

From Bipedal to Quadrupedal Locomotion, Experimental Realization of Lyapunov Approaches

Thesis by
Wen-Loong Ma

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

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended May 05, 2021

© 2021

Wen-Loong Ma

ORCID: 0000-0002-0115-5632

All rights reserved except where otherwise noted

To my parents and grandfather.

ACKNOWLEDGEMENTS

What a splendid adventure!

My goal for graduate school was to understand the whole process of making a robot. Now that I am on the other side of this trek and ready for another voyage, I want to thank my advisor, Aaron Ames, for guiding me through all the challenging years and for constructing the research environment for my ambitious dreams. Beyond the nonlinear control theorem, I learned from him; his passion and boldly rational vision have shown me how to navigate the ocean of knowledge without being disturbed by the fancy buzzwords and hot trends.

Further, it was an honor to be part of the AMBER group. I was lucky enough to work with the phase-I students, Ryan, Shishir, Huihua, Ayonga, and Matthew. They are the good examples of hard-working, creativity, and persistence, which set the ideal initial conditions for my research career. Those postmidnight hangouts after deadlines are the most memorable moments of my twenties. I owe much to Shishir for the “on-call” mentorship in developing the theorem of this thesis. I am grateful to have worked with Christian, Eric, Jenna, Noel, Drew, and Maegan to crack open the mysterious real-world robotics experiments and for their friendship.

My life in graduate school was made enjoyable because of all the crazy-minded peers I came across. The countless arguments with Fangzhou Xiao, Shaowu Pan, Wei Ding, Honglie Ning, Jin Sima, Yu Sun, etc., have stimulated various angles to reevaluate my research from the perspectives of biology, physics, aerospace, logic, and math. These discussions have widely opened my eyes and gradually broken my subconscious barriers. My gratitude extends to all my collaborators and those academic pioneers, Jessy Grizzle, Paulo Tabuada, Kaveh Akbari Hamed, Koushil Sreenath, Joel Burdick, John Doyle, Richard Murray, etc., for educating me not only knowledge but also the quality of an academic professional. Their light shone across the Pacific Ocean and guided me to this land of possibilities. I am particularly thankful for Kaveh’s and Joel’s coaching in the last few years of my PhD life.

Finally, I want to thank the delicious coffee stores in Pasadena, such as Copa Vida, Jameson Brown, Intelligentsia, Ambrose Cafe, and Starbucks, for brewing the refreshing drinks and setting the cosy environment to convert caffeine into control algorithms.

ABSTRACT

Possibly one of the most significant innovations of the past decade is the hybrid zero dynamics (HZD) framework, which formally and rigorously designs a control algorithm for robotic walking. In this methodology, Lyapunov stability, which is often used to certificate a dynamical system’s stability, was introduced to the control law design for a hybrid control system. However, the prerequisites of precise modeling to apply the HZD methodology can often be too restrictive to design controllers for uncertain and complex real-world hardware experiments. This thesis addresses the problem raised by noisy measurements and the intricate hybrid structure of locomotion dynamics.

First, the HZD methodology’s construction is based on the full-order, hybrid dynamics of legged locomotion, which can be intractable for control synthesis for high-dimensional systems. This thesis studies the general structure of hybrid control systems for walking systems, ranging from 1D hopping, 2D walking, 2D running, and 3D quadrupedal locomotion on rough terrains. Further, we characterize a walking behavior—gait—as a solution (execution) to a hybrid control system. To find these solutions, which represent a “gait,” we employed advanced numerical methods such as collocation methods to parse the solution-finding problem into the open- and closed-loop trajectory optimization problems. The result is that we can find versatile gaits for ten different robotic platforms efficiently. This includes bipedal running, bipedal walking on slippery surfaces, and quadrupedal robots walking on sloped terrains. The numerous solution-finding examples expand the applicability of the HZD framework towards more complex dynamical systems.

Further, for the uncertain and noisy real-world implementation, the exponential stability of the continuous dynamics is an ideal but restrictive condition for hybrid stability. This condition is especially challenging to satisfy for highly dynamical behaviors such as bipedal running, which loses ground support for a short period. This thesis observes the destabilizing effect of the noisy measurements of the phasing variable. By reformulating the traditional input-to-state stability (ISS) concept into phase-uncertainty to state stability, we are able to synthesize a robust controller for bipedal running on DURUS-2D. This time+state-based controller formally guarantees stability under noisy measurements and stabilizes the 1.75 m/s running experiments.

Lastly, robotic dynamics have long been characterized as the interconnection of rigid-body dynamics. We take this perspective one step further and incorporate controller design into the formulation of coupled control systems (CCS). We first view a quadrupedal robot as two bipedal robots connected via some holonomic constraints. In a dimensional reduction manner, we develop a novel optimization framework, and the computational performance is reduced to a few seconds for gait generation. Furthermore, we can design local controllers for each bipedal subsystem and still guarantee the overall system's stability. This is done by combining the HZD framework and the ISS properties to contain the disturbance induced by the other subsystems' inputs. Utilizing the proposed CCS methods, we will experimentally realize quadrupedal walking on various outdoor rough terrains.

PUBLISHED CONTENT AND CONTRIBUTIONS

Ambrose, E., W. Ma, and A. D. Ames (2021). “Towards the Unification of System Design and Motion Synthesis for High-Performance Hopping Robots”. In: *International Conference on Robotics and Automation (ICRA)*.

WM designed the optimization algorithm to simultaneously solve for a gait and the system parameters, and participated in the writing of the manuscript.

Ma, W., N. Csomay-Shanklin, and A. D. Ames (2021). “Coupled Control Systems: Periodic Orbit Generation With Application to Quadrupedal Locomotion”. In: *IEEE Control Systems Letters* 5.3, pp. 935–940. DOI: 10.1109/LCSYS.2020.3006963.

WM participated in the conception of the project, designed the optimization algorithm, led the experimental testing and the writing of the manuscript.

Ma, W. -L., N. Csomay-Shanklin, S. Kolathaya, et al. (2021). “Coupled Control Lyapunov Functions for Interconnected Systems, With Application to Quadrupedal Locomotion”. In: *IEEE Robotics and Automation Letters* 6.2, pp. 3761–3768. DOI: 10.1109/LRA.2021.3065174.

WM proved the main theorem about the coupled control Lyapunov function, developed the control algorithm and conducted the experiments, and led the writing of the manuscript.

Ma, W. -L., N. Csomay-Shanklin, and A. D. Ames (2020). “Quadrupedal Robotic Walking on Sloped Terrains via Exact Decomposition into Coupled Bipedal Robots”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4006–4011. DOI: 10.1109/IROS45743.2020.9341181.

WM derived the analysis for controlling quadrupedal robots walking on slope, participated in the experimental implementation, and led the writing of the manuscript.

Ma, Wen-Loong and Aaron D. Ames (2020). “From Bipedal Walking to Quadrupedal Locomotion: Full-Body Dynamics Decomposition for Rapid Gait Generation”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4491–4497. DOI: 10.1109/ICRA40945.2020.9196841.

WM derived the dynamic decomposition for gait generation, conducted the experiments, and led the writing of the manuscript.

Akbari Hamed, Kaveh., Wen-Loong. Ma, and Aaron. D. Ames (July 2019). “Dynamically Stable 3D Quadrupedal Walking with Multi-Domain Hybrid System Models and Virtual Constraint Controllers”. In: *2019 American Control Conference (ACC)*, pp. 4588–4595. DOI: 10.23919/ACC.2019.8815085.

W. M designed the optimization algorithm, defined the hybrid structure of quadrupedal dynamics, and participated in the writing of the manuscript.

Ma, W., K. A. Hamed, and A. D. Ames (2019). “First Steps Towards Full Model Based Motion Planning and Control of Quadrupeds: A Hybrid Zero Dynamics

- Approach”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5498–5503. DOI: 10.1109/IROS40897.2019.8968189. WM wrote the code for the gait generation of quadrupedal robots for three different motions, conducted outdoor experiments, and led the writing of the manuscript.
- Ma, W., Y. Or, and A D. Ames (2019). “Dynamic Walking on Slippery Surfaces : Demonstrating Stable Bipedal Gaits with Planned Ground Slippage*”. In: *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3705–3711. DOI: 10.1109/ICRA.2019.8793761. WM designed the controller, the gait optimization method, derived the hybrid structure of walking, conducted the experiments, and lead the writing of the manuscript.
- Reher, Jake, Wen-Loong Ma, and Aaron D. Ames (2019). “Dynamic Walking with Compliance on a Cassie Bipedal Robot”. In: *2019 18th European Control Conference (ECC)*, pp. 2589–2595. DOI: 10.23919/ECC.2019.8796090. WM designed the compliant walking gait and the feedforward controller, and participated in the experiments and the writing of the manuscript.
- Ambrose, E., W. Ma, C. Hubicki, et al. (2017). “Toward benchmarking locomotion economy across design configurations on the modular robot: AMBER-3M”. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1270–1276. DOI: 10.1109/CCTA.2017.8062633. WM designed the optimization algorithm for gait generation, participated in the conception of the project, and designed and conducted the experiments.
- Ma, Wen-Loong, Shishir Kolathaya, et al. (2017). “Bipedal Robotic Running with DURUS-2D: Bridging the Gap Between Theory and Experiment”. In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. HSCC '17. Pittsburgh, Pennsylvania, USA: ACM, pp. 265–274. ISBN: 978-1-4503-4590-3. DOI: 10.1145/3049797.3049823. URL: <http://doi.acm.org/10.1145/3049797.3049823>. WM designed running trajectory using optimization, participated in stability analysis for the continuous dynamics, participated in experimental design, and led the writing of the manuscript.
- Tabuada, P. et al. (2017). “Data-driven control for feedback linearizable single-input systems”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 6265–6270. DOI: 10.1109/CDC.2017.8264603. WM participated in the conception of the project, designed the experimental procedures, and participated in the writing of the manuscript.
- Ma, W., A. Hereid, et al. (2016). “Efficient HZD gait generation for three-dimensional underactuated humanoid running”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5819–5825. DOI: 10.1109/IROS.2016.7759856. WM designed the control and optimization methods for bipedal running, analyzed and prepared the data, and wrote the manuscript.

- Kolathaya, Shishir, Wen-Loong Ma, and Aaron D Ames (2015). “Composing Dynamical Systems to Realize Dynamic Robotic Dancing”. In: *Algorithmic Foundations of Robotics XI*. Springer, Cham, pp. 425–442.
WM designed and programmed the controller, constructed the real-time operating system environment, conducted the experiments, and participated in the writing of the manuscript.
- Mehra, A. et al. (2015). “Adaptive cruise control: Experimental validation of advanced controllers on scale-model cars”. In: *2015 American Control Conference (ACC)*, pp. 1411–1418. DOI: 10.1109/ACC.2015.7170931.
WM participated in the conception of the project, derived the safety controller for simulation, and participated in the writing of the manuscript.
- Ma, W., H. Zhao, et al. (May 2014). “Human-inspired walking via unified PD and impedance control”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5088–5094. DOI: 10.1109/ICRA.2014.6907605.
WM designed the walking controller, motor control algorithm, constructed the real-time operating system environment, conducted the experiments, and led the writing of the manuscript.
- Zhao, Hui-Hua et al. (2014). “Human-inspired multi-contact locomotion with AMBER2”. In: *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*. Best Paper Award Finalist of ICCPS 2014. IEEE, pp. 199–210. DOI: 10.1109/ICCPS.2014.6843723.
WM designed and programmed the controller, conducted the experiments, and participated in the writing of the manuscript.

TABLE OF CONTENTS

Acknowledgements	iv
Abstract	v
Published Content and Contributions	vii
Table of Contents	ix
List of Illustrations	xi
List of Tables	xvi
Chapter I: Introduction	1
1.1 Background	1
1.2 Contribution	2
1.3 Organization of Dissertation	3
Chapter II: Constrained Rigid-Body Dynamics	6
2.1 Coordinates and Notations	6
2.2 Holonomic Constraints and System-Level Degrees of Freedom	8
2.3 Continuous-Time Dynamics	9
2.4 Underactuated System	10
2.5 The ODE Form of Control-Affine Systems.	13
2.6 Hybrid System	14
2.7 Discrete Transitions	17
2.8 Solution to Hybrid Dynamics	19
2.9 Numerical Optimization	20
Chapter III: Controller Design for Bipedal Running	26
3.1 Running Dynamics (Open-Loop)	26
3.2 Virtual Constraint and Zero Dynamics	29
3.3 Gait Design via Closed-Loop Optimization	35
3.4 Bipedal Running with Input to State Stability	52
Chapter IV: Coupled Control System and Quadrupedal Applications	72
4.1 HZD, from Bipedal to Quadrupedal Robots	72
4.2 Coupled Control System and Its Solutions	81
4.3 CCS Optimization and Application to Quadrupedal Gaits	92
4.4 Coupled Control Lyapunov Functions	111
4.5 Coupled Mechanical Systems	118
4.6 Example 1. Feedback Linearization-Based CLFs	122
4.7 Example 2. Quadrupedal Walking with Model-Free CLFs	125
Chapter V: Conclusion and Future Work	130
5.1 Summary	130
5.2 Future Work	131
Bibliography	133

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 Robots developed to control in this thesis. From left to right: AMBER2 (2012-2014), AMBER3 (2015), AMBER3M (2016-2020), DURUS-2D (2015-2016), DURUS (2016), Cassie (2017-now), Vision60 v3.2 & v3.9 (2018-2020).	4
2.1 Configuration coordinates of AMBER robots, DURUS-2D running robot, DURUS, Cassie, and Vision 60, the quadrupedal robots.	6
2.2 Three fundamental contact problems modelled by holonomic constraints: a) point contact, $h(q) \in \mathbb{R}^3$ denotes the three-dimensional Cartesian position fixed to the world; b) line contact, $h(q) \in \mathbb{R}^{3+2}$ fixed one point of the contact line and the pitch, yaw angles of the body-fixed frame to the world; c) plane contact, $h(q) \in \mathbb{R}^6$ fixed the 6-dimensional body-fixed frame to the world.	8
2.3 Branched chain structure of a bipedal system—AMBER-3M, and a quadrupedal system—Vision 60.	10
2.4 Simple examples of overactuation in the presence of holonomic constraints: (a,b) sliding on a track; (c) pushing against a wall; (d) a quadrupedal robot standing on two feet.	11
2.5 Directed graph for two robotic systems.	15
2.6 A mixed sequence of walking tasks of a bipedal robot: flat ground walking, uphill walking, downhill walking, flat ground walking. On the right is the directed graph, where the arrows represent the corresponding edges. Each edge is associated with transition dynamics.	20
2.7 Left: the configuration coordinate of the 1D hopper. Right: The optimized trajectory from (2.51).	24
3.1 (a) The simulated running of the humanoid robot, DURUS, as a result of large-scale HZD optimization. (b) The directed cycle structure of the multi-domain hybrid system model for flat-footed humanoid running.	26
3.2 A block diagram showing the state-based feedback control structure.	34
3.3 Limit cycle of running at 2.0 m/s over 20 steps.	41
3.4 Snapshots of the DURUS running at 2.0 m/s.	42

3.5	The computation performance for generating gaits at each running velocity.	43
3.6	Multiple running gaits with forward velocity from 1.5 m/s to 3.0 m/s.	43
3.7	Slippage in the beginning of a step: pre-slip on the left and post-slip on the right.	44
3.8	On the left: The AMBER-3M with point foot, constrained to a planar rail to walk in a 2D environment on a treadmill. On the right: the model's configuration coordinates, with 3 global coordinates and 4 local coordinates.	46
3.9	Snapshots of one slippery walking step from <i>Experiment 3</i> . In the first three pictures, the left foot (stance foot) is slipping smoothly on the lubricated treadmill.	51
3.10	Phase portrait of 50 seconds' experimental data from <i>Experiment 3</i> , with walking speed (from left to right): 0.26m/s, 0.3m/s, 0.38m/s, 0.42m/s. Solid lines are for the desired values and dashed lines are for the actual measurements.	51
3.11	$MCOT^+$ of the experiments on AMEBR-3M. Those not included for certain speeds are failed experiments.	52
3.12	The spring-legged planar running biped, DURUS-2D, during take off (left) and while airborne (right).	52
3.13	(a) The model of DURUS-2D with two linear springs; (b) the directed cycle structure of the multi-domain hybrid system model for DURUS-2D running.	54
3.14	A block diagram showing the time-based feedback control structure.	59
3.15	Limit cycles of (a) simulation where time-based IO + state-based PD controller was applied for 100 steps; (b) Simulation where white noise was added to $\tau_v(q)$; (c) Experimental data. Note the solid lines are the designated gait from optimization.	67
3.16	Bounded tube around the designed periodic orbit.	67
3.17	Running tiles of simulation (top) vs. experiment (bottom) for one step.	68
3.18	Experimental setup for DURUS-2D running: 1) control station computer, 2) emergency stop, 3) four Li-Po batteries, 4) tripping harness, 5) treadmill control panel, 6) encoder wheel to measure treadmill speed, 7) treadmill.	68
3.19	Left and right foot height (ground clearance).	70

3.20	The actual vs. desired joint angles from the experiment.	70
3.21	The left column is the phase variables which are used to calculate time- and state-based outputs: $y^{d,t}(t) = y_v^d(\tau_v(t))$, $y^d(q) = y_v^d(\tau_v(q))$; the right column is output errors (virtual constraints) showing ultimate boundedness. (a, b, c, d) are from two simulations with controller given by (3.78). And (c, d) has a sinusoidal disturbance with 10% amplitude added to $\tau_v(q)$; (e, f) are from experiments.	71
4.1	(a) Vision 60 V3.2, 26 kg, 0.45m tall; (b) Vision 60 V3.9, 44 kg, 0.55m tall; (c) Configuration coordinates of both quadrupedal robots. Both robots were manufactured by Ghost Robotics.	72
4.2	The cyclic directed graph for the multi-domain hybrid dynamics of walking, ambling, and trotting gaits. The dashed lines represent a <i>relabeling</i> map (as given in (3.32)) that flips the left and right legs' contact attributes. We took these nomenclature from (Muybridge, n.d.).	76
4.3	On the top left is the snapshot for a 2×2 -domain walking gait; on the top right is a 2×2 -domain ambling gait; and on the bottom is a 4×2 -domain trotting gait. The symbol \square' is a left-right mirror of the contact attributes for domain \square . The orange-colored legs are the left-side legs whose contact points are highlighted by triangles; the blue-colored legs are the right-side legs with contact points highlighted by inverted triangles.	77
4.4	The periodic trajectories designed by NLP (4.7): walk (left), amble (middle), and trot (right).	79
4.5	Snapshots of the Vision 60 ambling in an outdoor environment, showing a full step of 2×2 domains of the amble gait.	80
4.6	The designed gaits (in red) from optimization/simulation vs. the experimental data (in cyan) in the form of phase portrait for <i>amble</i> . HR is short for hip roll, HP is for hip pitch, and k is for knee.	80
4.7	Conceptual illustration of the full-body dynamics decomposition, where the 3D quadruped — the Vision 60 — is decomposed into two constrained 3D bipedal robots.	82
4.8	A conceptual illustration of the full-body dynamics decomposition, where the 3D quadruped — the Vision 60 — is decomposed into two constrained 3D bipedal robots.	96

