrated in Fig. 3.1, human-in-the-loop online learning refers to the iterative process of querying a human for feedback and utilizing this feedback to select new actions. Concretely, the proposed framework balances preference optimization and characterization by the choice of acquisition function. If the objective is preference optimization, then the framework selects actions using

Thompson sampling, a regret minimization technique where *regret* is defined as the cumulative gap between the utility to the user of the optimal gait and of gaits selected by the algorithm. If the objective is preference characterization, the framework selects actions using *information gain*. Importantly, we restrict the information gain to a *region of interest* (ROI), which we define to be the region of actions excluding the gaits that make user feel unsafe or uncomfortable.

This chapter introduces the following three algorithms which comprise the preference-based learning framework: CoSpar for preference optimization [207]; LineCoSpar for preference optimization in high-dimensional action spaces [206]; and ROIAL for preference characterization constrained to a region of interest [128]. Moreover, the unification of these three algorithms is presented to obtain a single holistic framework. In this chapter, we will discuss the unified preference-based learning framework, followed by several experimental demonstrations of the framework towards preference optimization and characterization on the Atalante exoskeleton. In the subsequent chapter the methodology will be extended towards additional applications on other robotic platforms.

3.1 Preference-Based Learning Framework

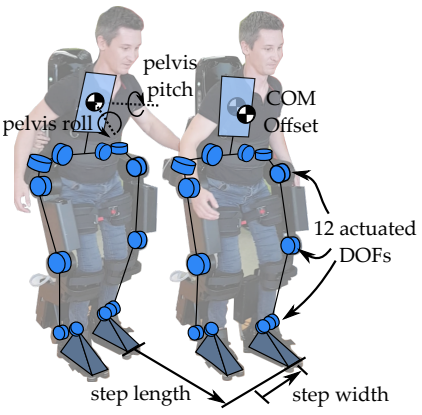
Problem Setup

Consider a robotic system with $v \in \mathbb{N}$ adjustable parameters. The possible range of each parameter is assumed to be upper and lower bounded, with these bounds defined as $a_i^{\max}, a_i^{\min} \in \mathbb{R}$, respectively, for each parameter $i = \{1, \ldots, v\}$. Also, since humans cannot easily distinguish between parameters with very similar values (often characterized as the minimum detectable change or just noticeable difference [23]), we restrict each parameter to belong to a discretized set with step sizes defined by $d_i \in \mathbb{R}_{>0}$. This restriction also helps with computational tractability.

Using the above notation, each individual dimension of the entire space of possible parameter combinations is defined as:

$$\mathcal{A}_i = \{a_i^{\min} + nd_i \mid n \in \mathbb{N}_{\geq 0} \text{ and } a_i^{\min} + nd_i \leq a_i^{\max}\}, \tag{3.1}$$

for $i = \{1, \ldots, v\}$. We refer to the entire space of discrete parameter combinations as the *action space* $\mathcal{A}$ with with cardinality $|\mathcal{A}| = \prod_{i=1}^{v} |\mathcal{A}_i|$. Each unique vector of parameter values is referred to as an *action*, denoted as $\boldsymbol{a} := [a_1, \ldots, a_v] \in \mathbb{R}^v$. Lastly, we assume that there exists some unknown



	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 3.2: 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.

reward function $r : \mathbb{R}^v \to \mathbb{R}$ which maps each action $\boldsymbol{a}$ to a latent reward, i.e., the human's valuation of the action. The restriction of r to $\mathcal{A}$ is denoted as the vectorized utility function $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{A}|}$.

In the context of exoskeleton walking, the action space consists of all possible gaits within a pre-computed gait library that is parameterized by the following gait features: step length (m), step cadence (steps/min), step width (m), maximum step height (m), maximum pelvis roll (deg), and maximum pelvis pitch (deg). The action space definitions for the gait libraries used throughout this chapter are outlined in Figure 3.2. Importantly, an additional gait parameter on the center of mass offset (m) of the combined patient-exoskeleton model is used for experiments with subjects with paraplegia since the weight distributions for those subjects are not well-approximated by the anthropomorphic data used to construct the patient models [225]. For all action spaces, the corresponding gait library is constructed using gaits generated from the *Partial Hybrid Zero Dynamics* method [8] described in Chapter 2. Thus, each gait is represented as a unique composition of exoskeleton joint-level trajectories, comprising the angular positions and velocities of each joint from one heel strike to the next heel strike.

Learning Objectives

The first learning objective, preference optimization, aims to minimize the regret, or the gap between the reward of selected actions and the true optimal action $\boldsymbol{a}^* = \text{argmax}_{\boldsymbol{a}\in\mathcal{A}}\, r(\boldsymbol{a})$. We formalize this optimization objective as follows:

Preference Optimization Objective:
Sequentially sample N actions, $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}$, to minimize cumulative regret:

$$\text{Regret}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N) = \sum_{t=1}^{N} \left(r(\boldsymbol{a}^*) - r(\boldsymbol{a}_t)\right).$$

Algorithms with finite-bounded regret result in $r(\boldsymbol{a}_N) \to r(\boldsymbol{a}^*)$ as $N \to \infty$.

To minimize regret in sequential online learning, in which an algorithm only receives feedback on iteratively-sampled actions, the algorithm must balance between exploration and exploitation [187]. Algorithms *exploit* by selecting actions that are expected to have high utility, while they *explore* the action space to avoid being trapped in local optima.

The second learning objective, preference characterization, aims to minimize error between the true underlying reward function, r, and the reward function learned after N iterations, $\hat{r}_N$. Importantly, unlike with regret minimization, characterization alone does not encourage avoiding low-utility actions. This is problematic for exoskeleton experiments because low-utility actions could make the user feel unsafe or uncomfortable. Thus, we modify the second learning objective to prioritize characterization within a *Region of Interest* (ROI). In the exoskeleton setting, we define the ROI as consisting of exoskeleton walking gaits that do not correspond to a user-provided label of "very bad," which the user is instructed to give when experiencing a gait that makes them feel unsafe or uncomfortable. The region of the action space $\mathcal{A}$ corresponding to the label "very bad" is termed the *Region of Avoidance* (ROA), and is the complement of the ROI. We formalize this ROI-based preference characterization problem as follows:

Preference Characterization Objective:
Sequentially sample N actions, $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}$, to minimize the reward function estimation error defined as:

$$\text{Error}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N) := \boldsymbol{\kappa}^\top |\boldsymbol{r} - \hat{\boldsymbol{r}}_N|,$$

where $\boldsymbol{\kappa} \in \{0,1\}^{|\mathcal{A}|}$ is a binary vector denoting which actions lie within the ROI, and $\boldsymbol{r}, \hat{\boldsymbol{r}} \in \mathbb{R}^{|\mathcal{A}|}$ are the true and learned utilites for all actions.

Modeling User Feedback

For exoskeleton gait generation, as in many real-world settings involving humans [18, 20, 26], it is challenging for people to reliably specify numerical scores or provide demonstrations. As such, we take advantage of three subjective feedback mechanisms: pairwise preferences, coactive suggestions, and ordinal labels. Here, we formally define and compare how each of these mechanisms is modeled. The process of querying the user for feedback is detailed in Alg. 1.

Preference Feedback. Preferences are defined as pairwise comparisons (i.e., "Does the user prefer action a or action b?"). Previous studies have found preferences to be more reliable than numerical scores in a range of domains, including information retrieval [48] and autonomous driving [26]. A preference between two actions $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, is denoted as $p = \boldsymbol{a}_1 \succ \boldsymbol{a}_2$ if action $\boldsymbol{a}_1$ is preferred, or $p = \boldsymbol{a}_2 \succ \boldsymbol{a}_1$ if action $\boldsymbol{a}_2$ is preferred. The probability of a user giving a pairwise preference $\boldsymbol{a}_1 \succ \boldsymbol{a}_2$ given an underlying reward function r is modeled using the likelihood function [51]:

$$\mathbb{P}(\boldsymbol{a}_1 \succ \boldsymbol{a}_2 \mid r(\boldsymbol{a}_1), r(\boldsymbol{a}_2)) = \begin{cases} 1 & \text{if } r(\boldsymbol{a}_1) \geq r(\boldsymbol{a}_2), \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

which captures the preference relations given ideal noise-free preference feedback. However, user preferences are expected to be corrupted by noise. Thus, we model the preferences as being contaminated by noise through the likelihood function:

$$\mathbb{P}(\boldsymbol{a}_1 \succ \boldsymbol{a}_2 \mid r(\boldsymbol{a}_1), r(\boldsymbol{a}_2)) = \psi\left(\frac{r(\boldsymbol{a}_1) - r(\boldsymbol{a}_2)}{c_p}\right), \tag{3.3}$$

where $\psi(\cdot) : \mathbb{R} \to (0,1)$ can be any monotonously-increasing activation function, and $c_p > 0$ quantifies noisiness in the preferences. The effect of c_p had

Algorithm 1 Obtain User Feedback (During Iteration i)

$n_s :=$ num. of samples per iter.
$n_b :=$ num. of past samples in user's memory buffer
$\binom{n_s+n_b}{2} :=$ number of pairwise comparisons per iteration
for $j = 1, \ldots, \binom{n_s+n_b}{2}$ **do**
    if User prefers first action ($\boldsymbol{a}_{j1}$) in comparison j **then**
        Record pairwise preference as $p_j = (\boldsymbol{a}_{j1} \succ \boldsymbol{a}_{j2})$
    else if User prefers second action ($\boldsymbol{a}_{j2}$) in comparison j **then**
        Record pairwise preference as $p_j = (\boldsymbol{a}_{j2} \succ \boldsymbol{a}_{j1})$
    else if No preference **then**
        $p_j = \emptyset$
    end if
    Append feedback to dataset: $\mathbf{D}_p = \mathbf{D}_p \cup p_j$
end for
for $j = 1, \ldots, n_s$ **do**
    if User has a suggestion regarding sampled action $\boldsymbol{a}_i^j$ **then**
        Record coactive action as $\bar{\boldsymbol{a}}_i^j$
        Append feedback to dataset: $\mathbf{D}_c = \mathbf{D}_c \cup (\bar{\boldsymbol{a}}_i^j \succ \boldsymbol{a}_i^j)$
    end if
end for
for $j = 1, \ldots, n_s$ **do**
    if User has an ordinal label regarding sampled action $\boldsymbol{a}_i^j$ **then**
        Record ordinal label as o_i^j
        Append feedback to dataset: $\mathbf{D}_c = \mathbf{D}_c \cup (\boldsymbol{a}_i^j, o_i^j)$
    end if
end for

on the preference likelihood function is illustrated in Figure 3.3a. Two example activation functions include the standard normal cumulative distribution function and the sigmoid function:

$$\psi_{\text{Gaus}}(x) := \int_{-\text{inf}}^{x} \text{Normal}(\gamma; 0, 1) d\gamma, \tag{3.4}$$

$$\psi_{\text{sig}}(x) := \frac{1}{1 + \exp(-x)}. \tag{3.5}$$

We empirically found that using $\psi_{\text{sig}}(x)$ resulted in improved performance because of its heavier-tailed distribution. The preference likelihood function with ψ_{sig} is illustrated in Figure 3.3a for preference noise parameters c_p. Note that the choice of c_p depends both on the expected preference noise as well as the range of the underlying utility function r.

A collection of pairwise preferences is denoted as:

$$\mathbf{D}_p := \{\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid k = 1, \ldots, K\}, \tag{3.6}$$

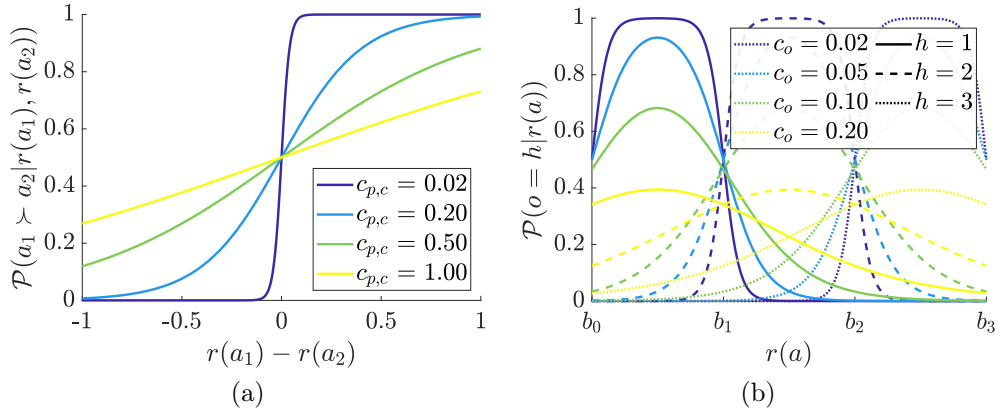


Figure 3.3: Illustration of the effect of the noise parameter on the likelihood function for a) preference and coactive feedback (c_p and c_c), and b) ordinal feedback (c_o).

where a_{k1} and a_{k2} denote the two actions corresponding to the k^{th} pairwise preference in a set of $K \in \mathbb{N}_{\geq 0}$ total comparisons. The likelihood function for the entire set $\mathbf{D}_p$ is then calculated as the product of each individual likelihood:

$$\mathbb{P}(\mathbf{D}_p \mid r) = \prod_{k=1}^{K} \mathbb{P}(\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid r(\boldsymbol{a}_{k1}), r(\boldsymbol{a}_{k2})). \tag{3.7}$$

This function is known as the preference likelihood function and is used later to approximate the posterior distribution.

Coactive Feedback. User suggestions, also known as coactive feedback, can be incorporated into the learning framework by treating user suggested improvements as implicit preferences. In this context, coactive feedback can be thought of as preference feedback between an action $\bar{\boldsymbol{a}}$ suggested by the user and the sampled action $\boldsymbol{a}$. In other words, the underlying assumption of coactive feedback is that $r(\bar{\boldsymbol{a}}) > r(\boldsymbol{a})$. This approach resembles the *coactive learning* framework [171], first proposed in [170], in which the user identifies an improved action as feedback to each presented action. The combination of preference and coactive feedback is termed *mixed-initiative* learning [127, 228]. Coactive learning has been applied to robot trajectory planning [106, 178], but was only first, to our knowledge, applied with preference learning in [207].

We denote a single user suggestion as $c = \bar{\boldsymbol{a}} \succ \boldsymbol{a}$. As with preferences, coactive feedback is modeled with the assumption that it is corrupted by noise:

$$\mathbb{P}(\bar{\boldsymbol{a}} \succ \boldsymbol{a} \mid r(\bar{\boldsymbol{a}}), r(\boldsymbol{a})) = \psi\left(\frac{r(\bar{\boldsymbol{a}}) - r(\boldsymbol{a})}{c_c}\right), \tag{3.8}$$

where $\psi(\cdot) : \mathbb{R} \to (0, 1)$ is the same activation function as with preference feedback, and $c_c > 0$ quantifies noisiness in the coactive suggestions. The hyperparameter c_c has an identical effect on the coactive likelihood function as c_p had on the preference likelihood function, illustrated in Figure 3.3a.

We recommend setting $c_c > c_p$ since in general, coactive feedback is more prone to noise compared to preference feedback. For example, in the case of exoskeleton user feedback, a user may provide a suggestion (i.e., "I would prefer a faster gait.") that would no longer be true once they experienced the walking (i.e., "Actually, I feel uncomfortable with the faster gaits."). However, incorporating suggestions increases the amount of information obtained from the same number of sampled actions, thus increasing the sample-efficiency of the algorithm.

A collection of user suggestions can either be stored with the preference dataset, or in their own dataset. For clarity, we will use the latter and define a set of user suggestions as:

$$\mathbf{D}_c := \{\bar{\boldsymbol{a}}_l \succ \boldsymbol{a}_l \mid l = 1, ..., L\},$$

where $\bar{\boldsymbol{a}}_1$ is the suggested action being compared to the sampled action $\boldsymbol{a}_l$, with this comparison being the l^{th} suggestion in a set of $L \in \mathbb{N}_{\geq 0}$ total suggested actions. As with user preferences, the coactive likelihood function is calculated as the product of each likelihood function for L user suggestions:

$$\mathbb{P}(\mathbf{D}_c \mid r) = \prod_{l=1}^{L} \mathbb{P}(\bar{\boldsymbol{a}}_l \succ \boldsymbol{a}_l \mid r(\bar{\boldsymbol{a}}_l), r(\boldsymbol{a}_l)). \tag{3.9}$$

Ordinal Feedback. Ordinal feedback assigns an ordered label to each sampled action, with the ordinal labels corresponding to one of $h \in \mathbb{N}$ ordinal categories. These labels partition the set of all possible actions into h sets denoted as $\mathbf{O}_i$ for $i = 1, \ldots, h$. We define the boundaries between ordinal categories using thresholds $-\infty = b_0 < b_1 < \cdots < b_h = \infty$. Thus, in the case of ideal noise-less feedback, ordinal labels are determined by the likelihood function:

$$\mathbb{P}\big((\boldsymbol{a}, o) \mid r(a)\big) = \begin{cases} 1 & \text{if } b_{o-1} \leq r(\boldsymbol{a}) < b_o, \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

where we denote $(\boldsymbol{a}, o)$ as the ordinal label $o \in \{1, \ldots, h\} \subset \mathbb{N}$ provided for the sampled action $\boldsymbol{a}$. As with preference and coactive feedback, we modify

this simplified likelihood function to account for noise as in [52]:

$$\mathbb{P}\big((\boldsymbol{a}, o) \mid r(\boldsymbol{a})\big) = \psi\left(\frac{b_o - r(\boldsymbol{a})}{c_o}\right) - \psi\left(\frac{b_{o-1} - r(\boldsymbol{a})}{c_o}\right), \tag{3.11}$$

with $\psi(\cdot) : \mathbb{R} \to (0, 1)$ an activation function as before, and $c_o > 0$ quantifying noisiness in the ordinal labels. The effect of c_o on the likelihood function is illustrated in Figure 3.3b. Since previous work has shown numerical scores (such as ordinal labels) to be less reliable than pairwise preferences for subjective human feedback, we set $c_o > c_c > c_p$ [110, 187].

We define a dataset of M ordinal labels as:

$$\mathbf{D}_o := \{(\boldsymbol{a}_m, o_m) \mid m = 1, ..., M\}, \tag{3.12}$$

with each ordinal label o_m given for the corresponding action $\boldsymbol{a}_m$ in a set of $M \in \mathbb{N}_{\geq 0}$ total user-provided labels. As with the preference and coactive likelihood functions, the ordinal likelihood function is calculated as the product of the M individual likelihoods:

$$\mathbb{P}(\mathbf{D}_o \mid r) = \prod_{m=1}^{M} \mathbb{P}\big((\boldsymbol{a}_m, o_m) \mid r(\boldsymbol{a}_m)\big). \tag{3.13}$$

Approximating the Utility as a Gaussian Process

The second component of the learning framework is to use the obtained user feedback to estimate the vectorized underlying utility function, $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{A}|}$, as $\hat{\boldsymbol{r}} \in \mathbb{R}^{|\mathcal{A}|}$. We compute $\hat{\boldsymbol{r}}$, as the maximum a posteriori (MAP) estimate:

$$\hat{\boldsymbol{r}} = \boldsymbol{r}_{\text{MAP}} := \underset{\boldsymbol{r} \in \mathbb{R}^{|\mathcal{A}|}}{\operatorname{argmax}}\, \mathbb{P}(\boldsymbol{r} \mid \mathbf{D}), \tag{3.14}$$

where $\mathbb{P}(\boldsymbol{r} \mid \mathbf{D})$ is the posterior distribution of $\boldsymbol{r}$ given the collection of user feedback $\mathbf{D} = \mathbf{D}_p \cup \mathbf{D}_c \cup \mathbf{D}_o$. We choose to approximate $\hat{\boldsymbol{r}}$ as a Gaussian process because it enables $\boldsymbol{r}$ to be modeled as a Bayesian posterior over a class of smooth, non-parametric functions. We approximate $\mathbb{P}(\boldsymbol{r} \mid \mathbf{D})$ as the Gaussian distribution Normal(μ, Σ), which is derived from the preference-based Gaussian process model of [51]. This allows us to approximate $\hat{\boldsymbol{r}} = \mu$.

In this section, we will first discuss how we model the Bayesian posterior $\mathbb{P}(\boldsymbol{r} \mid \mathbf{D})$, followed by how we approximate this distribution as Gaussian. The end result is an optimization problem which can be computationally expensive depending on the number of actions in $\mathcal{A}$ over which $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{A}|}$ inferred. Thus,

to remain tractable in high-dimensional action spaces, we only infer the reward function over a subset of actions, denoted $\boldsymbol{r}_{\mathbf{S}} \in \mathbb{R}^{|\mathbf{S}|}$ for some $\mathbf{S} \subset \mathcal{A}$. We will discuss this restriction at the end of this section.

Modeling the Posterior Probability

By assuming conditional independence of the feedback mechanisms, we can use Bayes rule to model the posterior of the utilities $\boldsymbol{r}$ as proportional to the product of the individual likelihood terms and the Gaussian prior:

$$\mathbb{P}(\boldsymbol{r}|\mathbf{D}) \propto \mathbb{P}(\mathbf{D}_p|\boldsymbol{r})\mathbb{P}(\mathbf{D}_c|\boldsymbol{r})\mathbb{P}(\mathbf{D}_o|\boldsymbol{r})\mathbb{P}(\boldsymbol{r}), \tag{3.15}$$

with each likelihood calculated as described in the Modeling User Feedback section. We define a Gaussian prior over $\boldsymbol{r}$ as:

$$\mathbb{P}(\boldsymbol{r}) = \frac{1}{(2\pi)^{\frac{|\mathcal{A}|}{2}} \left|\Sigma^{\text{pr}}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{r}^\top (\Sigma^{\text{pr}})^{-1}\boldsymbol{r}\right), \tag{3.16}$$

where $\Sigma^{\text{pr}} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ is the prior covariance matrix with $[\Sigma^{\text{pr}}]_{ij} = \mathcal{K}(\boldsymbol{a}_i, \boldsymbol{a}_j)$ and $\mathcal{K}$ being a kernel of choice. In our work, we select $\mathcal{K}$ to be the squared exponential kernel:

$$\mathcal{K}_{\text{SE}}(\boldsymbol{a}, \boldsymbol{a}') = \sigma^2 \exp\left(-\frac{(\boldsymbol{a} - \boldsymbol{a}')^2}{2l^2}\right), \tag{3.17}$$

where $\sigma \in \mathbb{R}$ is the output variance hyperparameter and $l \in \mathbb{R}^v$ is a vector of lengthscales for each dimension of $\mathcal{A}$. The output variance σ dictates the expected average distance the underlying function is away from its mean. The lengthscales l_i for $i = 1, \ldots, v$ are hyperparameters that dictate the expected "wiggliness" of the underlying function in each dimension.

Approximating the Posterior Distribution as Gaussian

Since we leverage more types of feedback than just pairwise preferences, we extend the method introduced in [51] towards approximating $\mathbb{P}(\boldsymbol{r} \mid \mathbf{D})$. This extension redefines the Gaussian distribution Normal(μ, Σ) as:

$$\mu = \boldsymbol{r}_{\text{MAP}} := \underset{\boldsymbol{r}}{\text{argmax}}\, \mathbb{P}(\boldsymbol{r} \mid \mathbf{D}), \tag{3.18}$$

$$\Sigma = ((\Sigma^{\text{pr}})^{-1} + \Lambda_{\text{MAP}})^{-1}, \tag{3.19}$$

where Λ_{MAP} is the matrix $\Lambda \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ evaluated at $\boldsymbol{r}_{\text{MAP}}$, with the composition of Λ extended to:

$$\begin{aligned}[\Lambda]_{ij} := &\sum_{k=1}^{K} \frac{-\partial^2 \ln \mathbb{P}\left(\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid \boldsymbol{r}(\boldsymbol{a}_{k1}), \boldsymbol{r}(\boldsymbol{a}_{k2})\right)}{\partial \boldsymbol{r}(\boldsymbol{a}_i)\boldsymbol{r}(\boldsymbol{a}_j)} \\ &+ \sum_{l=1}^{L} \frac{-\partial^2 \ln \mathbb{P}(\bar{\boldsymbol{a}}_l \succ \boldsymbol{a}_l \mid \boldsymbol{r}(\bar{\boldsymbol{a}}_l), \boldsymbol{r}(\boldsymbol{a}_l))}{\partial \boldsymbol{r}(\boldsymbol{a}_i)\boldsymbol{r}(\boldsymbol{a}_j)} \\ &+ \sum_{m=1}^{M} \frac{-\partial^2 \ln \mathbb{P}((\boldsymbol{a}_m, o_m) \mid \boldsymbol{r}(\boldsymbol{a}_m))}{\partial \boldsymbol{r}(\boldsymbol{a}_i)\boldsymbol{r}(\boldsymbol{a}_j)}. \end{aligned} \tag{3.20}$$

Here, $[\Lambda]_{ij}$ denotes the ij^{th} element of Λ for $i = 1, \ldots, |\mathcal{A}|$ and $j = 1, \ldots, |\mathcal{A}|$.

Solving for the MAP estimate using convex programming. Approximating $\mathbb{P}(\boldsymbol{r} \mid \mathbf{D})$ as a Gaussian distribution centered on $\boldsymbol{r}_{\text{MAP}}$ with the covariance matrix Σ is equivalent to the Laplace approximation of the functional:

$$\begin{aligned}\mathcal{F}(\boldsymbol{r}) = &- \sum_{k=1}^{K} \ln \mathbb{P}(\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid \boldsymbol{r}(\boldsymbol{a}_{k1}), \boldsymbol{r}(\boldsymbol{a}_{k2})) \\ &- \sum_{l=1}^{L} \ln \mathbb{P}(\bar{\boldsymbol{a}}_l \succ \boldsymbol{a}_l \mid \boldsymbol{r}(\bar{\boldsymbol{a}}_l), \boldsymbol{r}(\boldsymbol{a}_l)) \\ &- \sum_{m=1}^{M} \ln \mathbb{P}((\boldsymbol{a}_m, o_m) \mid \boldsymbol{r}(a_m)) + \frac{1}{2}\boldsymbol{r}^\top (\Sigma^{\text{pr}})^{-1}\boldsymbol{r}. \end{aligned} \tag{3.21}$$

The MAP estimate $\boldsymbol{r}_{\text{MAP}}$ is computed as the minimizer of $\mathcal{F}(\boldsymbol{r})$ using a convex program as outlined next.

Convex Program for Computing the MAP estimate. The convex program to solve for $\boldsymbol{r}_{\text{MAP}}$ is constructed as:

$$\boldsymbol{r}_{\text{MAP}} = \underset{\boldsymbol{r}\in\mathbb{R}^{|\mathcal{A}|}}{\text{argmin}} \ \mathcal{F}(\boldsymbol{r}), \tag{3.22}$$

with the first derivative terms of $\mathcal{F}(\boldsymbol{r})$ being:

$$\frac{\partial - \ln \mathbb{P}(\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid \boldsymbol{r}(\boldsymbol{a}_{k1}), \boldsymbol{r}(\boldsymbol{a}_{k2}))}{\partial \boldsymbol{r}(\boldsymbol{a}_i)} = \frac{-s_k(\boldsymbol{a}_i)}{c_p}\frac{\dot{\psi}(z_k)}{\psi(z_k)}, \tag{3.23}$$

$$\frac{\partial - \ln \mathbb{P}(\bar{\boldsymbol{a}}_l \succ \boldsymbol{a}_l \mid \boldsymbol{r}(\bar{\boldsymbol{a}}_l), \boldsymbol{r}(\boldsymbol{a}_l))}{\partial \boldsymbol{r}(\boldsymbol{a}_i)} = \frac{-s_l(\boldsymbol{a}_i)}{c_c}\frac{\dot{\psi}(z_l)}{\psi(z_l)}, \tag{3.24}$$

$$\frac{\partial - \ln \mathbb{P}((\boldsymbol{a}_m, o_m) \mid \boldsymbol{r}(\boldsymbol{a}_m))}{\partial \boldsymbol{r}(\boldsymbol{a}_i)} = \frac{1}{c_o}\frac{\dot{\psi}(z_{m1}) - \dot{\psi}(z_{m2})}{\psi(z_{m1}) - \psi(z_{m2})}, \tag{3.25}$$

and the second derivative terms:

$$\frac{\partial^2 - \ln \mathbb{P}(\boldsymbol{a}_{k1} \succ \boldsymbol{a}_{k2} \mid \boldsymbol{r}(\boldsymbol{a}_{k1}), \boldsymbol{r}(\boldsymbol{a}_{k2}))}{\partial \boldsymbol{r}(\boldsymbol{a}_i) \boldsymbol{r}(\boldsymbol{a}_j)} = \frac{s_k(\boldsymbol{a}_i) s_k(\boldsymbol{a}_j)}{c_p^2} \left(\frac{\dot{\psi}(z_k)^2}{\psi(z_k)^2} - \frac{\ddot{\psi}(z_k)}{\psi(z_k)} \right), \tag{3.26}$$

$$\frac{\partial^2 - \ln \mathbb{P}(\bar{\boldsymbol{a}}_l \succ \boldsymbol{a}_l \mid \boldsymbol{r}(\bar{\boldsymbol{a}}_l), \boldsymbol{r}(\boldsymbol{a}_l))}{\partial \boldsymbol{r}(\boldsymbol{a}_i) \boldsymbol{r}(\boldsymbol{a}_j)} = \frac{s_l(\boldsymbol{a}_i) s_l(\boldsymbol{a}_j)}{c_c^2} \left(\frac{\dot{\psi}(z_l)^2}{\psi(z_l)^2} - \frac{\ddot{\psi}(z_l)}{\psi(z_l)} \right), \tag{3.27}$$

$$\frac{\partial^2 - \ln \mathbb{P}((\boldsymbol{a}_m, o_m) \mid \boldsymbol{r}(\boldsymbol{a}_m))}{\partial \boldsymbol{r}(\boldsymbol{a}_i) \boldsymbol{r}(\boldsymbol{a}_j)} = \dots \tag{3.28}$$

$$\frac{1}{c_o^2} \left(\left(\frac{\dot{\psi}(z_{m1}) - \dot{\psi}(z_{m2})}{\psi(z_{m1}) - \psi(z_{m2})} \right)^2 - \left(\frac{\ddot{\psi}(z_{m1}) - \ddot{\psi}(z_{m2})}{\psi(z_{m1}) - \psi(z_{m2})} \right) \right), \tag{3.29}$$

where $\psi : \mathbb{R} \to (0, 1)$ represents any activation function with first derivative $\dot{\psi}(\cdot)$ and second derivative $\ddot{\psi}(\cdot)$. The activation function terms are defined as:

$$z_k = \left(\frac{\boldsymbol{r}(\boldsymbol{a}_{k1}) - \boldsymbol{r}(\boldsymbol{a}_{k2})}{c_p} \right), \tag{3.30}$$

$$z_l = \left(\frac{\boldsymbol{r}(\bar{\boldsymbol{a}}_l) - \boldsymbol{r}(\boldsymbol{a}_l)}{c_c} \right), \tag{3.31}$$

$$z_{m1} = \frac{b_{o_m} - \boldsymbol{r}(\boldsymbol{a}_m)}{c_o}, \tag{3.32}$$

$$z_{m2} = \frac{b_{o_m - 1} - \boldsymbol{r}(\boldsymbol{a}_m)}{c_o}. \tag{3.33}$$

Lastly, the indicator functions are defined as:

$$s_k(\boldsymbol{a}) = \begin{cases} +1 & \boldsymbol{a} = \boldsymbol{a}_{k1} \\ -1 & \boldsymbol{a} = \boldsymbol{a}_{k2} \\ 0 & \text{otherwise} \end{cases}, \tag{3.34}$$

$$s_l(\boldsymbol{a}) = \begin{cases} +1 & \boldsymbol{a} = \bar{\boldsymbol{a}}_l \\ -1 & \boldsymbol{a} = \boldsymbol{a}_l \\ 0 & \text{otherwise} \end{cases}. \tag{3.35}$$

Restricting the posterior distribution to a subset S

Solving for $\hat{\boldsymbol{r}}$ is computationally-expensive and can even be intractable for high-dimensional action spaces. Existing preference-based approaches optimize over the action space $\mathcal{A}$ by discretizing the entire space before beginning

the learning process. This results in m^v combinations from $m \in \mathbb{N}$ uniformly-spaced points (corresponding to actions) in each of the v dimensions of $\mathcal{A}$. The cardinality of this set is $|\mathcal{A}| = m^v$; larger m enables finer-grained search at a higher computational cost. The Bayesian preference model is updated over all $|\mathcal{A}|$ points during each iteration. This is intractable for settings with lots of parameters (large v) since computing the posterior over $|\mathcal{A}|$ points involves expensive matrix operations, such as inverting $\Sigma_t^{\text{pr}}, \Sigma_t \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$.