4.9	Top: Snapshots showing a full step of the ambling gait on an outdoor lawn. Bottom: The periodic trajectory produced by optimization (CCS-OPT) (in red) vs. the experimental tracking data (in cyan) vs. RaiSim simulation data (in green) in the form of phase portrait (limit cycle) using 18 seconds' data.	104
4.10	A comparison between the solution of bipedal walking dynamics obtained from the decomposition-based optimization and a simulated step of the full-order quadrupedal dynamics using the composed bipedal gaits; here MATLAB ODE45 was used.	106
4.11	Comparison between MuJoCo simulation (animated) and experiments. The upper two are for stepping in place, gait4; the lower two are logged for a full step of the ambling gait.	107
4.12	(a) The cyclic directed graph for the multi-domain hybrid dynamics on sloped terrain; (b) friction cone (pyramid) condition on some sloped terrain.	108
4.13	Phase portraits of the designed gaits (solid lines) vs. experimental data (transparent overlay) for quadrupedal walking on 13° (red) and $20^\circ \sim 25^\circ$ (blue) slopes. HR, HP, K are short for hip roll, hip pitch, and knee, respectively.	109
4.14	Full steps of gait tiles for the Vision 60 ambling in outdoor grasslands. The top two is comparison between simulation and experiments for a 13° slope. The bottom two are for walking on a ramp with varying slopes of $20^\circ \sim 25^\circ$	110
4.15	The quadrupedal robot, Vision 60 v3.9.	118
4.16	A cart-pole system with two inverted pendula, each is directly actuated by a motor. The mass of the cart and pendula are $2M$ and m , correspondingly. The length of both pendulum is l . We have the configuration coordinates as $(\xi, \theta_1, \theta_2)^\top$	119
4.17	The cart-pole simulation. data labeled FL (red) used the feedback linearization controller (4.98); data labeled CLF-QP (blue) used the controller (4.100); dark and light variants of the colors are used to distinguish between the first and second pendula. Both simulations show stability.	125

- 4.18 (a) The decomposition of a quadrupedal robot into two bipedal systems with toe indices labeled and (b) the configuration of the quadruped; each leg has a point contact toe and three rotational joints with motors. 126
- 4.19 Top: Snapshots showing a two full steps of the walking gait on an outdoor lawn. Bottom: Running gait in the RaiSim simulation environment. 128
- 4.20 Left three: Experimental (dashed transparent) and simulated (solid transparent) phase portraits for walking plotted against the desired values (solid). Right three: Simulated (transparent) vs. desired (solid) phase portraits for walking (red), hopping (green), and running (blue) behaviors. 129

LIST OF TABLES

<i>Number</i>	<i>Page</i>
4.1 Computing performance of gait generation. This is performed on a Linux machine with an i7-6820HQ CPU @2.70 GHz and 16 GB RAM.	106
4.2 Average torque inputs in experiments and simulations.	106

Chapter 1

INTRODUCTION

1.1 Background

Legged locomotion has been long described as “controlled falling.” This characterization reveals the key challenge of legged systems — stability. To accomplish stable walking on various terrains, many modelling and control strategies have been proposed and explored since the 1980s. Meanwhile, the majority of control methods are built to provide stability guarantees for certain, linear, or rather simple dynamical systems (Aëström and Kumar, 2014). Therefore, most work within legged locomotion involves some form of *model simplification* before designing a stabilizing controller.

The most fundamental modeling technique used in robotics are rigid-body dynamics, which is defined as “*the movement of systems of interconnected bodies under the action of external forces*” (Tsai, 1999). While a robot (modeled by rigid-body dynamics) appears to be the central component of legged locomotion, numerous other subcomponents are coupled with the robot itself through interconnections, such as terrain dynamics, compliant dynamics, impact dynamics, etc. The complexity of controlling such systems grows exponentially as the mathematical model details. There is a famous observation in statistics (Box, 1976): “*Since all models are wrong the scientist cannot obtain a "correct" one by excessive elaboration,*” which also captures this challenge of robotics. Therefore, a multi-link rigid-body system, with compliant components that are no more complicated than linear coil springs, and rigid contact/impact model is widely accepted as the “correct” model for robotics. We refer to this system as the *full-order system* in this thesis.

Among the approaches used in legged locomotion, a significant subset of the work lies in viewing walking dynamics as a reduction problem, wherein the complex real-world dynamics are *assumed* to be governed by the evolution of some reduced system. The linear invert pendulum model (LIPM) (S. Kajita, Kanehiro, et al., 2001; S. Kajita, Tani, and Kobayashi, 1990) assumes a bipedal system behaves as an invert pendulum with massless limbs. For high-performance behaviors such as running and hopping, a spring-loaded invert pendulum (SLIP) model is broadly applied to bipedal locomotion (Reinhard Blickhan and Full, 1993; R. Blickhan, n.d.;

Rezazadeh et al., 2015). It is also extended to the asymmetric-SLIP model to account for the heavy torso (Ioannis Poulakakis and Jessy W Grizzle, 2009). The centroidal dynamics (D. E. Orin, Goswami, and S.-H. Lee, 2013) captures the dominating effect of center of mass and angular momentum to control legged locomotion, including quadrupedal locomotion (Di Carlo et al., 2018). A prominent result of using simple models is the seminal work of Raibert’s hoppers and quadrupedal robots (M. Raibert and Tello, 1986). To maintain balance (an intuitive description of stability) and capture points, (Koolen et al., 2012) utilized a region to stabilize the LIPM-based systems, and the well-known Zero Moment Point (ZMP) method (Vukobratovic and Branislav Borovac, 2004) gives a robust but restrictive condition to prevent foot-rolling and further avoid falling. Yet these methods lack guarantees with respect to the full-order dynamics, and to mitigate erratic experimental performance, intensive parameter tuning is required.

Other works investigate this dimensional reduction by performing design of locomotion on the passive dynamics of the system. This can improve model fidelity and represent more physical details of the system; relevant methods include hybrid zero dynamics (HZD) methods (E. R. Westervelt, J. W. Grizzle, Chevallereau, et al., 2007; A. Ames, 2014) and other optimization-based approaches (Dalibard et al., 2013). By considering the rigid-body dynamics as a continuous-time system and the rigid impact with the ground as a discrete-time system, walking can be modelled as a hybrid system (Barton and C. K. Lee, 2002). Correspondingly, the HZD framework was then invented as a means to reduce the stability problem of the high-dimensional walking dynamics to the lower-dimensional zero dynamics on the hybrid zero dynamics manifold. From the perspective of Lyapunov, stability of the overall hybrid system is formally guaranteed in (A. Ames, Galloway, et al., 2014b). Many hardware results have been accomplished in the domain of legged locomotion, such as walking (J. Reher et al., 2016; Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, 2011) and running (Sreenath, H.-W. Park, I. Poulakakis, and J.W. Grizzle, 2013).

1.2 Contribution

The contribution of this thesis is twofold.

First, it enriches the application of HZD framework in the domains of 2D walking (Ambrose, W. Ma, C. Hubicki, et al., 2017), running (Wen-Loong Ma, Ayonga Hereid, et al., 2016; Wen-Loong Ma, Shishir Kolathaya, et al., 2017), monopedal

hopping (Ambrose, W. Ma, and A. D. Ames, 2021), walking with compliance (Jake Reher, Wen-Loong Ma, and Aaron D. Ames, 2019), and quadrupedal locomotion (W. Ma, K. A. Hamed, and A. D. Ames, 2019). Although the full-body hybrid dynamics of walking is believed to be “good enough” to represent the real-world hardware dynamics, we still face many uncertainties that can easily destroy stability, such as sensor noise, low-bandwidth compliant dynamics, etc. Therefore, (Wen-Loong Ma, Ayonga Hereid, et al., 2016) utilized direct-collocation methods to fast-finding periodic running gaits and formulated input-to-to state stability (ISS) analysis for the end result of robust running on DURUS-2D. See Fig. 1.1 for the development of the robots.

Secondly, when solving the control problem of quadrupedal locomotion from the hybrid control perspective (W. Ma, K. A. Hamed, and A. D. Ames, 2019), the computational complexity and experimental robustness is not effective in comparison with state-of-art simple model-based approaches. Therefore, we take inspirations from the seminal work of Raibert’s *virtual leg principle* (M. Raibert, Blankespoor, et al., 2008), and characterize quadrupedal locomotion as a collection of two connected bipedal subsystems. Through dynamics decomposition (Wen-Loong Ma and Aaron D. Ames, 2020), we were able to efficiently solve for quadrupedal gaits within a few seconds. It is also applied to sloped terrain walking (W. -L. Ma, Csomay-Shanklin, and A. D. Ames, 2020). These results are theoretically justified using the notion of coupled control systems (CCS) in (W. Ma, Csomay-Shanklin, and A. D. Ames, 2021), and Lyapunov stability analysis was used to synthesize local optimal controllers in (W. -L. Ma, Csomay-Shanklin, S. Kolathaya, et al., 2021) for each decoupled bipedal systems.

The goal of this thesis is to provide theoretical analysis for locomotion from Lyapunov’s perspective, and serve as a comprehensive introduction for experimental design of various robotic platforms (see Fig. 1.1 with a few examples).

1.3 Organization of Dissertation

This section provides a brief overview of the contents of each chapter.

Chapter 2: Constrained Rigid-Body Dynamics. This chapter details the modelling techniques and layout of the notations used for constructing the hybrid dynamics of legged locomotion. Using the conventional rigid-body dynamics and holonomic constraints, we can obtain the continuous-time equations of motion for the walking systems. Further we use plastic impact to describe the ground contact as a sim-

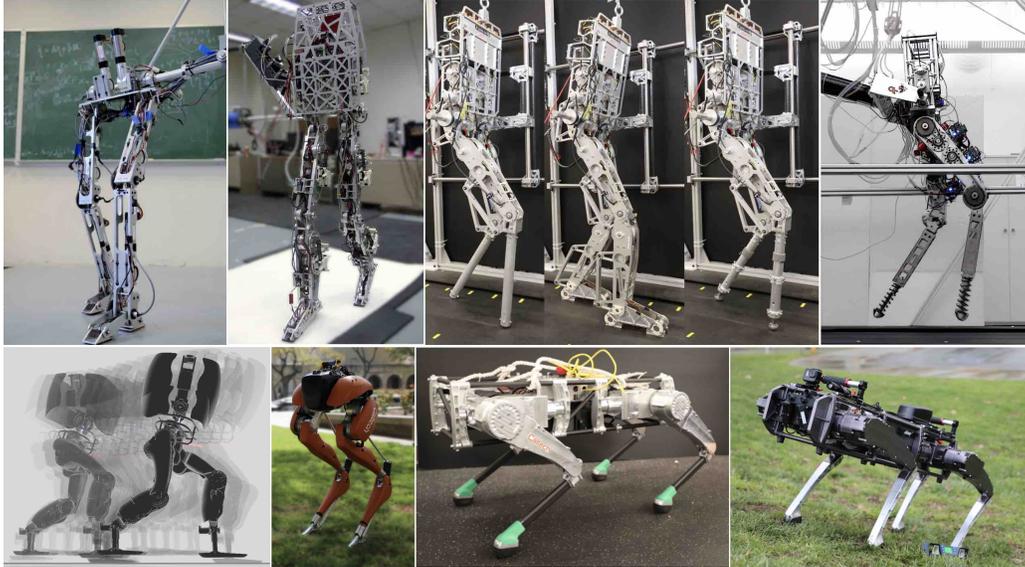


Figure 1.1: Robots developed to control in this thesis. From left to right: AMBER2 (2012-2014), AMBER3 (2015), AMBER3M (2016-2020), DURUS-2D (2015-2016), DURUS (2016), Cassie (2017-now), Vision60 v3.2 & v3.9 (2018-2020).

ple discrete-time dynamics. An alternating sequence of continuous and discrete dynamics is then regarded as a hybrid system, which is then represented by a set of differential algebra equations (DAEs). These concepts will be explained using multiple robots including *underactuated*, *fully-actuated*, and *overactuated* systems. We will finish this chapter with an optimization formulation that is used throughout this thesis to find a desired gait for a simple hopping robot, which is defined as the (numerical) solutions to the hybrid system.

Chapter 3: Controller Design for Bipedal Running. After formulating the dynamic equations for these legged systems, we are in place to control them. In this section, we introduce some key concepts to control legged locomotion from Lyapunov's perspective. This includes the virtual constraints, desired outputs, tracking control and Lyapunov stability. We first formally pose the control problem as a trajectory optimization problem, where a closed-loop system can be defined and a solution (gait) to the closed-loop system is found. These will be detailed using DURUS, a 3D underactuated humanoid robot, as an example. We will also use AMBER3M walking on slippery surface as an example the show the scalability of the optimization for high-performance locomotion. Ideally, tracking these solutions on the system with the pre-defined controller equates controlling the system as desired, which is true as we verified in simulation. However, the real-world suffers from

uncertainty such as inaccurate measurement. We then utilize the concept of input to state stability to analyze the tracking noise from the phasing variable for bipedal running on DURUS-2D. With the formally proven improved robustness, we can then realize bipedal running experimentally.

Chapter 4: Coupled Control System and Quadrupedal Applications. The aforementioned methods and application all build on full-order dynamics of these legged systems. While this is beneficial to realize high-performance robotics, it is not completely necessary. For example, multiple simplified models have been shown effective for quadrupedal locomotion. In this chapter we first show how to use off-line trajectory optimization to generate closed-loop trajectory for complex systems such as the quadrupedal robots of interest. Then we construct the concept of *coupled control systems*, which focuses on the control and optimization problem from a perspective of viewing the system as a set of subsystems coupled through control. In a concrete example, a quadrupedal robot can be viewed as two connected bipedal robots coupled through holonomic constraints. In the construction, we provide the stability conditions in the sense of Lyapunov and synthesize two algorithms. First, we proposed an optimization framework to rapidly generate gaits for the bipedal subsystem that can be reconstructed to quadrupedal gaits. Then we used coupled control Lyapunov functions to control each subsystem to achieve the stability of the full-order system. The result is Vision 60, version 3.2 and 3.9, two robots of different size and weight that can walk robustly indoor and outdoor on rough terrains.

Chapter 2

CONSTRAINED RIGID-BODY DYNAMICS

To formally construct the control problem of legged locomotion, we first convert the general *differential geometry equations* (DAE) formulation used in rigid-body dynamics into a control-friendly formulation — the *ordinary differential equation* (ODE) form $\dot{x} = f(x) + g(x)u$ — in this chapter. For this purpose, all internal variables need to be explicitly and uniquely determined by the state variables and control variables. Further, we employ the trajectory optimization technique to define and find numerical solutions to such systems. These solutions, i.e., *gaits*, are essentially what the controller is designed to drive the dynamics to in the content that follows.

2.1 Coordinates and Notations

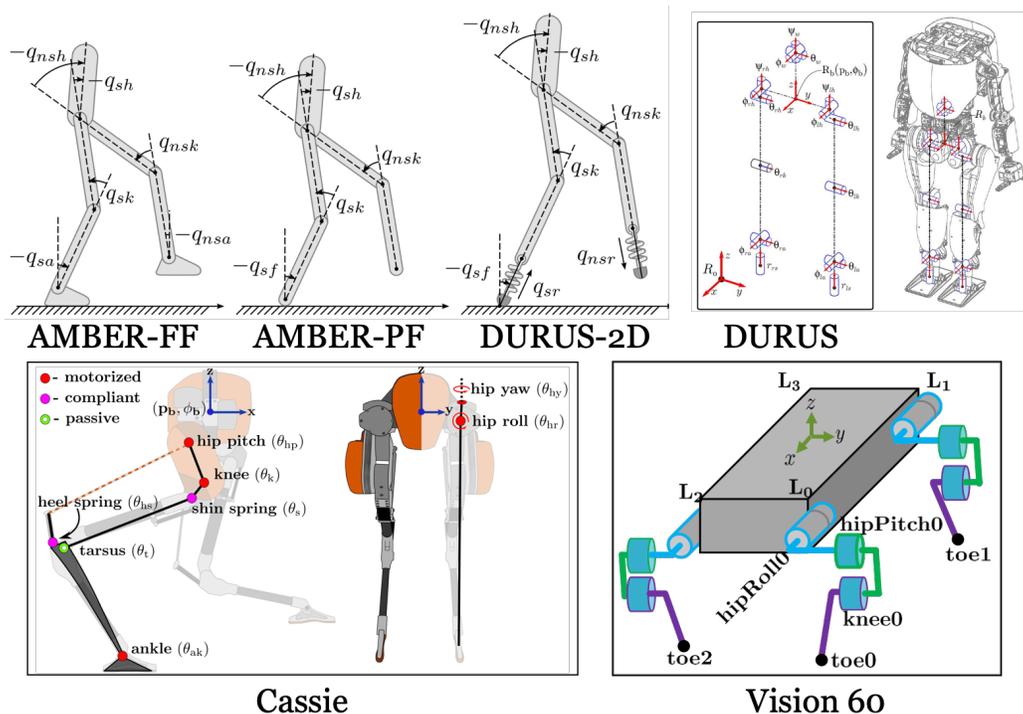


Figure 2.1: Configuration coordinates of AMBER robots, DURUS-2D running robot, DURUS, Cassie, and Vision 60, the quadrupedal robots.

The target of this section is to define the state coordinates for robotic systems such as those in Fig. 1.1. As shown in Fig. 2.1, we model each robot as a multi-link system.

The configuration coordinate is denoted by $q \in \mathcal{Q} \subset \mathbb{R}^n$, where n is the number of degrees of freedom (DOF) without considering any other constraints (referred to as *states-DOF*). Following (Jessy W. Grizzle et al., 2014), we use the floating-base convention for robotic systems. Each system starts from a *body* linkage as the base (normally picked as the torso link, or the ground for manipulation problems), with $\xi \in \mathbb{R}^3 \times \text{SO}(3)$ as the *global* Cartesian position and orientation of a frame attached to the *body* linkage. Then k -limbs are connected to the body, each of which has a few children links. These limbs are associated with *local* coordinates θ , including both prismatic joint length and revolute joint angles. Additionally, each robot is assumed to have m actuated joints and $m \leq n$. Here, ‘‘actuation’’ specifically refers to those joints included in the configuration coordinates that are driven by actuator modules, which are normally composed of a Brushless (BL) DC motor and a reduction gearbox. Concretely, we denote the input variable as $u \in \mathcal{U} \in \mathbb{R}^m$.

In summary, we use the following convention:

- $q = (\xi^\top, \theta^\top)^\top \in \mathbb{R}^n$ is used exclusively for a robot’s configuration coordinates, where ξ is the floating-base coordinates, θ is the ‘‘shape’’ coordinates representing the rotational and prismatic joints’ displacement, and their time-derivative is denoted as $\dot{q}, \dot{\xi}, \dot{\theta}$.
- y is for system-level outputs, and $\eta^\top = (y^\top, \dot{y}^\top)$.
- Normal form coordinates: $(\eta^\top, z^\top)^\top$, where z is the zero dynamics coordinates.

Notations. In this thesis, we denote that the set of natural numbers as \mathbb{N} ; the set of n by 1 real vectors as \mathbb{R}^n , the set of non-negative real numbers as \mathbb{R}_+ , the set of m by n matrices as $\mathbb{R}^{m \times n}$, an n by 1 vector whose elements are 1 as $\mathbf{1}_n$, a proper dimensional vector whose elements are 0s as $\mathbf{0}$, an n by n identity matrix as I_n , and an m by n zero matrix as $0_{m \times n}$. The Lie derivative of a function $f(x)$ along the vector field $g(\cdot)$ is defined as

$$L_{g(\cdot)}f(x) \triangleq \frac{\partial f(x)}{\partial x} g(\cdot). \quad (2.1)$$

Plus, some norm operators need to be defined. The Euclidean norm of a vector of proper dimension is $|\cdot|$, and we take

$$\|d\|_\infty \triangleq \sup_{t \geq 0} (|d(t)|). \quad (2.2)$$

The matrix norm induced by the Euclidean vector norm is $\|\cdot\|_2$, and the distance from a point (x, z) to a periodic orbit \mathcal{O} is defined as:

$$\|(x, z)\|_{\mathcal{O}} \triangleq \inf_{(x', z') \in \mathcal{O}} |(x, z) - (x', z')|. \quad (2.3)$$

2.2 Holonomic Constraints and System-Level Degrees of Freedom

We are interested in scenarios when the robots are shown in Fig. 1.1 interact with the world through rigid contacts, which are modeled as some form of *holonomic constraints* (for example, unilateral constraints, see (R. M. Murray et al., 1994)). In classical mechanics, *holonomic constraints* are defined relations between the position variables (Arnold, 1989). In this thesis, we denote a holonomic constraint as a function in the form of

$$h(q) \equiv c, \quad (2.4)$$

where $c \in \mathbb{R}^{n_h}$ is a constant, $h : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$, and the operator “ \equiv ”, identical equality of functions, means that $h(q) = 0$ is true for all $t \in \mathbf{I}$ that $q(t)$ is defined on.

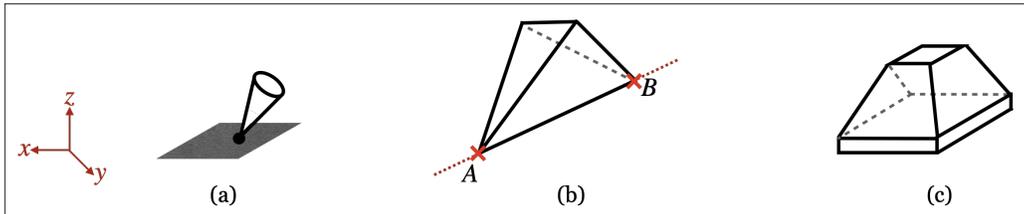


Figure 2.2: Three fundamental contact problems modelled by holonomic constraints: a) point contact, $h(q) \in \mathbb{R}^3$ denotes the three-dimensional Cartesian position fixed to the world; b) line contact, $h(q) \in \mathbb{R}^{3+2}$ fixed one point of the contact line and the pitch, yaw angles of the body-fixed frame to the world; c) plane contact, $h(q) \in \mathbb{R}^6$ fixed the 6-dimensional body-fixed frame to the world.

There are three fundamental contact problems in rigid-body dynamics: line contact, point contact, and plane contact. See Fig. 2.2. However, many complicated scenarios exist in reality, such as multiple contact points. Moving forward, we require that there be no redundancy when picking the representation of holonomic constraints. Mathematically, this means the *Contact Jacobian*,

$$J_h(q) \triangleq \frac{\partial h(q)}{\partial q} \in \mathbb{R}^{n_h \times n}, \quad (2.5)$$

is full rank. That is, $\text{Rank}(J_h) = \min(n_h, n)$. A simple example of redundancy is that in the line-contact problem shown in Fig. 2.2(b), either $(x_A, y_A, z_A, \theta_y, \theta_z) \equiv$

0, or $(x_A, y_A, z_A, x_B, y_B, z_B) \equiv 0$ can represent the contact condition. But the second formulation has one degree of redundancy, which is the result of rigid-body formulation: the distance between point A and B is a constant. Hence, we call $\text{Rank}(J_h)$ the *degree of contact constraint*. Recall the states-DOF n in Sec. 2.1, these holonomic contact constraints obviously change the *system-level degrees of freedom* (abbreviated as system-DOF). We then denote the total degrees of freedom as

$$\text{system-DOF} \triangleq n - \text{Rank}(J_h). \quad (2.6)$$

In consistence with this definition, we have $n \geq n_h$, and thus $\text{Rank}(J_h) = n_h$. Furthermore, this equality constraint $h(q)$ is enforced via *contact forces* (also known as constraint wrench) $\lambda \in \mathbb{R}^{n_h}$.

2.3 Continuous-Time Dynamics

The continuous-time dynamics of a robotic system will be expressed as a set of *differential algebra equations* (DAEs), which is composed of two parts: the equations of motion (EOMs), and some algebraic equations. Concretely,

$$D(q)\ddot{q} + C(q, \dot{q}) + g(q) = Bu + J_h^T(q)\lambda, \quad (2.7)$$

$$\text{s.t.} \quad h(q) \equiv 0. \quad (2.8)$$

We can obtain the EOMs in (2.7) by using the Euler–Lagrange equation, see (R. M. Murray et al., 1994; M. W. Spong, 1989; R. Featherstone, 2008). Here, $D(q) \in \mathbb{R}^{n \times n}$ represents the mass-inertia matrix, and is symmetric positive-definite; $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis matrix, and $g(q) \in \mathbb{R}^n$ contains the gravity terms. For the sake of notational simplicity, we denote

$$H(q, \dot{q}) \triangleq C(q, \dot{q}) + g(q)$$

throughout this thesis; the actuation matrix $B(q) \in \mathbb{R}^{n \times m}$ maps the motor input u from the control space \mathcal{U} to the configuration space \mathcal{Q} , which is a constant matrix; and $J_h \in \mathbb{R}^{n_h \times n}$ is the contact Jacobian, with $\lambda \in \mathbb{R}^{n_h}$ the contact force.

Remark 1. We note a special property of the mass-inertia matrix $D(q)$ when using the floating-base coordinate — the *branch-induced sparsity*. As detailed in (Roy Featherstone, 2010), there is a sparse structure for a multi-limbs robotic system such as a humanoid, or a quadrupedal robot. For example, as shown in Fig. 2.3, the

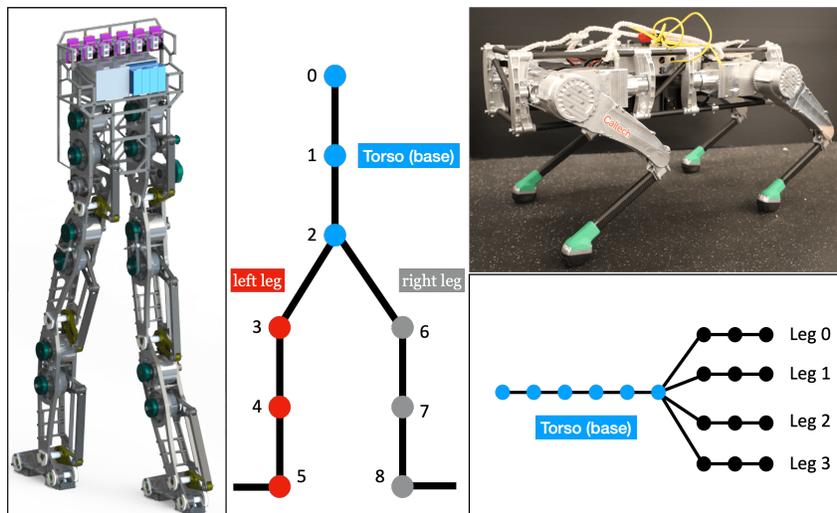


Figure 2.3: Branched chain structure of a bipedal system—AMBER-3M, and a quadrupedal system—Vision 60.

inertia matrix for a bipedal system and a quadrupedal system has the structure of

$$D_{\text{biped}} = \begin{bmatrix} D_{\xi} & D_{b_l} & D_{b_r} \\ D_{b_l}^{\top} & D_{\text{left}} & \mathbf{0} \\ D_{b_r}^{\top} & \mathbf{0} & D_{\text{right}} \end{bmatrix}, \quad D_{\text{quad}} = \begin{bmatrix} D_{\xi} & D_{b_0} & D_{b_1} & D_{b_2} & D_{b_3} \\ D_{b_0}^{\top} & D_{L_0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ D_{b_1}^{\top} & \mathbf{0} & D_{L_1} & \mathbf{0} & \mathbf{0} \\ D_{b_2}^{\top} & \mathbf{0} & \mathbf{0} & D_{L_2} & \mathbf{0} \\ D_{b_3}^{\top} & \mathbf{0} & \mathbf{0} & \mathbf{0} & D_{L_3} \end{bmatrix}. \quad (2.9)$$

This is particularly useful when formulating the coupled control problem of quadrupedal locomotion in Chapter 4.

2.4 Underactuated System

To eventually convert the DAEs given by (2.7)-(2.8) to an ODE form, we need to explicitly solve the internal variable λ with q, \dot{q}, u . But first, we need to systematically define the notion of *underactuated*, *fully-actuated*, and *overactuated* mechanical systems and their degrees.

Similar to how the contact force enters the EOMs of (2.7), $J(q)^{\top} \lambda$, the joint torque (or linear actuator's force) enters the system in the same way, Bu . Indeed, both are obtained via the *principle of virtual work* (Lanczos, 1986) (formally extended to dynamical systems as the *Lagrange–d'Alembert principle*, see (Arnold, 1989)), which is a specific application of the *method of Lagrange multiplier* (R. M. Murray et al., 1994, Chapter 6). Therefore, just as a complicated contact situation can cause redundancy among the contacts, the joint torque u could also “fight” with the contact

force λ (see Fig. 2.4(c) where the control variable u fights with the ground reaction force), which is the overactuation problem.

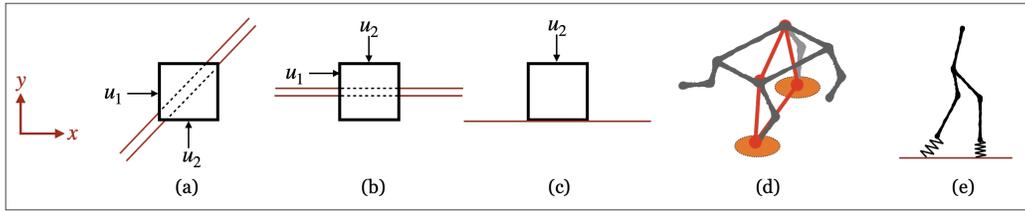


Figure 2.4: Simple examples of overactuation in the presence of holonomic constraints: (a,b) sliding on a track; (c) pushing against a wall; (d) a quadrupedal robot standing on two feet.

Define the following matrix as the *Actuation Jacobian*:

$$B_A \triangleq \begin{bmatrix} B^\top \\ J_h \end{bmatrix} \in \mathbb{R}^{(m+n_h) \times n}. \quad (2.10)$$

Note that $\text{Rank}(B_A) \leq n$.

Definition 1. If $\text{Rank}(B_A) < n$, we say the full system is *under-actuated*, and

$$n_U = n - \text{Rank}(B_A) \quad (2.11)$$

is defined as the *degrees of under-actuation for the full system*.

Definition 2. If $\text{Rank}(B_A) = n$, and $m + n_h > n$, the system is *overactuated*, and

$$n_O = m + n_h - \text{Rank}(B_A) \quad (2.12)$$

is defined as the *degrees of overactuation for the full system*.

Example.

- The system in Fig. 2.4 (a) has dynamics:

$$\begin{cases} m\ddot{x} = u_2 + \lambda \\ m\ddot{y} = u_1 - \lambda \\ \ddot{x} - \ddot{y} = 0 \end{cases} \Rightarrow B_A = \begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow n_O = 2 + 1 - 2 = 1.$$

- The system in Fig. 2.4 (b) has dynamics:

$$\begin{cases} m\ddot{x} = u_2 \\ m\ddot{y} = u_1 - \lambda \\ \ddot{y} = 0 \end{cases} \Rightarrow B_A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow n_O = 2 + 1 - 2 = 1.$$

Full system *overactuation* can be interpreted as:

- Control inputs fight with each other (Fig. 2.4(a));
- Control inputs fight with constraint force λ (Fig. 2.4(b,c)).

Definition 3. A system that is neither underactuated nor overactuated is said to be *fully actuated*.

Definition 4. If the actuation matrix B_A is not full rank (degenerate), i.e., $\text{Rank}(B_A) < \min(m + k, n)$, the system is *internally overactuated*. The *degree of internal overactuation* is defined as

$$n_{\text{IO}} = \min(m + n_h, n) - \text{Rank}(B_A).$$

The internally overactuation is often caused by some *closed-chain* structure, such as the double-support scenario of the quadrupedal locomotion with two-point contacts on the toes. As shown in Fig. 2.4(d), the red line indicates the closed-chain inside the system. Also, as shown in Fig. 2.4(e), right after an impact, the nonstance foot's spring needs some time to recover to the normal length while the stance spring is compressing itself. This results in a double-support domain, which also creates an internal chained structure and internal overactuation.

To uniquely determine λ with q, \dot{q}, u , we require the dynamics given by (2.7)-(2.8) have no overactuation or internally overactuation. In other words, J^h , B , and B_A are all full-rank, and $n_{\text{O}}, n_{\text{IO}} = 0$. This can be achieved by enforcing additional algebraic equations, such as setting some motor inputs to be 0. The reason is that a violation of this requirement will result in an underdetermined problem when solving for λ . This will be seen in the next section. In summary, we demand our problem to be either underactuated or fully-actuated.

Remark 2. It is unnecessary to have J_h , B , and the actuation matrix full-rank in real-world applications. For example, an all-wheel-drive (AWD) vehicle is apparently overactuated. Although it has no mathematical difference from a rear-wheel-drive vehicle in terms of rigid-body dynamics, the control robustness differs significantly. Further, a degenerate matrix formulation will make the stability analysis later using the traditional control language over complicated. This is beyond the scope of this thesis, but interested readers can find the optimization-based analysis for such problems in the literature of *control allocation* such as (Johansen and Fossen, 2013).

2.5 The ODE Form of Control-Affine Systems.

We now convert a DAE system given by (2.7)-(2.8) into a ODE system. We first note that (2.8) is an equality constraint placed on $q(t) \forall t \in \mathbf{I}$. To explicitly solve λ , we take the derivatives of (2.8) to get:

$$J_h(q)\dot{q} = 0; \quad (2.13)$$

$$\dot{J}_h(q, \dot{q})\dot{q} + J_h(q)\ddot{q} = 0. \quad (2.14)$$

Therefore, (2.7)-(2.8) became

$$\begin{cases} D(q)\ddot{q} + H(q, \dot{q}) = Bu + J_h^\top \lambda \\ \dot{J}_h(q, \dot{q})\dot{q} + J_h(q)\ddot{q} = 0 \end{cases}. \quad (2.15)$$

When compute the constraint wrench in (2.7), we can easily derive the following

$$\begin{aligned} (J_h D^{-1} J_h^\top) \lambda &= J_h D^{-1} H - J_h D^{-1} B u - \dot{J}_h \dot{q} \\ \Rightarrow \lambda &= (J_h D^{-1} J_h^\top)^{-1} \left(J_h D^{-1} H - J_h D^{-1} B u - \dot{J}_h \dot{q} \right), \end{aligned} \quad (2.16)$$

where we suppressed the arguments of $D(q)$, $H(q, \dot{q})$, $J_h(q)$ and $\dot{J}_h(q, \dot{q})$. Note that if J_h is not full rank, $J_h D^{-1} J_h^\top$ becomes a singular matrix, in which case we cannot uniquely determine λ .

Plugging (2.16) into (2.7), we have

$$\begin{aligned} D\ddot{q} + H &= B u + J_h^\top (J_h D^{-1} J_h^\top)^{-1} \left(J_h D^{-1} H - J_h D^{-1} B u - \dot{J}_h \dot{q} \right) \\ \Rightarrow \ddot{q} &= \underbrace{-D^{-1} H + D^{-1} J_h^\top (J_h D^{-1} J_h^\top)^{-1} \left(J_h D^{-1} H - \dot{J}_h \dot{q} \right)}_{\text{drift vector}} + \\ &\quad \left(I_n - D^{-1} J_h^\top (J_h D^{-1} J_h^\top)^{-1} J_h \right) D^{-1} B u. \end{aligned} \quad (2.17)$$

Another form often used in the control + robotics literature is the manipulator dynamics:

$$\underbrace{D\ddot{q} + H - J_h^\top (J_h D^{-1} J_h^\top)^{-1} \left(J_h D^{-1} H - \dot{J}_h \dot{q} \right)}_{\triangleq \bar{H}(q, \dot{q})} = \underbrace{\left(I_n - J_h^\top (J_h D^{-1} J_h^\top)^{-1} J_h D^{-1} \right)}_{\triangleq \bar{B}(q)} B u. \quad (2.18)$$

Further, we define the state space $\mathcal{X} = TQ \subseteq \mathbb{R}^{2n}$ with the state vector:

$$x \triangleq \begin{bmatrix} q \\ \dot{q} \end{bmatrix},$$

where TQ is the tangent bundle of the configuration space Q . Finally, we have the dynamical system that we wish to design controllers for:

$$\dot{x}(t) = f(x(t)) + g(x(t))u,$$

which will be abbreviated as

$$\dot{x} = f(x) + g(x)u \quad (2.19)$$

with

$$\begin{aligned} f(x) &\triangleq \begin{bmatrix} \dot{q} \\ -D^{-1}H + D^{-1}J_h^\top (J_h D^{-1} J_h^\top)^{-1} (J_h D^{-1} H - \dot{J}_h \dot{q}) \end{bmatrix}, \\ g(x) &\triangleq \begin{bmatrix} \mathbf{0} \\ (I_n - D^{-1}J_h^\top (J_h D^{-1} J_h^\top)^{-1} J_h) D^{-1} B \end{bmatrix}. \end{aligned} \quad (2.20)$$

2.6 Hybrid System

Having established the continuous-time dynamics for a particular contact scenario, we are in the position to define the hybrid model for walking. The motivation of using hybrid dynamics is straightforward: for legged locomotion, when the contact condition changes — such as adding or removing some contacts — the input (contact force, motor torque) changes, hence the dynamics change. The combination of these “domains”¹ of dynamics results in a *hybrid automata* (Barton and C. K. Lee, 2002), which is given by the following definition.

Definition 5. A *hybrid control system* is defined to be the tuple:

$$\mathcal{HC} = (\Gamma, \mathcal{D}, \mathcal{S}, \Delta, \mathbb{F}), \quad (2.21)$$

where,

- $\Gamma \triangleq \{\mathbb{V}, \mathbb{E}\}$ is a *directed graph* (or digraph) (see (Mesbahi and Egerstedt, 2010)), with \mathbb{V} the set of domain indices and \mathbb{E} the set of edges between two domains. Their elements are denoted as $v \in \mathbb{V}, e \in \mathbb{E}$, where $e \triangleq (v \rightarrow w)$ with $v, w \in \mathbb{V}$;
- $\mathcal{D} \triangleq \{\mathcal{D}_v\}_{v \in \mathbb{V}}$ is a set of admissible *domains*, where the state space is defined on;

¹“Domain” is interchangeably used with “phase” in this thesis.

- $\mathcal{S} \triangleq \{\mathcal{S}[e]\}_{e \in \mathbb{E}}$ is a set of *guards* (or switching surfaces), where $\mathcal{S} \subset \mathcal{D}_v \cap \mathcal{D}_w$, with $v, w \in \mathbb{V}$;
- $\Delta \triangleq \{\Delta[e]\}_{e \in \mathbb{E}}$ is a set of *reset maps* that maps the pre-edge states to the post-edge states. This will be detailed in the next section;
- $\mathbb{F} \triangleq \{\mathbb{F}_v\}_{v \in \mathbb{V}}$ is a set of control systems that determine the continuous-time dynamics on a domain \mathcal{D}_v .