Moreover, we cannot leverage existing work with high-dimensional Gaussian process learning since it requires quantitative feedback [113, 218]. Thus, to maintain computational tractability in high-dimensional action spaces, we simply restrict the posterior to only a subset of the discrete action space $\mathcal{A}$, denoted as the set $\mathbf{S} \subset \mathcal{A}$. The restricted utility function is denoted $\boldsymbol{r}_{\mathbf{S}} \in \mathbb{R}^{|\mathbf{S}|}$. Note that the specific composition of the subset $\mathbf{S}$ depends on the sampling strategy of the framework, and will thus be discussed in the next section.

Computing $\hat{\boldsymbol{r}}_{\mathbf{S}}$ has some slight modifications which we will present here. First, the likelihood functions are calculated only considering $\boldsymbol{a} \in \mathbf{S}$. Second, the Gaussian prior is redefined over $\boldsymbol{r}_{\mathbf{S}}$:

$$\mathbb{P}(\boldsymbol{r}_{\mathbf{S}}) = \frac{1}{(2\pi)^{\frac{|\mathbf{S}|}{2}}|\Sigma_{\mathbf{S}}^{\text{pr}}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{r}_{\mathbf{S}}^{\top}(\Sigma_{\mathbf{S}}^{\text{pr}})^{-1}\boldsymbol{r}_{\mathbf{S}}\right), \tag{3.36}$$

where $\Sigma_{\mathbf{S}}^{\text{pr}} \in \mathbb{R}^{|\mathbf{S}|\times|\mathbf{S}|}$ is the prior covariance matrix with $[\Sigma_{\mathbf{S}}^{\text{pr}}]_{ij} = \mathcal{K}(\boldsymbol{a}_{\mathbf{S}}^i, \boldsymbol{a}_{\mathbf{S}}^j)$ for the restricted set of actions $\boldsymbol{a}_{\mathbf{S}}$ in $\mathbf{S}$. Using these restricted likelihood terms and Gaussian prior, the posterior over $\mathbf{S}$ is modeled as:

$$\mathbb{P}(\boldsymbol{r}_{\mathbf{S}}|\mathbf{D}) \propto \mathbb{P}(\mathbf{D}_p|\boldsymbol{r}_{\mathbf{S}})\mathbb{P}(\mathbf{D}_c|\boldsymbol{r}_{\mathbf{S}})\mathbb{P}(\mathbf{D}_o|\boldsymbol{r}_{\mathbf{S}})\mathbb{P}(\boldsymbol{r}_{\mathbf{S}}), \tag{3.37}$$

and again approximated using the same procedure as presented in the Approximating the Utility as a Gaussian Process section to obtain Normal$(\mu_{\mathbf{S}}, \Sigma_{\mathbf{S}})$, where $\mu_{\mathbf{S}}$ and $\Sigma_{\mathbf{S}}$ denote the posterior mean and covariance over the subset $\mathbf{S}$. Lastly, $\hat{\boldsymbol{r}}_{\mathbf{S}}$ is computed as the solution to the minimization:

$$\hat{\boldsymbol{r}}_{\mathbf{S}} = \underset{\boldsymbol{r}\in\mathbb{R}^{|\mathbf{S}|}}{\text{argmin}} \ \mathcal{F}(\boldsymbol{r}). \tag{3.38}$$

Selecting New Actions

Since human-in-the-loop frameworks only collect user feedback for sampled actions, the process of selecting these actions is critical to the learning performance. In this section we will outline each of the two sampling techniques

used in the proposed preference-based learning framework: Thompson sampling and information gain across a region of interest.

Sampling method for regret minimization

There are many existing sampling methods aimed at regret minimization, including Thompson sampling [201] and upper confidence bound (UCB) algorithms [124]. Existing work has explored extending UCB algorithms for settings with relative feedback (such as pairwise preferences), aptly named Relative Upper Confidence Bound (RUCB) [241]. However, in our work we utilize Thompson Sampling since it slightly favors exploitation compared to RUCB in the presence of relative feedback [207].

The general concept of Thompson sampling is to determine $n_s \in \mathbb{N}$ actions to query by drawing n_s samples from a given distribution. Specifically, the probability of selecting any given action corresponds with its probability of maximizing the underlying utility function. Thus, as the uncertainty of the distribution shrinks, the action maximizing the mean of the distribution $\mathbb{P}(\boldsymbol{r} \mid \mathbf{D})$ has a higher probability of being selected.

Thompson sampling is conducted using the following procedure. In each iteration $i \in \mathbb{N}$, n_s samples are drawn from the distribution:

$$\boldsymbol{r}_i^k \sim \text{Normal}(\mu, \Sigma) \quad \forall k = 1, \ldots, n_s. \tag{3.39}$$

Then, the sampled actions $\{\boldsymbol{a}_i^1, \ldots, \boldsymbol{a}_i^{n_s}\}$ are selected to be the actions maximizing the drawn samples:

$$\boldsymbol{a}_i^k = \underset{\boldsymbol{a} \in \mathcal{A}}{\text{argmax}}\, \boldsymbol{r}_i^k(\boldsymbol{a}) \quad \forall k = 1, \ldots, n_s. \tag{3.40}$$

Note here that Thompson sampling can be performed identically when using the dimensionality reduction technique by limiting the selection of actions to $\boldsymbol{a} \in \mathbf{S}$ and considering $\mu_{\mathbf{S}}$ and $\Sigma_{\mathbf{S}}$.

Dimensionality reduction for regret minimization. While the restriction of the posterior to a subset $\mathbf{S} \subset \mathcal{A}$ addresses the issue of computational tractability, it is critical to construct $\mathbf{S}$ in a way that preserves regret minimization. Inspired from [118], we construct $\mathbf{S} := \mathbf{L} \cup \mathbf{V}$ as the union of a one-dimensional subspace $\mathbf{L} \subset \mathcal{A}$ with all previously sampled actions $\mathbf{V}$. Importantly, $\mathbf{L}$ is generated such that it intersects the most recent estimate of the optimal action ($\boldsymbol{a}^* =$

$\text{argmax}_{\boldsymbol{a}\in\mathbf{S}}\,\mu_{\mathbf{S}}$). While this dimensionality reduction technique limits the set of possible sampled actions to those within $\mathbf{S}$, [118] found that it yields state-of-the-art performance with significantly reduced computational complexity. Concretely, this approach reduces the model's covariance matrices $\Sigma_t^{\text{pr}}, \Sigma_t$ from size $|\mathcal{A}| \times |\mathcal{A}|$ to $|\mathbf{L} \cup \mathbf{V}| \times |\mathbf{L} \cup \mathbf{V}|$. Rather than growing exponentially in v, which is impractical for online learning, LineCoSpar's complexity is constant in the dimension v and linear in the number of iterations. Since queries are expensive in many human-in-the-loop robotics settings, the number of total iterations is typically low.

Sampling method for preference characterization

While there are many existing sampling approaches aimed at characterizing the latent reward function, we utilize a custom sampling approach [128] in order to: 1) prioritize sample efficiency; 2) utilize subjective feedback; 3) avoid low-utility actions; and 4) select actions that lead to easy to answer questions and reliable feedback. This custom acquisition function resembles information gain but limited to actions within a "Region of Interest" (ROI). This set of actions is denoted as $\mathbf{S}_{\text{ROI}} \in \mathbf{S} \subseteq \mathcal{A}$ and will be discussed more later.

Information gain selects actions that maximize the mutual information between the underlying utility function and the users feedback, thus learning the underlying utility function as efficiently as possible. It has been shown that selecting an entire sequence of actions to optimize this mutual information is NP-hard [4]. However, previous work has shown that state-of-the-art performance can be achieved via a greedy approach which only optimizes one action to compare with already sampled past actions. Thus, in our work we also only select one action at a time using information gain which is compared with past actions. It is possible to extend this method to more than one action sampled in each iteration but the problem quickly becomes intractable.

Information gain is conducted using the following procedure. In iteration $i \in \mathbb{N}$, n_s new actions are selected to be the solutions to the maximization problem:

$$\boldsymbol{a}_i^k = \underset{\boldsymbol{a}\in\mathbf{S}_{\text{ROI}}}{\text{argmax}}\, I(\boldsymbol{r}; o_i^k, \boldsymbol{p}_i^k | \mathbf{D}_p, \mathbf{D}_o, \boldsymbol{a}), \quad \forall k = 1, \ldots, n_s, \tag{3.41}$$

where o_i^k is the ordinal label associated with the sampled action $\boldsymbol{a}_i^k$, and $\boldsymbol{p}_i^k = \{p_i^1, \ldots, p_i^{n_s+n_b-1}\}$ is the set of pairwise preferences between $\boldsymbol{a}_i^k$ and each action

within the set $\{\{\{\boldsymbol{a}_i^1, \ldots, \boldsymbol{a}_i^{n_s}\} \setminus \boldsymbol{a}_i^k\} \cup \boldsymbol{a}_{\text{buffer}}\}$, where $\boldsymbol{a}_{\text{buffer}}$ represents the set of $n_b \in \mathbb{N}_{\geq 0}$ previously sampled actions stored in a "buffer." This set of buffer actions can be interpreted as the previously sampled actions that the user can reliably remember.

This maximization problem can be equivalently written in terms of information entropy:

$$\begin{aligned}\boldsymbol{a}_i^k = &\underset{\boldsymbol{a} \in \mathbf{S}_{\text{ROI}}}{\text{argmax}} \; H(o_i^k, \boldsymbol{p}_i^k | \mathbf{D}_p, \mathbf{D}_o, \boldsymbol{a}) \\ &- \mathbb{E}\left[\boldsymbol{r} | \mathbf{D}_p \mathbf{D}_o\right] \left[H(o_i^k, \boldsymbol{p}_i^k | \mathbf{D}_p, \mathbf{D}_o, \boldsymbol{a}, \boldsymbol{r})\right], \quad \forall k = 1, \ldots, n_s. \end{aligned} \tag{3.42}$$

In this expression, the first term can be interpreted as the model's uncertainty about a given action's ordinal label and preferences. The sampled action is aimed at maximizing this term because an action with high uncertainty regarding its feedback yields potentially valuable information regarding the underlying utility function. The second term can be interpreted as the user's expected uncertainty regarding the feedback. Therefore, the sampled action is aimed at yielding pairwise comparisons that are easy for the user to provide. By combining both terms, the sampled actions result in queries that are both informative and easy for users.

Limiting exploration to a region of interest. The region of interest (ROI) is defined as the set of actions $\mathbf{S}_{\text{ROI}} \in \mathbf{S} \subseteq \mathcal{A}$, restricted to the set $\mathbf{S}$ over which the posterior Normal$(\mu_{\mathbf{S}}, \Sigma_{\mathbf{S}})$ is constructed, that satisfy the criteria:

$$\mu_{\mathbf{S}}(\boldsymbol{a}) + \lambda \Sigma_{\mathbf{S}}(\boldsymbol{a}) > b_{\text{ROI}}. \tag{3.43}$$

The constant $b_{\text{ROI}} \in \mathbb{R}$ is the user-selected threshold that separates the ordinal categories belonging to the ROA and the ordinal categories belonging to the complement of the ROA. We term this complement the region of interest (ROI). The user also defines $\lambda \in \mathbb{R}$, a hyperparameter that determines the algorithm's conservatism in estimating the ROI. Smaller values of λ lead to more conservative estimates of the ROA.

3.2 Experimental Results: Low-Dimensional Preference Optimization

The first set of experiments deployed preference optimization across low-dimensional action spaces (gait libraries consisting of one or two gait parameters) and

Algorithm 2 CoSpar

procedure CoSpar($\mathcal{A}$ = action set, n_s = number of actions to select at each iteration, n_b = buffer size, (Σ, σ) = utility prior parameters)
  $\mathbf{D} = \emptyset$ ▷ Initialize feedback dataset
  Obtain prior $(\boldsymbol{\mu}_0, \Sigma_0)$ over $\mathcal{A}$ from (Σ, σ)
  for all $t = 1, 2, \ldots$ **do**
    for all $j = 1, \ldots, n_s$ **do**
      Sample utility function f_j from Normal$(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1})$
      Select action $\boldsymbol{a}_j(t) = \text{argmax}_{x \in \mathcal{A}} f_j(x)$
    end for
    Execute n_s actions
    Collect feedback using Algorithm 1
    Update Bayesian posterior over $\mathbf{D}$ to obtain Normal$(\boldsymbol{\mu}_t, \Sigma_t)$
  end for
end procedure

leveraged pairwise preference feedback and coactive feedback. This specific combination of user feedback and sampling was termed the CoSpar (Coactive Self-Sparring) algorithm [207]. CoSpar is a mixed-initiative approach, which both queries the user for preferences and allows the user to suggest improvements. For clarity, the algorithmic procedure is outlined in Alg. 2.

Simulation Results

The performance of CoSpar is first evaluated in two sets of simulations: (1) the compass-gait (CG) biped's COT,[1] and (2) a set of synthetic optimization objective functions.[2] In both cases, CoSpar efficiently converges to the optimum.

First, the results of the compass-gait simulations are illustrated in Figure 3.4. These simulations aim to identify the optimal step length with the underlying utility being the mechanical cost of transport, using only synthetic preference feedback. These synthetic preferences are determined by comparing MCOT values, calculated by simulating gaits for multiple step lengths, each at a fixed forward hip velocity of 0.2 m/s. These simulated gaits were synthesized via a single-point shooting partial hybrid zero dynamics method [221].

CoSpar is deployed for the CG biped with $n_s = 2$, $n_b = 0$, and without coactive

[1] Bayesian model's kernel: squared exponential with lengthscale = 0.025, signal variance = 0.0001, noise variance = 1e-8; preference noise (σ) = 0.01.

[2] Kernel: squared exponential with lengthscale = $[0.15, 0.15]$, signal variance = 0.0001, noise variance = 1e-5; preference noise (σ) = 0.01.

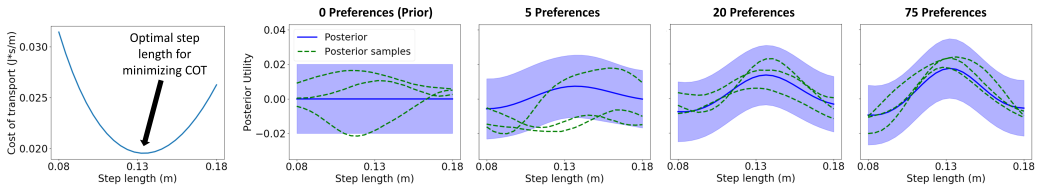
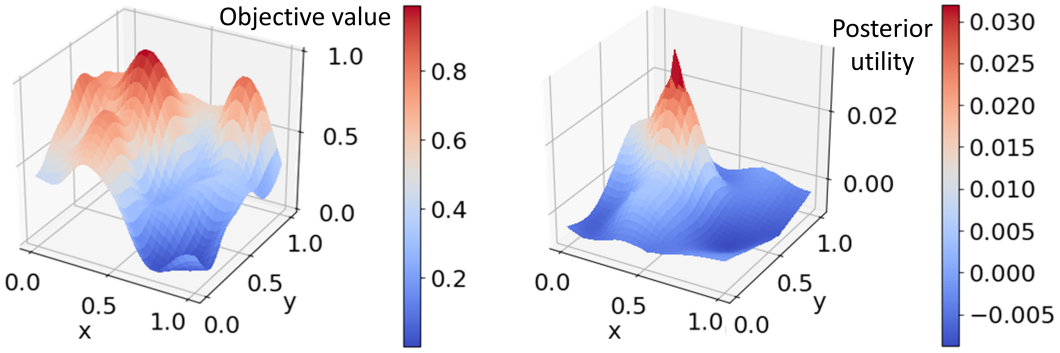


Figure 3.4: Illustration of preference-based learning applied towards identifying the optimal step length that maximizes MCOT for the CG biped in simulation. Leftmost: MCOT for the CG biped at different step lengths and a fixed 0.2 m/s velocity. Remaining plots: posterior utility estimates of CoSpar ($n_s = 2$, $n_b = 0$; without coactive feedback) after varying iterations of learning (posterior mean +/- 2 standard deviations). The plots each show 3 posterior samples, which lie in the high-confidence region (mean +/- 2 stds) with high probability.



(a) Objective function

(b) Model posterior

Figure 3.5: Demonstration of preference-based learning in simulation. a) Example synthetic 2D objective function. b) Utility model posterior learned after 150 iterations of CoSpar in simulation ($n_s = 1$; $n_b = 1$; coactive feedback). CoSpar prioritizes identifying and exploring the optimal region, rather than learning a globally-accurate utility landscape.

feedback. Note that without a buffer or coactive feedback, CoSpar reduces to Self-Sparring [185]. At each iteration, two new samples are drawn from the Bayesian posterior, and the resultant two step lengths are compared to elicit a preference. Using the new preferences, CoSpar updates its posterior over the utility of each step length. Figure 3.4 depicts the evolution of the posterior preference model, where each iteration corresponds to a preference between two new trials. With more preference data, the posterior utility increasingly peaks at the point of lowest COT. These results suggest that CoSpar can efficiently identify high-utility actions from preference feedback alone.

Second, CoSpar is demonstrated on synthetic 2D utility functions, such as the one shown in Figure 3.5a. Each utility function was generated from a Gaussian process prior on a 30-by-30 grid. These experiments evaluate the potential to scale CoSpar to higher dimensions and the advantages of coactive feedback. We compare three settings for CoSpar's (n, b) parameters: $(2, 0)$, $(3, 0)$, $(1, 1)$. For each setting—as well as with and without coactive feedback—we simulate CoSpar on each of the 100 random objective functions. In each case, the number of objective function evaluations, or experimental trials, was held constant at 150.

Coactive feedback is simulated using a 2nd-order differencing approximation of the objective function's gradient. If CoSpar selects a point at which both gradient components have magnitudes below their respective 50^{th} percentile thresholds, then no coactive feedback is given. Otherwise, we consider the higher-magnitude gradient component, and depending on the highest threshold that it exceeds (50^{th} or 75^{th}), simulate coactive feedback as either a 5% or 10% increase in the appropriate direction and dimension.

Figure 3.6 shows the simulation results. In each case, the mixed-initiative simulations involving coactive feedback improve upon those with only preferences. Learning is slowest for $n_s = 2, n_b = 0$ (Figure 3.6), since that case elicits the fewest preferences. Figure 3.5b depicts the utility model's posterior mean for the objective function in Figure 3.5a, learned in the simulation with $n_s = 1$, $n_b = 1$, and mixed-initiative feedback. In comparing Figure 3.5b to Figure 3.5a, we see that CoSpar learns a sharp peak around the optimum, as it is designed to converge to sampling preferred regions, rather than giving the user undesirable options by exploring elsewhere.

Human Subject Experiments on the Atalante Exoskeleton

After its validation in simulation, CoSpar was deployed on a lower-body exoskeleton, Atalante, in two personalized gait optimization experiments with human subjects (video: [190]). Both experiments aim to determine gait parameter values that maximize user comfort, as captured by preference and coactive feedback. The first experiment,[3] repeated for three able-bodied subjects, used CoSpar to determine the user's preferred step length, i.e., opti-

[3]Kernel: squared exponential with lengthscale = 0.03, signal variance = 0.005, noise variance = 1e-7; preference noise (σ) = 0.02.

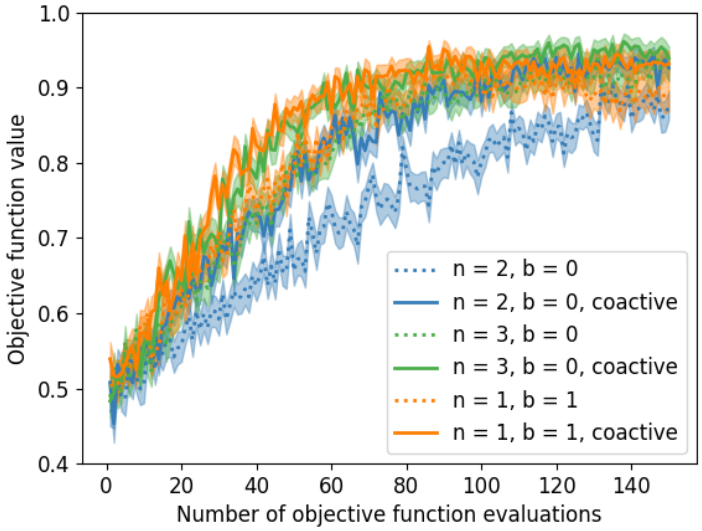


Figure 3.6: CoSpar simulation results on 2D synthetic objective functions, comparing CoSpar with and without coactive feedback for three parameter settings n_s and n_b. Mean +/- standard error of the objective values achieved over 100 repetitions. The maximal and minimal objective function values are normalized to 0 and 1. We see that coactive feedback always helps, and that $n_s = 2$, $n_b = 0$—which receives the fewest preferences—performs worst.

mizing over a one-dimensional feature space. The second experiment[4] demonstrates CoSpar's effectiveness in two-dimensional feature spaces, and optimizes simultaneously over two different gait feature pairs. Importantly, CoSpar operates independently of the choice of gait features. The subjects' metabolic expenditure was also recorded via direct calorimetry, but this data was uninformative of user preferences, as users are not required to expend effort toward walking.

One-Dimensional Action Space. In the first experiment, all three subjects walked inside the Atalante exoskeleton, with CoSpar selecting the gaits. We considered 15 equally-spaced step lengths between 0.08 and 0.18 meters, each with a precomputed gait from the gait library. Feature discretization was based on users' ability to distinguish nearby values. The users decided when to end each trial, so as to be comfortable providing feedback. Since users have difficulty remembering more than two trials at once, we used CoSpar with $n_s = 1$ and $n_b = 1$, which corresponds to asking the user to compare each current

[4]Same parameters as in [3] except for step duration lengthscale = 0.08 and step width lengthscale = 0.03.

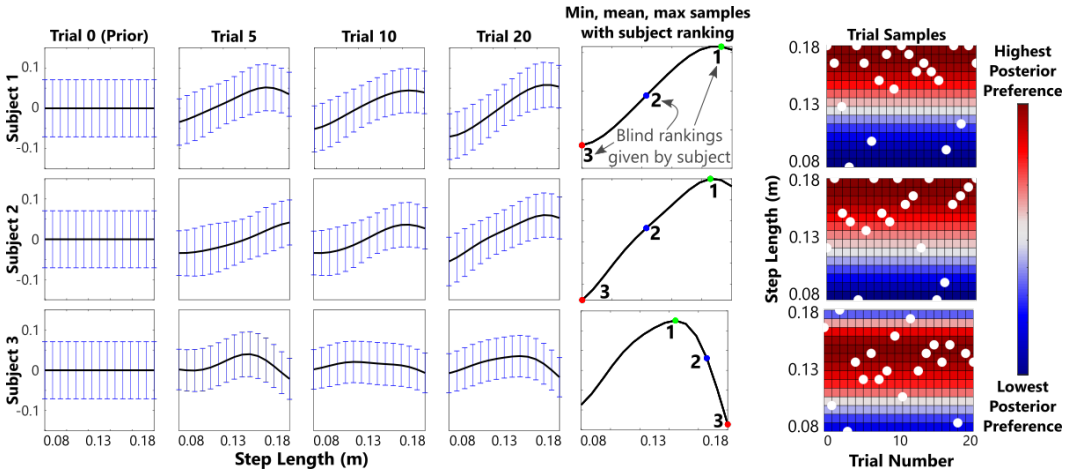


Figure 3.7: Experimental results for optimizing step length with three subjects (one row per subject). Columns 1-4 illustrate the evolution of the preference model posterior (mean +/- standard deviation), shown at various trials. CoSpar converges to similar but distinct optimal gaits for different subjects. Column 5 depicts the subjects' blind ranking of the 3 gaits sampled after 20 trials. The rightmost column displays the experimental trials in chronological order, with the background depicting the posterior preference mean at each step length. CoSpar draws more samples in the region of higher posterior preference.

trial with the preceding one. Additionally, we query the user for coactive feedback: after each trial, the user can suggest a longer or shorter step length (±20% of the range), a *slightly* longer or shorter step length (±10%), or no feedback.

Each participant completed 20 gait trials, providing preference and coactive feedback after each trial. Figure 3.7 illustrates the posterior's evolution over the experiment. After only five exoskeleton trials, CoSpar was already able to identify a relatively-compact preferred step length subregion. After the 20 trials, three points along the utility model's posterior mean were selected: the maximum, mean, and minimum. The user walked in the exoskeleton with each of these step lengths in a randomized ordering, and gave a blind ranking of the three, as shown in Figure 3.7. For each subject, the blind rankings match the preference posterior obtained by CoSpar, indicating effective learning of individual user preferences.

Two-Dimensional Action Spaces. We further demonstrate CoSpar's practicality to personalize over multiple features, by optimizing over two different

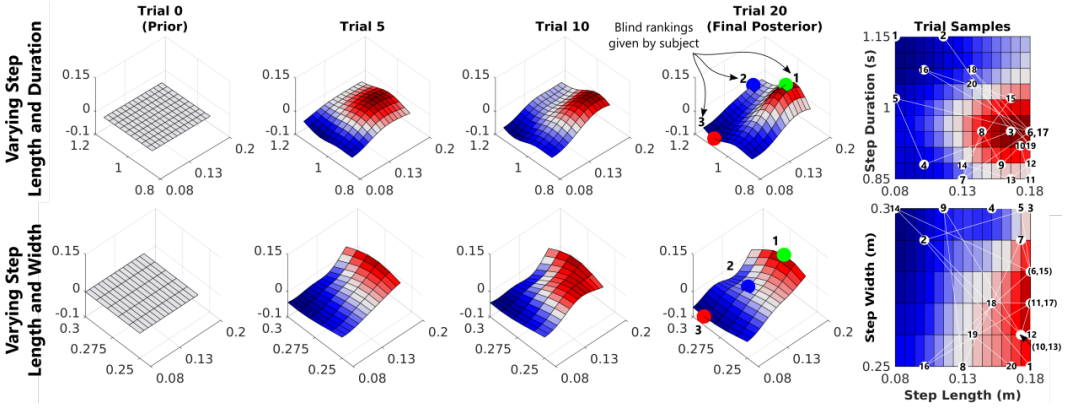


Figure 3.8: Experimental results from two-dimensional feature spaces (top row: step length and duration; bottom row: step length and width). Columns 1-4 illustrate the evolution of the preference model's posterior mean. Column 4 also shows the subject's blind ranking of the 3 gaits sampled after 20 trials. Column 5 depicts the experimental trials in chronological order, with the background as in Figure 3.7. CoSpar draws more samples in the region of higher posterior preference.

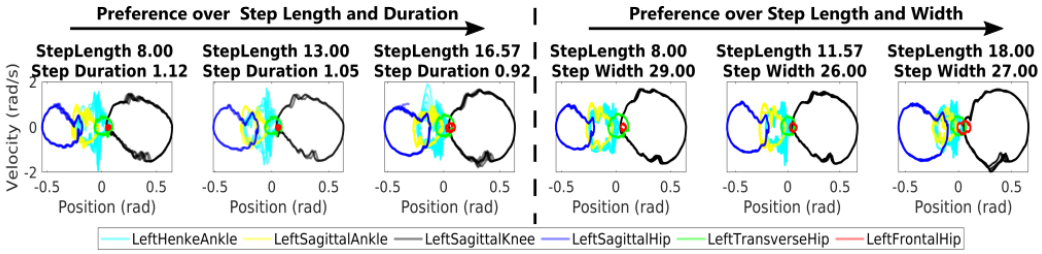


Figure 3.9: Experimental phase diagrams of the left leg joints over 10 seconds of walking. The gaits shown correspond to the maximum, mean, and minimum preference posterior values for both of subject 1's 2D experiments. For instance, subject 1 preferred gaits with longer step lengths, as shown by the larger range in sagittal hip angles in the phase diagram.

feature pairs: 1) step length and step duration and 2) step length and step width. The protocol of the 1D experiment was repeated for subject 1, with step lengths discretized as before, step duration discretized into 10 equally-spaced values between 0.85 and 1.15 seconds (with 10% and 20% modifications under coactive feedback), and step width into 6 values between 0.25 and 0.30 meters (20% and 40%). After each trial, the user was queried for both a pairwise preference and coactive feedback. Figure 3.8 shows the results for both feature spaces. The estimated preference values were consistent with a 3-sample blind ranking evaluation, suggesting that CoSpar successfully identified user-preferred parameters. Figure 3.9 displays phase diagrams of the gaits

Algorithm 3 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
    Normal$(\boldsymbol{\mu}_t, \Sigma_t)$ = posterior over points in $\mathbf{S}_t$, given $\mathbf{D}$
    Sample utility function $f_t \sim$ Normal$(\boldsymbol{\mu}_t, \Sigma_t)$
    Execute action $\boldsymbol{a}_t = \text{argmax}_{\boldsymbol{a} \in \mathbf{S}_t} f_t(\boldsymbol{a})$
    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 $\boldsymbol{a}_{t+1}^* = \text{argmax}_{\boldsymbol{a} \in \mathbf{V}_t} \boldsymbol{\mu}_t(\boldsymbol{a})$
  end for
end procedure

with minimum, mean, and maximum posterior utility values to illustrate the difference between preferred and non-preferred gaits.

3.3 Experimental Results: High-Dimensional Preference Optimization

Optimizing lower-body exoskeleton walking gaits for user comfort requires understanding users' preferences over a high-dimensional gait parameter space. However, the CoSpar algorithm only explored low-dimensional domains due to computational limitations. To learn user preferences in high dimensions, my thesis work also developed the LineCoSpar algorithm [206] outlined in Alg. 3, a human-in-the-loop preference-based framework that enables optimization over many parameters by iteratively exploring one-dimensional subspaces. Additionally, these experiments identify gait attributes that characterize broader preferences across users. In simulations and human trials, we empirically verify that LineCoSpar is a sample-efficient approach for high-dimensional preference optimization.

Simulation Results

We validate the performance of LineCoSpar in simulation using both standard Bayesian optimization benchmarks and randomly-generated polynomi-

als.[5] The simulations show that LineCoSpar is sample-efficient, converges to sampling higher-valued actions, and learns a preference relation function such that actions with higher objective values have high posterior utilities.

Standard Bayesian Optimization Benchmarks. We evaluated the performance of LineCoSpar on the standard Hartmann3 (H3) and Hartmann6 (H6) benchmarks (3 and 6 dimensions, respectively). We do not compare LineCoSpar to other optimization methods because there are no other preference-based Gaussian process methods that are tractable in high dimensions. As discussed, we focus on Gaussian process methods because they model smooth, non-parametric utility functions. We validate LineCoSpar with noiseless preferences and then demonstrate its robustness to noisy user preferences. Preferences are generated in simulation by comparing objective function values.

Under ideal preference feedback, $\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}$ if $r(\boldsymbol{a}_{k_1}) > r(\boldsymbol{a}_{k_2})$. The true objective values r are invisible to the algorithm, which observes only the preference dataset $\mathbf{D}$. Compared to CoSpar, LineCoSpar converges to sampling actions with higher objective values at a faster rate (Figure 3.10). Thus, LineCoSpar not only enables higher-dimensional optimization, but also improves speed and accuracy of learning.

Since human preferences may be noisy, we tested the algorithm's robustness to noisy preference feedback. In simulation, this is modeled via:

$$\mathbb{P}(\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}) = (1 + e^{-\frac{s_k}{c_p}})^{-1}, \tag{3.44}$$

where $s_k = r(\boldsymbol{a}_{k_1}) - r(\boldsymbol{a}_{k_2})$ and c_p is the hyperparameter for the preference noise level. As $c_p \to \infty$, the preferences approach uniform randomness (i.e., become noisier). Also, actions become less distinguishable when the distance between $r(\boldsymbol{a}_{k_1})$ and $r(\boldsymbol{a}_{k_2})$ decreases. This reflects human preference generation since it is more difficult to give consistent preferences between actions with similar utilities. By simulating noisy preferences, we demonstrate that LineCoSpar is robust to noisy feedback (see Figure 3.11).