Discussion. In application, we denote a specific contact condition as $h_v(q) \equiv c_v$ for each domain \mathcal{D}_v with $v \in \mathbb{V}$. Moreover, the satisfaction of this constraint defines the state space, i.e., the continuous dynamics on domain \mathcal{D}_v is given by

$$\dot{x} = f_v(x) + g_v(x)u, \quad x \in \mathcal{D}_v. \quad (2.22)$$

The guard \mathcal{S}_e associated an edge $e \triangleq (v \rightarrow w)$ with $v, w \in \mathbb{V}$ is normally defined by the change of states from $x \in \mathcal{D}_v$ to $x \in \mathcal{D}_w$. The control law $\{FG_v\}_{v \in \mathbb{V}}$ will be the main topic of the next chapter.

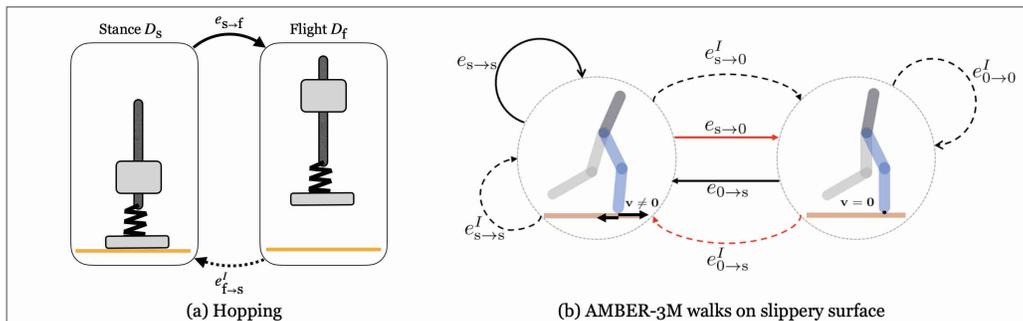


Figure 2.5: Directed graph for two robotic systems.

We now give two examples of formulating the hybrid model for low-dimensional legged systems.

A hopping machine. A canonical example of the hybrid dynamical system is hopping robots. In the simplified vertical hopping machine shown in Fig. 2.5(a), the robot hops up and down only. It then has two dynamic domains, the stance domain \mathcal{D}_s , where the foot (bottom link) stays on the ground, and the flight domain \mathcal{D}_f when the foot left the ground. Formally, this is defined as

$$\mathcal{D}_s \triangleq \{x \in TQ \mid z_{\text{foot}} = \dot{z}_{\text{foot}} = 0\}, \quad (2.23)$$

$$\mathcal{D}_f \triangleq \{x \in TQ \mid z_{\text{foot}} \geq 0\}, \quad (2.24)$$

where z_{foot} is the height of the foot. We then denote λ_{foot} as the ground reaction force exerting on the foot. Note that λ_{foot} can be explicitly solved with x, u using (2.16). Therefore, we have the domain indices $\mathbb{V} \triangleq \{s, f\}$, and the set of edges as

$$\mathbb{E} \triangleq \left\{ \underbrace{e_{s \rightarrow f}}_{\text{“lift”}}, \underbrace{e_{f \rightarrow s}}_{\text{“impact”}} \right\}, \quad (2.25)$$

where $e_{s \rightarrow f}$ marks the event that foot leaves from the ground, and $e_{f \rightarrow s}$ is the event that foot touch the ground from the air. Thus, we can use the boundary condition of each domain to define the guard conditions as:

$$\mathcal{S}[e_{s \rightarrow f}] \triangleq \{(x, u) \in \mathcal{D}_s \mid \lambda_{\text{foot}}(x, u) = 0\}, \quad (2.26)$$

$$\mathcal{S}[e_{f \rightarrow s}] \triangleq \{x \in \mathcal{D}_f \mid z_{\text{foot}} = 0, \dot{z}_{\text{foot}} < 0\}. \quad (2.27)$$

The transition dynamics for each edge $e \in \mathbb{E}$, are called *lift dynamics* $\Delta[e_{s \rightarrow f}]$, and the *impact dynamics* $\Delta[e_{f \rightarrow s}]$ will be detailed in the next section.

AMBER-3M walks on slippery surfaces. The cyclic directed graph of the multi-domain hybrid system for walking on a slippery surface is shown in Fig. 2.5(b). The solid lines are for transitions without non-stance foot impact events, and dashed lines are for transitions with impact events. The subscript s is for *walking with slippage*, 0 is *walking without slippage*, and the superscript *I* marks that the edge is equipped with an impact event. As shown, we have two types of dynamic domains: the stick domain \mathcal{D}_0 , where the contact foot stands on the ground without slippage, and the slip domain \mathcal{D}_s , where the contact foot slides the ground. Formally, this is given as

$$\mathcal{D}_0 = \{(x, u) \in TQ \times \mathcal{U} \mid z_s = \dot{z}_s = \dot{x}_s = 0, z_{ns} \geq 0, |\lambda_x^0| \geq \mu \lambda_z^0\}, \quad (2.28)$$

$$\mathcal{D}_s = \{(x, u) \in TQ \times \mathcal{U} \mid z_s = \dot{z}_s = 0, \dot{x}_s \neq 0, z_{ns} \geq 0, \lambda_z^s \geq 0\}, \quad (2.29)$$

where x_s, z_s represent the Cartesian position of the stance foot along the horizontal and vertical axes, accordingly; and z_{ns} is the height of the nonstance foot, λ_x, λ_z are the ground reaction forces along the horizontal and vertical axes. Hence, we have the domain indices $\mathbb{V} = \{0, s\}$. Correspondingly, we have the set of edges

$$\mathbb{E} = \{e_{0 \rightarrow s}, e_{s \rightarrow 0}, e_{s \rightarrow s}, e_{0 \rightarrow 0}^I, e_{0 \rightarrow s}^I, e_{s \rightarrow 0}^I, e_{s \rightarrow s}^I\}. \quad (2.30)$$

The superscript “*I*” denotes transition via impact, whereas its absence denotes stick \leftrightarrow slip transitions. We now define the guards, which are switching surfaces or conditions for transition between domains. The first guards are associated with the

smooth transitions between sticking and slipping domains:

$$\begin{aligned}\mathcal{S}[e_{0 \rightarrow s}] &= \{(x, u) \in \mathcal{D}_0 \mid |\lambda_x^0| = \mu \lambda_z^0\}, \\ \mathcal{S}[e_{s \rightarrow 0}] &= \{(x, u) \in \mathcal{D}_s \mid \dot{x}_s = 0, |\lambda_x^0| \leq \mu \lambda_z^0\}, \\ \mathcal{S}[e_{s \rightarrow s}] &= \{(x, u) \in \mathcal{D}_s \mid \dot{x}_s = 0, |\lambda_x^0| > \mu \lambda_z^0\}.\end{aligned}$$

Note that the last transition above associated with $e_{s \rightarrow s}$ is reversal of slip direction (cf. (Gamus and Yizhar Or, 2015)). The guards corresponding to transitions that involve sticking or slipping impacts are defined as:

$$\begin{aligned}\mathcal{S}[e_{0 \rightarrow 0}^I] &= \{(x, u) \in \mathcal{D}_0 \mid z_{ns} = 0, \dot{z}_{ns} < 0, |\Lambda_x^0| \leq \mu \Lambda_z^0\}, \\ \mathcal{S}[e_{s \rightarrow 0}^I] &= \{(x, u) \in \mathcal{D}_s \mid z_{ns} = 0, \dot{z}_{ns} < 0, |\Lambda_x^0| \leq \mu \Lambda_z^0\}, \\ \mathcal{S}[e_{0 \rightarrow s}^I] &= \{(x, u) \in \mathcal{D}_0 \mid z_{ns} = 0, \dot{z}_{ns} < 0, |\Lambda_x^0| > \mu \Lambda_z^0\}, \\ \mathcal{S}[e_{s \rightarrow s}^I] &= \{(x, u) \in \mathcal{D}_s \mid z_{ns} = 0, \dot{z}_{ns} < 0, |\Lambda_x^0| > \mu \Lambda_z^0\},\end{aligned}$$

where Λ_x, Λ_z are the impact impulses along the horizontal and vertical axes. These guards represent the conditions for sticking or slipping impacts as described above. Note that the overall non-smooth frictional dynamics may have special degenerate cases where the solution is inconsistent, indeterminate, or singular. These rare cases are known as *Painlevé paradox* (Champneys and Várkonyi, 2016; Yizhar Or, 2014), and lie beyond the scope of this work.

We remark that the sequence of motion in Fig. 2.5 that is marked by red color, has been controlled (see Sec. 3.3) and realized on the robot AMBER-3M, in (Wen-Loong Ma, 2019a).

2.7 Discrete Transitions

For legged locomotion, a change of contact condition² will result in either an event of *lift* or *impact*, or simultaneously both. We can characterize this “jump” between domains as *discrete transitional events*, which will be detailed in this section.

Impact-transition dynamics. Legged locomotion inevitably includes foot-impacts with the ground. This process can be modeled through complicated terrain dynamics such as the impact on granular media (Li, Zhang, and Goldman, 2013; A. H. Chang et al., 2017) and elastic or inelastic collisions. In this work, for the mathematical

² This not only includes foot impacts with the ground but also whenever a constraint ((2.8)) changes. For example, a hard stop of the spring vibration.

consistency with the assumption that the robots stand on the rigid ground — based on which the holonomic constraints can be defined — we model the foot impact as the *perfect inelastic collision* (also regarded as *plastic impact* in this thesis). Therefore, it obeys the *conservation of momentum*. And the positional variable remains unchanged through the event of an impact, and the velocity terms jump from the pre-impact quantity to the post-impact state. Formally, we model the impact dynamics from domain \mathcal{D}_w to domain \mathcal{D}_v — the transition dynamics on the edge $e_{w \rightarrow v}$ with $v, w \in \mathbb{V}$ — as:

$$\begin{cases} q_e^+ = q_e^- \\ \dot{q}_e^+ = \Delta[e_{w \rightarrow v}](q_e^-) \dot{q}_e^- \end{cases}, \quad (2.31)$$

where q_e^+, \dot{q}_e^+ are the pre-event states of the edge $e \triangleq e_{v \rightarrow w}$, i.e., these are the initial conditions of domain \mathcal{D}_v ; q_e^-, \dot{q}_e^- are the post-event states of the edge $e = e_{v \rightarrow w}$, i.e., these are the final conditions of domain \mathcal{D}_w . Note that we shorted the notation of $e_{w \rightarrow v}$ to e for simplicity in this section.

We now detail the transitional map $\Delta[e_{w \rightarrow v}]$. This map is obtained first via the law of momentum conservation:

$$D(q_e^-)(\dot{q}_e^+ - \dot{q}_e^-) = J_{h_v}(q_e^-)^\top \Lambda_e, \quad (2.32)$$

where $\Lambda_e \in \mathbb{R}^{n_{h_v}}$ is the impact impulse exerted on the contact surface. Entering domain \mathcal{D}_v means that the post-impact holonomic condition $h_v(q_e^+) \equiv c_v$ also holds. This yields:

$$J_{h_v}(q_e^+) \dot{q}_e^+ = \mathbf{0}. \quad (2.33)$$

Combining (2.31)-(2.33), we have the impact map as:

$$\begin{bmatrix} \dot{q}_e^+ \\ \Lambda_e \end{bmatrix} = \begin{bmatrix} D(q_e^-) & -J_{h_v}(q_e^-)^\top \\ J_{h_v}(q_e^-) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} D(q_e^-) \dot{q}_e^- \\ \mathbf{0} \end{bmatrix}, \quad (2.34)$$

which simultaneously gives the impulse and post-impact states, and this is referred to as the *impact dynamics*. As can be seen, this jump of states from \dot{q}_e^- to \dot{q}_e^+ results in a discrete transition. Additionally, removing some subset of the constraints defined on the previous domain, i.e., lifting, does not result in any impulse. Hence from (2.32), we then have the *lift transition dynamics* as:

$$\begin{cases} q_e^+ = q_e^- \\ \dot{q}_e^+ = \dot{q}_e^- \end{cases}. \quad (2.35)$$

We now unify (2.32) and (2.35) as the *transition dynamics* using state vectors. This is given as

$$x_v^+ = \Delta[e_{w \rightarrow v}](x_w^-). \quad (2.36)$$

The more general transition dynamics that involve impact with the slippery condition can be found in (W. Ma, Y. Or, and A D. Ames, 2019).

2.8 Solution to Hybrid Dynamics

Having established the multi-domain hybrid dynamics of a legged system as:

$$\mathcal{H} \triangleq \begin{cases} \dot{x} = f_v(x) + g_v(x)u & x \in \mathcal{D}_v, \forall v \in \mathbb{V} \\ x^+ = \Delta[e](x^-) & x^- \in \mathcal{S}[e], \forall e \in \mathbb{E} \end{cases}, \quad (2.37)$$

we hereby define the solutions to a multi-domain hybrid system and provide an optimization method to find the numerical solution. This section is an extension of the single-domain hybrid system given by (Aaron D. Ames et al., 2017).

Definition 6. A solution (execution) to the hybrid system \mathcal{H} is a tuple:

$$\chi = (\mathcal{V}, \mathcal{I}, \mathcal{C}), \quad (2.38)$$

where,

- $\mathcal{V} = (v_1, v_2, \dots)$ with $v_i \in \mathbb{V}$ and $i \in \mathbb{N}^+$, is a finite or countably infinite *domain sequence*³, and we denote its cardinality as N .
- $\mathcal{I} = (I_1, I_2, \dots)$ is a sequence of the time interval I_i with $i \in \mathbb{N}^+$, for each executed domain v_i . For each $v_i \in \mathcal{V}$, I_i is defined as:
 - $I_1 = [0, T_1]$, if $T_1 \neq \infty$, otherwise $I_1 = [0, \infty)$,
 - $I_i = [T_{i-1}, T_i]$, if $v_{i-1}, v_i \in \mathcal{V}$ and $T_i \neq \infty$,
 - $I_N = [T_{N-1}, T_i]$, if $i > 1$ and $T_i = \infty$
- $\mathcal{C} = (\{x_1(t), u_1(t)\}, \{x_2(t), u_2(t)\}, \dots)$ is a sequence of continuous trajectories for the states $x_i(t)$ and control inputs $u_i(t)$ that satisfies

$$\dot{x}_i(t) = f_{v_i}(x_i(t)) + g_{v_i}(x_i(t))u_i(t), \quad \forall t \in I_i.$$

We additionally require that for each $v_i \in \mathcal{V}$,

³Note that unlike a set, v_i and v_j with $i \neq j$ can be the same, i.e., the same domain can appear more than once in the solution's domain sequence. Further, note the sequence of domains \mathcal{V} is different from the set of domains \mathbb{V} from Def. 5.

- $x_i(t) \in \mathcal{D}_{v_i}, \quad \forall t \in I_i,$
- $x_i(T_i) \in \mathcal{S}[e_{v_i \rightarrow v_{i+1}}]$ and $\Delta[e_{v_i \rightarrow v_{i+1}}](x_i(T_i)) = x_{i+1}(T_i)$ if $v_{i+1} \in \mathcal{V}.$

We regard $x_0 \triangleq x_1(0)$ the initial condition of χ . A mixed sequence of walking tasks of a bipedal robot can serve as a good example of this definition. As shown in Fig. 2.6, we can define four types of domain: \mathcal{D}_0 for standing still, \mathcal{D}_f for walking on a flat surface, \mathcal{D}_u for walking uphill, \mathcal{D}_d for walking downhill. Hence we have the set of domain as $\mathcal{D} = \{\mathcal{D}_f, \mathcal{D}_u, \mathcal{D}_d\}$, and the set of guards $\mathcal{S} = \{\mathcal{S}[e_{f \rightarrow f}], \mathcal{S}[e_{f \rightarrow u}], \mathcal{S}[e_{u \rightarrow f}], \mathcal{S}[e_{u \rightarrow u}], \mathcal{S}[e_{f \rightarrow d}], \mathcal{S}[e_{d \rightarrow f}], \mathcal{S}[e_{d \rightarrow d}]\}$. We now assume there exists a solution to this hybrid system such that the robot starts from standing still, then walk two steps on flat ground, two steps uphill, two steps on the hill, then keeps walking on the flat ground. Therefore according to Def. 6, we have domain sequence as $\mathcal{V} = (\mathcal{D}_f, \mathcal{D}_f, \mathcal{D}_u, \mathcal{D}_u, \mathcal{D}_f, \mathcal{D}_f, \mathcal{D}_d, \mathcal{D}_d, \mathcal{D}_f, \mathcal{D}_f, \mathcal{D}_f, \dots).$ Correspondingly, we can obtain the sequence of smooth trajectories C .

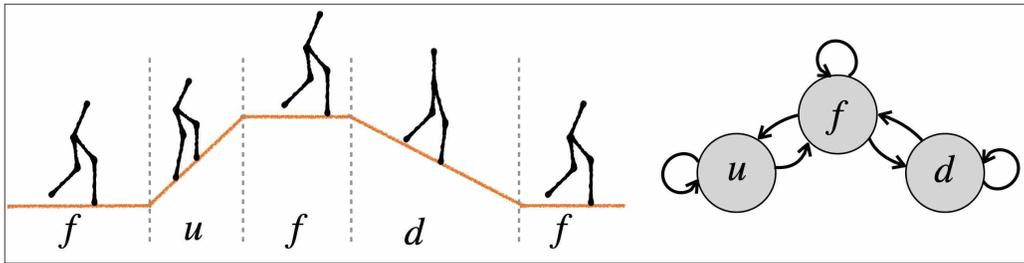


Figure 2.6: A mixed sequence of walking tasks of a bipedal robot: flat ground walking, uphill walking, downhill walking, flat ground walking. On the right is the directed graph, where the arrows represent the corresponding edges. Each edge is associated with transition dynamics.

2.9 Numerical Optimization

As given in (2.38), a solution to the hybrid dynamics is not necessarily a physically realizable trajectory. For example, the trajectory for the torque input $u(t) \forall t$ may violate the actuator's physical limitation. This section introduces the *numerical optimization* formulation of finding the solution to a nonlinear system in (2.37) that respects some inequality constraints. We will apply this formulation throughout this thesis. The history of finding solutions to nonlinear dynamics using numerical optimization is rich in all engineering areas. An exhaustive review of numerical optimization can be found in (Betts, 1998). We want to find a solution as (2.38) that also satisfies some inequality constraints in this section. In particular, we employ *direct collocation method* (Kelly, 2017; Ayonga Hereid, Shishir Kolathaya, et al.,

2014) to accomplish this goal. This method essentially transcribes the solution-finding problem into a set of algebraic equations via collocation schemes. This is an implicit Runge-Kutta method to solve the system dynamics simultaneously. There are many types of collocation schemes, but we will focus on a specific method: the *local direct collocation method*.

Unlike the single shooting method for the initial value problems (IVP) (Kiehl, 1994), where the solution $x(t) \forall t$ is obtained explicitly by integrating the dynamics from $t = t_0$, direct collocation formulation approximates the solution $x(t)$ implicitly by piecewise polynomials along the evolution of the nonlinear dynamics.

We first present the algorithm to find the solution $x(t) \forall t \in I_i$ using *defect constraints*. The basic procedures are:

Step1. We begin by defining a sequence of $K + 1$ discrete *nodes* along the time span of the trajectory:

$$T_{i-1} = t_0 < t_1 < t_2 < \dots < t_K = T_i, \quad (2.39)$$

which forms the basis of our discrete representation of the continuous dynamics, and we denote $\delta t^k = t_{k+1} - t_k$, where $k \in (0, 1, \dots, K_i)$. Note that K_i depends on the domain v_i , but we will omit the subscript i for the ease of notation.

Step2. We use a piecewise Hermite interpolation (cubic) polynomial to approximate the solution over each subinterval $[t_k, t_{k+1}]$ using the following cubic polynomial:

$$x(t) = C_0 + C_1 s + C_2 s^2 + C_3 s^3 \quad (2.40)$$

$s = \frac{t-t_k}{t_{k+1}-t_k} \in [0, 1]$ is the normalized time within the subinterval. The four coefficients C_0, C_1, C_2, C_3 are determined by the two boundary values x^k, x^{k+1} , which are given as decision variables of the optimization algorithm. Their derivatives \dot{x}^i and \dot{x}^{i+1} are computed by the continuous dynamics in (2.37).

Step3. Using these coefficients, we can compute the interpolated value of x at the center of the subinterval, i.e., the *collocation point*, as

$$\bar{x}_c^k = (x^k + x^{k+1})/2 + \delta t^k (\dot{x}^k - \dot{x}^{k+1})/8. \quad (2.41)$$

Similarly, the slope of the cubic polynomial at the center point is

$$\dot{\bar{x}}_c^k = -3(x^k - x^{k+1})/2\delta t^k - (\dot{x}^k + \dot{x}^{k+1})/4. \quad (2.42)$$

Step4. The *defect constraint* at the center of the subinterval is defined as the difference between the interpolated slope and the first-order derivatives of states at the center of the subinterval computed by the system dynamics, i.e.,

$$\zeta^k = f(\bar{x}_c^k) + g(\bar{x}_c^k)u_c^k - \dot{x}_c^k. \quad (2.43)$$

We now have the most important statement in this section: $(\bar{x}_c^k, \dot{x}_c^k)^T$ satisfies the dynamic constraints (C1) if and only if $\zeta^k = 0$. In fact, (2.43) is a nonlinear equality constraint posed on the nodes t^k and t^{k+1} .

Step5. Stack these defect constraints into a vector as

$$Z_i \triangleq \begin{bmatrix} \zeta^0 \\ \vdots \\ \zeta^{K-1} \end{bmatrix}, \quad (2.44)$$

in which Z becomes a function of the discretized states

$$\mathbf{x}_i \triangleq (x^0, x^1, \dots, x^K),$$

and control inputs

$$\mathbf{u}_i \triangleq (u^0, u_c^0, u^1, u_c^1, u^2, \dots, u^K).$$

Then the goal is to “find” a set of discretized states \mathbf{x}_i so that the defect $\Delta_i = 0$ for domain $v_i \in \mathcal{V}$. As a result, the interpolated piecewise polynomials are a valid approximation of the solution $x(t) \forall t \in I_i$. Note that there are two ways to improve the smoothness of control inputs: $u_c^k \equiv u^k$ and $u_c^k \equiv (u^k + u^{k+1})/2$, but these additional constraints are not required.

To find a solution to the hybrid system (2.38), we then encode the defect constraint (2.44) into a nonlinear program as follows:

$$\begin{aligned} \min_{\mathbf{X}} \quad & J(\mathbf{X}) & (2.45) \\ \text{s.t.} \quad & \text{(C1)} \quad Z_i(\mathbf{x}_i, \mathbf{u}_i) = 0 & \forall i \in (1, 2, \dots, N) \\ & \text{(C2)} \quad x_{i+1}^1 = \Delta[e](x_i^{K_i}) & \forall e \in \mathbb{E} \\ & \text{(C3)} \quad p(\mathbf{X}) \leq 0 \\ & \text{(C4)} \quad b(\mathbf{X}) = 0 \end{aligned}$$

where, $J(\mathbf{X})$ is a user-defined cost function; x_i^1 and $x_i^{K_i}$ are the first and last value of domain v_i and v_{i+1} , who are connected through edge e ; $p(\cdot)$ and $b(\cdot)$ are user-defined inequality and equality constraints. Further, the decision variable is given as

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{u}_1, \dots, \mathbf{u}_N, T_1, \dots, T_N).$$

We can then solve this optimization problem using nonlinear solvers such as SNOPT (Gill, W. Murray, and Saunders, 2005), IPOPT (Wächter and Biegler, 2006), GPOPS (Hager et al., 2019), PSOPT (Becerra, 2010),s and FROST (Ayonga Hereid and Aaron D. Ames, 2017). The collocation constraint $Z_i(\mathbf{X}) = \mathbf{0}$ is a stage-3, implicit Runge-Kutta method to solve dynamical systems. When the number of subinterval increases, the approximated solution becomes closer to the exact solution of the continuous dynamics. On the contrary, a shooting method finds a solution to the dynamical system by solving

$$x(t) = \int_{t_0}^{t_f} f(x(s)) + g(x(s))u(s) ds + x(0)$$

using explicit Runge-Kutta methods. The implementation of this methodology on simple and rigid systems is acceptable, see (Shishir Kolathaya, Wen-Loong Ma, and Aaron D Ames, 2015; H.-H. Zhao et al., 2014). However, it suffers from computational cost to maintain numerical stability when it comes to a high-dimensional problem. The collocation method instead simulates the dynamics implicitly because the solution at each sample time is picked by the optimization solver simultaneously. This made it expandable to high-dimensional and non-stiff systems. We will use a simple hopping robot example below to show this implementation.

Example. We now present a simple example, where we find the solution to a 1D hopping system using a nonlinear programming (NLP) toolbox —PSOPT(Becerra, 2010). In particular, we will utilize the local discretization methods to parse our problem.

The robot of interest is shown at Fig. 2.7, we write the equations of motion as

$$\begin{cases} \ddot{z}_m = \frac{u}{m_m} - g \\ \ddot{z}_r = -\frac{u}{m_r} - \frac{F_s}{m_r} - g \\ \ddot{z}_f = \frac{F_s}{m_f} - g \end{cases} \quad \forall x \in \mathcal{D}_f \quad (2.46)$$

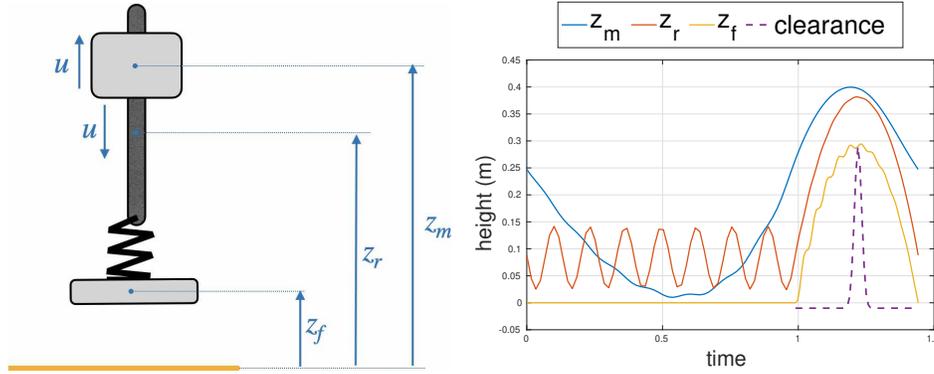


Figure 2.7: Left: the configuration coordinate of the 1D hopper. Right: The optimized trajectory from (2.51).

for the flight phase; and

$$\begin{cases} \ddot{z}_m = \frac{u}{m_m} - g \\ \ddot{z}_r = -\frac{u}{m_r} - \frac{F_s}{m_r} - g \\ \ddot{z}_f = 0 \end{cases} \quad \forall x \in \mathcal{D}_s \quad (2.47)$$

for the stance phase. The directed graph is given in Fig. 2.5 (a), and the domain definitions are given in (2.23). The other notation is explained as: the state variable is $x^\top = (q^\top, \dot{q}^\top)$; the configuration coordinate is $q^\top = (z_m, z_r, z_f)$; $m_m = 1.5$, $m_r = 3.5$, $m_f = 0.4$; is the mass of the motor, the body, and the foot, respectively; $F_s = k_s(z_r - z_f - L_0) + c_s(\dot{z}_r - \dot{z}_f)$ is the spring force, with $k_s = 8000$, $c_s = 4$, $L_0 = 0.09$ the stiffness, damping and the natural length of the coil spring, respectively.

We wish to generate a motion that can hop above 0.3 m with minimal motor force inputs. Therefore, we first parse the continuous dynamics given by (2.46) and (2.47) into the collocation (equality) constraints as given by (2.44). Because of the simplicity, the discrete dynamics that is shown abstractly in (2.31) is formulated into the following boundary conditions:

$$\begin{cases} x_f^1 - x_s^{K_s} = 0 \\ q_s^1 - q_f^{K_f} = 0 \\ (\dot{z}_m)_s^1 - (\dot{z}_m)_f^{K_f} = 0 \\ (\dot{z}_r)_s^1 - (\dot{z}_r)_f^{K_f} = 0 \\ (\dot{z}_f)_s^1 = 0 \end{cases} \quad .T \quad (2.48)$$

We then constrain the ending time of each domain as

$$\begin{cases} 0.1 \leq t_s^{K_s} \leq 1, \\ 0.1 \leq t_f^{K_f} \leq 3, \\ t_s^{K_s} - t_f^0 = 0. \end{cases} \quad (2.49)$$

To hop higher than 0.3 m, we utilized the following path constraints:

$$\begin{cases} (z_f)_s^0 = (\dot{z}_f)_s^0 = 0 \\ h_{\max} \exp(-\frac{1}{2}(\frac{\tau-c}{\sigma})^2) - 0.01 - z_f^i \leq 0 \quad \forall i \in (0, 1, \dots, K_f) \end{cases} \quad (2.50)$$

where $h_{\max} = 0.3$, and c is where we wish the highest hopping point to appear, and

$$\tau = \frac{t^i - t_f^0}{t_f^{K_f} - t_f^0}$$

is a parameterization of the time. We now have the open-loop trajectory optimization posed as

$$\begin{aligned} \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \sum_i |u^i|^2 \\ \text{s.t.} \quad & (2.44)(2.48)(2.49)(2.50). \end{aligned} \quad (2.51)$$

After 1.98 second, we obtained an optimal solution, i.e. the trajectory of the states \mathbf{x}_s , \mathbf{x}_f and the inputs \mathbf{u}_s , \mathbf{u}_f . We show the result in Fig. 2.7.

Note that benefit from the simplicity of this EOM in (2.46) and (2.47), we can use automatic differentiation (AD) solvers such as ADOL-C (Griewank, Juedes, and Utke, 1996) to get the Jacobians of the constraints. However, this simple structure is very rare in multi-body dynamics; obtaining the Jacobians using AD for the full-order dynamics of a humanoid robot demands much more computational resources. Therefore, we will not give the dynamics explicitly, and we will use *Wolfram Mathematica* to obtain the Jacobians from now on. Additionally, there are many software that can parse a multi-body structure such as the URDF⁴ into the general rigid-body dynamics format in (2.7), such as the *Screws* package in (R. M. Murray et al., 1994).

⁴<http://wiki.ros.org/urdf>

CONTROLLER DESIGN FOR BIPEDAL RUNNING

In this chapter, we design controllers and trajectories (gaits) for bipedal locomotion using the full-order hybrid dynamics. In particular, we study the gait design, Lyapunov stability, and input-to-state stability analysis for nonlinear control systems and apply them to a high-dynamic behavior, bipedal running, on two bipedal robots—DURUS-2D and DURUS.

3.1 Running Dynamics (Open-Loop)

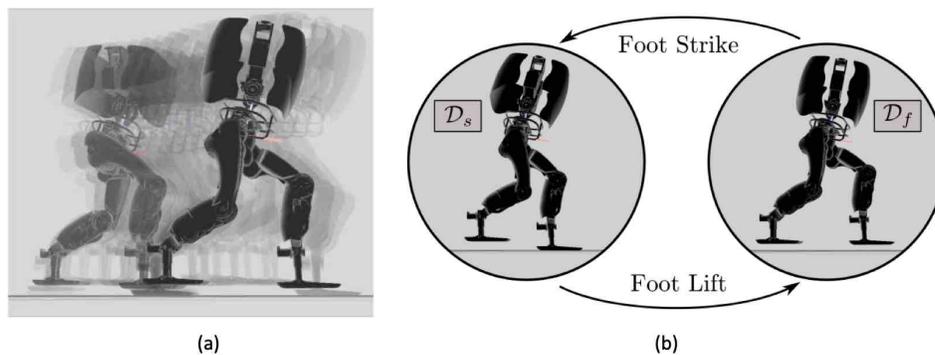


Figure 3.1: (a) The simulated running of the humanoid robot, DURUS, as a result of large-scale HZD optimization. (b) The directed cycle structure of the multi-domain hybrid system model for flat-footed humanoid running.

This section details the multi-domain hybrid dynamic model of 3D running on the DURUS robot.

Robot Model of 3D Humanoid—DURUS

A popular approach for robotic running is to utilize the Spring-Loaded Inverted Pendulum (SLIP) model (R. Blickhan, n.d.; Collins et al., 2005b; Rezazadeh et al., 2015), since the springs can be of assistance to improve energy efficiency and absorb the high-speed plastic impacts to protect the hardware. Inspired by the SLIP model, the three-dimensional DURUS robot (Fig. 2.1 (d)) is designed and built by SRI International for the study of high-efficiency multi-domain bipedal locomotion (Ayonga Hereid, Eric. Cousineau, et al., 2016; Jacob Reher et al., 2016). DURUS is an 80 kg, 23 degrees-of-freedom (DOFs) underactuated humanoid robot with 15 actuated joints and two passive linear springs attached to the end of each ankle joint

that compress perpendicular to the foot. The robot's upper body, which is used to balance the dynamics and better resemble human locomotion, is controlled by three orthogonal waist joints.

For the running model of DURUS, we use the generalized floating-base coordinates, $q = [p_b, \phi_b, \theta]^\top \in \mathcal{Q}$, of the robot, where $p_b \in \mathbb{R}^3$ is the Cartesian position and $\phi_b \in SO(3)$ is the orientation of the body base frame R_b , which is attached to the center of the pelvis link, with respect to the world frame; and as shown in Fig. 2.1 (d), the local joints are represented by

$$\theta = [\psi_w, \phi_w, \theta_w, \psi_{lh}, \phi_{lh}, \theta_{lh}, \theta_{lk}, \theta_{la}, \phi_{la}, r_{ls}, \psi_{rh}, \phi_{rh}, \theta_{rh}, \theta_{rk}, \theta_{ra}, \phi_{ra}, r_{rs}]^\top,$$

with the coordinates corresponding to the waist yaw, roll, pitch angles, left hip yaw, hip roll, hip pitch, knee pitch, ankle pitch, ankle roll angles and spring deflection, and the right hip yaw, hip roll, hip pitch, knee pitch, ankle pitch, ankle roll angles and spring deflection.

Hybrid Dynamics for Running

Due to the existence of both continuous and discrete dynamics, bipedal robot running is naturally modeled as a hybrid control system. The flat-footed running of DURUS is composed of two continuous domains: a *stance* domain, where the nonstance foot swings in the air while the stance foot stays on the ground, and a *flight* domain, where both feet are in the air (see Fig. 3.1 (b)). Therefore, as discussed in (5), the hybrid control system of DURUS running is defined as a *tuple*:

$$\mathcal{HC} = (\Gamma, \mathcal{D}, \mathcal{U}, \mathcal{S}, \Delta, \mathbb{FG}), \quad (3.1)$$

where,

- $\Gamma = \{\mathbb{V}, \mathbb{E}\}$ is a directed cycle with vertices $\mathbb{V} = \{s, f\}$, where s represents the stance domain and f represents the flight domain, and the edges $\mathbb{E} = \{s \rightarrow f, f \rightarrow s\}$,
- $\mathcal{D} = \{\mathcal{D}_s, \mathcal{D}_f\}$ is a set of admissible domains of continuous dynamics,
- $\mathcal{U} = \{\mathcal{U}_s, \mathcal{U}_f\}$ is a set of admissible controls,
- $\mathcal{S} = \{\mathcal{S}_s \subset \mathcal{D}_s, \mathcal{S}_f \subset \mathcal{D}_f\}$ is a set of *guards*,
- $\Delta = \{\Delta_{s \rightarrow f}, \Delta_{f \rightarrow s}\}$ is a set of smooth *reset maps* representing the discrete dynamics,

- $\mathbb{FG} = \{(f_s, g_s), (f_f, g_f)\}$ is a set of affine control systems

$$\dot{x} = f_v(x) + g_v(x)u,$$

defined on \mathcal{D}_v for all $v \in \mathbb{V}$, with $x = (q, \dot{q})$ being the system states.

The directed cycle Γ is depicted in the Fig. 3.1 (b). The construction of individual elements of (3.1) will be presented in the remainder of this section.

Stance Domain. During the stance domain, the stance foot remains flat on the ground. Often we use holonomic constraints to model the foot contact with the ground (Jessy W. Grizzle et al., 2014). Here, we define the holonomic constraints of the stance domain as

$$h_s(q) \triangleq \begin{bmatrix} p_{sf}(q) \\ \phi_{sf}(q) \end{bmatrix} \in \mathbb{R}^6, \quad (3.2)$$

with p_{sf} the position and ϕ_{sf} the orientation of the *stance foot*. Given the physical properties of each link, the unconstrained dynamics of the stance domain \mathcal{D}_s is given by

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J_s^T(q)F, \quad (3.3)$$

where, $D(q)$ is the inertia matrix, $H(q, \dot{q})$ contains the Coriolis, gravity, and spring forces terms, $J_s(q)$ is the Jacobian of the holonomic constraints, and F is a *wrench* containing the ground constraint forces and moments. The holonomic constraints are guaranteed via enforcing the second-order derivative of h_s to be zero:

$$J_s(q)\ddot{q} + \dot{J}_s(q, \dot{q})\dot{q} = 0. \quad (3.4)$$

Thus the affine control system (f_s, g_s) can be determined by combining (3.3) and (3.4). The manifold of the stance domain is determined by unilateral constraints, which could be formulated as a vector of admissible conditions, $A_s(q, \dot{q}, u)$. These conditions include positive non-stance foot height, positive normal ground force, etc. In other words, we have

$$\mathcal{D}_s = \{(q, \dot{q}, u) \in T\mathcal{Q} \times \mathcal{U}_s \mid A_s(q, \dot{q}, u) \geq 0\}. \quad (3.5)$$

Further, the guard condition of the stance domain is defined as the normal ground force crosses zero, i.e.,

$$\mathcal{S}_{s \rightarrow f} = \{(q, \dot{q}, u) \in T\mathcal{Q} \times \mathcal{U}_s \mid F^z(q, \dot{q}, u) = 0\}. \quad (3.6)$$

Because there is no impact involved during the transition, the reset map from the stance domain to flight domain, $\Delta_{s \rightarrow f}$, is an identity map.

Flight Domain. Since there is no ground contact during the flight domain, the continuous dynamics of the domain is determined by the unconstrained Euler-Lagrangian equation:

$$D(q)\ddot{q} + H(q, \dot{q}) = B(q)u. \quad (3.7)$$

The admissible conditions of the flight domain are defined so that both feet are above the ground, i.e., $A_f(q) = (h_{sf}(q), h_{nsf}(q))$. Therefore, we have

$$\mathcal{D}_f = \{(q, \dot{q}, u) \in TQ \times \mathcal{U}_f \mid A_f(q) \geq 0\}. \quad (3.8)$$

Accordingly, the transition from the flight to stance domain occurs when the non-stance foot strikes the ground, i.e.,

$$\mathcal{S}_{f \rightarrow s} = \{(q, \dot{q}, u) \in TQ \times \mathcal{U}_f \mid h_{nsf}(q) = 0, \dot{h}_{nsf}(q, \dot{q}) < 0\}. \quad (3.9)$$

Both dynamics in (3.3) and (3.7) can be converted to the following form:

$$\dot{x} = f_v(x) + g_v(x)u, \quad (3.10)$$

and the derivation was given in (2.20).