Randomly-Generated Functions. We also tested LineCoSpar using randomly-generated v-dimensional polynomials (for $v = 6$) as objective functions: $r(\boldsymbol{a}) =$

[5] The code is at https://github.com/myracheng/linecospar. All experiments use the squared exponential kernel with lengthscale 0.15 in every dimension, signal variance 1e−4, noise variance 1e−5, and preference noise 0.005.

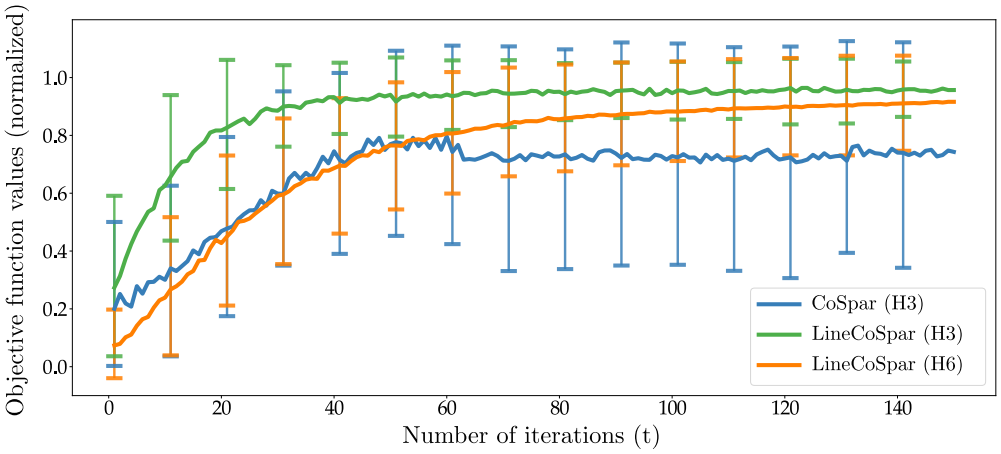


Figure 3.10: Convergence to higher values on standard benchmarks. Mean objective value ± SD using H3 and H6, averaged over 100 runs. The sampled actions converge to higher objective values at a faster rate with LineCoSpar, which has an improved sampling approach and activation function. It is intractable to run CoSpar on a 6-dimensional space.

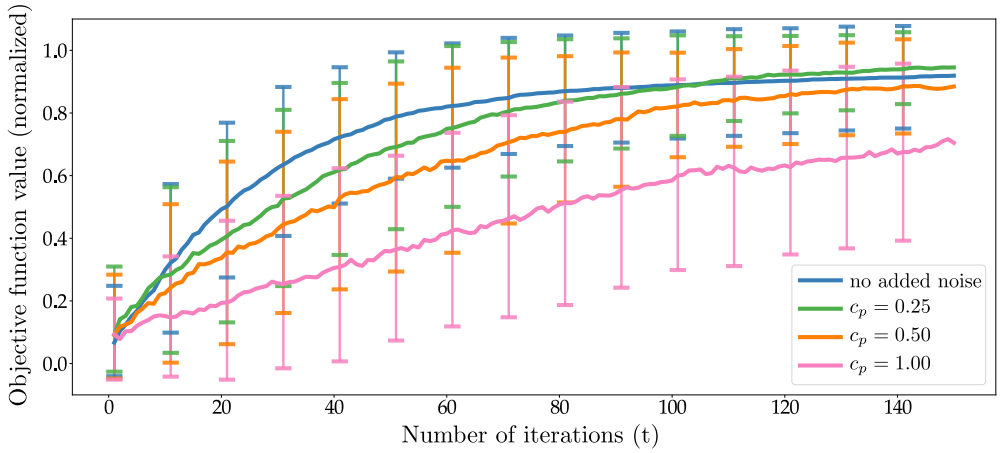


Figure 3.11: Robustness to noisy preferences. Mean objective value ± SD of the action $\boldsymbol{a}_{\max}$ with the highest posterior utility. This is averaged over 100 runs using LineCoSpar on H6 with varying preference noise, as quantified by c_p. Higher performance correlates with less noise (lower c_p). The algorithm is robust to noise to a certain degree ($c_p \leq 0.5$).

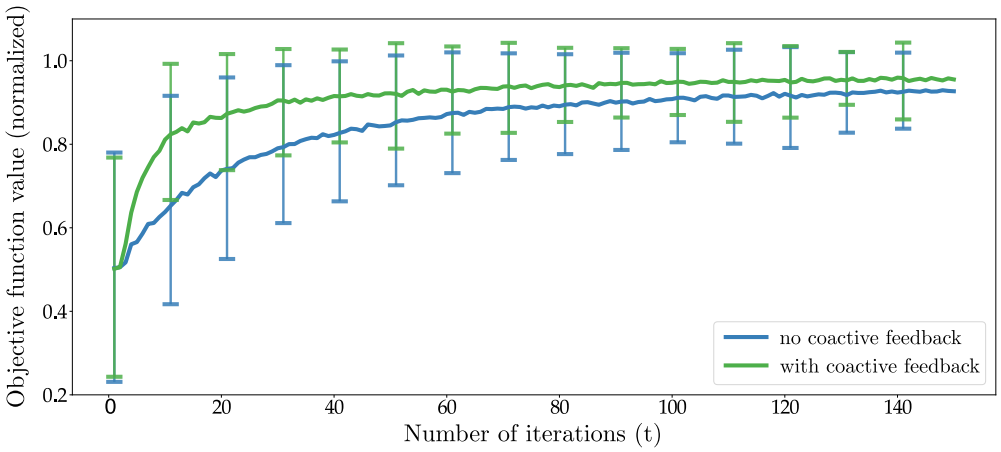


Figure 3.12: Coactive feedback improves convergence. Mean objective value ± SD of the sampled actions using random functions. This is averaged over 1000 runs using LineCoSpar on 100 randomly-generated six-dimensional functions ($v = 6$). The sampled actions converge to high objective values in relatively few iterations, and coactive feedback accelerates this process.

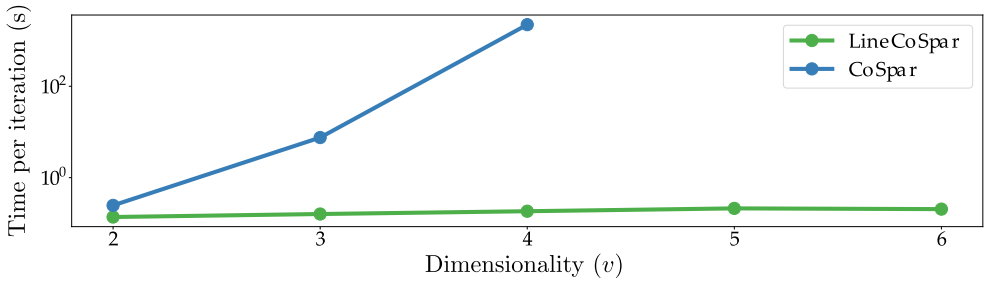


Figure 3.13: Curse of dimensionality for CoSpar. Average time per iteration of CoSpar vs. LineCoSpar. The y-axis is on a logarithmic scale. For LineCoSpar, the time is roughly constant in the number of dimensions v, while the runtime of CoSpar increases exponentially. For $v = 4$, the duration of a CoSpar iteration is inconvenient in the human-in-the-loop learning setting, and for $v \geq 5$, it is intractable.

$\sum_{i=1}^{v} c_i \sum_{j=1}^{v} d_j a_j$, where a_j denotes the j^{th} element of $\boldsymbol{a}$, and $c_i, d_i, i \in \{1, \ldots, v\}$ are sampled independently from the uniform distribution Uniform$(-1, 1)$. The dimensions' ranges and discretizations match those in the exoskeleton experiments, so that these simulations approximate the number of human trials needed to find optimal gaits.

Table 3.1: Gait parameters identified by LineCoSpar as optimizing for user comfort for six able-bodied subjects.

Sub.	Height (m)	Mass (kg)	SL (m)	SD (s)	SW (m)	SH (m)	PR (deg)	PP (deg)	Validation Accuracy (%)
1	1.85	89.9	0.0835	0.943	0.278	0.0674	6.38	10.9	**75**
2	1.668	69.2	0.136	1.04	0.285	0.0679	6.41	12.4	**100**
3	1.635	51.2	0.137	0.922	0.279	0.0688	8.56	11.4	**100**
4	1.795	73.6	0.127	0.989	0.268	0.065	6.68	12.7	**25**
5	1.625	55.9	0.161	1.05	0.258	0.0689	7.32	13.2	**100**
6	1.66	65	0.177	1.11	0.256	0.0663	7.71	13.5	**100**

Coactive feedback was simulated for each sampled action $\boldsymbol{a}_t$ by finding an action $\boldsymbol{a}'_t$ with a higher objective value that differs from $\boldsymbol{a}_t$ along only one dimension. The action $\boldsymbol{a}'_t$ is determined by randomly choosing a dimension in $\{1, \ldots, v\}$ and direction (positive or negative), and taking a step from $\boldsymbol{a}_t$ along this vector. If the resulting action $\boldsymbol{a}'_t$ has a higher objective value, it is added to the dataset $\mathbf{D}$ as $\boldsymbol{a}'_t \succ \boldsymbol{a}_t$. This is a proxy for the human coactive feedback acquired in the exoskeleton experiments described below, in which the user can suggest a dimension and direction in which to modify an action to obtain an improved gait.

Figure 3.12 displays LineCoSpar's performance over 100 randomly-generated polynomials (10 repetitions each) with computation time shown in Figure 3.13. The results demonstrate that LineCoSpar samples high-valued actions within relatively few iterations (≈ 20 with coactive feedback).

Human Subject Experiments on the Atalante Exoskeleton

After the performance of LineCoSpar was demonstrated in simulation, the algorithm was experimentally deployed on the lower-body exoskeleton Atalante to optimize six gait parameters for six able-bodied users (see Table 3.1 for results and [188] for a video).

Experimental Procedure. LineCoSpar optimized exoskeleton gaits for six self-identified able-bodied subjects over six gait parameters: step length, step duration, step width, maximum step height, pelvis roll, and pelvis pitch. These parameters were chosen from the pre-computed gait library because they are relatively intuitive for users to understand when giving coactive feedback. The parameter ranges, respectively, are: 0.08-0.18 meters, 0.85-1.15 seconds, 0.25-0.3 meters, 0.065-0.075 meters, 5.5-9.5 degrees, and 10.5-14.5 degrees.

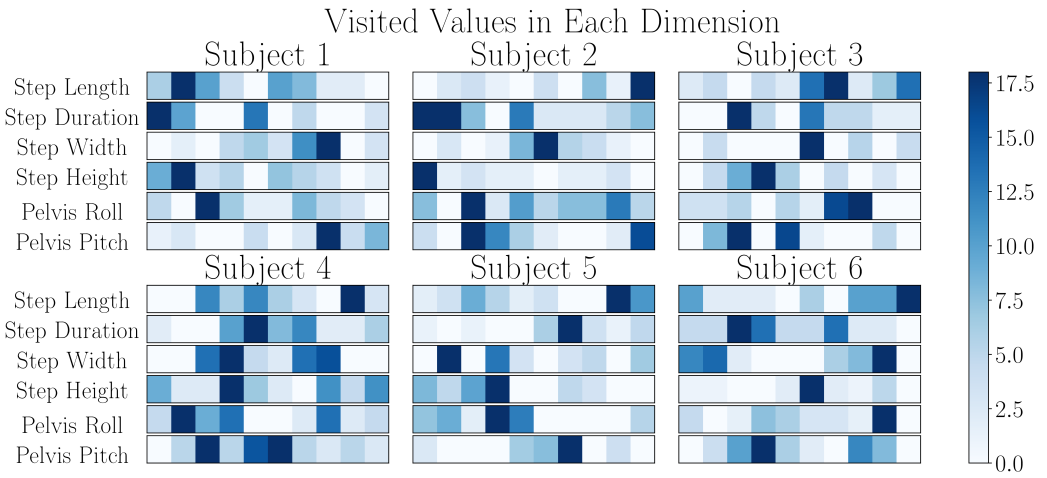


Figure 3.14: Exploration vs. exploitation in human trials. Each row depicts the distribution of a particular gait parameter's values across all gaits that the subject tested. Each dimension is discretized into 10 bins. Note that the algorithm explores different parts of the action space for each subject. These visitation frequencies exhibit a statistically-significant correlation with the posterior utilities across these regions (Pearson's p-value = 1.22e-10).

All subjects were volunteers without prior exoskeleton exposure. For each subject, the testing procedure lasted approximately two hours, with one hour of setup and one hour of exoskeleton testing. The setup consisted of explaining the procedure (including how to provide preference and coactive feedback), measuring subject parameters, and adjusting the thigh and shank length of the exoskeleton to the subject. During the testing, the subjects had control over initiating and terminating each instance of exoskeleton walking and were instructed to try each walking gait until they felt comfortable giving a preference. The subjects could choose to test each gait multiple times to confirm their preference. They could also specify "no preference" between two gait trials, in which case no new information was added to the dataset $\mathbf{D}$.

After completing 30 trials (including trials with no preference, but not including voluntary gait repetitions), the subject began a set of "validation" trials; for consistency, the subject was not informed of the start of the validation phase. Validation consisted of six additional trials and yielded four pairwise preferences, each between the posterior-maximizing action $\boldsymbol{a}_{\max}$ and a randomly-generated action. This validation step verifies that $\boldsymbol{a}_{\max}$ is preferred over other parameter combinations across the search space.

Algorithm 4 ROIAL Algorithm

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

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

Experimental Results. Figure 3.14 shows that the LineCoSpar algorithm both explores across the gait parameter space and exploits regions with higher posterior utility. Over time, LineCoSpar increasingly samples actions concentrated in regions of the search space that are preferred based on previous feedback. This results in a significant correlation between visitation frequencies and posterior utilities across these regions (Pearson's p-value = 1.22e-10).

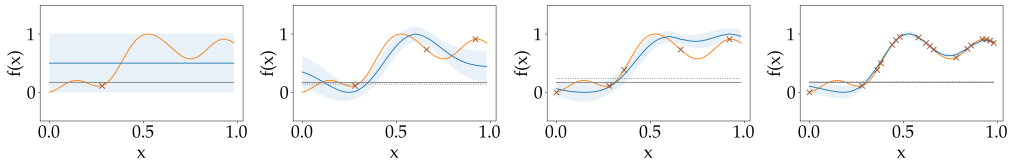
For each subject, Table 3.1 lists the parameters of the predicted optimal gaits, $\boldsymbol{a}_{\max}$, identified by LineCoSpar. Table 3.1 also illustrates the results of the validation trials for each subject. These results show that $\boldsymbol{a}_{\max}$ was predominantly preferred over the randomly-selected actions during validation. For four of the six subjects, all four validation preferences matched the posterior, while the other subjects matched three and one of the four preferences, respectively.

3.4 Experimental Results: Preference Characterization

Since optimizing every exoskeleton user's comfort from scratch would be time-consuming and inefficient, the final presented algorithm, termed ROIAL (Region of Interest Active Learning) [128], is aimed at preference characterization. For clarity, the procedure is outlined in Alg. 4.

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



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

Figure 3.15: 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.

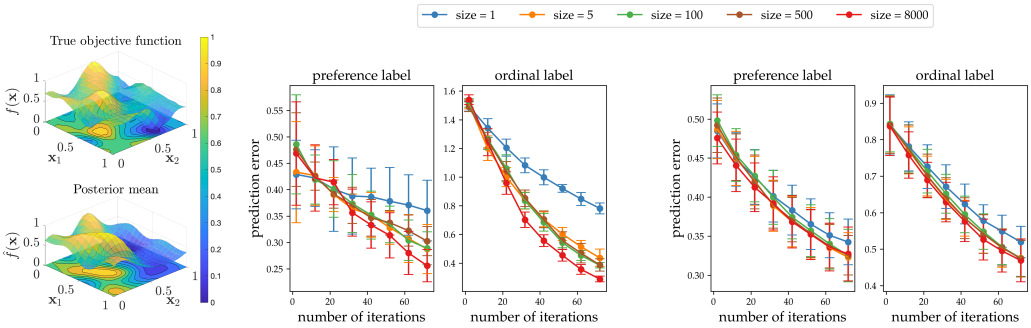
when the true utility function is unknown. The average ordinal prediction error is defined as $\overline{\text{Error}}(N) := \frac{1}{N}\sum_{k=1}^{N} |o_k^{pred} - o_k^{true}|$, and all simulations use $h = 5$ ordinal categories.[6]

Figure 3.15 illustrates the algorithm for a 1D objective function. Initially, ROIAL samples widely across the action space (Figure 3.15a-3.15c). As seen by comparing iterations 5 and 20 (Figure 3.15c-3.15d), 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).

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 3.16a provides an example of a 3D posterior, Figure 3.16b depicts the average performance for H3 over 10 simulation repetitions, and Figure 3.16c 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.

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. Figure 3.17a 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

[6]Unless otherwise stated, hyperparameters are held constant across simulations and experiments, and their values can be found in https://github.com/kli58/ROIAL.

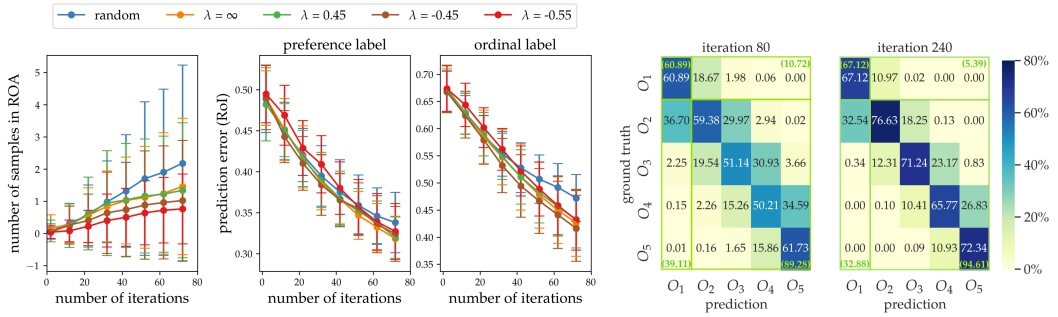


(a) Synthetic function posterior (b) Hartmann3 prediction error (c) Synthetic function prediction error

Figure 3.16: 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 ± 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.

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 (Figure 3.17a). Additionally, the confusion matrix (Figure 3.17b) shows that the algorithm usually predicts either the correct ordinal label or an adjacent ordinal category. The ROI prediction accuracy (green text in Figure 3.17b) indicates that ROIAL predicts whether points belong to the ROI with relatively-high accuracy.

Since user feedback is expected to be noisy, we evaluate the algorithm's robustness to noisy feedback generated from the distributions $\mathbb{P}((\boldsymbol{a}, o) \mid r) = \psi_o\left(\frac{\tilde{b}_o - r(\boldsymbol{a})}{\tilde{c}_o}\right) - \psi_o\left(\frac{\tilde{b}_{o-1} - r(\boldsymbol{a})}{\tilde{c}_o}\right)$ and $\mathbb{P}(\boldsymbol{a}_1 \succ \boldsymbol{a}_2 \mid r) = \psi_p\left(\frac{r(\boldsymbol{a}_1) - r(\boldsymbol{a}_2)}{\tilde{c}_p}\right)$ for ordinal and preference feedback, respectively, with true ordinal thresholds $\{\tilde{b}_j | j = 1, \ldots, h-1\}$ and simulated noise parameters $\tilde{c}_p$ and $\tilde{c}_o$. We set $\tilde{c}_o > \tilde{c}_p$ 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 (Figure 3.18).



(a) Number of samples in the ROA and prediction error in the ROI

(b) Confusion matrices

Figure 3.17: 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 ± 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 ± 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.

Human Subject Experiments on the Atalante Exoskeleton

After demonstrating ROIAL's performance in simulation, we experimentally deployed it on the Atalante exoskeleton[7] (video: [193]).

Four gait library parameters were selected for these experiments: step length (SL) in meters, step duration (SD) in seconds, maximum pelvis roll (PR) in degrees, and maximum pelvis pitch (PP) in degrees. These parameters were selected because exoskeleton users frequently suggested modifications to SL, SD, and PR in the preference optimization experiments presented earlier, 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

[7]Hyperparameters can be found in the repository https://github.com/kli58/ROIAL

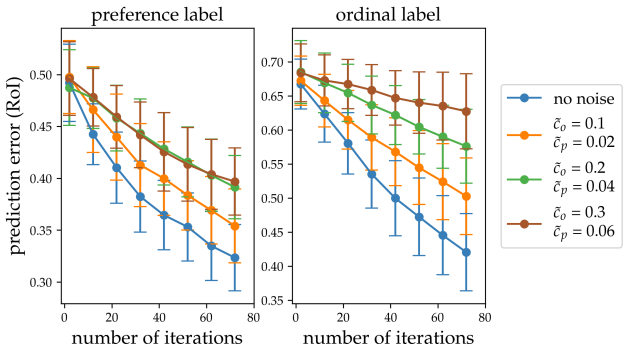


Figure 3.18: Effect of noisy feedback. The ordinal and preference noise parameters, $\tilde{c}_0$ and $\tilde{c}_p$, range from 0.1 to 0.3 and 0.02 to 0.06, respectively. All cases use a random subset size of 500 and $\lambda = -0.45$, and each simulation uses 1,000 random actions to evaluate label prediction. Plots show means $\pm$ standard deviation.

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.

Figure 3.19 depicts the results of the validation phase for all three subjects. These results show a reliable correlation between the predicted categories and

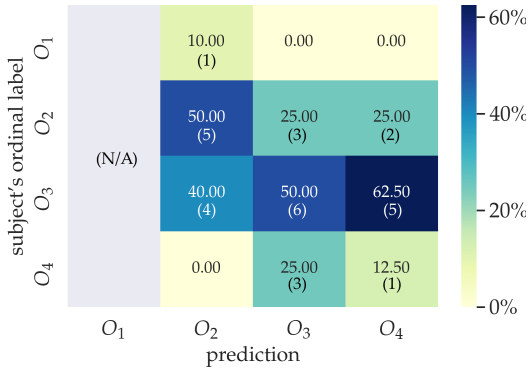


Figure 3.19: Confusion matrix of the validation phase results for all three subjects. The first column is grey because actions in the ROA ($\mathbf{O}_1$) were purposefully avoided to prevent subject discomfort. Percentages are normalized across columns. Parentheses show the numbers of gait trials in each case.

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 3.17b). Overall, these results suggest that ROIAL can yield reliable preference landscapes within a moderate number of samples.

Figure 3.20 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[8]. 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.

[8]GitHub Repo: https://github.com/kli58/ROIAL

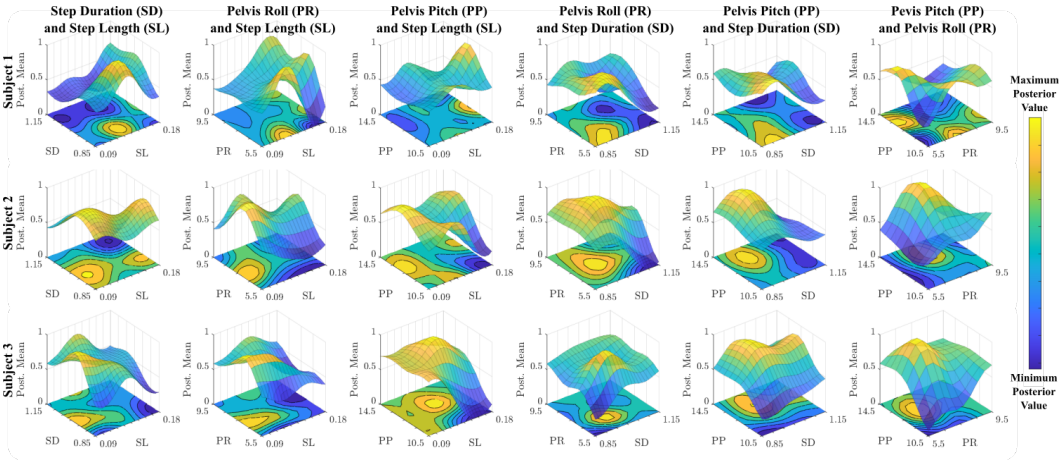


Figure 3.20: 4D posterior mean utility across exoskeleton gaits. Utilities are plotted over each pair of gait space parameters, with the values averaged over the remaining 2 parameters in each plot. Each row corresponds to a subject: Subject 1 is the most experienced exoskeleton user, Subject 2 is the second-most experienced user, and Subject 3 never used the exoskeleton prior to the experiment.

3.5 Experimental Results: Patients with Paraplegia

Having evaluated the preference-based learning framework across both learning objectives with non-disabled participants, 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[163], 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.

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 3.2 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 3.21.

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 3.21 and the gaits identified as optimal illustrated via gait tiles in the bottom row. These second experimental sessions were 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.

3.6 Experimental Results: 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

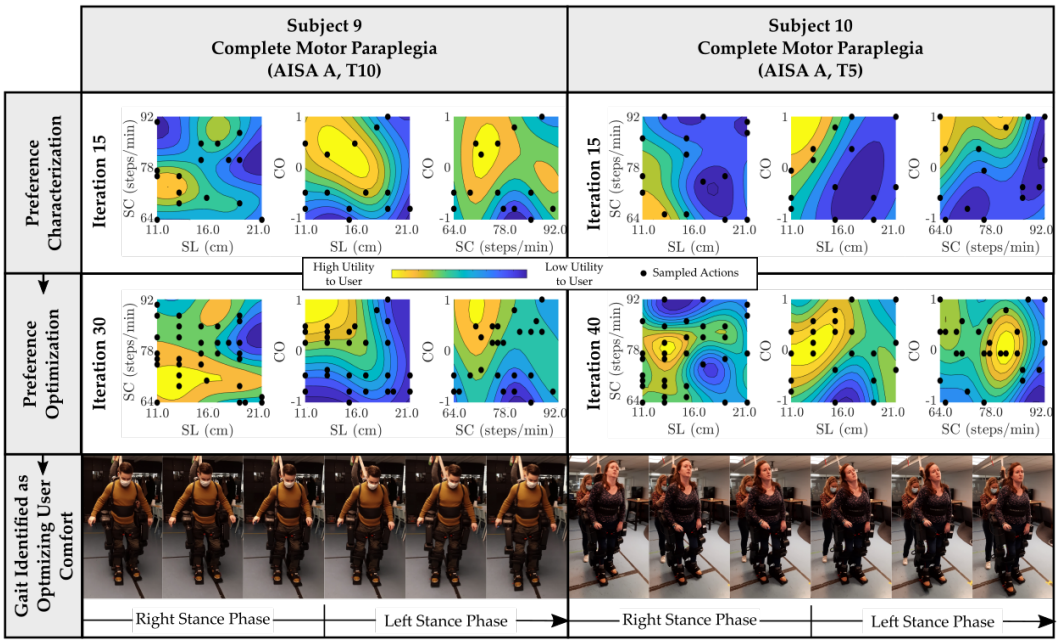


Figure 3.21: Experimental results of unified framework during exoskeleton walking for subjects with paraplegia. We illustrate the experimental results from applying the learning framework towards preference characterization and preference optimization for two subjects with complete motor paraplegia. 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.

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 3.2, and illustrated in the top row of Figure 3.22.

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 3.22. 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 3.22. 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 3.22.

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 was $[20 \text{ deg}, 0.9 \text{ s}, 0.875 \text{ s}, 15 \text{ mm}, 5 \text{ cm}, 5 \text{ cm}]$. 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 3.2. This is striking, considering that the action space (defined in Figure 3.2) 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.

3.7 Summary

To systematically explore the space of possible gaits, 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 subjec-

tive 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.

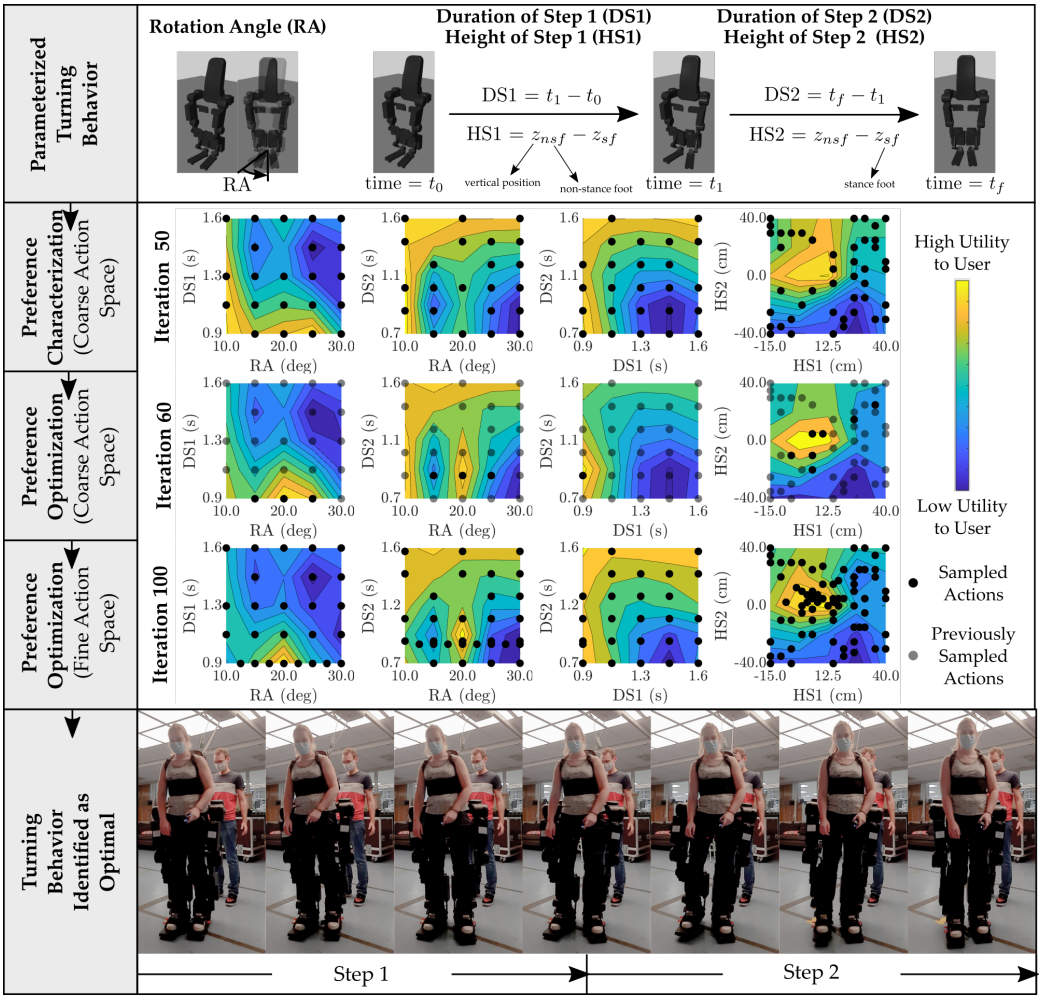


Figure 3.22: Experimental results of unified framework during exoskeleton turning for a non-disabled subject. To demonstrate the learning framework's application-agnostic nature, we applied it to sequentially characterize and optimize user comfort during exoskeleton turning. First, we defined the action 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.

Chapter 4

EXTENDING APPLICATIONS OF PREFERENCE-BASED LEARNING

Potential applications of preference-based learning extend beyond optimizing user comfort during exoskeleton locomotion. This extension comes from generalizing the previously presented learning framework, shown in Figure 3.1, to the procedure illustrated below in Figure 4.1. Overall, the work presented in this chapter discovers that the generalized preference-based learning framework is extremely powerful towards optimizing robotic behavior for any situation where the underlying reward function is difficult to characterize numerically. Specifically, throughout this chapter, my work experimentally demonstrates the generalized learning framework towards three robotic applications. Importantly, the robotic behaviors are still realized using theoretically-motivated methods from nonlinear control theory, with the learning merely shaping the experimental behavior. This provides a systematic approach towards realizing experimentally stable, robust, and performant behaviors without relying heavily on heuristics. Also, by leveraging only *subjective* feedback, we take advantage of a human operator's natural ability to recognize *good* robotic behavior—something that is challenging to capture numerically.