Transition Map. The reset map from the flight to the stance domain incorporates the impact dynamics when the non-stance foot hits the ground, during which the joint velocities undergo discrete changes due to new contact constraints. Given the pre-impact states (q^-, \dot{q}^-) , the post-impact states $(q^+, \dot{q}^+) = \Delta_{f \rightarrow s}(q^-, \dot{q}^-)$ are determined by assuming a perfectly plastic impact of the rigid body. Since the position terms do not change through the impact, we have the impact equation, which determines the discrete changes of velocities as

$$\begin{bmatrix} D(q^-) & -J_s^\top(q^-) \\ J_s(q^-) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \delta F \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix} \quad (3.11)$$

where δF is a vector of impulsive contact wrenches.

3.2 Virtual Constraint and Zero Dynamics

Given the running model, this section introduces the basic concepts used in this chapter, including virtual constraints, phasing variables, and zero dynamics. We will use DURUS-3D running (Fig. 3.1) as an example to concretize these constructions.

Desired behavior. To achieve stable locomotion, we first need to decide *what to control*, which is the *feature* (actual output, y^a) that we intend to drive to some *desired behavior* (desired output, y^d) that we will design through *trajectory optimization*. The actual output is a function of the system's states, given as

$$y^a = y^a(x); \quad (3.12)$$

and the desired behavior is represented by a Bézier polynomial

$$y^d = \beta(\tau, \alpha), \quad (3.13)$$

where α is the (constant) coefficients for the polynomial, and this trajectory is parameterized by a **phasing variable** τ , and it monotonically increases over time.

Phasing variable. Locomotion gaits, viewed as a set of desired trajectories, are often modulated as functions of a phase variable to eliminate the dependence on the time-based (Villarreal and R. D. Gregg, 2014). We will discuss two types of phasing variables in this manuscript, one is a **time-based** phasing variable

$$\tau_t = \tau(t). \quad (3.14)$$

Correspondingly, the desired output (3.13) will be a function of time. We also define a **state-based** phasing variable as

$$\tau_s = \tau(q). \quad (3.15)$$

Correspondingly, the desired output (3.13) will be a function of states. In the content that follows, we will omit the state-based subscript $_s$ since we will not talk about time-based setting until Sec. 3.4.

Virtual constraint. To achieve stable locomotion, we first need to establish the following goal

$$y^d \rightarrow y^a \quad \text{as} \quad \text{time } t \rightarrow \infty, \quad (3.16)$$

which is further described as driving the *output* (virtual constraint):

$$y = y^a(x) - y^d(\tau, \alpha) \quad (3.17)$$

to 0 as $t \rightarrow \infty$.

Since α is a set of constant parameters, we will suppress this argument from $y^d(\tau, \alpha)$ to have $y^d(\tau)$.

Output dynamics. The dynamic relation between the output y , and an input u , is referred as *output dynamics*. For the robotic systems considered in this dissertation, we are particularly interested in two types of outputs, the *relative degree 1* output:

$$\begin{aligned} \dot{y} &= \frac{dy}{dt} = \frac{\partial y}{\partial x} \dot{x} \\ &= \underbrace{\frac{\partial y}{\partial x} f(x)}_{\triangleq L_f y} + \underbrace{\frac{\partial y}{\partial x} g(x)}_{\triangleq L_g y} u, \end{aligned} \quad (3.18)$$

and the *relative degree 2* output when $L_g y = 0$:

$$\begin{aligned} \ddot{y} &= \frac{d^2 y}{dt^2} = \frac{\partial L_f y(x)}{\partial x} \dot{x} \\ &= \underbrace{\frac{\partial L_f y(x)}{\partial x} f(x)}_{\triangleq L_f^2 y} + \underbrace{\frac{\partial L_f y(x)}{\partial x} g(x)}_{\triangleq L_g L_f y} u. \end{aligned} \quad (3.19)$$

where f , g is given in (3.10). Rigid-body dynamics have no more than a relative-degree two outputs. All systems of interest in this thesis only have less than two relative degrees. In particular, *any output that is only a function of the configuration coordinate $y(q)$, is relative-degree two*. Note that when compliance such as series-elastic actuator is involved, some of the global coordinates (w.r.t. world frame) can have more than two relative degrees.

The goal is to design a control scheme u to drive the difference between the actual output and the desired output to zero. Before doing so, let us take the bipedal running as an example.

Example. For DURUS running Fig. 3.1(a), we first pick the actual outputs $y^a(q, \dot{q})$. In the stance domain \mathcal{D}_s , the forward velocity of the center of mass (COM) is chosen as the relative degree one output

$$y_{1,s}^a = v_{com}^x(q, \dot{q})$$

to regulate the forward velocity of the robot, and the (vector) relative degree two outputs are defined as

$$y_{2,s}^a(q) = (\theta_{sk}, \phi_{slr}, \theta_{sh}, \psi_{sh}, \phi_{sa}, \phi_w, \theta_w, \psi_w, \theta_{nsk}, \phi_{nsh}, \theta_{nsh}, \phi_{nsf}, \theta_{nsf}, \psi_{nsf})^\top.$$

In the flight domain \mathcal{D}_f , actual outputs consist of only the relative degree two outputs

$$y_{2,f}^a(q) = (\theta_{sk}, \phi_{slr}, \theta_{sh}, \phi_{sf}, \theta_{sf}, \psi_{sf}, \phi_w, \theta_w, \psi_w, \theta_{nsk}, \phi_{nsh}, \theta_{nsh}, \phi_{nsf}, \theta_{nsf}, \psi_{nsf})^\top.$$

In particular, $\phi_{slr} = \phi_{sh} - \phi_{nsh}$ is the stance leg roll angle, $\phi_{sf}, \theta_{sf}, \psi_{sf}$ and $\phi_{nsf}, \theta_{nsf}, \psi_{nsf}$ are the orientations (Euler angles) of the stance and non-stance foot, respectively. Other outputs are simply joint angles, as shown in Fig. 2.1 (d).

We want to design a running motion with constant forward velocity, hence the desired velocity of COM is a constant v_d , i.e.,

$$y_{1,s}(q, \dot{q}, v_d) = y_{1,s}^a(q, \dot{q}) - v_d. \quad (3.20)$$

And the desired relative degree two outputs $y_{2,v}^d(\tau(q), \alpha_v)$ are represented by seventh-order Bézier polynomials parameterized by a set of parameters α_v with $v \in \{s, f\}$. The virtual constraints on \mathcal{D}_v became:

$$y_{2,v}(q, \alpha_v) = y_{2,v}^a(q) - y_{2,v}^d(\tau(q), \alpha_v), \quad (3.21)$$

where $\tau(q)$ is a monotonic state-based parameterization of time, defined as

$$\tau(q) = \frac{p_b^x - p_0}{p_1 - p_0}, \quad (3.22)$$

where p_b^x is the x -position of the floating base's frame R_b , and the parameters p_1 and p_0 are the corresponding values, by design, at the beginning and the end of each (cyclic) step. This setting allows the phasing variable to parameterize the desired trajectory over one step from 0 to 1, i.e., $\tau \in [0, 1]$. In particular, the desired outputs of the stance and non-stance foot orientations are set to be zero respectively to keep the feet being flat throughout the step. Note that if an output is defined for both domains, the coefficients of the corresponding desired Bézier polynomials must be the same.

As given in (3.18) and (3.19), we can summarize (3.20) and (3.21) as

$$\begin{bmatrix} \dot{y}_{1,v} \\ \ddot{y}_{2,v} \end{bmatrix} = \underbrace{\begin{bmatrix} L_f y_{1,v} \\ L_f^2 y_{2,v} \end{bmatrix}}_{\mathcal{L}_v(q, \dot{q})} + \underbrace{\begin{bmatrix} L_f^2 y_{1,v} \\ L_g L_f y_{2,v} \end{bmatrix}}_{\mathcal{A}_v(q)} u \quad (3.23)$$

where we call $\mathcal{A}_v(q)$ the *decoupling matrix*.

Feedback Linearization-Based Control

As often practiced for nonlinear control systems, we can construct a feedback linearization-based control law for bipedal locomotion. To drive the output:

$$y_v = \begin{bmatrix} y_{1,v} \\ y_{2,v} \end{bmatrix} \rightarrow 0$$

for each \mathcal{D}_v with $v \in \{s, f\}$, we can utilize the input-output feedback linearization control law

$$u_v^\varepsilon = -\mathcal{A}_v^{-1} (\mathcal{L}_v + \mu_v^\varepsilon), \quad (3.24)$$

which is often shorted as *IO controller*. The control structure is illustrated in Fig. 3.2.

For running dynamics, we have the decoupling matrix \mathcal{A}_v as:

$$\mathcal{A}_s = \begin{bmatrix} L_{g_s} y_{1,s}(q, \dot{q}) \\ L_{g_s} L_{f_s} y_{2,s}(q) \end{bmatrix}, \quad \mathcal{A}_f = L_{g_f} L_{f_f} y_{2,f}(q)$$

respectively, and

$$\mathcal{L}_s = \begin{bmatrix} 0 \\ L_{f_s} L_{f_s} y_{2,s}(q) \end{bmatrix}, \quad \mathcal{L}_f = L_{f_f} L_{f_f} y_{2,f}(q).$$

With the given control law, we have the output dynamics become

$$\begin{bmatrix} \dot{y}_{1,s} \\ \ddot{y}_{2,s} \end{bmatrix} = -\mu_s^\varepsilon, \quad \ddot{y}_{2,f} = -\mu_f^\varepsilon \quad (3.25)$$

for the stance and flight domain respectively, where μ_v^ε can be chosen so that the outputs converge to zero exponentially at a rate of $\varepsilon > 0$. In particular, we define

$$\mu_s^\varepsilon = \begin{bmatrix} \varepsilon y_{1,s}(q, \dot{q}, v_d) \\ 2\varepsilon \dot{y}_{2,s}(q, \dot{q}, \alpha_s) + \varepsilon^2 y_{2,s}(q, \alpha_s) \end{bmatrix}, \quad (3.26)$$

$$\mu_f^\varepsilon = 2\varepsilon \dot{y}_{2,f}(q, \dot{q}, \alpha_f) + \varepsilon^2 y_{2,f}(q, \alpha_f). \quad (3.27)$$

Denote some new coordinates as

$$\eta_s \triangleq \begin{bmatrix} y_{1,s} \\ y_{2,s} \\ \dot{y}_{2,s} \end{bmatrix}, \quad \eta_f \triangleq \begin{bmatrix} y_{2,f} \\ \dot{y}_{2,f} \end{bmatrix},$$

and $\eta = (\eta_s, \eta_f)^\top$, we can see the exponential stabilizing effect of the η -dynamics according to the following definition.

Definition 7. A smooth function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is an *exponentially stabilizing control Lyapunov function (ES-CLF)* for

$$\dot{\eta} = f(\eta) + g(\eta)u,$$

if there exists constants $c_1, c_2, c_3 > 0$ such that

$$\begin{aligned} c_1 |\eta|^2 &\leq V(\eta) \leq c_2 |\eta|^2 \\ \inf_u (L_f V(\eta) + L_g V(\eta)u + c_3 V(\eta)) &\leq 0 \end{aligned} \quad (3.28)$$

for all $x \in \mathcal{X}$. If there further exists a constant $\varepsilon \in (0, 1)$ such that

$$\begin{aligned} c_1 |\eta|^2 &\leq V(\eta) \leq \frac{c_2}{\varepsilon^2} |\eta|^2 \\ \inf_u (L_f V(\eta) + L_g V(\eta)u + \frac{c_3}{\varepsilon} V(\eta)) &\leq 0 \end{aligned} \quad (3.29)$$

for all $\eta \in \mathcal{Y}$, then $V(\eta)$ is a **rapidly exponentially stabilizing control Lyapunov function (RES-CLF)**.

and we can pick a Lyapunov candidate as

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

for the dynamics in (3.25) under control input (3.26) and (3.27). Remark that these controllers only stabilize the η -dynamics, with $k < n$. For the underactuated full-order system, we need some other conditions to establish its stability property, which is introduced below.

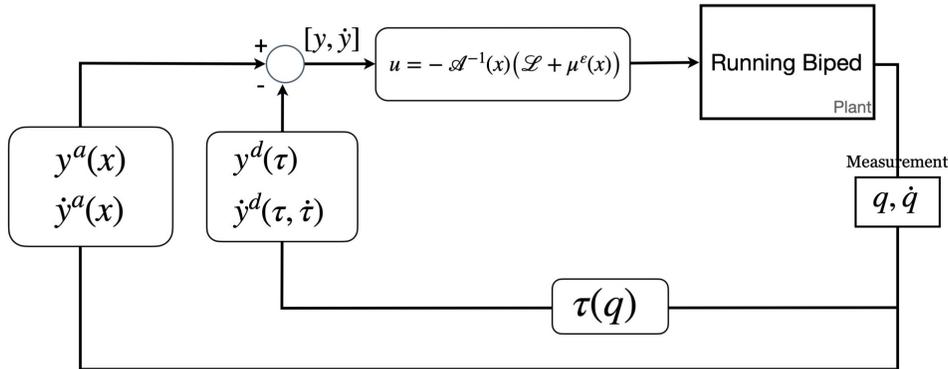


Figure 3.2: A block diagram showing the state-based feedback control structure.

Hybrid Zero Dynamics Invariance

The fundamental theorem gives the condition of the overall system's stability in the sense of Lyapunov in the HZD condition, (A. Ames, Galloway, et al., 2014a), which will be a basis for the development of the methods in this thesis. We will briefly introduce it here.

Any admissible state-based feedback controller, such as (3.24), that has been applied to the control system, \mathbb{FG} , yields the following closed-loop continuous system:

$$\dot{x} = f_v^{\text{cl}}(x) = f_v(x) + g_v(x)u_v(x, \alpha_v)$$

forward invariant on the (partial) zero dynamics surface:

$$\mathcal{Z}_v \triangleq \{(q, \dot{q}) \in \mathcal{D}_v \mid y_{2,v}(q) = \dot{y}_{2,v}(q, \dot{q}) = 0\}. \quad (3.30)$$

But due to the impact dynamics, the invariance of the zero dynamics surface is not necessarily guaranteed by the controller. Particularly, it is impossible to enforce the relative degree one output to be invariant through impact due to the changes in the velocity at the impact event.

Therefore, a submanifold \mathcal{Z}_v is impact-invariant if

$$\Delta_e(x) \in \mathcal{Z}_{v^+}, \quad \forall x \in \mathcal{S}_e \cap \mathcal{Z}_v \quad (3.31)$$

for each $e \triangleq (v \rightarrow v^+) \in \mathbb{E}$. The resulting hybrid invariant submanifold is referred as a *hybrid zero dynamics* (HZD), $\mathcal{H}|_{\mathcal{Z}}$. Further, $\mathcal{Z} = \bigcup_{v \in \mathcal{V}} \mathcal{Z}_v$ is hybrid invariant if it is invariant over all domains of continuous dynamics and discrete dynamics. In fact, the restricted reduced dimensional dynamics are independent of control input.

According to (A. Ames, Galloway, et al., 2014a, Thm.2) if there exists a RES-CLF for the output dynamics in (3.25), which drives the actual output to the desired trajectory exponentially, then there exists an exponentially stable periodic orbit of the hybrid zero dynamics manifold given in (3.30). We can have an exponentially stable orbit for the full-order system. With the feedback controller defined, as given in (3.24), (3.26), and (3.27), the mission is to find the parameter set α that $y^a(q)$ converges to $y^d(\alpha, \tau)$ exponentially. In other words, the goal of designing a gait is to find a set of parameters $\alpha = \{v_d, \alpha_s, \alpha_f\}$ that ensures the existence of a periodic orbit for the system (3.1) and the (partial) hybrid zero dynamics surface, \mathcal{Z}_v is invariant through the discrete impact dynamics.

The process of finding α is then formulated as a nonlinearly constrained optimization problem in Sec. 3.3 subject to the HZD condition.

3.3 Gait Design via Closed-Loop Optimization

We can now use optimization tools to design a gait (periodic trajectory), a periodic solution to the closed-loop system. In this section, we first use bipedal running as an example to present the details of closed-loop optimization, including the equality

and inequality constraints. Then we study the energy efficiency of these generated running behaviors. We will lastly use the tool developed to realize bipedal walking on slippery surfaces.

Closed-Loop Optimization for Running

Building on top of (2.45), we first define the dynamic constraints for hybrid systems for the purpose of *finding periodic solution to the open-loop dynamics* (3.10) as follows.

1. Continuous Dynamic Constraint.

We first need the optimization algorithm to find a trajectory of the states and inputs for all time. Follow the setting in Sec. 3.10 and notations from (3.1), we have $N = 2$ for the two-domain running, and pick $K_1 = K_2 = 15$ for the discretization of horizon. We then have the first equality constraint for continuous dynamics (as in (2.45)):

$$Z_v(\mathbf{x}_v, \mathbf{u}_v) = \mathbf{0} \quad \forall v \in \mathcal{V} \triangleq (s, f). \quad (\text{C.1})$$

2. Discrete Dynamic Continuity.

To guarantee the trajectories $C_v \triangleq \{\mathbf{x}_v, \mathbf{u}_v\}$ of the neighbor domains belong to the same flow, we have the following equality constraint for discrete dynamic continuity

$$x_f^0 - \Delta[e_{s \rightarrow f}](x_s^{K_1}) = 0 \quad (\text{C.2})$$

In essence, (C.1) and (C.2) pose the hybrid dynamics

$$\begin{cases} \dot{x} = f_v(x) + g_v(x)u & x \in \mathcal{D}_v \\ x^+ = \Delta[e_{v_i \rightarrow v_{i+1}}](x^-) \end{cases},$$

where $v \in \mathcal{V} \triangleq (s, f)$, into a set of equality constraints. The advantage is that we can pre-compute the symbolic Jacobians of these constraints and hence accelerate the evaluation time, and the problem is made robust against stiff systems (see (Kiehl, 1994; Ascher and Petzold, 1998)).

3. Periodic Continuity.

To find a periodic solution, we need to make sure the final condition is “stitched” to the initial condition. Still, since there is a jump in the velocity

terms due to the impact, as in (3.11), we will need to relate them through the impact map. But before doing so, a key concept needs to be introduced, the relabelling matrix $\mathcal{R} : \mathcal{Q} \rightarrow \mathcal{Q}$. Since all of the legs' configuration coordinates are labelled as *stance* $'''_s$ and *nonstance* $'''_{ns}$, and after the impact, the nonstance leg will become a stance leg, we need to mirror (relabel) the states accordingly. For example, the pitch angles of the stance and nonstance joints need to be flipped, and the roll and yaw angles of stance and nonstance legs need to be flipped with a negative sign. We then modify (3.11) with the notation from (C.2) to have an equality constraint for *periodic continuity*:

$$\begin{bmatrix} D(q_f^{K_f}) & -J_s^\top(\mathcal{R}q_f^{K_f}) \\ J_s(\mathcal{R}q_f^{K_f}) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_s^1 \\ \delta F \end{bmatrix} = \begin{bmatrix} D(\mathcal{R}q_f^{K_f})\dot{q}_f^{K_f} \\ 0 \end{bmatrix}, \quad (\text{C.3})$$

or simply,

$$x_s^0 - \Delta[\text{f} \rightarrow \text{s}]\mathcal{R}x_f^{K_f} = 0. \quad (3.32)$$

To establish Lyapunov stability analysis, we need to have the system in the closed-loop form, which is open-loop dynamics with a parameterized feedback controller such as in (3.24). We now present the closed-loop dynamics in the optimization setting for the control problem of interest.

4. Output Dynamics.

Since the controller in (3.24) yields output dynamics in (3.25), we can equivalently enforce the output dynamics as

$$\begin{cases} \dot{y}_{1,v}(\dot{x}, x) + \varepsilon y_{1,v}(x, v_d) = 0 \\ \ddot{y}_{2,v}(\dot{x}, x) + 2\varepsilon \dot{y}_{2,v}(x) + \varepsilon^2 y_{2,v}(x, \alpha_v) = 0 \end{cases} \quad (\text{C.4})$$

according to (3.26) and (3.27), where $i = 0, 1, \dots, K_v$, with $v \in (s, f)$. Remark the here we have introduced the most essential parameter α_v that represents the desired output (trajectory) of the solution, as given in (3.20) and (3.21).

5. Hybrid Invariance.

As established in (3.31), we need the HZD condition to reduce the complexity of stability analysis from the full-order system to the passive dynamics on the hybrid zero dynamics manifold \mathcal{Z}_v . Formally,

$$\begin{aligned} y_{2,v}(x_v^0, \alpha_v) &= \dot{y}_{2,v}(x_v^0, \dot{x}_v^0, \alpha_v) = 0 \\ y_{2,v}(x_v^{K_v}, \alpha_v) &= \dot{y}_{2,v}(x_v^{K_v}, \dot{x}_v^{K_v}, \alpha_v) = 0 \quad \forall v \in \mathcal{V} \end{aligned} \quad (\text{C.5})$$

which is a boundary condition.

We now introduce constraints that are essential to robotics and legged locomotion.

6. Feasibility Conditions.

We can use some optimization algorithm, such as direct collocation with (C.1)—(C.5), to find a solution to a dynamical system. However, these solutions might not be feasibly realizable on hardware. For example, although a solution $x(t)$, $t \in [0, T]$ satisfies $\dot{x}(t) = f(x(t))$, there is no guarantee that this trajectory will not penetrate an obstacle on in the way. We can use a set of *feasibility conditions* (also known as path constraint) to enforce these conditions. These conditions include the electric motors’ torque saturation, each joint’s feasible workspace that is specified by the hardware limitation, given by

$$\begin{aligned} -\mathbf{u}_{\max} &\leq u^i \leq \mathbf{u}_{\max}, \\ \mathbf{q}_{\min} &\leq q^i \leq \mathbf{q}_{\max}, \\ -\mathbf{v}_{\max} &\leq \dot{q}^i \leq \mathbf{v}_{\max}. \end{aligned}$$

We also need to enforce the friction pyramid condition¹ to prevent the stance feet from slipping on the ground. Then, these conditions can be configured in the following condition:

$$\begin{cases} u^i \in \mathcal{U}, x^i \in \mathcal{X} \\ \mu F_z(x^i, u^i) - |F_x(x^i, u^i)| \geq 0 \\ \mu F_z(x^i, u^i) - |F_y(x^i, u^i)| \geq 0 \end{cases} \quad \forall i \in (0, 1, \dots, K_v), \forall v \in \mathcal{V} \quad (\text{C.6})$$

where μ is the static friction coefficient, and $F_{(\cdot)}$ is the ground reaction force along certain axis.

Remark that for experimental success, these conditions can often be manually tuned to be more restrictive, for example, a smaller friction constant and a smaller torque bound can give the actuator more “freedom to save” the robot from failures under disturbance and uncertainty.

7. Foot clearance.

For legged locomotion, ground clearance for the nonstance foot is necessary. This is a heuristic constraint that users normally need to tune to get a “natural-looking” gait eventually. Denote the height of the nonstance foot (or feet,

¹ A computationally efficient but more restrictive version of the friction cone condition.

sometime there can be two feet in the air, such as the flight phase) as $h_{\text{nsf},z}(q)$, we can specify a lower bound $h_v^l(t)$ for $v \in \mathcal{V}$. The foot clearance conditions become:

$$h_v^l(t^i) - h_{\text{nsf},z}(q^i) \leq \mathbf{0}, \quad \forall i \in (0, 1, \dots, K_v), \forall v \in \mathcal{V}. \quad (\text{C.7})$$

We now have completed the optimization formulation for finding a solution to the closed-loop dynamics of a running bipedal robot. Formally,

$$\begin{aligned} & \underset{\mathbf{X}}{\text{argmin}} \quad J(\mathbf{X}) & (3.33) \\ \text{s.t.} \quad & (\text{C.1}) \quad Z_v(\mathbf{x}_v, \mathbf{u}_v) = \mathbf{0} \quad \forall v \in \mathcal{V} \\ & (\text{C.2}) \quad x_f^0 - \Delta[e_s \rightarrow f](x_s^{K_1}) = 0 \\ & (\text{C.3}) \quad x_s^0 - \Delta[f \rightarrow s]\mathcal{R}x_f^{K_f} = 0 \\ & (\text{C.4}) \quad \begin{cases} \dot{y}_{1,v}(\dot{x}, x) + \varepsilon y_{1,v}(x, v_d) = 0 \\ \ddot{y}_{2,v}(\dot{x}, x) + 2\varepsilon \dot{y}_{2,v}(x) + \varepsilon^2 y_{2,v}(x, \alpha_v) = 0 \end{cases} \\ & (\text{C.5}) \quad y_{2,v}^0 = y_{2,v}^{K_v} = \dot{y}_{2,v}^0 = \dot{y}_{2,v}^{K_v} = 0 \quad \forall v \in \mathcal{V} \\ & (\text{C.6}) \quad \begin{cases} u^i \in \mathcal{U}, x^i \in \mathcal{X} \\ \mu F_z(x^i, u^i) - |F_x(x^i, u^i)| \geq 0 \\ \mu F_z(x^i, u^i) - |F_y(x^i, u^i)| \geq 0 \end{cases} \quad \forall i \in (0, 1, \dots, K_v), \forall v \in \mathcal{V} \\ & (\text{C.7}) \quad h_v^l(t^i) - h_{\text{nsf},z}(q^i) \leq \mathbf{0}, \quad \forall i \in (0, 1, \dots, K_v), \forall v \in \mathcal{V} \end{aligned}$$

where $J(\mathbf{X})$ is the cost function. A common choice is the total energy consumption, given as $\sum_{i=0}^N \langle Bu^i, \dot{q}^i \rangle$. With this formulation, we can employ some optimization tools such as IPOPT (Wächter and Biegler, 2006) to solve the gait parameters α_v and initial condition $x(0)$.

We now apply this gait generation method to two behaviors: studying the energy efficiency of running motion and realizing bipedal walking on slippery surfaces.

Towards High-Energy Efficiency of Running

Marc Raibert's technical report "Dynamically Stable Legged Locomotion, 1989" (M. H. Raibert et al., n.d.) summarized that "*the running speed of a legged system depends upon the frequency and length of its steps*" based on the study of his legendary hopper. To better understand this phenomenon in 3D bipedal-legged systems, this section documents a statistical result based on the stable running gaits obtained from an optimization-based gait generation framework, i.e., how the

optimizer should respond when asked to generate faster gaits to maximize its energy efficiency. Bipedal running is an essential benchmark for humanoid control for many mathematical and practical reasons. Unlike walking, running is an inherently underactuated control problem (M. Spong, 1998). Whenever the robot leaves the ground, it fundamentally loses its ability to actuate all degrees of freedom and is at the mercy of its ballistic trajectory. It is also a multi-domain hybrid control problem (H. Zhao et al., 2016). Further, the high power demands push the practical limits of humanoid actuators. This section presents 3D running via hybrid zero dynamics (HZD) (A. D. Ames, 2014; E. R. Westervelt, J. W. Grizzle, C. Chevallereau, et al., 2007) on a simulated underactuated model of the humanoid robot, DURUS. The running gaits emerge from a large-scale gait optimization of the full-order system dynamics, a previously developed tool for 3D walking with the DURUS hardware (Ayonga Hereid, Eric. Cousineau, et al., 2016). We report the success of this toolset as a milestone toward 3D running.

The earliest example of running controllers was developed using a set of highly successful heuristics. Examples include the Raibert hoppers (M. H. Raibert et al., n.d.) and the ARL-Monopod II (Ahmadi and Buehler, 2006). Decades later, Honda’s humanoid robot, ASIMO (Sakagami et al., 2002), claims running speeds up to 2.5 m/s without revealing its control method. Other methods have been employed to achieve stable running in simulation by constraining the robot’s dynamics to a reduced-order model (W. C. Martin, Wu, and Geyer, 2015) and even achieving simulated high-speed turning (Wensing and D. Orin, 2014). Researchers have also generated running gaits for simulated robots with various degrees of freedom. From simple point-mass models (Srinivasan and Ruina, 2006) to planar hopping models (Xi, Yesilevskiy, and Remy, 2015) to planar humanoid models (Mombaur, 2009) (which even certify stability inside the optimization). Recent work optimized an ATLAS model for 3D locomotion by reasoning about the centroidal dynamics (Hongkai Dai, Valenzuela, and Russ Tedrake, 2014). Graphics researchers even successfully created 3D running controllers after running an extensive evolutionary algorithm (Geijtenbeek, Panne, and Stappen, 2013). Here, we seek a method that generates optimal running gaits within the hybrid zero dynamics framework to leverage its formal guarantees regarding stability.

In an effort to embrace underactuation in locomotion with formal control methods, hybrid zero dynamics (HZD) was developed on multiple successful robotic walking implementations (H. Park et al., 2012; A. D. Ames, 2014). It was ultimately imple-

mented to produce planar running on the spring-legged robot, MABEL (Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, 2013), and is being extended to non-planar cases (Kaveh Akbari Hamed and Jessy W. Grizzle, 2013). While there are formal mathematical underpinnings to HZD, practical implementation requires a gait optimization which considers the full-order dynamics of the system. Traditionally, these gait optimizations have become increasingly unreliable with robots as complex as humanoids. In prior work, the authors presented a collocation-based HZD optimization formulation to produce stable 3D humanoid walking (Ayonga Hereid, Eric. Cousineau, et al., 2016). Here, we further extend this approach to 3D running on DURUS. Moreover, we show that the optimization is sufficiently reliable that we generated 25 gaits at various running speeds. This library of 3D gaits allows us, for the first time, to observe trends in energy costs and running strategies in humanoid running. Prior work has given physical insight into mechanisms of speed adjustment, such as step length and step frequency (M. H. Raibert et al., n.d.). Here, we can assess whether these strategies manifest when optimizing 3D running with humanoid complexity. We also present a cost-of-transport vs. speed curve for 3D running, commonly reported in animal locomotion studies, which we can now tractably generate for humanoid running.

With the proposed optimization method, we generated multiple stable 3D running gaits for DURUS with velocities varying from 1.5 m/s to 3.0 m/s. This section will focus on one of the simulated running gaits in detail first, then the statistical analysis of the NLP performance, and all of the running gaits will be summarized.

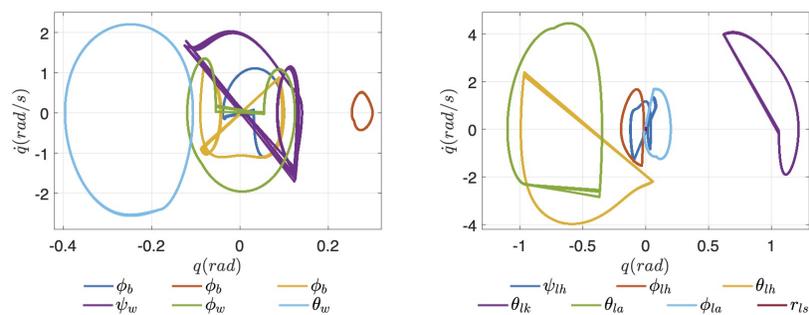


Figure 3.3: Limit cycle of running at 2.0 m/s over 20 steps.

Running at 2 m/s. With the constraints configured as explained in (3.33) and the large-scale IPOPT NLP solver developed by COIN-OR, a 3D running gait is solved after 722 iterations and 374 seconds of computation, with dual infeasibility converged to 9.0×10^{-4} , and constraints violation 1.8×10^{-7} . This particular gait runs

at 2.0 m/s. Note that we categorize each running gait based on the x component of the COM velocity during the flight domain. The specific cost of transport (SCOT) (Collins et al., 2005b) is calculated in simulation as 0.90, the maximum angular velocity of all joints is 4.4 rad/s, peak torque is 446 N m, and peak power is 1.1 kW. A running tile is shown in Fig. 3.4, the limit cycle of each joint is also shown in Fig. 3.3. Only one leg is shown because of the symmetric motion. We have verified the stability of this running gait by numerically computing the eigenvalues of the linearization of the Poincarémap that is restricted to the zero dynamics about the Poincarésection where $p_b^x = 0$. The magnitude of its eigenvalues are [0.414, 0.083, 0.031, 0.006, 0.000]. All values smaller than 1 indicate asymptotic stability obtained from this running framework (see (Morris and J. W. Grizzle, 2005) for details).



Figure 3.4: Snapshots of the DURUS running at 2.0 m/s.

Efficient 3D Running Gaits Generation. We now present a working framework to generate stable running gaits for 3D bipedal robots reliably and efficiently (see (Wen-Loong Ma, 2016b) for the simulated running). For the 25 gaits that the optimization found, we documented the computation time and iterations it needs (see Fig. 3.5). As a result, it takes 609 iterations and 323 seconds on average for the optimizer to find a feasible solution². Note that the threshold of the dual infeasibility of the NLP is set to be 10^{-3} . The number of grid points is chosen to be 15 for both the stance and flight domains. All constraints and physical limitations are configured based on the hardware capability, and the constraint violation converged below 10^{-6} .

A significant benefit of this method’s computational efficiency is that it offers the flexibility to refine the running behavior and adjust the model parameters in practice, paving the way to actual experimental realizations. In practice, however, generating candidate gaits reliably will requires some minor heuristic tweaks to the constraints. For instance, by simply modifying the forward velocity constraint and fine-tuning

² This algorithm runs on a Ubuntu14.04 machine, equipped with an Intel® Xeon® processor E3-1246 V3 and 32 GB of RAM.

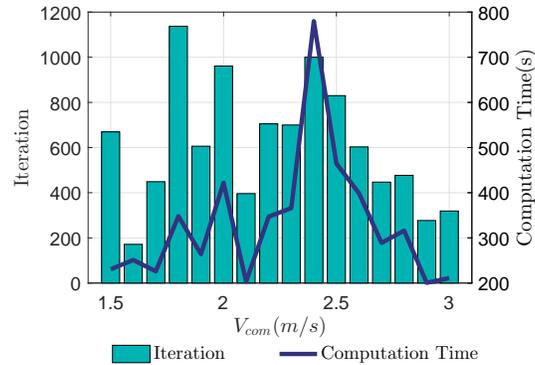


Figure 3.5: The computation performance for generating gaits at each running velocity.

a few constraints to adjust the running appearance, the suggested gait generation method can find running gaits that satisfy all the physical limitations reliably.

Once a reliably solvable formulation is engineered, we can solve for many gaits and inspect the solutions for trends in energy-efficient locomotion. The Specific Cost of Transport, which quantifies the energy efficiency of transportation, is embedded as the objective by the optimizer. Notably, as shown in Fig. 3.6 (a), for a faster running gait, the optimization tended to generate gaits with higher SCOT, which aligns with the sense that for a particular running pattern, faster locomotion requires greater energy cost. Further, as shown in Fig. 3.6 (c) and Fig. 3.6 (b), when the optimizer is commanded to find incrementally faster gaits (while still minimizing SCOT), the optimizer increases both the running frequency and step length to achieve a faster running speed, which agrees with Raibert’s hypothesis and indicates that both are key factors to fast running (M. H. Raibert et al., n.d.).

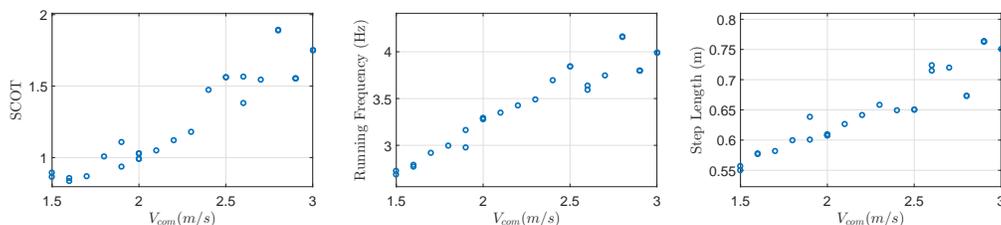


Figure 3.6: Multiple running gaits with forward velocity from 1.5 m/s to 3.0 m/s.

Dynamic Walking with Planned Slippage

Tremendous progress in realizing robust bipedal robot locomotion has been achieved in the last decade. This is in part due to the successful combination of theoretical

modeling and analysis using the framework of hybrid systems (A. Ames, 2014; Jessy W. Grizzle et al., 2014), application of advanced methods of nonlinear control (Khalil, 2002; E. R. Westervelt, J. W. Grizzle, Chevallereau, et al., 2007), as well as careful mechanical design and hardware implementation on various experimental platforms such as AMBER-3M (Ambrose, W. Ma, C. Hubicki, et al., 2017), DURUS (J. P. Reher et al., 2016), and Cassie (Da et al., 2016). Underlying these results, along with the successes for robots using other paradigms such as ZMP (Vulobratović and B. Borovac, 2004; R. Tedrake et al., 2015) and spring-loaded inverted pendulum (SLIP) based models (Ioannis Poulakakis and Grizzle, 2007; Vejdani et al., 2015), is the assumption that the foot does not slip. Thus, in all of these cases, the foot acts as a stationary pivot point. While this assumption may easily hold in sterile laboratory environments where the floors can be chosen with sufficiently high friction, it becomes impractical on natural outdoor terrains; wherein there is a plethora of slippery or slightly granulated irregular surfaces. Success in challenging the stationary contact point assumption includes multi-contact walking (H.-H. Zhao et al., 2014) and bipedal running (Wen-Loong Ma, Shishir Kolathaya, et al., 2017; Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, 2013).

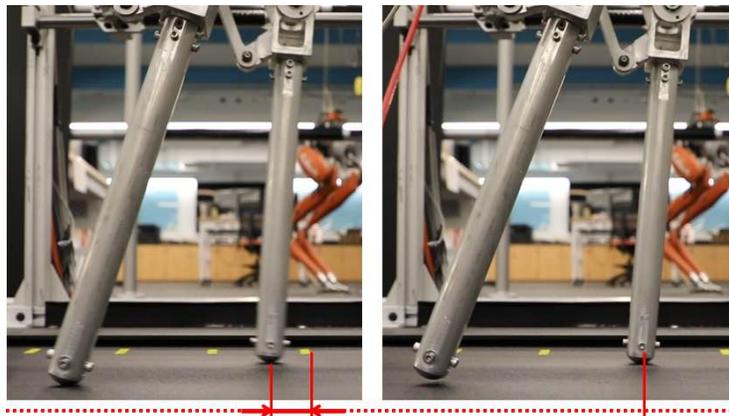


Figure 3.7: Slippage in the beginning of a step: pre-slip on the left and post-slip on the right.