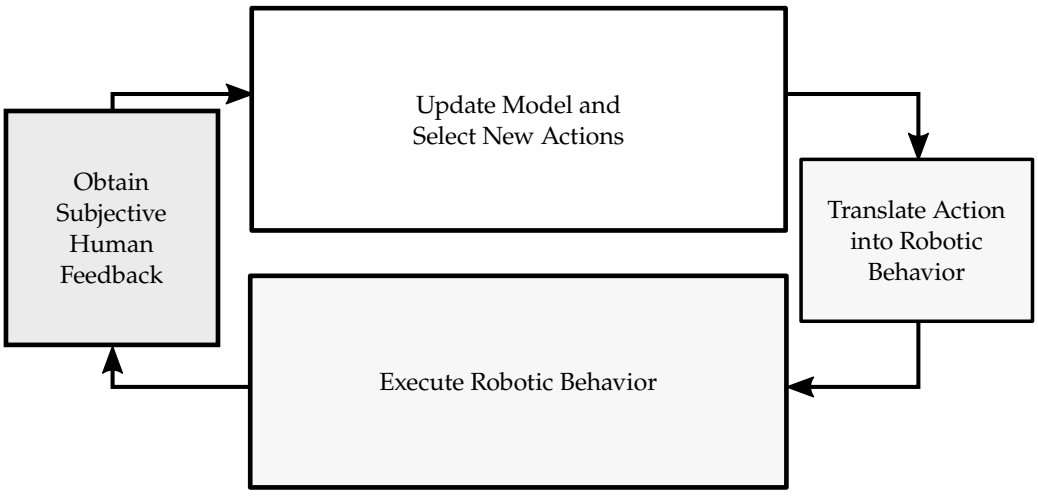


Figure 4.1: Generalized procedure for applying preference-based learning to robotic applications.

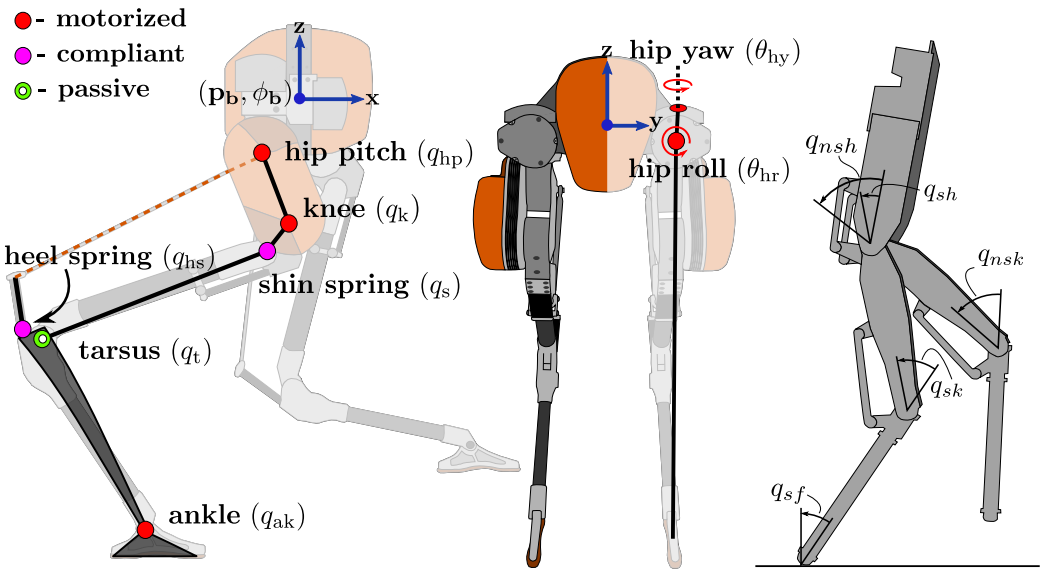


Figure 4.2: Configuration of the 22 DOF (using an unpinned model) Cassie robot[164] (left) and configuration of the 5 DOF (using a pinned model) planar robot AMBER-3M[7] (right).

First, I present the results of applying preference-based learning towards achieving stable and robust locomotion on the planar biped AMBER-3M (illustrated in Figure 4.2) by directly tuning the essential constraints of the HZD optimization problem [208]. Second, I will demonstrate preference-based learning for tuning low-level control parameters of a control Lyapunov function based quadratic program (CLF-QP), resulting in stable and robust bipedal locomotion on both AMBER-3M and Cassie (also illustrated in Figure 4.2) [59]. Lastly, I will discuss the application of preference-based learning towards tuning the performance/safety trade-off of a control barrier function with experimental demonstrations on the Unitree quadrupedal platform A1 [56].

Related Work. The "last-mile mission" of realizing bipedal locomotion in the real world was historically solved by intensive parameter tuning. This parameter tuning process is typically arduous and nonintuitive, thereby inevitably affecting the scalability of translating theory to hardware in a practical setting. To circumvent this engineering empiricism, the field of machine learning has approached bipedal locomotion from different perspectives, including reinforcement leaning and imitation learning. Reinforcement learning simplifies the process of "learning to walk" [147] without prior knowledge [46, 81, 92, 142], but because this methodology relies on a carefully crafted reward function, the behavior is exclusively determined by its construction. This motivates the

second method, imitation learning, which infers the underlying reward function from expert demonstrations [102, 202, 232]. While both methods have demonstrated promising results, they heavily rely on physical engines such as Bullet [58], MuJoCo [203], and RaiSim [101]. As realistic as these rigid-body-dynamics based simulation environments have become, they still struggle with rough-terrain dynamics such as elastic impacts, slipping contacts, and granular media. These differences become more apparent when transferred to real-world systems.

Instead of relying on just one field, my research explores combining the successes of both: the formality of stability from control theory and the ability to learn the relationship between complex parameter combinations and their resulting locomotive behavior from machine learning. This is accomplished by integrating preference-based learning with control theoretic approaches such as the HZD method of gait generation, CLF-QP based controllers, and control barrier function controllers. The result is optimal walking on hardware, shaped by pairwise preferences from the operator (i.e., the user prefers gait A over gait B). Preferences are a particularly useful feedback mechanism for parameter tuning because they are able to capture the notion of "general goodness" without a predefined reward function. This is particularly important for bipedal locomotion due to the lack of commonly agreed upon numerical metric of good or even stable walking in the community [78, 154, 213, 223].

4.1 PBL for User-Guided Gait Generation

During the HZD trajectory optimization parameter tuning process, expert operators typically tune $\boldsymbol{a}_{\min} \in \mathbb{R}^v$ and $\boldsymbol{a}_{\max} \in \mathbb{R}^v$, denoting the bounds of the essential constraints, in the hopes of guiding the HZD optimization towards a solution that maximizes the operators' subjective metric of "good" walking. Since the construction of these constraints is often essential towards achieving experimental robustness, we term them *essential constraints*. Traditionally, essential constraints consist of gait features such as average velocity, step length, foot clearance, and impact velocity.

Often, practitioners derive intuition on how to shape essential constraints from years of experience. One example of how this intuition relates to stability is Raibert-type controllers [157], which tune the relationship between step length and walking velocity based on a simplified model. Instead of relying on

Table 4.1: Action Space Definition: HZD optimization problem essential constraint bounds.

Essential Constraint	Bounds $[a_i^{\min}, a_i^{\max}]$	Disc. d_i
Average Forward Velocity (m/s)	$[0.3, 0.6]$	0.05
Clearance Tau $(\cdot)$	$[0.4, 0.7]$	0.1
Minimum Foot Clearance (m)	$[0.05, 0.19]$	0.02
Impact Velocity (m/s)	$[-0.8, -0.2]$	0.1
Step Length (m)	$[0.2, 0.4]$	0.05

this manual tuning process, this section explores the use of preference-based learning to systematically tune the essential constraints. To do so, we express the essential constraints as:

$$\boldsymbol{a} - \boldsymbol{\varepsilon} \preceq e(X) \preceq \boldsymbol{a} + \boldsymbol{\varepsilon}, \tag{4.1}$$

where $\boldsymbol{a} \in \mathbb{R}^v$ consists of $v \in \mathbb{N}$ constraint values, and $\boldsymbol{\varepsilon} \in \mathbb{R}^v$ defines the equality tolerance for each constraint. The associated learning objective is thus preference optimization, with the underlying utility function denoted as before by $r : \mathbb{R}^v \to \mathbb{R}$. For the specific application of preference-based learning presented in this section, the components of $\boldsymbol{a}$ are selected to be: 1) average forward velocity of the torso (m/s); 2) phase variable value at which to enforce minimum foot clearance, τ_c; 3) minimum nonstance foot clearance enforced at τ_c (m); 4) downward velocity enforced at impact (m/s); and 5) step length, i.e., the forward distance between swing foot and stance foot at impact (m). These constraints define the search space of possible parameter combinations $\mathcal{A}$, a discretization of $\mathbb{R}^v$, as given in Table 4.1.

Benefits of Preference-Based Learning towards Gait Generation

The traditional hand-tuning process requires a human operator to make assumptions about the underlying utility function r, which is difficult given the following: the non-intuitive relationship between parameter combinations and the resulting experimental behavior; and the need to account for numerous factors including stability, robustness to perturbations/model uncertainty, and visual appearance. Additionally, r admits no obvious mathematical description; eliminating the use of reward-based tuning methods.

Alternatively, preference-based learning mainly relies on pairwise preferences, thereby taking advantage of a human's natural ability to combine many factors into a single judgment of "better" or "worse." Although this requires the

human to provide feedback, there are two major benefits of this approach: 1) the duration of the tuning process is reduced significantly compared to hand-tuning; and 2) pairwise preferences are much easier for a naïve user to provide compared to manually navigating the complex search space of parameter combinations.

In this section, the LineCoSpar algorithm is experimentally deployed (open-source code[1]) to tune the 5 essential constraints outlined in Table 4.1 on the planar robot AMBER-3M [6]. This custom research platform has three interchangeable lower-limb configurations: flat-foot, point-foot, and spring-foot. We specifically selected this platform because of its engineering reliability [135], enabling consistent data collection to isolate the effects of various gaits in the learning process. The controller for AMBER-3M is implemented on an off-board i7-6700HQ CPU @ 2.6GHz with 16 GB RAM, which computes desired torques and communicates them with the motor drivers. The motor driver communication and the control logic run at ∼1kHz, each on a separate core.

Experimental Procedure

In the experiments, walking gaits are generated by the HZD-based method presented in Chapter 2. We take $y^a(q) \triangleq q^a \subset \mathbb{R}^4$ as the position of the four motorized joints of AMBER-3M, $\tau(q)$ to be the linearized forward hip position, and use a 5^{th}-order Bézeir polynomial ($\alpha \in \mathbb{R}^{4\times 6}$) to describe the desired output trajectories. Additionally, the cost function is selected to be the mechanical cost of transport (MCOT), a common metric for locomotion efficiency:

$$\text{MCOT} = \int_{t_0}^{t_f} \frac{P(t)}{mgv} dt, \tag{4.2}$$

where $P(t) = \sum_{i=1}^{4} |u_i(t)\dot{q}_i^a(t)|$ is the 2-norm sum of power.

The average optimization run time is 0.1 second per iteration, with each gait averaging 160 iterations. The experimental procedure is illustrated in Figure 4.3. In our experiments, the learning was conducted for $n_s = 2$ and $n_b = 0$, corresponding to two gaits being compared in each iteration. This was chosen because we empirically found that operators sometimes had difficulty remembering the details of more than two gaits at a time, leading to the most

[1]Repository found at: https://github.com/maegant/ICRA2021-LearningHZD

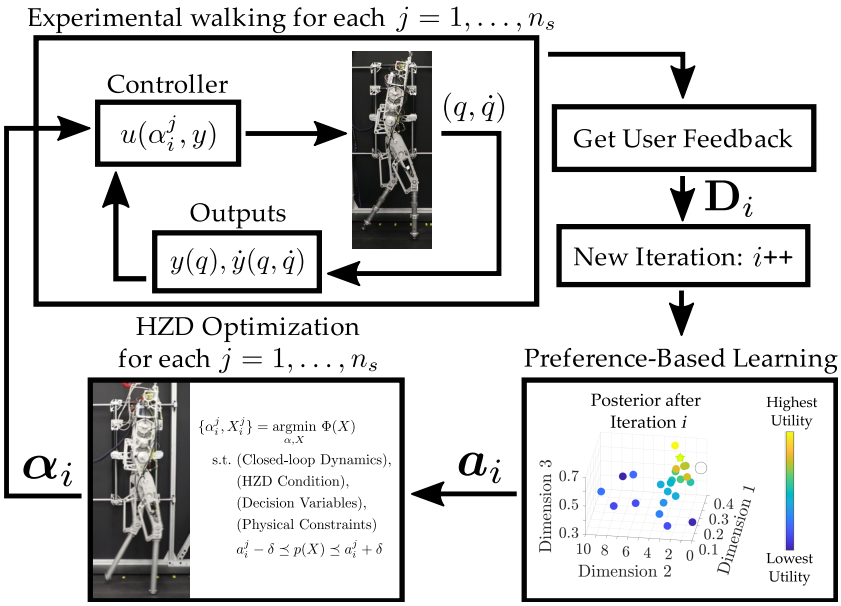


Figure 4.3: The experimental procedure is illustrated in terms of each iteration i with n_s denoting the number of gaits compared in each iteration. The experiments presented in this work used $n_s = 2$. Using this notation, the set of n_s actions given to the HZD optimization is denoted: $\boldsymbol{a}_i = \{a_i^1, \ldots, a_i^{n_s}\}$. The resulting n_s sets of Bézier coefficients given to the controller are denoted $\boldsymbol{\alpha}_i = \{\alpha_i^1, \ldots, \alpha_i^{n_s}\}$.

reliable preference feedback when $n_s = 2$. Note that other applications may benefit in a higher n_s, which would increase the rate of learning.

Each trial began by initializing AMBER-3M in a static double-support configuration, starting the treadmill, and attempting to push the robot into the designed periodic orbit. If the resultant dynamics were not stable, extra precaution was taken to give the gait the best chance at succeeding. Once the gait reached its orbit, the robot was released and the robustness of the gait to various disturbances was investigated. After both gaits were executed on the physical robot, a preference was collected from the human operator observing the physical realization of the walking. In some iterations, video footage was also reviewed before giving a preference. The criteria used to determine preferences between gaits were the following (in order of prioritization): 1) Capable of walking; 2) Robust to perturbations in treadmill speeds; 3) Robust to external disturbance; 4) Does not exhibit harsh noise (e.g., during impact); and 5) Is visually appealing (intuitive judgment from operator).

The entire experimental procedure was completed for two configurations of the robot: 1) the point-foot configuration, AMBER3M-PF (1.373 m, 21.3 kg); and

2) the spring-foot configuration, AMBER3M-SF (1.430 m, 23.5 kg) [6]. The learning framework is first demonstrated on AMBER3M-PF, with the corresponding rigid point-foot model used in the gait generation. To emphasize the scalability of the method, the exact procedure is repeated on AMBER3M-SF, but while intentionally not accounting for changes in the robot model and instead still generating gaits assuming the rigid-body model. Furthermore, the gaits are executed on AMBER3M-SF using the same controller with unmodified gains. Historically, robots with compliance are difficult to generate gaits for because of the resulting complexities which include: increased degrees of freedom of the system; the addition of a double support domain to the hybrid dynamics; and increased stiffness of the dynamics. Past success with compliant bipeds has relied on sophisticated models [88]. Therefore, the fact that the method yields stable walking despite the unmodeled compliance highlights its effectiveness.

Experimental Results

A summary of the experimental results[2] is illustrated in the supplementary video [191]. A visualization of the gait tuning process is provided in Figure 4.4).

The experiment with AMBER3M-PF was run for 30 iterations and sampled 27 unique gaits. The final posterior over the 27 executed actions is illustrated in the top row of Figure 4.5. Since gaits quickly met the first criterion of being able to walk, preferences were mainly dictated based on the robustness and appearance of the experimental walking. The initial gaits tried on hardware, although optimal subject to the imposed constraints, resulted in inferior trajectory tracking and power consumption. As the algorithm progressed, the gaits became significantly smoother, more robust to disturbance, and energy efficient. This is exemplified in Figure 4.6 which illustrates the gaits corresponding to the minimum, a middle, and the maximum posterior utility; the iterations corresponding to when these gaits were first sampled is 1, 21, and 26, respectively. In Figure 4.6, we note significantly lower velocity overshoot for all of the limbs and tighter tracking shown in the phase portraits for the gaits with higher posterior utility. It is also interesting to note the framework's suc-

[2]Additional videos and material available at https://maegant.github.io/ICRA2021-LearningHZD/ and the final obtained posterior provided with the framework code in the repository https://github.com/raisimTech/raisimlib.

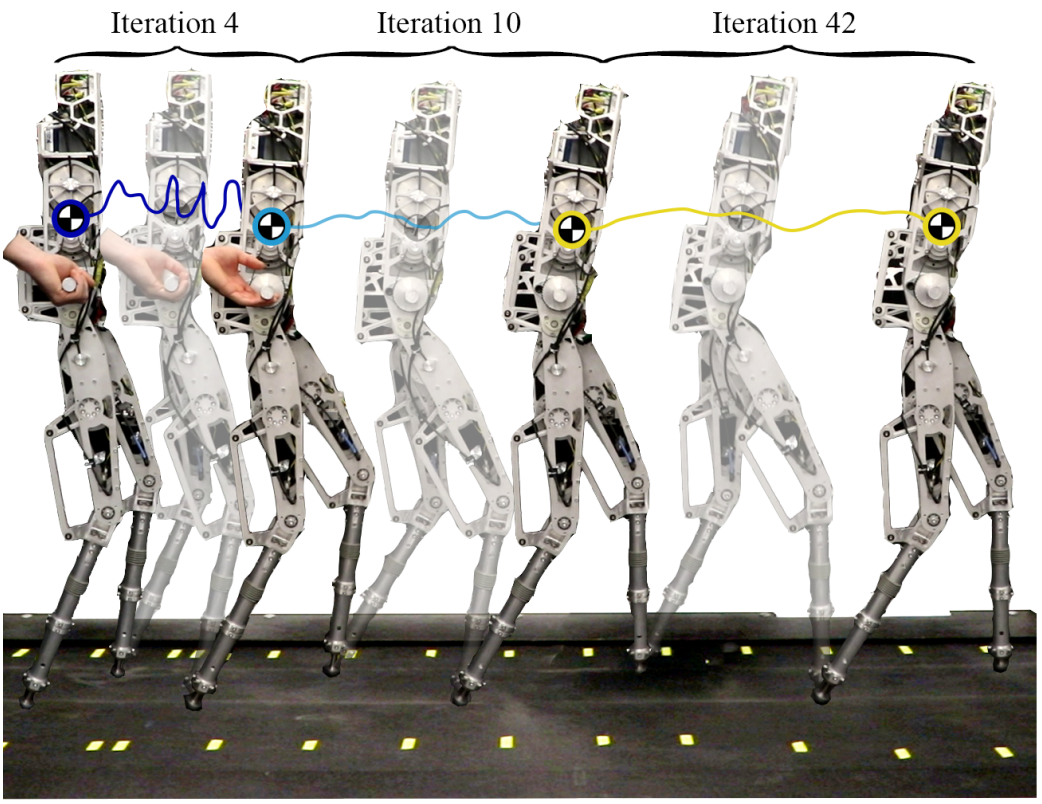


Figure 4.4: Through 50 iterations of experiments, the proposed combination of preference-based learning and HZD optimization transforms failed gaits into robust walking on the AMBER-3M robot with a pair of compliant legs.

cess at improving the efficiency of the experimental walking: a latent property which is discernible to the human operator even though it is not immediately measured. This improvement is demonstrated by the MCOT values of the three gaits in Figure 4.6: 0.74, 0.95, and 0.26, respectively.

When the procedure was repeated on AMBER3M-SF, many of the initial gaits were unable to walk due to the unmodeled compliance. Thus, gaits exhibiting periodic walking were strongly preferred. This second experiment was conducted for 50 iterations and sampled 37 unique gaits with the obtained posterior illustrated in the bottom row of Figure 4.5. Again, three gaits are selected for further discussion corresponding to the minimum, a middle, and the maximum posterior utility values. Gait tiles and phase portraits for these are again shown in Figure 4.6. The iterations when these gaits were first sampled are 4, 10, and 42. Once again, the algorithm converges to gaits with superior trajectory tracking and lower MCOT (1.16, 0.38, and 0.33, respectively).

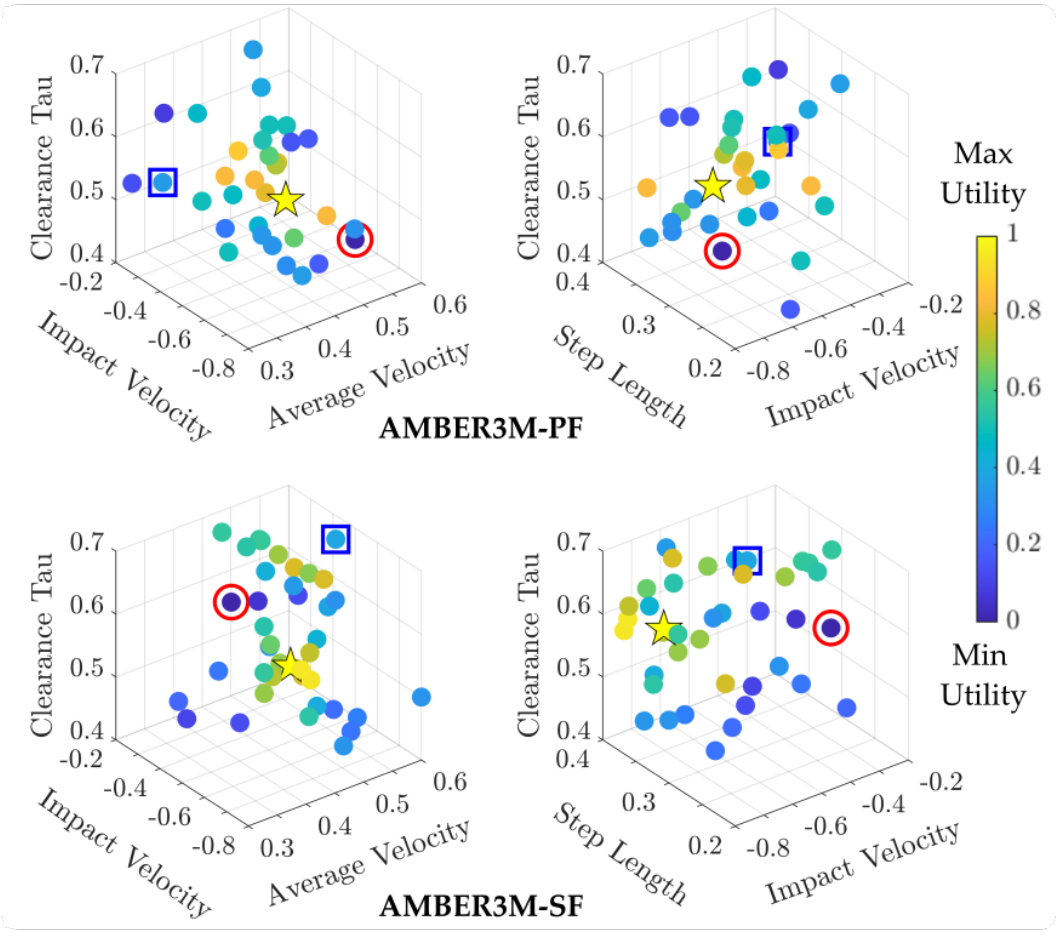


Figure 4.5: The final obtained utilities for the visited actions for the rigid model (top row) and spring model (bottom), with the posterior utility functions averaged over the two dimensions not shown on each subplot. The optimal action is illustrated by the yellow star ($[0.4399, 0.5425, 0.0759, -0.6040, 0.3190]$ for AMBER3M-PF and $[0.4105, 0.5930, 0.0833, -0.7020, 0.3504]$ for AMBER3M-SF). The other two actions depicted in Figure 4.6 are denoted with a red circle (worst gait) and a blue square (middle gait).

4.2 PBL for User-Guided Gain Tuning of a CLF-QP Controller

Experimental demonstration of complex robotic behaviors also relies on finding the correct controller gains. As with tuning essential constraint bounds, this painstaking process is typically completed by a domain expert, requiring deep knowledge of the relationship between parameter values and the resulting behavior of the system. Even when such knowledge is possessed, it can take significant effort to navigate the nonintuitive landscape of possible parameter combinations. To combat this challenge, researchers have developed systematic ways to tune gains for specific controller types [93, 94, 226, 239]. However, for controllers where the input/output relationship between parameters and

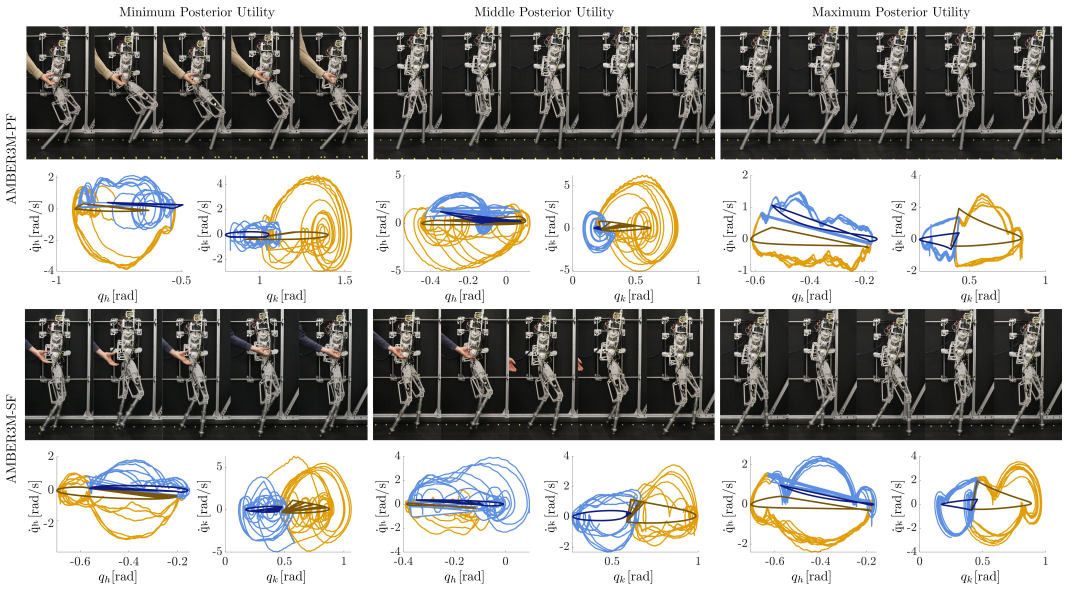


Figure 4.6: Gait tiles with increasing posterior utility values from left to right are shown for the the rigid model (top) and spring model (bottom). The phase portraits of the hip (q_h) and knee (q_k) of the stance leg (blue) and swing leg (yellow) are shown below each corresponding gait, plotted over 10 seconds of data. The phase portraits clearly indicate that for both AMBER3M-PF and AMBER3M-SF the gaits evolved to be more experimentally robust.

the resulting behavior is unclear, this can be prohibitively difficult. These difficulties are especially prevalent in the setting of bipedal locomotion, due to the extreme sensitivity of the stability of the system with respect to controller gains.

Recently, machine learning techniques have been implemented to alleviate the process of hand-tuning gains in a controller agnostic way by systematically navigating the entire parameter space [32, 111, 138]. More specifically, Bayesian optimization techniques have been applied to learning gait parameters and controller gains for various bipedal systems [43, 155]. However, these techniques rely on a carefully constructed predefined reward function. Furthermore, it is often the case that different desired properties of the robotic behavior are conflicting and therefore cannot be simultaneously optimized.

To alleviate the gain tuning process and enable the use of complex controllers for naïve users, this section explores the use of preference-based learning to systematically searching the controller parameter space and realize stable and robust experimental walking. Specifically, the general PBL methodology is demonstrated towards tuning gains of a CLF-QP$^+$ controller on the

AMBER bipedal robot, as well as an ID-CLF-QP$^+$ controller on the Cassie bipedal robot, requiring the learning framework to operate in a much higher-dimensional space.

Nominal Controller

It was shown in [10] that control Lyapunov functions (CLFs) are capable of stabilizing locomotion through the hybrid zero dynamics (HZD) framework, with [67] demonstrating how this can be implemented as a quadratic program (QP), allowing the problem to be solved in a pointwise-optimal fashion even in the face of feasibility constraints. However, achieving robust walking behavior on physical bipeds can be an arduous process due to complexities such as compliance, under-actuation, and narrow domains of attraction. One such controller that has recently demonstrated stable locomotion on the 22 degree of freedom (DOF) Cassie biped, as shown in Figure 4.2, is the ID-CLF-QP$^+$ [161]. A brief introduction of this controller is presented in this subsection.

Overall, the desired outputs were optimized using the FROST toolbox [87], where stability of the gait was ensured in the sense of Poincaré via HZD theory [220]. This was done first for AMBER, in which one walking gait was designed using a pinned model of the robot [7], and then on Cassie for 3D locomotion using the motion library found in [161] consisting of 171 walking gaits for speeds in 0.1 m/s intervals on a grid for sagittal speeds of $v_x \in [-0.6, 1.2]$ m/s and coronal speeds of $v_y \in [-0.4, 0.4]$ m/s.

CLFs, and specifically rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs), were introduced as methods for achieving (rapidly) exponential stability on walking robots [14]. This control approach has the benefit of yielding a control framework that can provably stabilize periodic orbits for hybrid system models of walking robots, and can be realized in a pointwise optimal fashion. In this section, we consider only outputs which are *vector relative degree* 2. Specifically, the *virtual constraints*, or outputs are of the form as in (2.16):

$$y_\alpha(x) = y_2^a(x) - y_2^d(\tau(x), \alpha) \tag{4.3}$$

with the goal of driving $y_\alpha \to 0$. Thus, differentiating (4.3) twice with respect to the dynamics results in:

$$\ddot{y}_\alpha(x) = L_f^2 y_\alpha(x) + L_g L_f y_\alpha(x) u,$$

where $L_f^2 y_\alpha(x) : \mathsf{T}\mathcal{Q} \to \mathbb{R}^{n_o}$ and $L_g L_f y(x) : \mathsf{T}\mathcal{Q} \to \mathbb{R}^{n_o}$ represent the Lie derivatives of the outputs with respect to the vector fields $f(x)$ and $g(x)$. Assuming that the system is feedback linearizeable, we can invert the decoupling matrix, $L_g L_f y(x)$, to construct a preliminary control input:

$$u = (L_g L_f y_\alpha(x))^{-1} \left(\nu - L_f^2 y_\alpha(x)\right), \tag{4.4}$$

which renders the output dynamics to be $\ddot{y}_\alpha = \nu$. With the auxiliary input ν appropriately chosen, the nonlinear system can be made exponentially stable. Assuming the preliminary controller (4.4) has been applied to our system, and defining $\eta = [y, \dot{y}]^\top$ we have the following output dynamics [103]:

$$\dot{\eta} = \underbrace{\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}}_{\mathbf{F}} \eta + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{\mathbf{G}} v. \tag{4.5}$$

With the goal of constructing a CLF using (4.5), we evaluate the continuous time algebraic Ricatti equation (CARE):

$$\mathbf{F}^\top \mathbf{P} + \mathbf{P}\mathbf{F} + \mathbf{P}\mathbf{G}\mathbf{R}^{-1}\mathbf{G}^\top \mathbf{P} + \mathbf{Q} = \mathbf{0}, \tag{CARE}$$

which has a solution $\mathbf{P} \succ 0$ for any $\mathbf{Q} = \mathbf{Q}^\top \succ 0$ and $\mathbf{R} = \mathbf{R}^\top \succ 0$. From the solution of (CARE), we can construct a rapidly exponentially stabilizing CLF (RES-CLF) [14]:

$$V(\eta) = \eta^\top I_\varepsilon \mathbf{P} I_\varepsilon \eta, \quad I_\varepsilon = \begin{bmatrix} \frac{1}{\varepsilon} I & 0 \\ 0 & I \end{bmatrix}, \tag{4.6}$$

where $0 < \varepsilon < 1$ is a tunable parameter that drives the (rapidly) exponential convergence. Any feedback controller, u, which can satisfy the convergence condition:

$$\dot{V}(\eta) = L_f V(\eta) + L_g V(\eta) u \leq -\frac{1}{\varepsilon} \underbrace{\frac{\lambda_{min}(\mathbf{Q})}{\lambda_{max}(\mathbf{P})}}_{\gamma} V(\eta), \tag{4.7}$$

will then render rapidly exponential stability for the output dynamics (4.3). To enforce (4.7), a quadratic program (CLF-QP) [67], with (4.7) as an inequality constraint can be posed.

Implementing this controller on physical systems, which are often subject to additional constraints such as torque bounds or friction limits, suggests that

relaxation for the inequality constraint should be used. The introduction of relaxation and the need to reduce torque chatter on physical hardware lead to the following relaxed (CLF-QP) with incentivized convergence in the cost [161]:

CLF-QP$^+$:

$$u^* = \underset{u\in\mathbb{R}^m}{\operatorname{argmin}} \quad \|L_f^2 y_\alpha(x) + L_g L_f y_\alpha(x) u\|^2 + w_{\dot{V}} \dot{V}(x,u) \tag{4.8}$$
$$\text{s.t.} \quad u_{min} \preceq u \preceq u_{max}$$

In order to avoid computationally expensive inversions of the model sensitive mass-inertia matrix, and to allow for a variety of costs and constraints to be implemented, a variant of the (CLF-QP) termed the (ID-CLF-QP) was introduced in [161]. This controller is used on the Cassie biped, with the decision variables $X = [\ddot{q}^\top, u^\top, \lambda_c^\top]^\top \in \mathbb{R}^{39}$:

ID-CLF-QP$^+$:

$$X^* = \underset{X\in\mathbb{X}_{ext}}{\operatorname{argmin}} \quad \|A(x)X - b(x)\|^2 + \dot{V}(q,\dot{q},\ddot{q}) \tag{4.9}$$
$$\text{s.t.} \quad D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_c(q)^\top \lambda_c$$
$$u_{min} \preceq u \preceq u_{max}$$
$$\lambda_c \in \mathcal{AC}(X) \tag{4.10}$$

---

where $\lambda_c \geq 0$ is a vector of constraint wrenches corresponding to the forces arising from the holonomic constraint $J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q} = 0$. Notably, the holonomic constraint has been moved into the cost with terms $A(x)$ and $b(x)$ as a weighted soft constraint, in addition to a feedback linearizing cost, and a regularization for the nominal X^* from the HZD optimization. Interested readers are referred to [161] for the full (ID-CLF-QP$^+$) formulation.

Parameterization of CLF-QP

As with the previous applications of preference-based learning, each action is defined as $\boldsymbol{a} = [a_1, ..., a_v] \in \mathcal{A} \subset \mathbb{R}^v$ for a $v-$dimensional parameter space. In this specific application, let $\mathbf{Q} = \mathbf{Q}(\boldsymbol{a})$, $\varepsilon = \varepsilon(\boldsymbol{a})$, and $w_{\dot{V}} = w_{\dot{V}}(\boldsymbol{a})$ denote a parameterization of our control tuning variables, which will subsequently be learned. Each gain a_i for $i = 1, \ldots, v$ is discretized into d_i values, leading to

Table 4.2: Action Space Definition: (ID-)CLF-QP$^+$ control parameter bounds.

CASSIE		
	Pos. Bounds	Vel. Bounds
$\mathbf{Q}$ Pelvis Roll (ϕ_x)	a_1:[2000, 12000]	a_7:[5, 200]
$\mathbf{Q}$ Pelvis Pitch (ϕ_y)	a_2:[2000, 12000]	a_8:[5, 200]
$\mathbf{Q}$ Stance Leg Length ($\|\phi^{st}\|_2$)	a_3:[4000, 15000]	a_9:[50, 500]
$\mathbf{Q}$ Swing Leg Length ($\|\phi^{sw}\|_2$)	a_4:[4000, 20000]	a_{10}:[50, 500]
$\mathbf{Q}$ Swing Leg Angle (θ^{sw}_{hp})	a_5:[1000, 10000]	a_{11}:[10, 200]
$\mathbf{Q}$ Swing Leg Roll (θ^{sw}_{hr})	a_6:[1000, 8000]	a_{12}:[5, 150]

AMBER				
	Pos. Bounds	Vel. Bounds		Bounds
$\mathbf{Q}$ Knees	a_1:[100, 1500]	a_3:[10, 300]	ε	a_5:[0.08, 0.2]
$\mathbf{Q}$ Hips	a_2:[100, 1500]	a_4:[10, 300]	$w_{\dot{V}}$	a_6:[1, 5]

an overall search space of actions given by the set $\mathcal{A}$ with cardinality $|\mathcal{A}| = \prod_{i=1}^{v} d_i$. For the AMBER robot, $v = 6$ with discretizations $d = [4, 4, 5, 5, 4, 5]$, resulting in the following parameterization:

$$\mathbf{Q}(\boldsymbol{a}) = \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & \mathbf{Q}_2 \end{bmatrix}, \quad \begin{aligned} \mathbf{Q}_1 &= \mathrm{diag}([a_1, a_2, a_2, a_1]), \\ \mathbf{Q}_2 &= \mathrm{diag}([a_3, a_4, a_4, a_3]), \end{aligned}$$
$$\varepsilon(\boldsymbol{a}) = a_5, \qquad w_{\dot{V}}(\boldsymbol{a}) = a_6,$$

which satisfies $\mathbf{Q}(\boldsymbol{a}) \succ 0$, $0 < \varepsilon(\boldsymbol{a}) < 1$, and $w_{\dot{V}}(\boldsymbol{a}) > 0$ for the choice of bounds, as summarized in Table 4.2. Because of the simplicity of AMBER, we were able to tune all associated gains for the CLF-QP$^+$ controller. For Cassie, however, the complexity of the ID-CLF-QP$^+$ controller warranted only a subset of parameters to be selected. Namely, $v = 12$ with $d_i = 8$, resulting in:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & \mathbf{Q}_2 \end{bmatrix}, \quad \begin{aligned} \mathbf{Q}_1 &= \mathrm{diag}([a_1, \ldots, a_{12}]), \\ \mathbf{Q}_2 &= \bar{\mathbf{Q}}, \end{aligned}$$

with $\bar{\mathbf{Q}}$, ε, and $w_{\dot{V}}$ remaining fixed and predetermined by a domain expert. From this definition of $\mathbf{Q}$, we can split our output coordinates $\eta = (\eta_t, \eta_{nt})$ into *tuned* and *not-tuned* components, where $\eta_t \in \mathbb{R}^{12}$ and $\eta_{nt} \in \mathbb{R}^6$ correspond to the $\mathbf{Q}_1$ and $\mathbf{Q}_2$ blocks in in $\mathbf{Q}$.

In this section, the LineCoSpar algorithm is leveraged, but with the addition of ordinal labels as an additional feedback mechanism to improve sample-

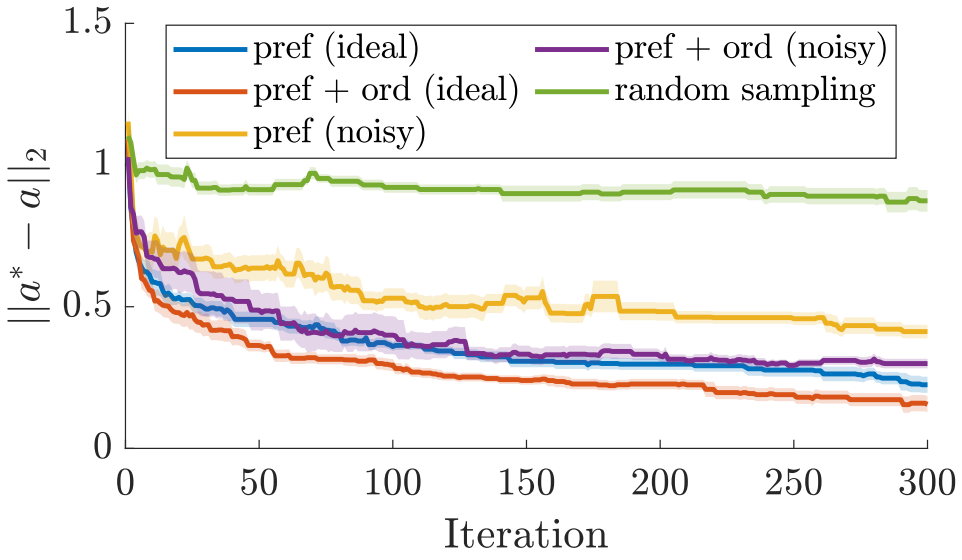


Figure 4.7: Simulated results averaged over 10 runs, demonstrating the capability of preference-based learning to optimize over large action spaces, specifically the one used for experiments with Cassie. Shaded region depicts standard error.

Table 4.3: (ID-)CLF-QP$^+$ control parameters identified by the learning framework as optimal.

AMBER	$[750, 100, 300, 100, 0.125, 2]$
Cassie	$[2400, 1700, 4200, 5600, 1700, 1200, 27, 40, 120, 56, 17, 7]$

efficiency. As mentioned earlier, this algorithm is aimed at regret minimization, with $r : \mathcal{A} \to \mathbb{R}$ denoting the underlying utility function of the human operator mapping each action to a subjective measure of "good," and $\boldsymbol{a}^*$ denoting the action maximizing r.

To demonstrate the learning, a simple example was constructed of the same dimensionality as the parameter space being investigated on Cassie ($v = 12, d = 8$), where the utility was modeled as $r(\boldsymbol{a}) = \|\boldsymbol{a} - \boldsymbol{a}^*\|_2$ for some $\boldsymbol{a}^*$. Feedback was automatically generated for both ideal noise-free feedback as well as for noisy feedback (correct feedback given with probability 0.9). The results of the simulated algorithm, illustrated in Figure 4.7, show that the learning framework quickly samples actions near $\boldsymbol{a}^*$, even for an action space as large as the one used in the experiments with Cassie. The simulated results also show that ordinal labels improve convergence, motivating their use in the final experiment.

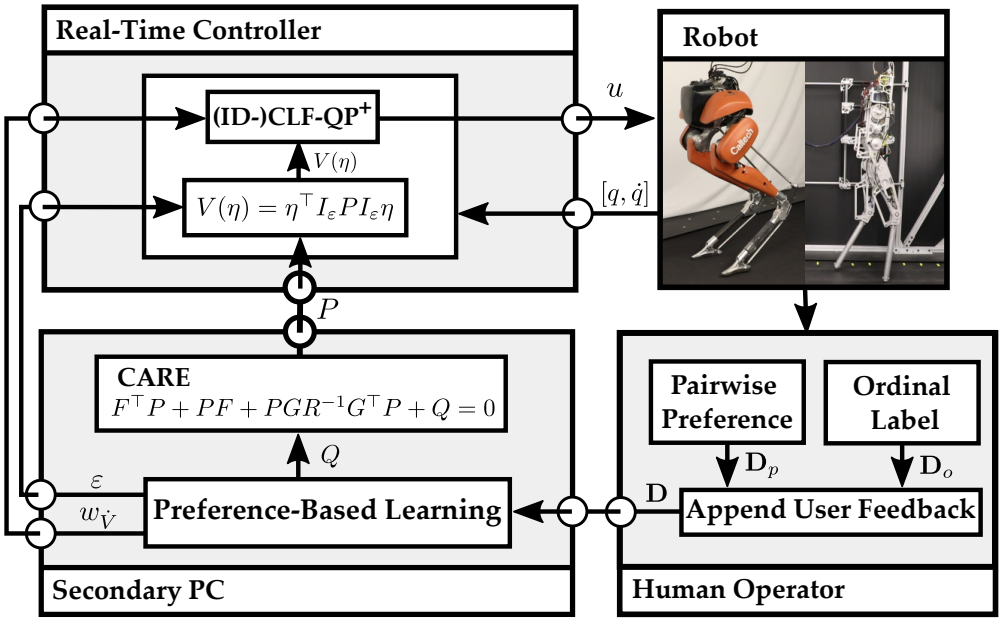
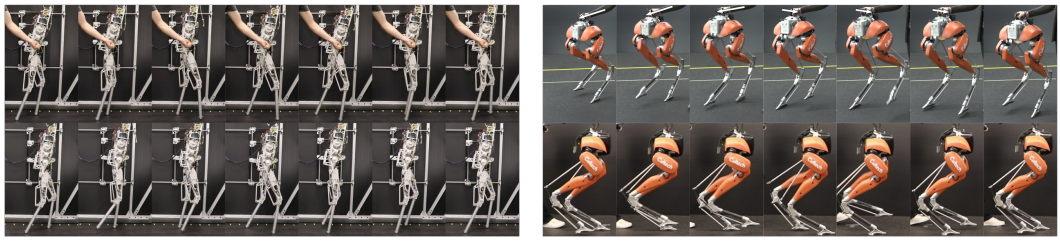


Figure 4.8: The experimental procedure, notably the communication between the controller, physical robot, human operator, and learning framework.

Experimental Results

Preference-based learning was applied to the realization of optimization-based control on two separate robotic platforms: the planar biped AMBER, and the 3D biped Cassie, as can be seen in the video [189]. As illustrated in Figure 4.8, the experimental procedure had four main components: the physical robot (either AMBER or Cassie), the controller running on a real-time PC, a human operator providing feedback, and a secondary PC running the learning algorithm. Each action was tested for approximately one minute, during which the behavior of the robot was evaluated in terms of both performance and robustness. User feedback in the form of pairwise preferences and ordinal labels was obtained after testing each action via the respective questions: "Do you prefer this behavior more or less than the last behavior?" and "Would you give this gait a label of very bad, neutral, or very good?". After user feedback was collected for the sampled controller gains, the posterior was inferred over all of the uniquely sampled actions, which took up to 0.5 seconds. The experiment with AMBER was conducted for 50 iterations, lasting one hour, and the experiment with Cassie was conducted for 100 iterations, lasting two hours. The duration of the experiments was scaled based on the size of the respective action spaces, and trials were terminated when satisfactory behaviors had been sampled.



(a) (b)

Figure 4.9: Gait tiles illustrating the experimental results of the preference-based learning framework towards tuning CLF-QP based controller gains using subjective operator feedback: (a) gaits corresponding to low (top) and high (bottom) underlying utilities on AMBER and (b) the gait identified as optimal as demonstrated by robustness testing (top) and good tracking (bottom) on Cassie.

Results with AMBER – CLF-QP$^+$. The CLF-QP$^+$ controller was implemented on an off-board i7-6700HQ CPU @ 2.6GHz with 16 GB RAM, which solved for desired torques and communicated them with the ELMO motor drivers on the AMBER robot at 2kHz. During the first half of the experiment, the algorithm sampled a variety of gains causing behavior ranging from instantaneous torque chatter to induced tripping due to inferior output tracking. It is important to note that none of the initial sampled values led to unassisted walking. By the end of the experiment however, the algorithm had sampled 3 gains which were deemed "very good," which resulted in stable walking. The final learned best actions found by the algorithm are reported in Table 4.3. Gait tiles for an action deemed "very bad," as well as the learned best action are shown in Figure 4.9a. Additionally, tracking performance for the two sets of gains is seen in Figure 4.10a, where the learned best action tracks the desired behavior to a better degree.

Results with Cassie – ID-CLF-QP$^+$. To test the capability of the learning method towards tuning more complex controllers, the preference-based learning method was applied for tuning the gains of the ID-CLF-QP$^+$ controller for the Cassie bipedal robot. The ID-CLF-QP$^+$ controller was implemented on the on-board Intel NUC computer, which was running a PREEMPT_RT kernel. The software runs on two ROS nodes, one of which communicate state information and joint torques over UDP to the Simulink Real-Time xPC, and one of which runs the controller. Each node is given a separate core on the CPU, and is elevated to real-time priority. Preference-based learning was run on an

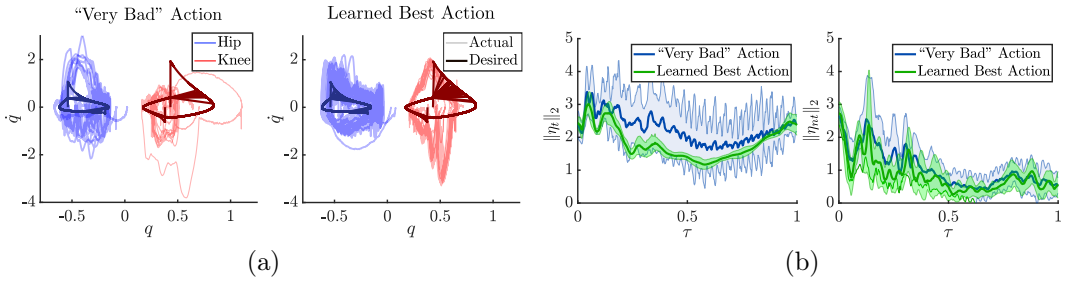


Figure 4.10: Comparison of the (ID-)CLF-QP$^+$ controller performance for gaits identified by the preference-based learning framework as corresponding to the lowest (denoted as a "Very Bad" Action) and highest (denoted as the "Learned Best Action") values of the posterior mean. Specifically, the comparisons are illustrated via (a) phase portraits for the AMBER experiments and (b) output Error of η_t (left) and η_{nt} (right) for the Cassie experiment.

external computer and was connected to the ROS master over WiFi. Actions were updated in real-time; once an action was selected, it was sent to Cassie via a rosservice call, where, upon receipt, the robot immediately updated the corresponding gains. As rosservice calls are blocking, multithreading their receipt and parsing was necessary in order to maintain real-time performance.

To demonstrate repeatability, the experiment was conducted twice on Cassie: once with a domain expert, and once with a naïve user. In both experiments, a subset of the Q matrix from (CARE) was tuned with coarse bounds given by a domain export, as reported in Table 4.2. These specific outputs were chosen because they were deemed to have a large impact on the performance of the controller. Some metrics used to determine preferences were the following: no torque chatter, no drift in the floating base frame, responsiveness to desired directional input, no violent impacts, no harsh noise, and naturalness of walking. At the start of the experiments, there was significant torque chatter and wandering, with the user having to regularly intervene to recenter the global frame. As the experiments continued, the walking noticeably improved. At the conclusion of 100 iterations, the posterior was inferred over all uniquely visited actions. The action corresponding with the maximum utility – believed by the algorithm to result in the most user preferred walking behavior – was further evaluated for tracking and robustness. In the end, this learned best action coincided with the walking behavior that the user preferred the most.

Features of this optimal action, compared to a worse action sampled in the beginning of the experiments, are outlined in Figure 4.10. In terms of quantifiable improvement, the difference in tracking performance is shown in Figure

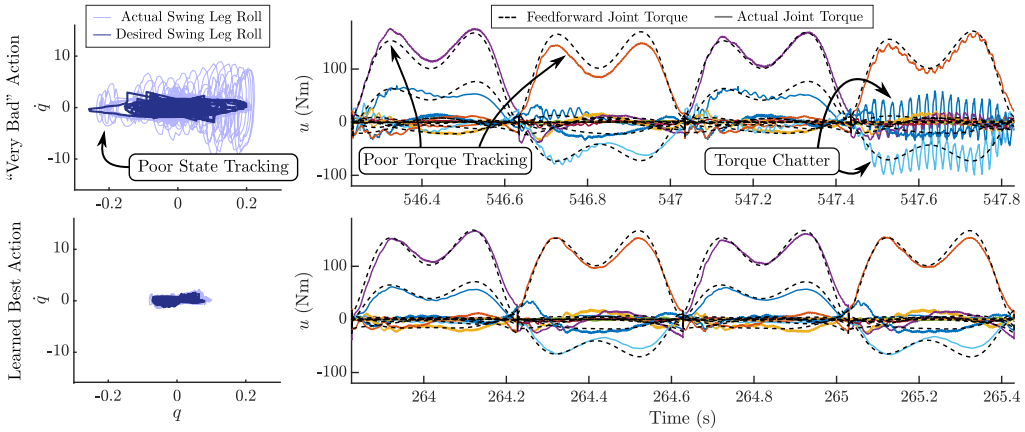


Figure 4.11: Phase plots and torques commanded by the ID-CLF-QP$^+$ in the naïve user experiments with Cassie. For torques, each colored line corresponds to a different joint, with the black dotted lines being the feedforward torque. The gains corresponding to a "very bad" action (top) yield torques that exhibit poor tracking on joints and torque chatter. On the other hand, the gains corresponding to the learned optimal action (bottom) exhibit much better tracking and no torque chatter.

4.10b. The magnitude of the tuned parameters, η_t, illustrates the improvement that preference-based learning attained in tracking the outputs it intended to. At the same time, the tracking error of the constant parameters, η_{nt}, shows that the outputs that were not tuned remained unaffected by the learning process. This quantifiable improvement is further illustrated by the commanded torques in Figure 4.11, which show that the optimal gains result in much less torque chatter and better tracking as compared to the other gains.

Limitations and Future Work

The main limitation of the current formulation of preference-based learning is that the action space must be predefined with set bounds. In the context of controller gains, these bounds are difficult to know *a priori* since the relationship between the gains and the resulting behavior is unpredictable. Future work to address this problem involves modifications to the learning framework to shift action space based on the user's preferences. Furthermore, the current framework limits the set of potential new actions to the set of actions discretized by d_i for each dimension i. As such, future work also includes adapting the granularity of the action space based on the uncertainty in specific regions.

4.3 PBL for User-Guided Gain Tuning of a Control Barrier Function

Modern robotics strives to develop safe and performant systems. Yet controllers designed to provide robust safety guarantees often result in conservative behavior, and tuning these controllers to find the ideal trade-off between performance and safety typically requires domain expertise or a carefully constructed reward function. Thus, the final application of preference-based learning explores how to systematically balance performance and robust safety by integrating *safety-aware* Preference-Based Learning (PBL) with Control Barrier Functions (CBFs). The overall procedure is illustrated in Figure 4.12.

For complex safety-critical systems, Control Barrier Functions (CBFs) have become a popular tool for the constructive synthesis of model-based controllers that endow nonlinear systems with rigorous guarantees of safety [15, 17, 96]. As these safety guarantees are susceptible to inaccuracies in the models of a system's dynamics, actuators, and sensors, approaches have been proposed to deal with model uncertainty [45, 198, 199, 217], disturbances [5, 50, 54, 107, 120, 167], and measurement errors [55, 61, 196]. These approaches can work well when deployed independently, but can be extremely conservative systems when used in conjunction. In practice, achieving performant behaviors with these methods is accomplished by conceding theoretical safety guarantees and tuning controller robustness parameters.

To reduce the burden on experts in controller tuning, we seek to incorporate the preference-based learning (PBL) framework into the design of safety-critical control systems. For applications with actions that may be classified as safe or unsafe, *safety-critical* PBL algorithms have been demonstrated to

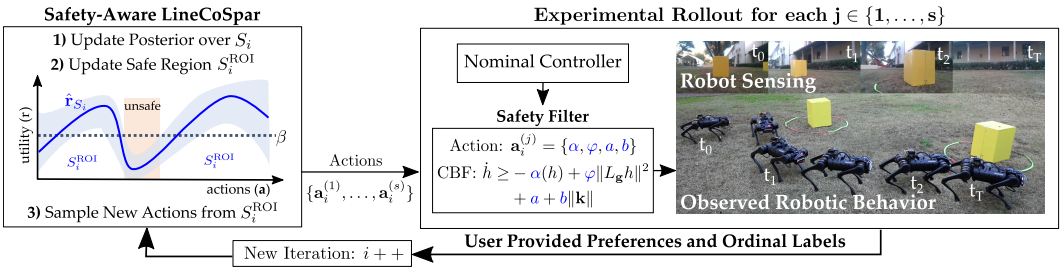


Figure 4.12: An overview of the Safety-Aware Preference-Based Learning design paradigm. Safety-Aware LineCoSpar is used to generate actions which are rolled out in experiments as parameters of the CBF-based safety filter to obtain user preferences and safety ordinal labels which are then used to update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

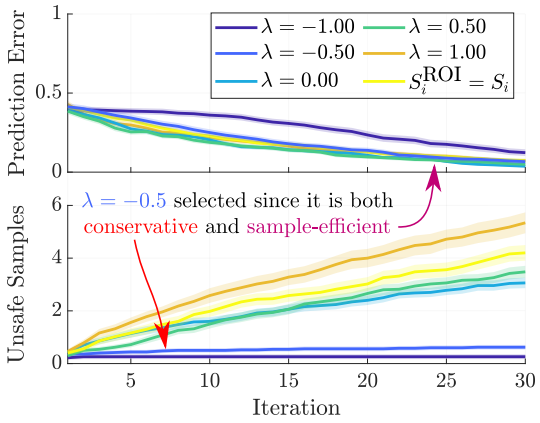


Figure 4.13: A comparison of SA-LineCoSpar and standard LineCoSpar on a synthetic utility function (drawn from the Gaussian prior) averaged over 50 runs with standard error shown by the shaded region. The safety-aware criteria reduces the number of sampled unsafe actions with a minimal effect on the prediction error, defined as $|\hat{\boldsymbol{a}}_i^* - \boldsymbol{a}^*|$ with $\hat{\boldsymbol{a}}_i^* \triangleq \text{argmax}_{\boldsymbol{a}}\, \hat{\boldsymbol{r}}_{S_i}$ and $\boldsymbol{a}^* \triangleq \text{argmax}_{\boldsymbol{a}}\, r(\boldsymbol{a})$.

prevent unsafe actions from being sampled [29, 184, 186]. However, these safety-critical algorithms require worst-case approximations which may cause performant and safe actions to be characterized as catastrophically unsafe. Thus, this section also formulates a *safety-aware* approach to PBL that generally avoids unsafe actions without being overly conservative.

Accounting for Safety in Learning Framework

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 [29, 184, 186] considered the setting where actions below a prespecified safety threshold are catastrophic and must be avoided at all cost. In this section, since the utilized controllers 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, the ROIAL algorithm is leveraged (presented earlier in Chapter 3), which uses ordinal labels to identify a *region of interest* (ROI) in $\mathcal{A}$. In this section, the ROI is defined to be the actions labeled as

Algorithm 5 SA-LineCoSpar

s uniform random actions ($\mathbf{V}_1 \subset \mathcal{A}$), corresponding feedback ($\mathbf{D}_1$),
for $i = 2, \dots, N$ **do**
    Update posterior over $\mathbf{V}_{i-1}$
    $\hat{\boldsymbol{a}}^*_{i-1} \leftarrow \text{argmax}_{\boldsymbol{a} \in \mathbf{V}_{i-1}} \hat{\boldsymbol{r}}_{\mathbf{V}_{i-1}}(\boldsymbol{a})$
    $\mathbf{L}_i \leftarrow$ New linear subspace intersecting $\hat{\boldsymbol{a}}^*_{i-1}$
    Construct subspace $\mathbf{S}_i = \mathbf{L}_i \cup \mathbf{V}_{i-1}$
    Update the model posterior over $\mathbf{S}_i$
    Determine region of interest $\mathbf{S}_i^{\text{ROI}}$
    for $j = 1, \dots, s$ **do**
        $r^{(j)} \sim \mathcal{N}(\hat{\boldsymbol{r}}_{\mathbf{S}_i}, \boldsymbol{\Sigma}_{\mathbf{S}_i})$
        $\boldsymbol{a}_i^{(j)} \leftarrow \text{argmax}_{\boldsymbol{a} \in \mathbf{S}_i^{\text{ROI}}} r^{(j)}$
    end for
    Deploy $\{\boldsymbol{a}_i^{(1)}, \dots \boldsymbol{a}_i^{(s)}\}$ on system
    $\mathbf{V}_i \leftarrow \mathbf{V}_{i-1} \cup \{\boldsymbol{a}_i^{(1)}, \dots \boldsymbol{a}_i^{(s)}\}$
    $\mathbf{D}_i \leftarrow \mathbf{D}_{i-1} \cup$ new prefs. $\cup$ new ord. labels
end for

"safe." In each iteration i we estimate an ROI within the set $\mathbf{S}_i$ as:

$$\mathbf{S}_i^{\text{ROI}} = \{\boldsymbol{a} \in \mathbf{S}_i \mid \hat{\boldsymbol{r}}_{\mathbf{S}_i}(\boldsymbol{a}) + \lambda \boldsymbol{\sigma}_{\mathbf{S}_i}(\boldsymbol{a}) > \beta_{\text{ROI}}\}, \tag{4.11}$$

where $\hat{\boldsymbol{r}}_{\mathbf{S}_i}(\boldsymbol{a})$ and $\boldsymbol{\sigma}_{\mathbf{S}_i}(\boldsymbol{a})$ are the posterior mean and standard deviation, respectively, evaluated at the action $\boldsymbol{a} \in \mathbf{S}_i$. As before, 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 $\mathbf{S}_i^{\text{ROI}}$ is added to LineCoSpar by only considering actions in $\mathbf{S}_i^{\text{ROI}}$ during Thompson sampling. We refer to this as Safety-Aware LineCoSpar (SA-LineCoSpar), with the full algorithm outlined in Alg. 5.

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-LineCoSparto 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:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{v} + \mathbf{d}(t)), \tag{4.12}$$

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:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{k}(\mathbf{x}) + \mathbf{d}(t)). \tag{4.13}$$

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 $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, \boldsymbol{\rho}) \geq 0\}, \tag{4.14}$$

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 (4.13) satisfies $\mathbf{x}(t) \in \mathcal{C}$ for all $t \geq 0$. The system (4.13) 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 11 (Control Barrier Functions (CBF) [15])**.** The function h is a Control Barrier Function (CBF) for (4.12) on $\mathcal{C}$ if there exists $\alpha \in \mathcal{K}_\infty^{\mathrm{e}}$ [3] such that for all $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{v} \in \mathbb{R}^m} \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, \boldsymbol{\rho})\mathbf{f}(\mathbf{x})}_{L_{\mathbf{f}} h(\mathbf{x}, \boldsymbol{\rho})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, \boldsymbol{\rho})\mathbf{g}(\mathbf{x})}_{L_{\mathbf{g}} h(\mathbf{x}, \boldsymbol{\rho})} \mathbf{v} > -\alpha(h(\mathbf{x}, \boldsymbol{\rho})). \tag{4.15}$$

While it may be possible to synthesize controllers that render a given set $\mathcal{C}$ safe in the presence of disturbances [107], this may result in overly-conservative behavior. Instead, we consider how safety properties degrade with disturbances via the following definition.

[3] 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 $\lim_{r \to -\infty} \alpha(r) = -\infty$.

Definition 12 (Input-to-State Safety [120])**.** The system (4.13) 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:

$$\mathcal{C}_\delta = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, \boldsymbol{\rho}) \geq -\gamma(\delta)\}, \tag{4.16}$$

is forward invariant. The function h is an Input-to-State Safe Control Barrier Function (ISSf-CBF) for (4.12) 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 $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{v} \in \mathbb{R}^m} L_{\mathbf{f}} h(\mathbf{x}, \boldsymbol{\rho}) + L_{\mathbf{g}} h(\mathbf{x}, \boldsymbol{\rho})\mathbf{v} - \varphi \|L_{\mathbf{g}} h(\mathbf{x}, \boldsymbol{\rho})\|^2 > -\alpha(h(\mathbf{x}, \boldsymbol{\rho})). \tag{4.17}$$

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 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 13 (Measurement-Robust Control Barrier Functions [61])**.** The function h is a *Measurement-Robust Control Barrier Function* (MR-CBF) for (4.12) 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 $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{v} \in \mathbb{R}^m} L_{\mathbf{f}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}}) + L_{\mathbf{g}} h(\mathbf{x}, \widehat{\boldsymbol{\rho}})\mathbf{v} - a - b\|\mathbf{v}\| > -\alpha(h(\mathbf{x}, \widehat{\boldsymbol{\rho}})). \tag{4.18}$$

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

Theorem 8. *Consider the set $\mathcal{C}$ defined in (4.14).*

1. *If h is a CBF for (4.12) 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 (4.13) is safe with respect to $\mathcal{C}$.*
2. *If h is an ISSf-CBF for (4.12) on $\mathcal{C}$ with parameter φ and $\widehat{\boldsymbol{\rho}} = \boldsymbol{\rho}$, then there exists a controller $\mathbf{k}$ such that (4.13) 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 $\|\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}\| \leq \varepsilon$ for some $\varepsilon \in \mathbb{R}_{\geq 0}$. Then there exists $\underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ such that if h is an MR-CBF for (4.12) 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 (4.13) is safe with respect to $\mathcal{C}$.*

In particular, consider the following cascaded nonlinear control-affine system resulting as a modification of (4.12):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\boldsymbol{\kappa}(\boldsymbol{\xi}), \qquad \dot{\boldsymbol{\xi}} = \mathbf{f}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}) + \mathbf{g}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi})\mathbf{u}, \tag{4.19}$$

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 (4.12) 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 (4.14), 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 $\mu \in \mathbb{R}_{\geq 0}$ such that for any continuous, bounded signal $\mathbf{s} : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$, the closed-loop system:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}) + \mathbf{g}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi})\boldsymbol{\pi}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{s}(t)), \tag{4.20}$$

satisfies the following implication:

$$\|\boldsymbol{\kappa}(\boldsymbol{\xi}(0)) - \mathbf{s}(0)\| \leq \mu \implies \|\boldsymbol{\kappa}(\boldsymbol{\xi}(t)) - \mathbf{s}(t)\| \leq \mu, \quad t \in \mathbb{R}_{\geq 0}. \tag{4.21}$$

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 (4.12) and $\|\boldsymbol{\kappa}(\boldsymbol{\xi}(0)) - \mathbf{k}(\mathbf{x}(0))\| \leq \mu$, then we have that the controller $\boldsymbol{\pi}$ ensures $\|\boldsymbol{\kappa}(\boldsymbol{\xi}(t)) - \mathbf{k}(\mathbf{x}(t))\| \leq$

μ for $t \in \mathbb{R}_{\geq 0}$. With this assumption in mind, we may study the ISSf behavior of the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{k}(\mathbf{x}) + \mathbf{d}(t)), \qquad \dot{\boldsymbol{\xi}} = \mathbf{f}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}) + \mathbf{g}_{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi})\boldsymbol{\pi}(\mathbf{x}, \boldsymbol{\xi}, \mathbf{k}(\mathbf{x})), \quad (4.22)$$

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

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 (4.22). This is formalized in the following theorem:

Theorem 9. *Given the set $\mathcal{C}$ defined in (4.14), 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 $\|\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}\| \leq \varepsilon$ for some $\varepsilon \in \mathbb{R}_{\geq 0}$. Then there exists $\underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ such that if h satisfies:*

$$\sup_{\mathbf{v} \in \mathbb{R}^m} L_{\mathbf{f}}h(\mathbf{x}, \widehat{\boldsymbol{\rho}}) + L_{\mathbf{g}}h(\mathbf{x}, \widehat{\boldsymbol{\rho}})\mathbf{v} - \varphi\|L_{\mathbf{g}}h(\mathbf{x}, \widehat{\boldsymbol{\rho}})\|^2 - a - b\|\mathbf{v}\| > -\alpha(h(\mathbf{x}, \widehat{\boldsymbol{\rho}})), \quad (4.23)$$

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 (4.22) 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[4]. As in [79], (4.23) 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.

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

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 [74].

[4]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 4.4: 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.

Additionally, this safety filter is a Second-Order Cone Program (SOCP) [37] for which an array of solvers exist including ECOS [63].

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 9 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 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: [195]). The action space $\mathcal{A}$ and learning hyperparameters are defined in Table 4.4. A unicycle model was used as the simplified model (4.12) with the nominal

controller $\mathbf{k}_{\text{nom}}$:

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\cos\phi & 0\\ \sin\phi & 0\\ 0 & 1\end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \left(\underbrace{\begin{bmatrix}v\\ \omega\end{bmatrix}}_{\mathbf{v}} + \mathbf{d}(t) \right), \quad \mathbf{k}_{\text{nom}}(\mathbf{x}) = \begin{bmatrix} K_v d_{\text{g}} + C \\ -K_\omega(\sin\phi - (y_{\text{g}} - y)/d_g) \end{bmatrix}, \tag{4.24}$$

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:

$$h(\mathbf{x}, \boldsymbol{\rho}_i) = d_{\text{obs},i} - r_{\text{obs}} - \zeta\cos(\phi - \theta_i), \tag{4.25}$$

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 (4.24). In practice, infeasibilities 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 [39], which realizes a stable walking gait for (4.22) 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. SA-LineCoSpar

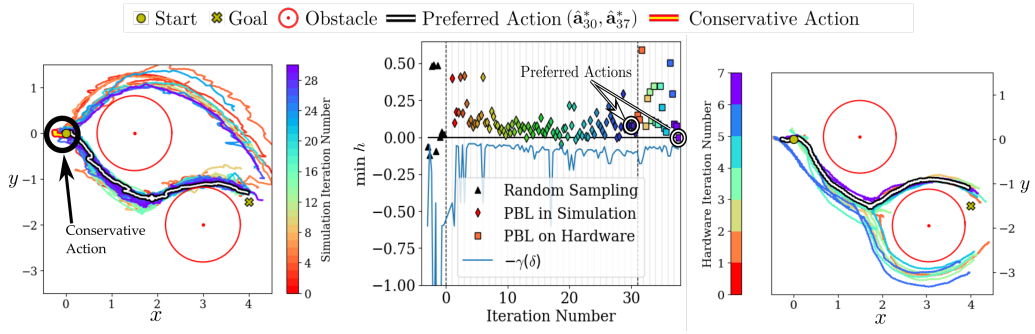


Figure 4.14: Illustration of the robotic behavior throughout the learning process. (Left) Actions sampled during simulation in 30 iterations with 3 new actions in each iteration. The preferred action, $\hat{\boldsymbol{a}}_{30} = (3, 0.6, 0.5, 0.015)$, is shown in black and white. A conservative action, $\boldsymbol{a} = (2, 0.5, 0.0651, 0.485)$, is indicated by the black circle, where a and b were determined by estimating the Lipschitz coefficients present in the proof of Theorem 9. 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.

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

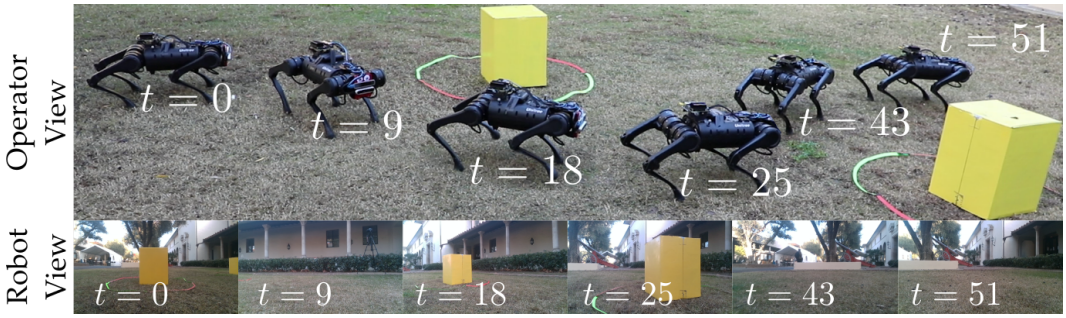


Figure 4.15: The preferred action, $\hat{\mathbf{a}}^*_{40} = (5, 0.1, 0.4, 0.02)$, after simulation, indoor experiments, and 3 additional iterations of 3 actions in an outdoor environment is shown alongside views from the onboard camera.

be seen in the supplementary video [195].

4.4 Summary

In this chapter, the preference-based learning framework was demonstrated towards three extending robotic applications. First, it was applied to tuning constraint bounds directly in the HZD gait generation process, resulting in stable, robust, and efficient walking on the AMBER-3M robot with both rigid and compliant point feet. Second, it was applied towards tuning control parameters of an ID-CLF-QP$^+$ controller, which yielded performant locomotion on the AMBER-3M biped and the 22-DOF Cassie platform. Lastly, the learning framework was demonstrated towards systematically selecting parameters of a control barrier function to yield both performant and safe locomotion on the Unitree A1 quadruped.

Chapter 5

CLINICAL CONSIDERATIONS FOR LOWER-BODY ASSISTIVE DEVICES

The final chapter of my thesis explores three clinical considerations for lower-body assistive devices: 1) How can variable assistance be achieved on lower-body exoskeletons (such as the Atalante exoskeleton) for subjects with motor incomplete paraplegia and how does this mode of control influence the users associated metabolic expenditure; 2) How can we study potential latent factors underlying user-preferred exoskeleton locomotion; and 3) How can we utilize musculoskeletal models in the gait generation process to systematically generate motions that naturally align with that of humans. The goal of addressing these questions is to better understand the clinical benefits associated with robotic assistive devices and their associated control methods.

These considerations are each individually motivated. First, towards variable assistance, existing work has shown that providing some freedom of movement around a nominal gait, instead of rigidly following it, accelerates the spinal learning process of people with a walking impediment when using a lower body exoskeleton. Moreover, since physical deconditioning is a common symptom of SCI, many SCI rehabilitation techniques are aimed at facilitating physical exercise [105]. As such, an important clinical consideration that has yet to be sufficiently explored is how variable assistance and their corresponding controllers influence metabolic expenditure. Second, as discussed in Chapter 3, better understanding the factors underlying user-preferred exoskeleton locomotion would reduce the burden on user-customization. Instead, prior information could be leveraged to predict exoskeleton gaits that are comfortable for new users. Thus, in the second section of this chapter we explore how to study potential latent factors underlying user-preferred locomotion. Lastly, there exists prior work towards leveraging musculoskeletal models to synthesize natural walking motions [70]. However, these techniques are often heuristic, so instead we explore how to integrate muscle model constraints directly in the gait generation optimization problem to systematically synthesize stable and natural walking gaits.

5.1 Studying Metabolic Expenditure during Variable Assistance

While the full assistance approach to dynamically stable exoskeleton locomotion (as presented in Chapter 2) enables crutch-less exoskeleton walking, it is no longer optimal when exoskeleton technology is extended to patients who are recovering muscle functionality. For patients who are trying to strengthen recovering muscles, partial assistance would be more appropriate than full assistance. A previous study showed that permitting partial assistance and variability during step training enhanced stepping recovery after a complete spinal cord transection in adult mice [42]. The study also hypothesized that a fixed trajectory training strategy would drive the spinal circuitry toward a state of learned helplessness. These "assist-as-needed" algorithms, which have also been explored in other publications [41, 180, 235], utilize velocity field control to provide gentle guidance at a constant rate towards the desired walking trajectory.

In [80], a novel framework was presented for variable assistance on lower body exoskeletons, based upon safety-critical control methods. This method leverages tools from controlled set invariance [22, 35]—in particular, control barrier functions [16, 17]—to enable assist-as-needed strategies while guaranteeing coherence of the walking pattern. The method allows users to control their own motions when they are performing well (i.e., staying in a tube around a nominal trajectory) but intervene when they are not, so as to maintain a functional walking pattern. This approach, therefore, takes motivation from the growing area of safety-critical control [2, 17, 216], and extends its application to exoskeletons with experimental demonstration with multiple subjects.

While the details of the variable assistance method are omitted in this thesis, the experimental results demonstrating how variable assistance influences

Figure 5.1: Photos of the eight non-disabled subjects who participated in the experimental evaluation.

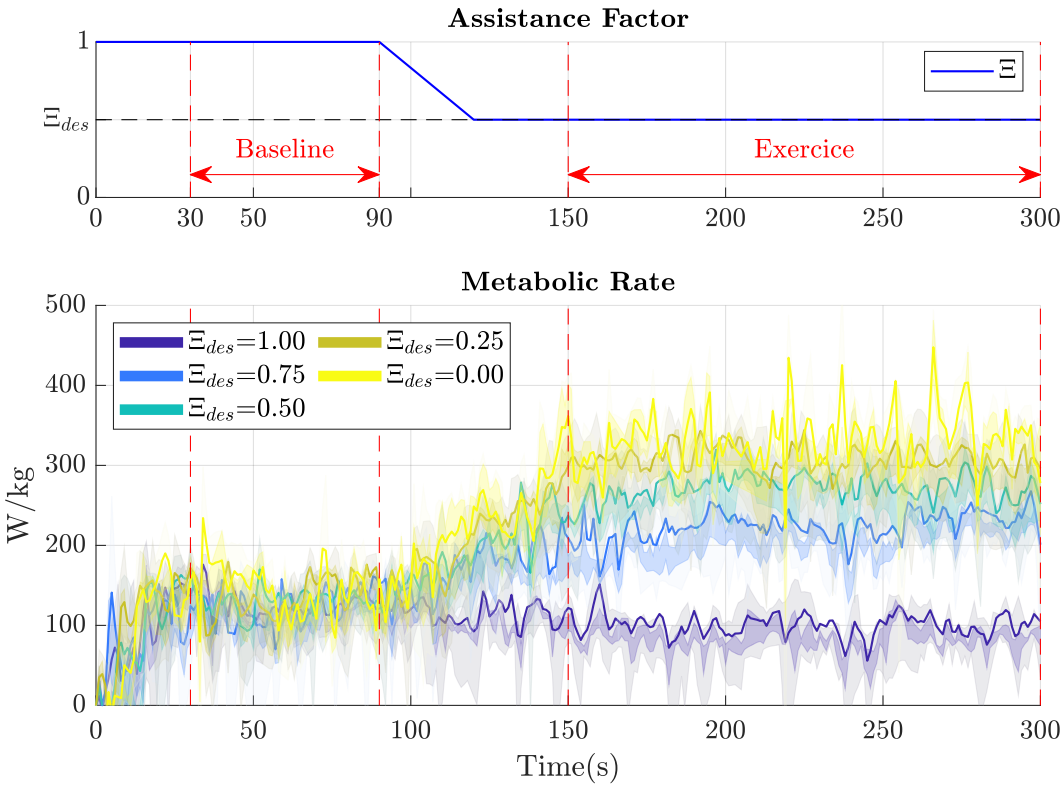


Figure 5.2: Metabolic expenditure experimental setup. (Top) Variable assistance testing procedure. (Bottom) Metabolic rates as aligned with the testing procedure.

metabolic expenditure for exoskeleton users is presented. Specifically, the variable assistance controller was demonstrated in three separate experiments. First, the entire framework was tested with eight non-disabled human subjects of masses and heights from 58kg to 91kg and 1.62m to 1.93m (shown in Figure 5.1). Then, the framework was tested over a larger set of assistance factors for a single subject.

Experimental Results Across Eight Subjects

The experimental testing conducted for non-disabled subjects consisted of walking trials lasting five minutes each. The format of each trial is shown in Figure 5.2 and is as follows. First, 90 seconds of walking with full assistance, then 30s of transitioning to the desired level of assistance and finally 180s of walking at that desired assistance factor. "Full Assistance" corresponds to an assistance factor $\Xi = 1$, which is equivalent to the baseline controller without the variable assistance controller. "Partial Assistance" corresponds to $\Xi = 0.5$, i.e., $q_{bound} = \pm 4$deg where q_{bound} denotes the range of motion allowed in the

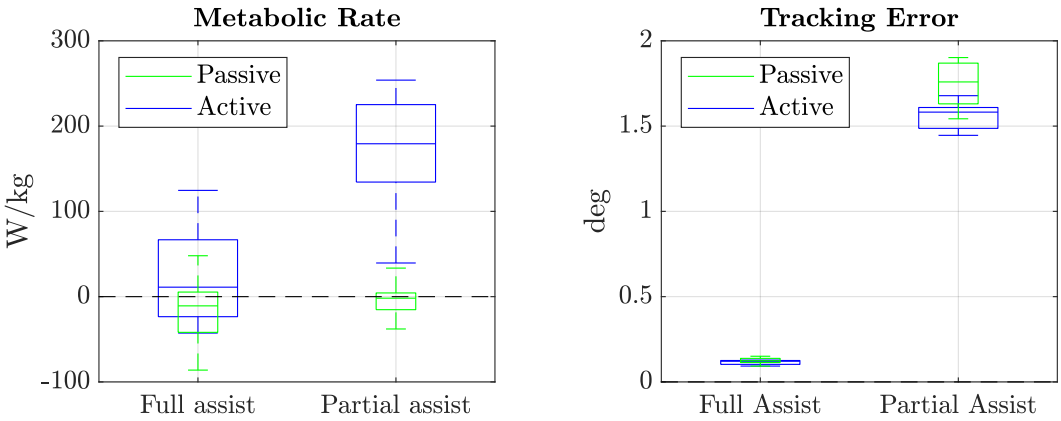


Figure 5.3: Comparison between tracking accuracy and subject power consumption. The passive data correspond to the subject not doing anything. The active data correspond to the subject trying to follow the nominal gait. Full assist correspond to nominal PID control around the gait, whereas partial assist corresponds to ±4deg wide virtual guides.

hip and knee joints (around the nominal gait trajectory) of the lower-body exoskeleton before the variable assistance controller takes over the motion of the exoskeleton. Beside the subject model parameters, the gait parameters were the same for all subjects. The step length and duration were chosen to be 0.16m and 0.8s, respectively.

In order to demonstrate the effectiveness of the variable assistance framework, four trials were conducted per subject. The first two trials were one with Full Assistance and one with Partial Assistance, where the subjects were asked to be completely passive and let the exoskeleton do the work. The same two trials were then repeated but this time asking the subjects to: "Do whatever feels necessary to track the nominal gait."

Human metabolic expenditure was recorded for all subjects as it provides critical insight into how much effort the user is exerting. The metabolic rate was determined from oxygen and carbon dioxide exchange rates as measured by a COSMED K4b2. The exchange rates were converted to a metabolic rate using the equation developed by Brockway et al. [38]. When calculating the metabolic rate, the average metabolic rate recorded over the baseline part of every trial was subtracted from the average rate of the exercise part to isolate the part of the total metabolic power used for compensating for the varying levels of assistance.

The results for all eight subjects are summarized in Figure 5.3. This figure

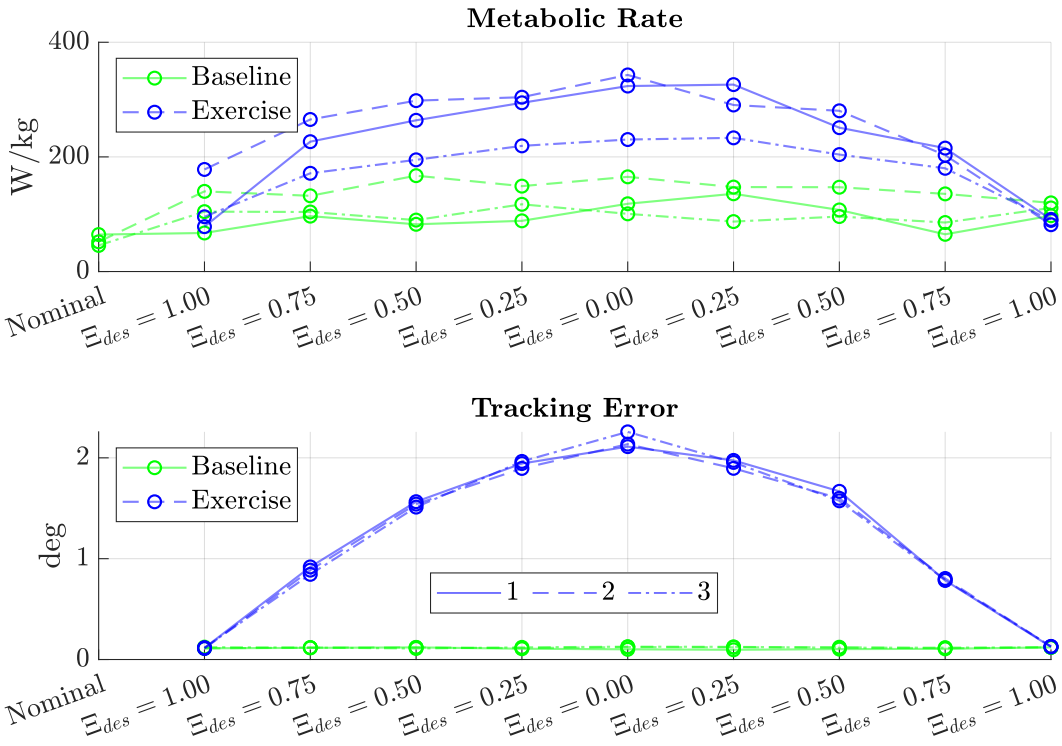


Figure 5.4: Raw metabolic rate and tracking error in chronological order for the baseline and exercise segments as defined in Figure 5.2. The step length and duration, respectively, are 0.16m and 0.8s.

shows that when the subject was passive, the metabolic rate remained consistent between full assistance and partial assistance. The metabolic rate when passive also is consistently lower than the metabolic rate of the subjects when active at partial assistance. An interesting observation is that the metabolic rate of the subjects when active at full assistance is not much different from that of the subjects when passive. This suggests that the subjects do not feel the need to provide more energy than necessary when the exoskeleton is already providing full assistance. On the other hand, partial assistance incentivises users to contribute to the tracking of the gait which translates into an increase in metabolic rate as expected. Finally, note that on average, the subjects were able to improve the accuracy of tracking in Partial Assist when actively trying.

Experimental Results of Varying the Assistance Factor for One Subject

The testing procedure for the final experiment was the same as discussed previously and shown in Figure 5.2 but was repeated with a larger set of

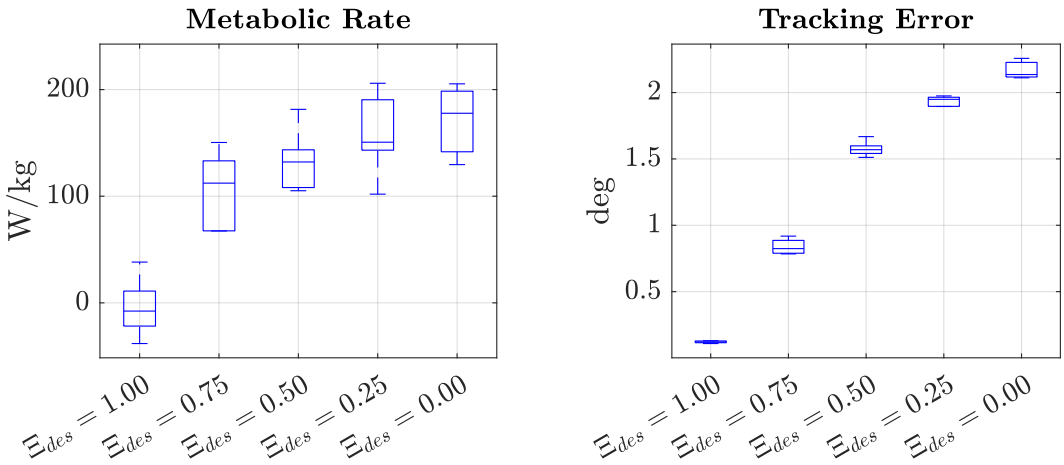


Figure 5.5: Comparison between tracking accuracy and subject normalized power consumption.

assistance factors. The trials were done, in order, with assistance factors $\Xi \in \{1.0, 0.75, 0.5, 0.25, 0.0, 0.25, 0.5, 0.75, 1.0\}$. A five minute break was taken in between each trial to let the subject return to a resting metabolic rate. The subject also completed one five minute trial while walking on the treadmill at the same velocity as during the trials to compare the subject's nominal walking metabolic rate with that of the exoskeleton testing. This entire procedure was repeated on three consecutive days with the same subject.

The metabolic power consumption as well as the average tracking error for each segment is reported in Figure 5.4. The subject's average resting oxygen and carbon dioxide exchange rates, measured at the start of testing, are subtracted from the recorded exchange rates of each trial. Interestingly, it can be seen that the baseline metabolic rate is relatively consistent between all trials and that the data is symmetric around the 0.00 assistance factor trial. This confirms that the increase in exercise metabolic rate for lower assistance factors is due to the lowered assistance and not exhaustion of the subject.

Figure 5.5 presents the metabolic rates of the exercise part normalized by the baseline ones for the different values of assistance factor, as well as the corresponding tracking errors. These normalized values indicate a clear trend: *The normalized metabolic rate and the normalized tracking error increase as the assistance factor decreases.*

5.2 Analysis of User-Preferred Exoskeleton Locomotion

In addition to optimizing exoskeleton walking gaits for individual user comfort, as discussed in Chapter 3, a clinical consideration towards lower-body exoskeleton locomotion is to better understand the utility functions underlying human preferences. As such, my work aims to derive insight from obtained user preferences and apply this knowledge towards improving gait synthesis.

As discussed throughout the thesis, the cost function $\Phi(X)$ of the gait generation optimization problem largely influences the behavior of the walking gaits that it generates; however, the user's cost function Φ_{human} underlying her preferences is poorly-understood. This section aims to describe the relationship between gaits and user preferences through the underlying cost function Φ_{human}, so that future gait synthesis can be streamlined towards user-preferred walking. Thus, we aim to identify key terms in Φ_{human} that numerically account for the preferences captured during the preference-based learning procedure conducted in [206].

In the experiments conducted in [206], all walking gaits that were executed on the exoskeleton were flat-footed. Thus, by analyzing the center of mass (CoM) and center of pressure (CoP), the patient-exoskeleton system can be approximated by a Linear Inverted Pendulum Model (LIPM). This allows us to analyze the underlying utility function Φ_{human} using the cost structure from [182]. However, before discussing the structure of Φ_{human}, we will first introduce the concepts of Zero Moment Point (ZMP) and LIPM.

Zero Moment Point. The Zero Moment Point (ZMP) is a widely-used notion of stability for bipedal robots that is defined as *the point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes* [213]. When the ZMP exists outside of the "support polygon," i.e., the convex hull of the stance foot (or stance feet in the double-support domain), the robot experiences foot roll.

Static and Dynamic Stability. For a full discussion, refer to pg. 7 of [221]. In general, static stability is the condition in which the CoM and CoP never leave the support polygon. In contrast, quasi-static stability relaxes this condition on the CoM and only requires that the CoP remains inside the support polygon. For dynamic stability, the CoP lies on the boundary of the support polygon for a portion of the gait.

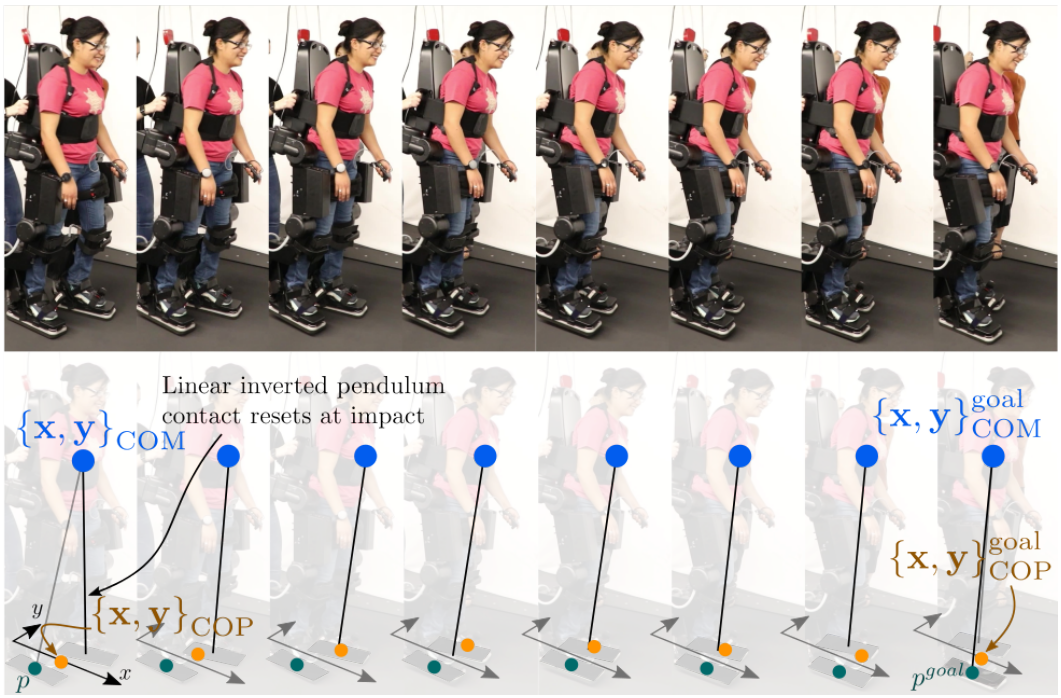


Figure 5.6: Illustration of a single step with the overlayed LIPM model.

Linear Inverted Pendulum Model (LIPM). The LIPM is a low-dimensional dynamical system for reduced-order gait generation. The LIPM model assumes constant height of the center of mass, as well as zero angular momentum. The dynamics of the LIPM [112] are:

$$m\ddot{x}_{\text{CoM}} = \frac{mg}{z_0}(x_{\text{CoM}} - x_{\text{CoP}}), \tag{5.1}$$

$$m\ddot{y}_{\text{CoM}} = \frac{mg}{z_0}(y_{\text{CoM}} - y_{\text{CoP}}), \tag{5.2}$$

where $\{x, y\}_{\text{CoM}}$ are the x and y positions of the CoM at constant height z_0, and $\{x, y\}_{\text{CoP}}$ denote the x and y positions of the CoP. For planar horizontal ground walking, the ZMP is mathematically equivalent to the CoP. The CoP was experimentally obtained using the four 3-axis force sensors on the bottom of the exoskeleton's feet.

Fitting the LIPM Cost Function to User Preferences

Since flat-foot level-ground walking is well captured by the Linear Inverted Pendulum Model (LIPM), the cost function used in the LIPM to generate desirable walking behavior may explain the users' utility functions underlying their exoskeleton gait preferences. As defined in [182], the LIPM cost function

is:

$$\begin{aligned}\Phi_{\text{LIPM}} = w_1||x^{goal}_{\text{CoM}} - x_{\text{CoM}}||^2 + w_2||\dot{x}_{\text{CoM}}||^2 + \\ w_3||\dot{x}_{\text{CoP}}||^2 + w_4||p^{\text{goal}}_x - p_x||^2 + \\ w_1||y^{\text{goal}}_{\text{CoM}} - y_{\text{CoM}}||^2 + w_2||\dot{y}_{\text{CoM}}||^2 + \\ w_3||\dot{y}_{\text{CoP}}||^2 + w_4||p^{\text{goal}}_y - p_y||^2, \end{aligned} \tag{5.3}$$

where $\{x, y\}^{\text{goal}}_{\text{CoM}}$ denotes the CoM goal position in the x and the y directions, $\{\dot{x}, \dot{y}\}_{\text{CoP}}$ denotes the velocity of the CoP in the x and y directions, $\{\dot{x}, \dot{y}\}_{\text{CoM}}$ is the velocity of the CoM, $p^{\text{goal}}_{\{x,y\}}$ denotes the next stance foot position in the x and y directions, and $p_{\{x,y\}}$ denotes the x and y positions of the swing foot (Figure 5.6).

We hypothesize that $\Phi_{\text{human}}(w)$ can be captured as a function of the weights $w := \{w_i\}, i \in \{1, \ldots, 4\}$. Therefore, we fit the weights w of Φ_{LIPM} to the validation-stage preference data collected in [206], i.e., the preferences between the most-preferred gaits (gaits with parameters $\boldsymbol{a}_{\max}$) and each of the random gaits presented during the validation phase[1]. The weights w were optimized via the quadratic program:

$$\begin{aligned} w* = &\underset{w}{\text{argmin}} \;\; ||w|| \\ \text{s.t.} &\begin{bmatrix} \delta^{(1)}_1 & \delta^{(1)}_2 & \delta^{(1)}_3 & \delta^{(1)}_4 \\ & & \vdots & \\ \delta^{(n)}_1 & \delta^{(n)}_2 & \delta^{(n)}_3 & \delta^{(n)}_4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} < 0, \end{aligned} \tag{5.4}$$

where n denotes the number of pairwise preferences, and:

$$\delta_i = \left(||x^{\text{pref}}_{(i,x)}||^2 + ||x^{\text{pref}}_{(i,y)}||^2\right) - \left(||x^{\text{not pref}}_{(i,x)}||^2 + ||x^{\text{not pref}}_{(i,y)}||^2\right) \tag{5.5}$$

$$\begin{aligned} x_{(1,x)} &= x^{goal}_{\text{CoM}} - x_{\text{CoM}} & x_{(1,y)} &= y^{goal}_{\text{CoM}} - y_{\text{CoM}} \\ x_{(2,x)} &= \dot{x}_{\text{CoM}} & x_{(2,y)} &= \dot{y}_{\text{CoM}} \\ x_{(3,x)} &= \dot{x}_{\text{CoP}} & x_{(3,y)} &= \dot{y}_{\text{CoP}} \\ x_{(4,x)} &= p^{goal}_x - p_x & x_{(4,y)} &= p^{\text{goal}}_y - p_y. \end{aligned}$$

We use subject-wise holdout (leave-one-out) cross-validation across the subjects to verify the reliability of the fit. The average weights across all six holdout fits are: $w_1 = -0.1266$, $w_2 = 0.1363$, $w_3 = -0.0944$, and $w_4 = 1.0662$.

[1]Cost function fitting and CoP/CoM plotting code can be found at:
https://github.com/myracheng/linecospar/tree/master/gaitAnalysis

Table 5.1: Predictive power of each cost function term on the obtained exoskeleton user preferences.

Cost Function	**Correctly predicted preferences per subject (%)**					
	1	**2**	**3**	**4**	**5**	**6**
$\Phi_{\textbf{LIPM}}$ **(holdout)**	75	100	62.5	75	12.5	87.5
$\Phi_{\textbf{LIPM}}$	75	87.5	62.5	75	62.5	100
$\Phi_{\textbf{dynamic}}$	100	100	50	75	12.5	37.5
$\Phi_{\textbf{static}}$	50	75	37.5	50	100	75

We quantify the predictive power of each fitted cost function on the users' utility functions using the rank consistency between the cost function values and the preference data. Table 5.1 shows the predictive power of Φ_{LIPM} on the preferences, as well as the predictive power of two other cost functions, Φ_{static} and Φ_{dynamic}, respectively, defined as:

$$\Phi_{\text{static}} = ||\{x, y\}_{\text{CoM}} - \{x, y\}_{\text{CoP}}||^2, \tag{5.6}$$

$$\Phi_{\text{dynamic}} = ||p^{\text{goal}}_{\{x,y\}} - p_{\{x,y\}}||^2. \tag{5.7}$$

These two metrics are directly opposed: while Φ_{dynamic} is the term from Φ_{LIPM} that promotes dynamic stability, Φ_{static} penalizes dynamic stability in favor of static stability. This is because in the LIPM dynamics, the acceleration of $\{x, y\}_{\text{CoM}}$ approaches zero as Φ_{static} approaches zero. We find that Φ_{LIPM} and Φ_{dynamic} capture the preferences of five of the six subjects, while Φ_{static} completely predicts the preferences of the single outlier, subject 5.

Figure 5.7 further illustrates this difference. The largest discrepancy between Φ_{dynamic} and Φ_{static} is that of subject 1 and subject 5. The preferences of subject 1 align with dynamic stability, while the preferences of subject 5 align with static stability. The diametric opposition between the cost function terms predicting these users' preferences reflects inconsistencies across users' gait utility functions. This suggests that there is most likely no single metric that entirely captures all users' underlying utilities. Thus, it is important to generate a variety of gaits that satisfy the cost functions reflecting different users' preferences.

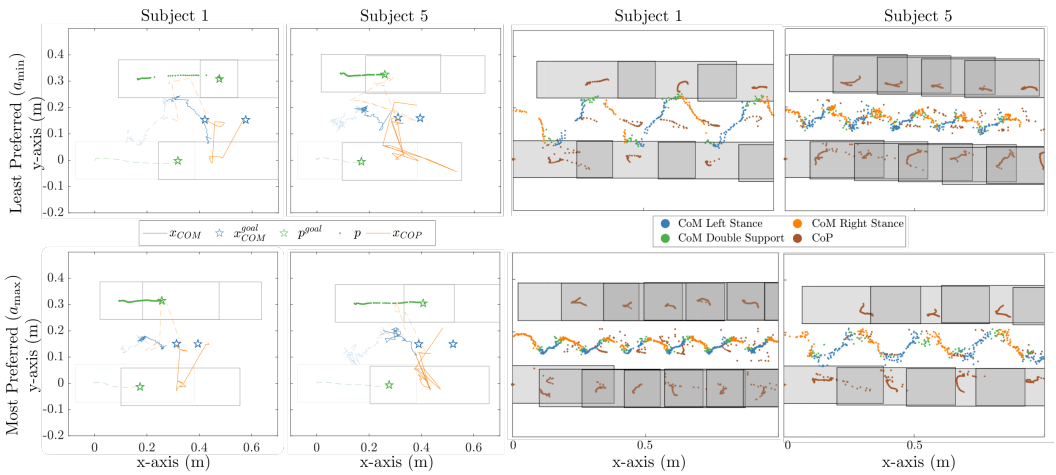


Figure 5.7: Contrasting preferences across two subjects regarding the trade-off between more and less dynamically-stable gaits. While all of the exoskeleton gaits are dynamically stable, both the least preferred gait ($a_{\min}$) of subject 1 and the most preferred gait ($a_{\max}$) of subject 5 exhibit behavior closer to statically-stable gaits. Subject 1 preferred dynamic gaits with a large difference between x_{CoP} and x_{CoM}; in contrast, subject 5 preferred gaits in which x_{CoP} closely followed the center of mass. Rectangles represent the exoskeleton feet.

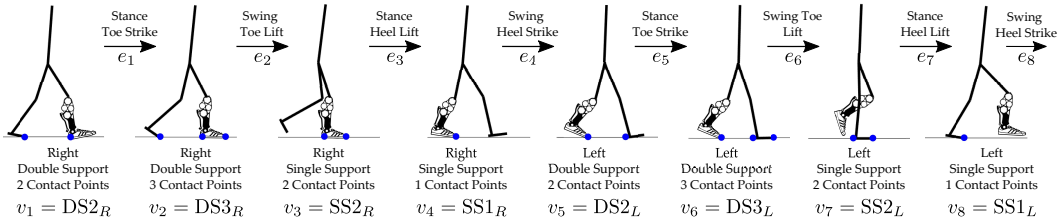


Figure 5.8: A complete gait cycle from right heel strike to right heel strike. The gait cycle is described using the directed cycle $\Gamma = (V, E)$ with the vertices $V = \{v_1, \ldots, v_8\}$ and edges $E = \{e_1, \ldots, e_8\}$ illustrated in the figure. The naming convention is based on the stance 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.

5.3 Incorporating Musculoskeletal Models into the Gait Generation Optimization Problem

Lastly, this section explores the addition of musculoskeletal models directly into the gait generation process to intuitively shape the resulting behavior. Specifically, a multi-domain hybrid system model (as illustrated in Figure 5.8) is constructed that combines the system dynamics with muscle models to represent natural multicontact walking. Provably stable walking gaits can then be generated for this model via the hybrid zero dynamics (HZD) method.

In this section, the integrated framework is specifically applied towards achieving multicontact locomotion on a dual-actuated transfemoral prosthesis, AMPRO3, for two subjects. The results demonstrate that enforcing muscle model constraints produces gaits that yield natural locomotion (as analyzed via comparison to motion capture data and electromyography). Moreover, gaits generated with our framework were strongly preferred by the non-disabled prosthetic users as compared to gaits generated with the nominal HZD method, even with the use of systematic tuning methods. We conclude that the novel approach of combining robotic walking methods (specifically HZD) with muscle models successfully generates anthropomorphic robotic-assisted locomotion.

Related Work

Existing approaches towards realizing natural walking include modifying the HZD method to obtain gaits that resemble walking recorded by motion capture [8, 237] and optimizing joint-level trajectories for experimental metrics such as electromyography (EMG) signals and metabolic expenditure [84, 236]. While these methods yield natural behavior, they are data-driven and thereby heavily rely on the quality and quantity of the data. Moreover, such data is behavior-specific and not always accessible. A separate approach that does not rely on gait generation is to directly control the walking based on real-time EMG feedback [21, 53, 97, 215, 229]. While this methodology also successfully yields natural locomotion, it has no theoretical guarantees of stability and relies on careful tuning of the musculoskeletal model.

This section instead presents an alternative approach based upon hybrid system models of locomotion that utilize musculoskeletal models—to the best of my knowledge, this is the first time these two modeling paradigms have been combined. This proposed integrated framework both enjoys the theoretical guarantees of stability via the HZD method, while also achieving natural locomotion via the musculoskeletal models. Since humans usually self-select gaits that are physiologically and mechanically energy efficient [219], the hypothesis made in this section is that generating stable gaits that satisfy muscle model constraints would naturally lead to more anthropomorphic and efficient behavior that respects physiological limits.

This hypothesis will be evaluated by first generating multicontact walking gaits utilizing the HZD method coupled with musculoskeletal models, followed by

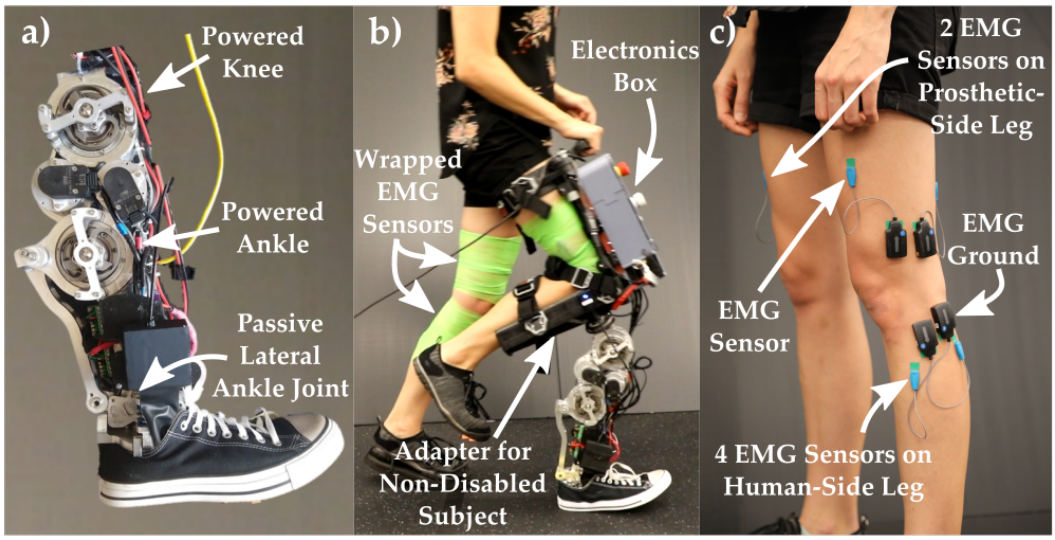


Figure 5.9: Experimental setup. a) AMPRO3 prosthesis, b) Non-disabled subject wearing the device during multicontact locomotion, c) placement of the surface mount electrodes for electromyography (EMG).

the experimental implementation on a dual-actuated transfemoral prosthesis, AMPRO3, shown in Figure 5.9. The experimental results demonstrate that the novel combination of muscluloskeletal models with HZD results in natural multicontact locomotion, as quantified by comparisons with motion capture data and via electromyography (EMG).

Muscle Model

First, we introduce how a single muscle-tendon unit (MTU) is modeled. Later, details will be provided on how to extend these muscle models to multiple muscles and incorporate them into the Hybrid Zero Dynamics (HZD) gait generation framework.

Muscle-tendon Unit (MTU). Each muscle is modeled as a two-element Hill-type muscle-tendon unit [71] with a contractile element (CE) and a series elastic element (SE) as shown in Figure 5.10a. The constant parameters of each muscle are defined in [70, 71].

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

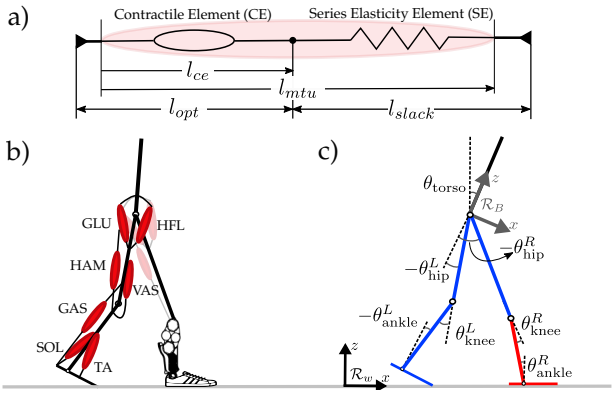


Figure 5.10: Musculoskeletal modeling setup. a) A single muscle tendon unit (MTU) consists of a contractile element (CE) and a series elasticity element (SE). The length of CE and SE is denoted by l_{ce} and l_{se}. At the reference angle (θ_{ref}), these lengths are equal to $l_{ce} = l_{opt}$ and $l_{se} = l_{slack}$. 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.

MTU length as a function of q:

$$l_{mtu}(q) = l_{opt} + l_{slack} - \sum_{j=1}^{j_N} \Delta l_{mtu}(\theta_j), \tag{5.8}$$

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 [70]. 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.5.10b. The individual change in length due to a single joint, $\Delta l_{mtu}(\theta) \in \mathbb{R}$, is given by:

$$\Delta l_{mtu}(\theta) = \begin{cases} \rho r_0(\theta - \theta_{ref}), & \text{for hip} \\ \rho r_0 \big[\sin(\theta - \theta_{max}) & \\ \quad - \sin(\theta_{ref} - \theta_{max}) \big], & \text{otherwise.} \end{cases} \tag{5.9}$$

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:

$$f_l(l_{ce}) = \exp\left(\log(c)\left|\frac{l_{ce} - l_{opt}}{l_{opt} w}\right|^3\right), \tag{5.10}$$

$$f_v(v_{ce}) = \begin{cases} \frac{v_{max} - v_{ce}}{v_{max} + K v_{ce}}, & \text{if } v_{ce} < 0 \\ N + \frac{(N-1)(v_{max} + v_{ce})}{7.56 K v_{ce} - v_{max}}, & \text{if } v_{ce} \geq 0 \end{cases} \tag{5.11}$$

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:

$$f_{se}(l_{se}) = \begin{cases} \left(\frac{l_{se} - l_{slack}}{l_{slack}(\varepsilon_{ref})}\right)^2, & \text{if } l_{se} \geq l_{slack} \\ 0, & \text{otherwise} \end{cases} \tag{5.12}$$

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:

$$F_{ce}(l_{ce}, v_{ce}, s) = s\, F_{max}\, f_l(l_{ce})\, f_v(v_{ce}), \tag{5.13}$$

$$F_{se}(l_{se}) = F_{max}\, f_{se}(l_{se}), \tag{5.14}$$

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 muscle activation is assumed 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:

$$u_m = r(\theta)\, F_m, \tag{5.15}$$

$$r(\theta) = \begin{cases} r_0, & \text{for hip} \\ r_0\, cos(\theta - \theta_{max}), & \text{otherwise} \end{cases} \tag{5.16}$$

where $r(\theta) \in \mathbb{R}$ is the length of the MTU lever-arm based on r_0 (previously defined in Eq. 5.9), 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 [70].

Gait Generation with the Integrated Framework

Next, an integrated framework is proposed that enforces the various muscle-tendon unit properties (introduced in the preceding subsection) directly into the HZD gait generation framework introduced in Chapter 2. First, we will present the details of the integrated framework. Then, its effect on the gait generation process is demonstrated 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 earlier, we construct a non-

linear 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 $X = [X_{\text{NLP}}, X_{\text{MUSC}}]^\top$ 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 [90].

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 (2.8); C3 constrains the decision variables as $X_{\min} \preceq X \preceq X_{\max}$; 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:

Muscle Model Constraints:

C5. $\{F_m^{(i)} = F_{ce}^{(i)}(l_{ce}^{(i)}, v_{ce}^{(i)}, l_{se}^{(i)}, s^{(i)}),\ \forall i = 1, \ldots, 10\}$

C6. $\{F_m^{(i)} = F_{se}(l_{se}^{(i)}),\ \forall i = 1, \ldots, 10\}$

C7. $\{l_{ce}^{(i)} + l_{se}^{(i)} = l_{mtu}(q)^{(i)},\ \forall i = 1, \ldots, 10\}$

C8. $\{l_{ce}^{(i)} = \int v_{ce}^{(i)} dt,\ \forall i = 1, \ldots, 10\}$

C9. $u_{\text{hip}}^{L} = u_m^{(1h)} + u_m^{(2)} + u_m^{(3)}$

C10. $u_{\text{knee}}^{L} = u_m^{(1k)} + u_m^{(4k)} - u_m^{(5)}$

C11. $u_{\text{ankle}}^{L} = u_m^{(4a)} + u_m^{(6)} - u_m^{(7)}$

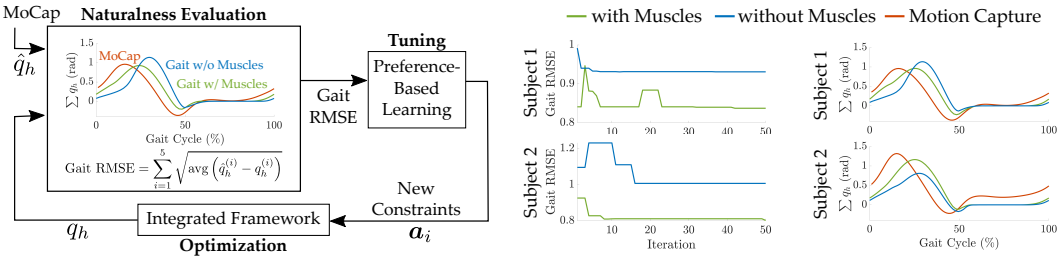
C12. $u_{\text{hip}}^{R} = u_m^{8} + u_m^{9} - u_m^{10}$

where $i = 1, \ldots, 10$ denotes a specific muscle out of the ten muscles we consider, illustrated in Figure 5.10b. 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 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. 5.15 with different reference angles in Eq. 5.16.

Evaluation of the Integrated Framework

Optimization setup. To evaluate the 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 [87].



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

Figure 5.11: Results of gait generated with and without the muscle models. a) Gait generation and tuning procedure. Note that the MoCap data are taken from [44] 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.

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

$$\text{Gait RMSE} = \sum_{i=1}^{5} \sqrt{\text{avg}\left(\hat{q}_h^{(i)} - q_h^{(i)}\right)}, \tag{5.17}$$

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 [44] and matched to subjects by height and weight.

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 Figure 5.11a. Specifically, the LineCoSpar algorithm is used 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, the optimization is warm-started with the solution from the current 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. The search space of the algorithm is constructed with the muscle model parameters outlined in Table 5.2.

Table 5.2: Action Space Definition: Musculoskeletal model and gait generation optimization problem constraints.

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

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 Figure 5.11b. 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 (Figure 5.11b-c). This highlights the advantage of including muscle models in the gait generation, as it guided the optimization to find more natural solutions.

Experimental Demonstration on AMPRO3

The two gaits obtained in the automated tuning procedure as having the lowest gait RMSE (with and without the inclusion of muscle model constraints) were experimentally deployed 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 [194].

Experiment Procedure. During the experiments, a non-disabled human user wore AMPRO3 using an adapter on the right leg (Figure 5.9b). 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 [237].

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

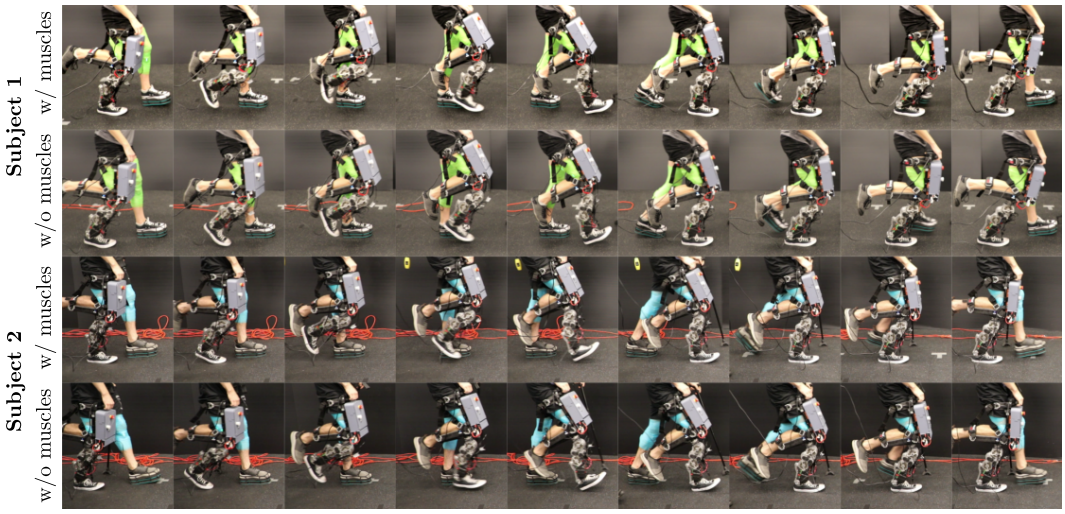


Figure 5.12: Gait tiles of experimental demonstration on AMPRO3 for gaits generated without or with muscle model constraints for two subjects.

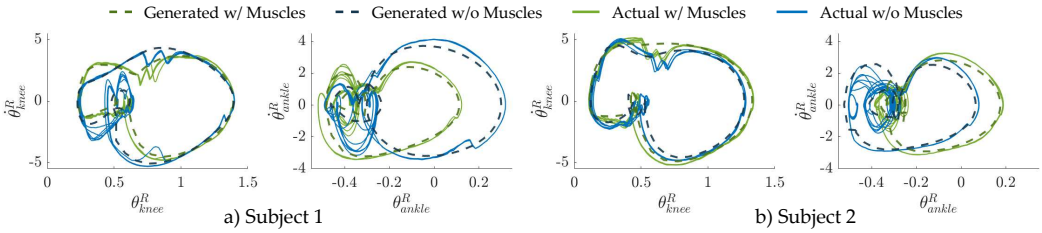


Figure 5.13: Limit cycles illustrating the periodic stability achieved during experimental multicontact locomotion (10s of data plotted).

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 Figure 5.9c.

Experiment Results. A visualization of the experimental behaviors is provided in Figure 5.12 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 Figure 5.13 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 Figure 5.14. We also calculated the RMSE between the EMG

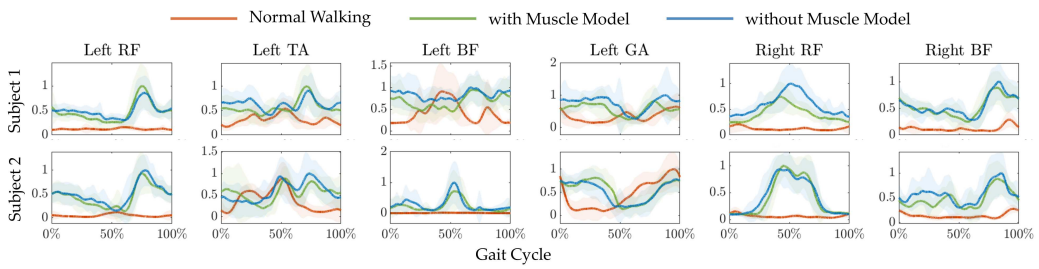


Figure 5.14: EMG activity normalized over a full gait cycle for normal walking, prosthetic walking with gaits generated with or without the muscle model.

activity of the generated gaits and normal walking, defined as:

$$\text{EMG RMSE} = \sum_{i=1}^{6} \sqrt{\text{avg}\left(\hat{s}_{\text{EMG}}^{(i)} - s_{\text{EMG}}^{(i)}\right)}, \tag{5.18}$$

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.

5.4 Summary

Throughout the thesis, locomotive stability and user-customization are considered to be two of the most important clinical considerations. However, there exist many other important factors, including the three explored in this chapter. First, metabolic cost of transport was analyzed on the Atalante exoskeleton using a method of control that enabled variable assistance. This work found that metabolic cost of transport was only influenced when exoskeleton users had the ability to contribute to the walking motion. Second, features of user-preferred exoskeleton gaits were extracted in an attempt to understand the mechanisms dictating user comfort. Specifically, two oppos-

ing measures of dynamicity were identified, implying that exoskeleton users either prefer walking gaits that are dynamically or statically stable. These considerations help to inform the synthesis of new exoskeleton gaits that directly maximize user comfort. Lastly, musculoskeletal model constraints were directly enforced 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. It was found that incorporating the muscle model guides the optimization problem towards uncovering periodic orbits that resemble natural bipedal locomotion.

Chapter 6

CONCLUSIONS

In conclusion, this dissertation seeks to improve lower-body assistive devices by systematically approaching locomotive robustness and user-customization. The individual contributions of the presented work are as follows:

- **Systematic approaches towards achieving and certifying stable and robust bipedal locomotion.** Chapter 2 presented two overall approaches towards addressing robustness of nominal walking gaits to uncertain impact events due to either uncertain human-robot models or environments. Overall, the first approach (minimizing the induced matrix norm of the saltation matrix directly in the gait generation process) demonstrated improved experimental robustness compared to the baseline gait generation methodology. Further, the second approach (an input-to-state stability perspective on walking) provided theoretic certificates of locomotive robustness to uncertain impact events. Both of these approaches help to better understand the theoretical conditions underlying provably robust walking.

- **A preference-based learning framework for optimizing user comfort during exoskeleton locomotion.** Chapter 3 proposed a principled, sample-efficient preference-based learning framework for optimizing and characterizing individual exoskeleton user comfort. The details of this learning framework were provided, along with simulation and experimental demonstrations of the corresponding algorithms.

- **Extending demonstrations for how the preference-based learning framework can be utilized for additional robotic applications.** Chapter 4 applies the preference-based learning framework towards general robotic applications which illustrate the efficacy of the methodology towards tasks that relies on expert operator tuning. These applications include tuning constraint bounds directly in the HZD gait generation process, tuning control parameters of an ID-CLF-QP$^+$ controller, and tuning control barrier function parameters. The experimen-

tal demonstrations of these applications were performed on a planar biped, a 3D biped, and a quadrupedal biped. Overall, the chapter demonstrates that preference-based learning is an efficient method for quickly translating complex robotic behaviors to the real world by leveraging a human's natural ability to judge *good* behavior.

- **Experimental insight into additional clinical considerations for realizing beneficial locomotion on lower-body assistive devices.** Finally, Chapter 5 presented three tangential clinical considerations for lower-body assistive devices: variable assistance and its effect on metabolic expenditure; latent factors underlying user-preferred locomotion; and the embedding of musculoskeletal model constraints in the gait generation process. Overall, all three of these considerations are important for the clinical viability of lower-body assistive devices.

6.1 Future Work

Future work for the contributions of this thesis are summarized as follows. First, future work towards better understanding the mathematical conditions underlying locomotive robustness includes directly evaluating δ-robustness in the gait generation process to systematically generate periodic orbits that are robust to uncertain terrain. Moreover, one could apply the stochastic notion of ISS (ISSp) to the gait synthesis framework to systematically generate nominal walking trajectories that have probabilistic guarantees of robustness for reasonable estimates of uncertain terrain. Additionally, since the probabilistic guarantees associated with ISSp are currently loose, it is also likely that tighter bounds may be obtained by exploiting particular structure in the disturbance distribution and dynamics. An important question here is how to choose Lyapunov function V, and the supermartingale W, to obtain the tightest probabilistic bound possible.

Second, towards user-customization, future steps include incorporating increasingly fine-grained subjective feedback mechanisms into the learning framework such as scaled preference feedback [224]. The hypothesis is that this fine-grained feedback would improve the sample-efficiency. Also, as user preferences may change over time, creating a learning framework that accounts for these adaptations is also an important future research direction. Lastly, more extensive studies are needed involving subjects with paraplegia, whose

preferences likely differ from those of able-bodied subjects. Such future clinical trials would not only improve user comfort during exoskeleton rehabilitation, but would provide the data required to unlock a deeper understanding of the science of walking. For example, given a large set of preference data, one could apply tensor decomposition techniques to discover invariant subspaces among the gait parameters. Such knowledge could accelerate learning of personalized gaits by guiding exploration.

Finally, there are several next steps for investigating clinical considerations associated with assisted locomotion. First, while the work presented in this thesis studied how variable assistance influences metabolic cost of the exoskeleton user, further studies are needed to determine how assisted locomotion influences muscle activity directly. This can be accomplished by recording exoskeleton users' electromyography during variable assistance. However, care is needed towards insuring that the electrical motor noise does not influence the accuracy of the electromyographic signals, and that the results are repeatable regardless of placement error. Second, the aforementioned exhaustive studies on user-preference would provide sufficient data to make conclusive claims about attributes of preferred exoskeleton walking gaits. Lastly, the musculoskeletal models presented in this thesis leveraged physiological parameters (reference lengths, angle, etc.) from [70], which was intended for a non-disabled subject with different height and weight. Thus, 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 [200]. Such prediction accuracy would further allow for targeted muscle behavior of the user for rehabilitation applications.

Bibliography

[1] A Primer on Bézier Curves. https://pomax.github.io/bezierinfo/, 2023.

[2] 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*, volume 13. Cambridge, MA, USA, 2017.

[3] 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. *IEEE Access*, 5:9919–9934, 2017.

[4] Nir Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(1), 2012.

[5] 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. *IEEE Control Systems Letters*, 2021.

[6] Eric Ambrose, Wen-Loong Ma, Christian Hubicki, and Aaron D. Ames. Toward benchmarking locomotion economy across design configurations on the modular robot: AMBER-3M. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1270–1276, 2017. doi: 10.1109/CCTA.2017.8062633.

[7] Eric Ambrose, Wen-Loong Ma, Christian Hubicki, and Aaron D. Ames. Toward benchmarking locomotion economy across design configurations on the modular robot: AMBER-3M. In *Conference on Control Technology and Applications (CCTA)*, pages 1270–1276. IEEE, 2017.

[8] Aaron D. Ames. Human-inspired control of bipedal walking robots. *Transactions on Automatic Control*, 59(5):1115–1130, 2014.

[9] Aaron D. Ames and Ioannis Poulakakis. Hybrid zero dynamics control of legged robots. In *Bioinspired Legged Locomotion: Models, Concepts, Control and Applications*, pages 292–331. Elsevier, 2017. URL http://ames.caltech.edu/HZD_bookchapter.pdf.

[10] Aaron D. Ames and Matthew Powell. Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs. In *Control of Cyber-Physical Systems*, pages 219–240. Springer, 2013.

[11] Aaron D. Ames, Ryan W. Sinnet, and Eric Wendel. Three-dimensional kneed bipedal walking: A hybrid geometric approach. In *International Workshop on Hybrid Systems: Computation and Control*, pages 16–30. Springer, 2009.

[12] Aaron D. Ames, Ramanarayan Vasudevan, and Ruzena Bajcsy. Human-data based cost of bipedal robotic walking. In *International Conference on Hybrid Systems: Computation and Control*, pages 153–162, 2011.

[13] Aaron D. Ames, Eric A. Cousineau, and Matthew J. Powell. Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In *International Conference on Hybrid Systems: Computation and Control*, pages 135–144. ACM, 2012.

[14] Aaron D. Ames, Kevin Galloway, Koushil Sreenath, and Jessy W. Grizzle. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *Transactions on Automatic Control*, 59(4):876–891, 2014.

[15] Aaron D. Ames, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Conference on Decision & Control (CDC)*, pages 6271–6278. IEEE, 2014.

[16] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[17] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *European Control Conference (ECC)*, pages 3420–3431. IEEE, 2019.

[18] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[19] Khairul Anam and Adel Ali Al-Jumaily. Active exoskeleton control systems: State of the art. *Procedia Engineering*, 41:988–994, 2012.

[20] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[21] Samuel K. Au, Paolo Bonato, and Hugh Herr. An EMG-position controlled system for an active ankle-foot prosthesis: an initial experimental study. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 375–379. IEEE, 2005.

[22] Jean-Pierre Aubin. *Viability theory*. Springer Science, 2009.

[23] Alejandro F. Azocar and Elliott J. Rouse. Stiffness perception during active ankle and knee movement. *Transactions on Biomedical Engineering*, 64(12):2949–2956, 2017.

[24] Sai K. Banala, Seok Hun Kim, Sunil K. Agrawal, and John P. Scholz. Robot assisted gait training with active leg exoskeleton (ALEX). *Transactions on Neural Systems and Rehabilitation Engineering*, 17(1):2–8, 2008.

[25] 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)*, pages 6728–6731. IEEE, 2015.

[26] Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Sinahal, and Anca D. Dragan. Do you want your autonomous car to drive like you? In *International Conference on Human-Robot Interaction (HRI)*, pages 417–425. IEEE, 2017.

[27] 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. *Spinal Cord*, 56(2): 106–116, 2018.

[28] Andrea L. Behrman and Susan J. Harkema. Physical rehabilitation as an agent for recovery after spinal cord injury. *Physical Medicine and Rehabilitation Clinics of North America*, 18(2):183–202, 2007.

[29] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with Gaussian processes. In *International Conference on Robotics and Automation (ICRA)*, 2016.

[30] Michael Bernhardt, Martin Frey, Gery Colombo, and Robert Riener. Hybrid force-position control yields cooperative behaviour of the rehabilitation robot LOKOMAT. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 536–539. IEEE, 2005.

[31] John E.A. Bertram. Constrained optimization in human walking: cost minimization and gait plasticity. *Journal of Experimental Biology*, 208 (6):979–991, 2005.

[32] Mauro Birattari and Janusz Kacprzyk. *Tuning metaheuristics: a machine learning perspective*, volume 197. Springer, 2009.

[33] 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.

[34] 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 *Conference on Robot Learning*, pages 1177–1190. PMLR, 2020.

[35] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Springer, 2008.

[36] Susan A. Bloomfield, Walter Jerry Mysiw, and Rebecca Dorothy Jackson. Bone mass and endocrine adaptations to training in spinal cord injured individuals. *Bone*, 19(1):61–68, 1996.

[37] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[38] J.M. Brockway. Derivation of formulae used to calculate energy expenditure in man. *Human Nutrition: Clinical Nutrition*, 41(6):463–471, 1987.

[39] Jonas Buchli, Mrinal Kalakrishnan, Michael Mistry, Peter Pastor, and Stefan Schaal. Compliant quadruped locomotion over rough terrain. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 814–820. IEEE, 2009.

[40] Samuel A. Burden, Thomas Libby, and Samuel D. Coogan. On contraction analysis for hybrid systems. *arXiv preprint arXiv:1811.03956*, 2018.

[41] Lance L. Cai, Andy J. Fong, Yongqiang Liang, Joel Burdick, and V. Reggie Edgerton. Assist-as-needed training paradigms for robotic rehabilitation of spinal cord injuries. In *International Conference on Robotics and Automation (ICRA)*, pages 3504–3511. IEEE, 2006.

[42] Lance L. Cai, Andy J. Fong, Chad K. Otoshi, Yongqiang Liang, Joel W. Burdick, Roland R. Roy, and V. Reggie Edgerton. Implications of assist-as-needed robotic step training after a complete spinal cord injury on intrinsic strategies of motor learning. *Journal of Neuroscience*, 26(41): 10564–10568, 2006.

[43] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1):5–23, 2016.

[44] 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. *Journal of Biomechanics*, 119:110320, 2021.

[45] 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 and dynamics. In *American Control Conference (ACC)*, pages 3683–3690. IEEE, 2021.

[46] Guillermo A. Castillo, Bowen Weng, Ayonga Hereid, Zheng Wang, and Wei Zhang. Reinforcement learning meets hybrid zero dynamics: A case study for rabbit. In *International Conference on Robotics and Automation (ICRA)*, pages 284–290. IEEE, 2019.

[47] Guillermo A. Castillo, Bowen Weng, Terrence C. Stewart, Wei Zhang, and Ayonga Hereid. Velocity regulation of 3D bipedal walking robots with uncertain dynamics through adaptive neural network controller. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 7703–7709. IEEE, 2020.

[48] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30(1):1–41, 2012.

[49] Christine Chevallereau, Dalila Djoudi, and Jessy W. Grizzle. Stable bipedal walking with foot rotation through direct regulation of the zero moment point. *Transactions on Robotics*, 24(2):390–401, 2008.

[50] Jason J. Choi, Donggun Lee, Koushil Sreenath, Claire J. Tomlin, and Sylvia L. Herbert. Robust control barrier–value functions for safety-critical control. In *Conference on Decision and Control (CDC)*, pages 6814–6821. IEEE, 2021.

[51] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 137–144, 2005.

[52] Wei Chu, Zoubin Ghahramani, and Christopher K.I. Williams. Gaussian processes for ordinal regression. *Journal of Machine Learning research*, 6(7), 2005.

[53] 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 *International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 557–563. IEEE, 2020.

[54] Andrew Clark. Control barrier functions for complete and incomplete information stochastic systems. In *American Control Conference (ACC)*, pages 2928–2935. IEEE, 2019.

[55] Ryan K. Cosner, Andrew W. Singletary, Andrew J. Taylor, Tamás G. Molnár, Katherine L. Bouman, and Aaron D. Ames. Measurement-robust control barrier functions: Certainty in safety with uncertainty in state. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 6286–6291. IEEE, 2021.

[56] Ryan K. Cosner, Maegan Tucker, Andrew 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*, pages 1020–1033. PMLR, 2022.

[57] Ryan K. Cosner, Preston Culbertson, Andrew J. Taylor, and Aaron D. Ames. Robust safety under stochastic uncertainty with discrete-time control barrier functions. *arXiv preprint arXiv:2302.07469*, 2023.

[58] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2019.

[59] Noel Csomay-Shanklin, Maegan Tucker, Min Dai, Jenna Reher, and Aaron D. Ames. Learning controller gains on bipedal walking robots via user preferences. In *International Conference on Robotics and Automation (ICRA)*, pages 10405–10411. IEEE, 2022. URL http://dx.doi.org/10.1109/ICRA46639.2022.9811541.

[60] Hongkai Dai and Russ Tedrake. Optimizing robust limit cycles for legged locomotion on unknown terrain. In *Conference on Decision and Control (CDC)*, pages 1207–1213. IEEE, 2012.

[61] Sarah Dean, Andrew Taylor, Ryan K. Cosner, Benjamin Recht, and Aaron D. Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning (CoRL)*, pages 654–670. PMLR, 2021.

[62] Gülçin Demirel, HuÈrriyet Yilmaz, Nurdan Paker, and Selma OÈnel. Osteoporosis after spinal cord injury. *Spinal Cord*, 36(12):822–825, 1998.

[63] Alexander Domahidi, Eric Chu, and Stephen Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076. IEEE, 2013.

[64] Rudolfs Drillis and Renato Contini. *Body segment parameters*. New York University, School of Engineering and Science Research Division, NY, 1966.

[65] Péter Fankhauser, Marco Hutter, Christian Gehring, Michael Bloesch, Mark A. Hoepflinger, and Roland Siegwart. Reinforcement learning

of single legged locomotion. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 188–193. IEEE, 2013.

[66] Ryan J. Farris, Hugo A. Quintero, and Michael Goldfarb. Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals. *Transactions on Neural Systems and Rehabilitation Engineering*, 19(6):652–659, 2011.

[67] Kevin Galloway, Koushil Sreenath, Aaron D. Ames, and Jessy W. Grizzle. Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332, 2015.

[68] Gabriel García, Robert Griffin, and Jerry Pratt. MPC-based locomotion control of bipedal robots with line-feet contact using centroidal dynamics. In *International Conference on Humanoid Robots (Humanoids)*, pages 276–282. IEEE, 2021.

[69] Rachel Gehlhar, Maegan Tucker, Aaron J. Young, and Aaron D. Ames. A review of current state-of-the-art control methods for lower-limb powered prostheses. *Annual Reviews in Control*, 2023.

[70] Hartmut Geyer and Hugh Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *Transactions on Neural Systems and Rehabilitation Engineering*, 18(3):263–273, 2010.

[71] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1529):2173–2183, 2003.

[72] 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. *Rehabilitation Psychology*, 48(3):157, 2003.

[73] Christopher Glocker and Franz Pfeiffer. Dynamical systems with unilateral contacts. *Nonlinear Dynamics*, 3(4):245–259, 1992.

[74] 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)*, pages 897–902. IEEE, 2018.

[75] 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)*, pages 4730–4737. IEEE, 2019.

[76] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford University Press, 2020.

[77] Jessy W. Grizzle, Gabriel Abba, and Franck Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *Transactions on Automatic Control*, 46(1):51–64, 2001.

[78] Jessy W. Grizzle, Christine Chevallereau, Ryan W. Sinnet, and Aaron D. Ames. Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8):1955–1988, 2014.

[79] Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, Ayonga Hereid, Omar Harib, Matthieu Masselin, Jessy W. Grizzle, and Aaron D. Ames. Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons. In *International Conference on robotics and automation (ICRA)*, pages 2804–2811. IEEE, 2018.

[80] Thomas Gurriet, Maegan Tucker, Alexis Duburcq, Guilhem Boeris, and Aaron D. Ames. Towards variable assistance for lower body exoskeletons. *IEEE Robotics and Automation Letters*, 5(1):266–273, 2019. URL http://dx.doi.org/10.1109/LRA.2019.2955946.

[81] Sehoon Ha, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. Learning to walk in the real world with minimal human effort. *arXiv preprint arXiv:2002.08550*, 2020.

[82] Kaveh Akbari Hamed, Brian G. Buss, and Jessy W. Grizzle. Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. *The International Journal of Robotics Research*, 35(8):977–999, 2016.

[83] Samar Hamid and Ray Hayek. Role of electrical stimulation for rehabilitation and regeneration after spinal cord injury: an overview. *European Spine Journal*, 17(9):1256–1269, 2008.

[84] 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. *Transactions on Neural Systems and Rehabilitation Engineering*, 29:944–952, 2021.

[85] Omar Harib, Ayonga Hereid, Ayush Agrawal, Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, M. Eva Mungai, Mattieu Masselin, and Aaron D. Ames. Feedback control of an exoskeleton for paraplegics: Toward robustly stable, hands-free dynamic walking. *Control Systems Magazine*, 38(6):61–87, 2018.

[86] 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. *The Lancet*, 377(9781):1938–1947, 2011.

[87] Ayonga Hereid and Aaron D. Ames. FROST: Fast robot optimization and simulation toolkit. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 719–726. IEEE, 2017.

[88] 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 *International Conference on Hybrid Systems: Computation and Control*, pages 263–272, 2014.

[89] Ayonga Hereid, Christian M. Hubicki, Eric A. Cousineau, Jonathan W. Hurst, and Aaron D. Ames. Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking. In *International Conference on Robotics and Automation (ICRA)*, pages 5734–5740. IEEE, 2015.

[90] 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 *International Conference on Robotics and Automation (ICRA)*, pages 1447–1454. IEEE, 2016.

[91] Ayonga Hereid, Christian M Hubicki, Eric A. Cousineau, and Aaron D. Ames. Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization. *Transactions on Robotics*, 34(2):370–387, 2018.

[92] Kentarou Hitomi, Tomohiro Shibata, Yutaka Nakamura, and Shin Ishii. Reinforcement learning for quasi-passive dynamic walking of an unstable biped robot. *Robotics and Autonomous Systems*, 54(12):982–988, 2006.

[93] Hhakan Hjalmarsson and Thom Birkeland. Iterative feedback tuning of linear time-invariant MIMO systems. In *Conference on Decision and Control (CDC)*, volume 4, pages 3893–3898. IEEE, 1998.

[94] Weng Khuen Ho, Chang Chieh Hang, and Lisheng S. Cao. Tuning of PID controllers based on gain and phase margin specifications. *Automatica*, 31(3):497–502, 1995.

[95] Daan G.E. Hobbelen and Martijn Wisse. A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *Transactions on Robotics*, 23(6):1213–1224, 2007.

[96] Kerianne L. Hobbs, Mark L. Mote, Matthew C.L. Abate, Samuel D. Coogan, and Eric M. Feron. Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems. *Control Systems Magazine*, 43(2):28–65, 2023.

[97] Carl D. Hoover and Kevin B. Fite. A configuration dependent muscle model for the myoelectric control of a transfemoral prosthesis. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6. IEEE, 2011.

[98] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[99] Yildirim Hurmuzlu and Dan B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The International Journal of Robotics Research*, 13(1):82–92, 1994.

[100] Yildirim Hurmuzlu, Frank GéNot, and Bernard Brogliato. Modeling, stability and control of biped robots—a general framework. *Automatica*, 40(10):1647–1664, 2004.

[101] Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *Robotics and Automation Letters*, 3(2):895–902, 2018. URL www.raisim.com.

[102] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

[103] Alberto Isidori. *Nonlinear Control Systems, Third Edition*. Communications and Control Engineering. Springer, 1995. ISBN 978-1-4471-3909-6. doi: 10.1007/978-1-84628-615-5. URL https://doi.org/10.1007/978-1-84628-615-5.

[104] Alberto Isidori, S. Shankar Sastry, Petar V. Kototovic, and Christopher I. Byrnes. Singularly perturbed zero dynamics of nonlinear systems. *Transactions on Automatic Control*, 37(10):1625–1631, 1992.

[105] Patrick L. Jacobs and Mark S. Nash. Exercise recommendations for individuals with spinal cord injury. *Sports Medicine*, 34(11):727–751, 2004.

[106] Ashesh Jain, Shikhar Sharma, Thorsten Joachims, and Ashutosh Saxena. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research*, 34(10):1296–1313, 2015.

[107] Mrdjan Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, 96:359–367, 2018.

[108] Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[109] Zhong-Ping Jiang and Yuan Wang. A converse Lyapunov theorem for discrete-time systems with disturbances. *Systems & Control Letters*, 45: 49–58, 2002.

[110] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Special Interest Group on Information Retrieval (SIGIR)*, volume 5, pages 154–161, 2005.

[111] Myungsoo Jun and Michael G. Safonov. Automatic PID tuning: An application of unfalsified control. In *International Symposium on Computer Aided Control System Design*, pages 328–333. IEEE, 1999.

[112] 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 *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1620–1626. IEEE, 2003.

[113] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, pages 295–304. PMLR, 2015.

[114] Jacques Kerdraon, Jean Gabriel Previnaire, Maegan Tucker, Pauline Coignard, Willy Allegre, Emmanuel Knappen, and Aaron D. Ames. Evaluation of safety and performance of the self balancing walking system Atalante in patients with complete motor spinal cord injury. *Spinal Cord Series and Cases*, 7(1):71, 2021. URL http://dx.doi.org/10.1038/s41394-021-00432-3.

[115] 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. *Journal of Neuroengineering and Rehabilitation*, 16(1):1–17, 2019.

[116] Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586*, 2019.

[117] 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. *PloS one*, 12(9):e0184054, 2017.

[118] 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)*, pages 3429–3438. PMLR, 2019.

[119] Paula Kocina. Body composition of spinal cord injured adults. *Sports Medicine*, 23(1):48–60, 1997.

[120] Shishir Kolathaya, J. Reher, Ayonga Hereid, and Aaron D. Ames. Input to state stabilizing control Lyapunov functions for robust bipedal robotic locomotion. In *American Control Conference*, pages 2224–2230. IEEE, 2018.

[121] Nathan J. Kong, Joe Payne, George Council, and Aaron M. Johnson. The salted Kalman filter: Kalman filtering on hybrid dynamical systems. *Automatica*, 131:109752, 2021.

[122] Harold J. Kushner. *Stochastic Stability and Control*. Academic Press, 1967.

[123] Harold J. Kushner. Some Basic Ideas in Stochastic Stability. *Annals of Economics and Social Measurement*, 3(1):85–90, 1974.

[124] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[125] Remco I. Leine and Henk Nijmeijer. *Dynamics and bifurcations of non-smooth mechanical systems*, volume 18. Springer Science & Business Media, 2013.

[126] Johannes Lengler. Drift Analysis. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, Natural Computing Series, pages 89–131. Springer International Publishing, 2020.

[127] James C. Lester, Brian A. Stone, and Gary D. Stelling. Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User Modeling and User-Adapted Interaction*, 9(1-2):1–44, 1999.

[128] 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 *International Conference on Robotics*

and Automation (ICRA), pages 3212–3218. IEEE, 2021. URL http://dx.doi.org/10.1109/ICRA48506.2021.9560840.

[129] 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 *International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[130] Winfried Lohmiller and Jean-Jacques E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[131] John Lygeros, Karl Henrik Johansson, S. Shankar Sastry, and Magnus Egerstedt. On the existence of executions of hybrid automata. In *Conference on Decision and Control (CDC)*, volume 3, pages 2249–2254. IEEE, 1999.

[132] John Lygeros, Karl Henrik Johansson, Slobodan N. Simic, Jun Zhang, and S. Shankar Sastry. Dynamical properties of hybrid automata. *Transactions on automatic control*, 48(1):2–17, 2003.

[133] Wen-Loong Ma, Hui-Hua Zhao, Shishir Kolathaya, and Aaron D. Ames. Human-inspired walking via unified PD and impedance control. In *International Conference on Robotics and Automation (ICRA)*, pages 5088–5094. IEEE, 2014.

[134] 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 *International Conference on Hybrid Systems: Computation and Control*, pages 265–274, 2017.

[135] Wen-Loong Ma, Yizhar Or, and Aaron D. Ames. Dynamic walking on slippery surfaces: Demonstrating stable bipedal gaits with planned ground slippage. In *International Conference on Robotics and Automation (ICRA)*, pages 3705–3711. IEEE, 2019.

[136] Wen-Loong Ma, Noel Csomay-Shanklin, and Aaron D. Ames. Coupled control systems: Periodic orbit generation with application to quadrupedal locomotion. *Control Systems Letters*, 5(3):935–940, 2020.

[137] Ian R. Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research (IJRR)*, 30(3):265–279, 2011.

[138] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR tuning based on Gaussian process global optimization. In *International Conference on Robotics and Automation (ICRA)*, pages 270–277. IEEE, 2016.

[139] Robert D. McAllister and James B. Rawlings. Stochastic exponential stability of nonlinear stochastic model predictive control. In *Conference on Decision and Control (CDC)*, pages 880–885. IEEE, 2021.

[140] Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Springer, 1993.

[141] Katja Daniela Mombaur. *Stability optimization of open-loop controlled walking robots*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2001.

[142] Jun Morimoto, Gordon Cheng, Christopher G. Atkeson, and Garth Zeglin. A simple reinforcement learning algorithm for biped walking. In *International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3030–3035. IEEE, 2004.

[143] 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 *Conference on Decision and Control (CDC)*, pages 4199–4206. IEEE, 2005.

[144] N. Eric Naftchi, Anna T. Viau, G. Heiner Sell, and Edward W. Lowman. Mineral metabolism in spinal cord injury. *Archives of Physical Medicine and Rehabilitation*, 61(3):139–142, 1980.

[145] Kemal Nas, Levent Yazmalar, Volkan Şah, Abdulkadir Aydın, and Kadriye Öneş. Rehabilitation of spinal cord injuries. *World Journal of Orthopedics*, 6(1):8, 2015.

[146] Sergey G. Nersesov, VijaySekhar Chellaboina, and Wassim M. Haddad. A generalization of Poincaré's theorem to hybrid and impulsive dynamical systems. In *American Control Conference (ACC)*, volume 2, pages 1240–1245. IEEE, 2002.

[147] NeurIPS 2019: Learn to Move - Walk Around, 2019. URL https://www.aicrowd.com/challenges/neurips-2019-learning-to-move-walk-around.

[148] Susan B. O'Sullivan and Thomas J. Schmitz. Physical rehabilitation: Assessment and treatment, 1994.

[149] Victor Paredes and Ayonga Hereid. Dynamic locomotion of a lower-limb exoskeleton through virtual constraints based ZMP regulation. In *Dynamic Systems and Control Conference*, volume 84270. American Society of Mechanical Engineers, 2020.

[150] Victor C. Paredes and Ayonga Hereid. Resolved motion control for 3D underactuated bipedal walking using linear inverted pendulum dynamics

and neural adaptation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 6761–6767. IEEE, 2022.

[151] Hae-Won Park, Alireza Ramezani, and Jessy W Grizzle. A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *Transactions on Robotics*, 29(2):331–345, 2012.

[152] Joe Payne, Nathan J. Kong, and Aaron M. Johnson. The uncertainty aware salted Kalman filter: State estimation for hybrid systems with uncertain guards. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 8821–8828. IEEE, 2022.

[153] Lawrence Perko. *Differential Equations and Dynamical Systems*, volume 7. Springer Science & Business Media, 2013.

[154] Jerry E. Pratt and Russ Tedrake. Velocity-based stability margins for fast bipedal walking. In *Fast Motions in Biomechanics and Robotics*, pages 299–324. Springer, 2006.

[155] Akshara Rai, Rika Antonova, Seungmoon Song, William Martin, Hartmut Geyer, and Christopher Atkeson. Bayesian optimization using domain knowledge on the ATRIAS biped. In *International Conference on Robotics and Automation (ICRA)*, pages 1771–1778. IEEE, 2018.

[156] Marc H. Raibert. Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case. *Transactions on Systems, Man, and Cybernetics*, pages 451–463, 1984.

[157] Marc H. Raibert. *Legged robots that balance*. MIT press, 1986.

[158] J. Reher, Eric A. Cousineau, Ayonga Hereid, Christian M. Hubicki, and Aaron D. Ames. Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS. In *International Conference on Robotics and Automation (ICRA)*, pages 1794–1801. IEEE, 2016.

[159] J. 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*, pages 400–415. Springer, 2020.

[160] Jenna Reher and Aaron D. Ames. Dynamic walking: Toward agile and efficient bipedal robots. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:535–572, 2021.

[161] Jenna Reher and Aaron D. Ames. Inverse dynamics control of compliant hybrid zero dynamic walking. In *International Conference on Robotics and Automation (ICRA)*, pages 2040–2047. IEEE, 2021.

[162] 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. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12, 2019.

[163] Timothy T. Roberts, Garrett R. Leonard, and Daniel J. Cepela. Classifications in brief: American spinal injury association (ASIA) impairment scale. *Clinical Orthopaedics and Related Research*, 475(5):1499, 2017.

[164] Agility Robotics. https://www.agilityrobotics.com/robots#cassie, Last accessed on 2021-09-14, 2017.

[165] David F. Rogers and James Alan Adams. *Mathematical elements for computer graphics*. McGraw-Hill, Inc., 1989.

[166] 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*, pages 91–100. PMLR, 2022.

[167] Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. Verification and control for finite-time safety of stochastic systems via barrier functions. In *Conference on Control Technology and Applications (CCTA)*, pages 712–717. IEEE, 2019.

[168] Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. A barrier function approach to finite-time stochastic system verification and control. *Automatica*, 125:109439, 2021.

[169] Nebahat Sezer, Selami Akkuş, and Fatma Gülçin Uugurlu. Chronic complications of spinal cord injury. *World Journal of Orthopedics*, 6(1): 24, 2015.

[170] Pannaga Shivaswamy and Thorsten Joachims. Online structured prediction via coactive learning. In *International Conference on Machine Learning (ICML)*, pages 59–66, 2012.

[171] Pannaga Shivaswamy and Thorsten Joachims. Coactive learning. *Journal of Artificial Intelligence Research*, 53:1–40, 2015.

[172] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[173] Shubham Singh, Ryan P. Russell, and Patrick M. Wensing. Efficient analytical derivatives of rigid-body dynamics using spatial vector algebra. *Robotics and Automation Letters*, 7(2):1776–1783, 2022.

[174] Ryan W. Sinnet and Aaron D. Ames. 2D bipedal walking with knees and feet: A hybrid control approach. In *Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3200–3207. IEEE, 2009.

[175] Patrick Slade, Mykel J. Kochenderfer, Scott L. Delp, and Steven H. Collins. Personalizing exoskeleton assistance while walking in the real world. *Nature*, 610(7931):277–282, 2022.

[176] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified MPC framework for whole-body dynamic locomotion and manipulation. *Robotics and Automation Letters*, 6(3): 4688–4695, 2021.

[177] Martha Freeman Somers. *Spinal cord injury: functional rehabilitation.* Prentice Hall, 2001.

[178] Thane Somers and Geoffrey A Hollinger. Human–robot planning and learning for marine data collection. *Autonomous Robots*, 40(7):1123–1137, 2016.

[179] Eduardo D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and Optimal Control Theory*, pages 163–220. Springer, 2008.

[180] Shraddha Srivastava, Pei-Chun Kao, Seok Hun Kim, Paul Stegall, Damiano Zanotto, Jill S. Higginson, Sunil K. Agrawal, and John P. Scholz. Assist-as-needed robot-aided gait training improves walking function in individuals following stroke. *Transactions on Neural Systems and Rehabilitation Engineering*, 23(6):956–963, 2014.

[181] Jacob Steinhardt and Russ Tedrake. Finite-time regional verification of stochastic non-linear systems. *The International Journal of Robotics Research (IJRR)*, 31(7):901–923, 2012.

[182] Benjamin J. Stephens and Christopher G. Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *International Conference on Humanoid Robots*, pages 52–59. IEEE, 2010.

[183] Yanan Sui and Joel Burdick. Clinical online recommendation with subgroup rank feedback. In *Conference on Recommender Systems*, pages 289–292, 2014.

[184] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 997–1005. PMLR, 2015.

[185] Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. *arXiv preprint arXiv:1705.00253*, 2017.

[186] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 4781–4789. PMLR, 2018.

[187] Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5502–5510, 2018.

[188] Supplemental Video for "Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits." https://youtu.be/c6a0kXMyML0, 2020.

[189] Supplemental Video for "Learning controller gains on bipedal walking robots via user preferences." https://youtu.be/jMX5a_6Xcuw, 2022.

[190] Supplemental Video for "Preference-based learning for exoskeleton gait optimization." https://youtu.be/-27sHXsvONE, 2020.

[191] Supplemental Video for "Preference-based learning for user-guided HZD gait generation on bipedal walking robots." https://youtu.be/rLJ-m65F6C4, 2021.

[192] Supplemental Video for "Robust bipedal locomotion: Leveraging saltation matrices for gait optimization." https://youtu.be/BZu-9UStG2E, 2023.

[193] Supplemental Video for "ROIAL: Region of interest active learning for characterizing exoskeleton gait preference landscapes." https://www.youtube.com/watch?v=04lMJmKmZrQ, 2021.

[194] Supplementary Video for "Natural multicontact walking for robotic assistive devices via musculoskeletal models and hybrid zero dynamics." https://www.youtube.com/watch?v=gOhZlTypNIs, 2022.

[195] Supplementary Video for "Safety-aware preference-based learning for safety-critical control." https://youtu.be/QEuwRDTG7TE, 2022.

[196] 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)*, pages 300–305. IEEE, 2018.

[197] Yang Tang, Xiaotai Wu, Peng Shi, and Feng Qian. Input-to-state stability for nonlinear systems with stochastic impulses. *Automatica*, 113: 108766, 2020.

[198] Andrew J. Taylor and Aaron D. Ames. Adaptive safety with control barrier functions. In *American Control Conference (ACC)*, pages 1399–1405. IEEE, 2020.

[199] Andrew J. Taylor, Andrew Singletary, Yisong Yue, and Aaron D. Ames. Learning for safety-critical control with control barrier functions. *Proceedings of Machine Learning Research (PMLR)*, 120:708–717, 2020.

[200] Nitish Thatte, Helei Duan, and Hartmut Geyer. A method for online optimization of lower limb assistive devices with high dimensional parameter spaces. In *International Conference on Robotics and Automation (ICRA)*, pages 5380–5385. IEEE, 2018.

[201] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25 (3/4):285–294, 1933.

[202] Sashank Tirumala, Sagar Gubbi, Kartik Paigwar, Aditya Sagi, Ashish Joglekar, Shalabh Bhatnagar, Ashitava Ghosal, Bharadwaj Amrutur, and Shishir Kolathaya. Learning stable manoeuvres in quadruped robots from expert demonstrations. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1107–1112. IEEE, 2020.

[203] 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*, pages 5026–5033. IEEE, 2012.

[204] John Tsinias. Stochastic input-to-state stability and applications to global feedback stabilization. *International Journal of Control*, 71(5): 907–930, 1998.

[205] Maegan Tucker and Aaron D. Ames. An input-to-state stability perspective on robust locomotion. *arXiv preprint*, 2023.

[206] 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 (IROS)*, pages 3423–3430. IEEE, 2020. URL http://dx.doi.org/10.1109/IROS45743.2020.9341416.

[207] 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)*, pages 2351–2357. IEEE, 2020. URL http://dx.doi.org/10.1109/ICRA40945.2020.9196661.

[208] 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 *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. URL http://dx.doi.org/10.1109/ICRA48506.2021.9561515.

[209] Michael R. Tucker, Jeremy Olivier, Anna Pagel, Hannes Bleuler, Mohamed Bouri, Olivier Lambercy, José del R. Millán, Robert Riener, Heike Vallery, and Roger Gassert. Control strategies for active lower extremity prosthetics and orthotics: A review. *Journal of Neuroengineering and Rehabilitation*, 12(1):1, 2015.

[210] Edwin H.F. van Asseldonk and Herman van der Kooij. Robot-aided gait training with lopes. In *Neurorehabilitation Technology*, pages 379–396. Springer, 2012.

[211] 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. *Disability and Rehabilitation*, 43(14):1982–1988, 2021.

[212] Jean Ville. Étude critique de la notion de collectif, 1939.

[213] Miomir Vukobratović and Branislav Borovac. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

[214] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[215] Jing Wang, Oliver A. Kannape, and Hugh M. Herr. Proportional EMG control of ankle plantar flexion in a powered transtibial prosthesis. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 1–5. IEEE, 2013.

[216] Li Wang, Aaron D. Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *Transactions on Robotics*, 33(3):661–674, 2017.

[217] Li Wang, Evangelos A. Theodorou, and Magnus Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *International Conference on Robotics and Automation (ICRA)*, pages 2460–2465. IEEE, 2018.

[218] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas, et al. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1778–1784, 2013.

[219] Robert L. Waters and Sara Mulroy. The energy expenditure of normal and pathologic gait. *Gait & Posture*, 9(3):207–231, 1999.

[220] Eric R. Westervelt, Jessy W. Grizzle, and Daniel E. Koditschek. Hybrid zero dynamics of planar biped walkers. *Transactions on Automatic Control*, 48(1):42–56, 2003.

[221] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[222] Richard L. Wheeden and Antoni Zygmund. *Measure and Integral: an Introduction to Real Analysis*, volume 308. CRC Press, 2015. URL https://faculty.sdu.edu.cn/_resources/group1/M00/00/3F/CgECYWPLk2uATCbtAEt7JSqmvww777.pdf.

[223] Pierre-Brice Wieber. On the stability of walking systems. In *International Workshop on Humanoid and Human Friendly Robotics*, 2002.

[224] Nils Wilde, Erdem Bıyık, Dorsa Sadigh, and Stephen L. Smith. Learning reward functions from scale feedback. In *Conference on Robot Learning (CoRL)*, 2021.

[225] David A Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[226] Willy Wojsznis, John Gudaz, Terry Blevins, and Ashish Mehta. Practical approach to tuning MPC. *ISA Transactions*, 42(1):149–162, 2003.

[227] Jamie Wolff, Claire Parker, Jaimie Borisoff, W. Ben Mortenson, and Johanne Mattie. A survey of stakeholder perspectives on exoskeleton technology. *Journal of Neuroengineering and Rehabilitation*, 11(1):169, 2014.

[228] 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*, pages 167–174. ACM, 2001.

[229] Sai-Kit Wu, Garrett Waycaster, and Xiangrong Shen. Electromyography-based control of active above-knee prostheses. *Control Engineering Practice*, 19(8):875–882, 2011.

[230] Xinyu Wu, Du-Xin Liu, Ming Liu, Chunjie Chen, and Huiwen Guo. Individualized gait pattern generation for sharing lower limb exoskeleton robot. *Transactions on Automation Science and Engineering*, 15(4): 1459–1470, 2018.

[231] Zhaoming Xie, Glen Berseth, Patrick Clary, Jonathan Hurst, and Michiel van de Panne. Feedback control for Cassie with deep reinforcement learning. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1241–1246. IEEE, 2018.

[232] Zhaoming Xie, Patrick Clary, Jeremy Dao, Pedro Morais, Jonathan Hurst, and Michiel van de Panne. Iterative reinforcement learning based design of dynamic locomotion skills for cassie. *arXiv preprint arXiv:1903.09537*, 2019.

[233] Aaron J. Young and Daniel P. Ferris. State of the art and future directions for lower limb robotic exoskeletons. *Transactions on Neural Systems and Rehabilitation Engineering*, 25(2):171–182, 2016.

[234] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[235] Damiano Zanotto, Paul Stegall, and Sunil K. Agrawal. Adaptive assist-as-needed controller to improve gait symmetry in robot-assisted gait training. In *International Conference on Robotics and Automation (ICRA)*, pages 724–729. IEEE, 2014.

[236] 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. *Science*, 356(6344):1280–1284, 2017.

[237] Huihua Zhao, Ayonga Hereid, Wen-loong Ma, and Aaron D. Ames. Multi-contact bipedal robotic locomotion. *Robotica*, 35(5):1072–1106, 2017.

[238] Huihua Zhao, Jonathan Horn, J. Reher, Victor Paredes, and Aaron D. Ames. First steps toward translating robotic walking to prostheses: A nonlinear optimization based control approach. *Autonomous Robots*, 41 (3):725–742, 2017.

[239] Li Zheng. A practical guide to tune of proportional and integral (PI) like fuzzy controllers. In *International Conference on Fuzzy Systems*, pages 633–640. IEEE, 1992.

[240] James Zhu, Nathan J. Kong, George Council, and Aaron M. Johnson. Hybrid event shaping to stabilize periodic hybrid orbits. In *International Conference on Robotics and Automation (ICRA)*, pages 01–07. IEEE, 2022.

[241] 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)*, pages 10–18. PMLR, 2014.