The section aims to address this fundamental assumption of no slippage by embracing its violation while still demonstrating the ability to achieve stable walking experimentally. In legged robots, foot slippage is often treated as an external disturbance which should be avoided at the gait planning stage (T.-H. Chang and Hurmuzlu, 1993; Shuuji Kajita et al., 2004), or detected and recovered in real-time by feedback control at the experimental implementation stage (Kaneko et al., 2005;

Vázquez and Velasco-Villa, 2013). Some of the most famous examples are Boston Dynamics' robots BigDog (*Big Dog Walking* 2010) and SpotMini (*Spotmini Walking* 2016) successfully recovering from slippage. Conversely, legged animals across a wide range of scales show impressive adaptability to slippery surfaces on natural terrains. Stick insects when confronted with a slippery surface modulate their motor outputs to produce normal walking gaits, despite a drastic change in the loads that these limbs experience (Gruhn, Zehl, and Büschges, 2009). Slippage in the bipedal running of Guinea fowl has been studied in (Clark and Higham, 2011), showing that falling on slippery surfaces is a strong function of both speed and limb posture at touchdown. Several works in human biomechanics literature study the conditions that cause slipping (Moyer et al., 2006), its consequences (Tinetti and Williams, 1997) and dynamics (Strandberg and Lanshammar, 1981). Finally, (Spence et al., 2007) has measured feet motion in galloping gaits of horses on outdoor racing terrains and found a significant phase of hoof slippage.

Recent theoretical work has incorporated slippage into classic simple planar models of legged locomotion both in passive dynamics and actuated walking — the rimless wheel (Gamus and Yizhar Or, 2015), compass biped (Gamus and Yizhar Or, 2015; Gamus and Yizhar Or, 2013), and SLIP (Yizhar Or and Moravia, 2016). The models use Coulomb's friction law and account for stick-slip transitions and friction-bounded inelastic impacts, which add complexity to the system's multi-domain hybrid dynamics. By investigating the influence of friction on both passive dynamics down a slope and open-loop actuated walking, it has been found in (Gamus and Yizhar Or, 2015; Gamus and Yizhar Or, 2013) that upon decreasing the friction coefficient, periodic solutions with stick-slip transitions begin to evolve while their orbital stability decreases until reaching stability loss when the friction is too low. Nonetheless, stability can be recovered when adding simple PD control to track a reference trajectory. In addition, it has been found in (Gamus and Yizhar Or, 2015; Gamus and Yizhar Or, 2013; Yizhar Or and Moravia, 2016) that periodic solutions with slipping impact showed a significant reduction in the energetic cost of transport compared to their no-slip counterparts. Yet, these promising theoretical results have never been tested and implemented experimentally on legged robots.

In this section, we bridge this gap by presenting an experimental realization of stable planar bipedal robotic walking on a slippery surface.

About the Robot. For the bipedal robot AMBER3-PF (PF is short for *point foot*, see Fig. 3.8), the configuration space is chosen as $q \in Q \subseteq \mathbb{R}^n$, where n is the

number of unconstrained degrees of freedom (DOF), i.e. without considering contact constraints. Using the *floating base convention*, we have $q = (q_b, q_l)$, where $q_b \in \mathbb{R}^3$ is the global coordinate of the body-fixed frame attached to the base linkage (torso), and $q_l \in \mathbb{R}^4$ is the local coordinate representing rotational joint angles. For planar walking on AMBER3-PF, it is chosen as $q_b = (p_x, p_y, \theta_y)$, where p_x, p_y are the Cartesian positions of the torso and θ_y is the angle between the torso and world. The local coordinates are chosen as $q_l = (q_{sk}, q_{sh}, q_{nsh}, q_{nsk})$, each representing the stance knee, stance hip, non-stance hip, and non-stance knee joint angle. Further, the continuous-time state space $\mathcal{X} = TQ \subseteq \mathbb{R}^{2n}$ has coordinates $x = (q^\top, \dot{q}^\top)^\top$. The control inputs $u \in \mathcal{U} \subseteq \mathbb{R}^m$ represents the actuator torques, with m the total number of motors. For AMBER3-PF, we have 4 motors on both knee and hip joints. This indicates under-actuated dynamics for AMBER-3M walking.

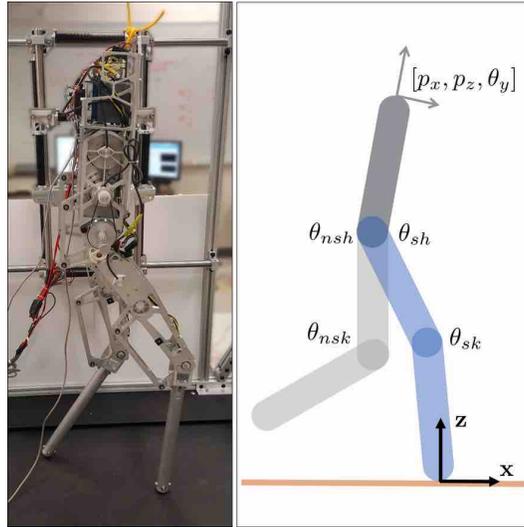


Figure 3.8: On the left: The AMBER-3M with point foot, constrained to a planar rail to walk in a 2D environment on a treadmill. On the right: the model's configuration coordinates, with 3 global coordinates and 4 local coordinates.

Continuous-time Dynamics for Stick/Slip Domains. The kinematic constraint of zero normal displacement of the stance foot reads as $z_s(q) = 0$. An additional no-slip constraint in tangential direction occurs only in the *stick* domain, and is given by $x_s(q) = x_0$. For a particular continuous domain $(q, \dot{q}) \in \mathcal{D}_v$, as introduced in Sec. 2.6 and Fig. 2.5 (b), the dynamics is formulated as

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J_x^\top(q)\lambda_x + J_z^\top(q)\lambda_z \quad (3.34)$$

where $D(q)$, $H(q, \dot{q})$, $B(q)$ are given by the physical parameters of the robot and thus remain the same across all continuous domains. In addition, the Jacobian matrices (constraint gradient vectors) in (3.3) are defined as

$$J_x(q) = \frac{\partial x_s(q)}{\partial q} \quad \text{and} \quad J_z(q) = \frac{\partial z_s(q)}{\partial q},$$

and λ_x, λ_z are the tangential and normal forces enforcing the contact constraints. In the domain of sticking contact, expressions for the contact forces can be obtained by augmenting the second time-derivative of the holonomic constraints:

$$\dot{J}(q, \dot{q})\dot{q} + J(q)\ddot{q} = 0, \quad \text{where } J(q) = \begin{pmatrix} J_x(q) \\ J_z(q) \end{pmatrix}. \quad (3.35)$$

Eliminating \ddot{q} from (3.34) and substituting into (3.35), one can solve for the constraint forces under sticking contact (cf. (R. M. Murray et al., 1994; Gamus and Yizhar Or, 2015)):

$$\begin{pmatrix} \lambda_x^0 \\ \lambda_z^0 \end{pmatrix} = \left(JD^{-1}J^\top \right)^{-1} \left(JD^{-1}(H - Bu) - \dot{J}\dot{q} \right), \quad (3.36)$$

where the dependencies on q, \dot{q}, u in (3.36) are suppressed for brevity. The forces must satisfy Coulomb's inequality of dry friction:

$$|\lambda_x^0(q, \dot{q}, u)| \leq \mu \lambda_z^0(q, \dot{q}, u), \quad (3.37)$$

where μ is the *coefficient of friction*. When μ is too low ($\mu \leq 0.1$ is assumed in this section), slippage of the stance foot in tangential direction begins to evolve, $\dot{x}_s = J_x \dot{q} \neq 0$. In this case, the equation of motion (3.34) still holds while the tangential constraint in (3.35) is no longer valid. Instead, the following two equations should be augmented with (3.34):

$$\dot{J}_z(q, \dot{q})\dot{q} + J_z(q)\ddot{q} = 0, \quad (3.38)$$

$$\lambda_x = -\text{sgn}(\dot{x}_s)\mu\lambda_z. \quad (3.39)$$

The tangential force during slippage reaches its maximal magnitude while opposing the slip direction. Note that we do not distinguish here between static and dynamic friction coefficients for simplicity. Combining (3.34) and (3.38) to obtain expressions for the constraint forces during slippage (Gamus and Yizhar Or, 2015):

$$\lambda_z^s(q, \dot{q}, u) = \left(J_z D^{-1} (J_z - \text{sgn}(\dot{x}_s) \mu J_x)^\top \right)^{-1} \left(JD^{-1}(H - Bu) - \dot{J}_z \dot{q} \right), \quad (3.40)$$

$$\lambda_x^s(q, \dot{q}, u) = -\text{sgn}(\dot{x}_s)\mu\lambda_z. \quad (3.41)$$

The inequality constraints for slippage are $\lambda_z \geq 0$ and $\dot{x}_s \neq 0$. Finally, in both domains the non-stance foot must stay above the ground, $z_{ns}(q) \geq 0$. Additionally, for a particular domain $v \in \{0, s\}$, we can convert the dynamics (3.34) and constraint forces in (3.36) or (3.40) into an affine control system in the state space as:

$$\dot{x} = f_v(x) + g_v(x)u \quad \forall x \in \mathcal{D}_v. \quad (3.42)$$

Discrete Dynamics. The reset maps associated with non-impacting transitions $\Delta[e_{0 \rightarrow s}]$, $\Delta[e_{s \rightarrow 0}]$, $\Delta[e_{s \rightarrow s}]$ are simply an identity matrix: $x^+ = x^-$, where x^- , x^+ are the pre-event and post-event states. This means that the transition is smooth in state space. In the case of collision of the non-stance foot, the transition involves impact which induces an instantaneous velocity jump $\dot{q}^+ = \Delta[e]\dot{q}^-$. The impulse-momentum balance reads as follows

$$D(q_c)(\dot{q}^+ - \dot{q}^-) = J(q_c)\Lambda = J_x(q_c)^\top \Lambda_x + J_z(q_c)^\top \Lambda_z, \quad (3.43)$$

where q_c is the robot's configuration at collision and $\Lambda = (\Lambda_x, \Lambda_z)^\top$ are tangential and normal impulses at the colliding foot. (Note that one has to interchange the stance and non-stance variables right before impact, so that J_x, J_z are associated with velocities of the colliding foot.) The commonly used model is that of perfectly inelastic impact. Assuming zero tangential and normal contact velocities at the post-impact state gives $J(q_c)\dot{q}^+ = 0$. Combining this with (3.43), one obtains the contact impulse and post-impact velocity as:

$$\Lambda^0 = \begin{pmatrix} \Lambda_x^0 \\ \Lambda_z^0 \end{pmatrix} = -(JD^{-1}J^\top)^{-1}J\dot{q}^-$$

$$\dot{q}^+ = \left(I - D^{-1}J^\top(JD^{-1}J^\top)^{-1}J \right) \dot{q}^-$$

where I is the identity matrix and D, J are evaluated at $q = q_c$. This is the sticking impact law, associated with reset maps $\Delta[e]$ for transition edges $e_{0 \rightarrow 0}^I, e_{s \rightarrow 0}^I$. This solution holds only if the impulses satisfy the frictional inequality $|\Lambda_x^0| \leq \mu\Lambda_z^0$. Otherwise, a slipping impact occurs where $J_z\dot{q}^+ = 0$ while $J_x\dot{q}^+ \neq 0$. The impulses are thus related as $\Lambda_x = -\text{sgn}(J_x(q_c)\dot{q}^+)\mu\Lambda_z$. Combining this with (3.43), one obtains:

$$\Lambda_z^s = -(J_z D^{-1} \tilde{J}^\top)^{-1} J_z \dot{q}^-$$

$$\dot{q}^+ = \left(I - D^{-1} \tilde{J}^\top (J_z D^{-1} \tilde{J}^\top)^{-1} J_z \right) \dot{q}^-$$

where $\tilde{J} = J_z - \text{sgn}(J_x(q_c)\dot{q}^+)\mu J_x$. This slipping impact law is associated with reset maps $\Delta[e]$ for transition edges $e_{0 \rightarrow s}^l, e_{s \rightarrow s}^l$.

Trajectory Optimization. To generate a slippery walking gait, we formulate this control problem as an implicit trajectory optimization problem. As was done in (3.33), we have

$$\begin{aligned} \min_{\alpha, x_i, \dot{x}_i, u_i} \quad & \sum u_i^\top u_i \quad i = 0, 1, 2, \dots, K \\ \text{s.t.} \quad & \text{(C.1)-(C.7)} \end{aligned} \quad (3.44)$$

C.8 slipping feasibility

with K the total number of collocation points, and the target is to minimize torque inputs. To yield a slipping gait, we additionally include feasibility constraints (C.8) from definitions in (2.28). In our formulation, we pre-specified a specific ordered sequence of transitions, indicated by the red line in Fig. 3.6 (b). Additionally, since a smoother state trajectory is preferred for experiment robustness, we further forced the static parameters to be the same across all domains. It is worthwhile to mention, this constraint is feasible if and only if the transition between domains within one step does not involve any jump in states. This yields a uniform trajectory for the multi-domain walking dynamics.

Optimal Gaits. Solving the optimization problem (3.44), we obtained a two-domain slippery walking gait with slippage on the stance foot 3 cm, shown with snapshots in Fig. 3.7. The $MCOT^+$ from optimization is given as 0.001. The *positive only mechanical cost of transport* is calculated using

$$MCOT^+ = \frac{\bar{P}^+}{mgv} \quad (3.45)$$

with m the total mass, g the gravitational acceleration, v the average walking speed, and \bar{P}^+ is the mean value of $P^+ = \{P_i^+\}_{i=1}^N$ with $i \in \{1, 2, 3 \dots N\}$ and N is the total number of sample points. The positive only power at sample time t_i is computed by $P_i^+ = \sum_{k=1}^4 \max(u_i(k) \cdot \dot{q}_i(k), 0)$, with $u_i \in \mathbb{R}^4$ and $q_i \in \mathbb{R}^4$ torques and velocities of the actuated joints at time t_i .

For a fair comparison against *sticky walking*, we simulate the slippery gait-based controller in a sticky environment, i.e., the ground has a much higher friction coefficient so that no slipping can happen. After 20 ~ 30 steps, the walking converged into a new stable patten. with $MCOT^+$ being 0.0024. which is 140% less energy efficient than walking on a slippery surface. Further at its steady state, the

non-stance foot's velocities changed from $(0.563, -0.359)\text{m/s}$ to $(0, 0)\text{m/s}$ through the sticky impact. The body kinetic energy changed from 6.87J to 5.84J. However, the original optimal slippery gait has a non-stance foot impact velocity changing from $(0.371, -0.237)\text{m/s}$ to $(0.251, 0)\text{m/s}$, and kinetic energy changing from 3.00J to 2.63J. This agrees with the theories on energy efficiency in (Gamus and Yizhar Or, 2015).

Experiment. AMBER-3M is a modular testbed to study planar bipedal locomotions. Its robustness and durability were validated in multiple experiments (Ambrose, W. Ma, C. Hubicki, et al., 2017; Tabuada et al., 2017). In this section, we particularly studied the slippery walking behavior on the point foot version (with total mass 21.6kg). As detailed in (Ambrose, W. Ma, C. Hubicki, et al., 2017), planar walking is achieved by constraining the robot on a planar rail structure and walking on a treadmill (Fig. 3.7). Further experimental details will be presented in this section.

To begin, we placed a demonstration walking gait designed for a sticky surface on a slippery surface covered by some lubricant. This gait has been shown to be robust over countless trials via public demonstrations, and it seldom fell, i.e., loss of stability. However, a few drops of lubricant easily disabled its walking capability. To clarify, we consider falling and hitting the mechanical limits of the testbed both as failures. Later, we conducted four different experimental setups. For *experiment 1, 2, and 3*, we increased the amount of lubricant on the treadmill to induce different slippery walking behaviors and completely removed the lubricant for *experiment 0*. For each fixed environmental setup, we manually increased the treadmill speed to trigger different walking speeds on the slippery surface. We logged 50 seconds' data (sampling period 3 ms) for each experiment to calculate the energy economy. Fig. 3.10 shows the phase portrait for *experiment 3* which has the most slippery surface. The result is AMBER-3M is capable of walking stably on different slipping conditions, including on a sticky surface, proving its robustness and adaptability to uncertainties induced by differences between simulation and experiment. See (Wen-Loong Ma, 2019a) for the walking on slippery surfaces.

Energy Economy. Previous research (Ambrose, W. Ma, C. Hubicki, et al., 2017) on AMBER-3M with a circular boom has benchmarked the energy economy of walking controllers. In this research, due to the slippage of stance foot, it became too noisy to measure the movement of the *center of mass*. In this work, we used the measure (3.45) for $MCOT^+$. See Fig. 3.11 for the energy results. Note that we only provide *positive only power* because AMBER-3M's hardware cannot do power-regeneration

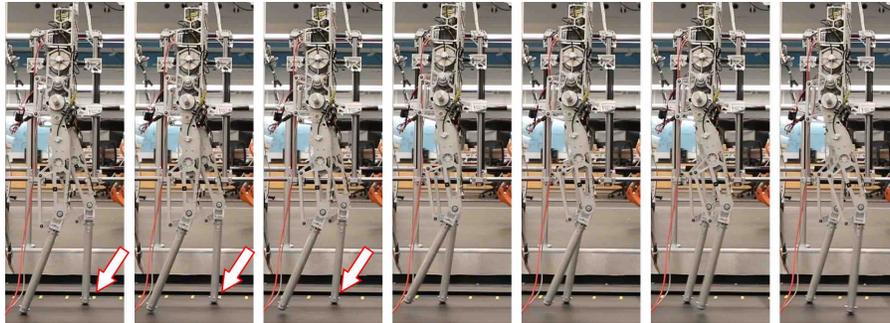


Figure 3.9: Snapshots of one slippery walking step from *Experiment 3*. In the first three pictures, the left foot (stance foot) is slipping smoothly on the lubricated treadmill.

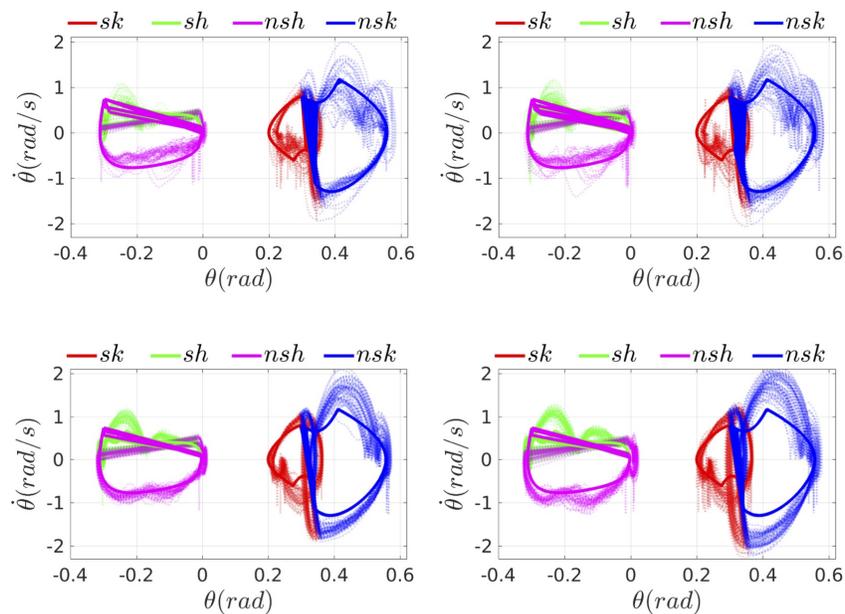


Figure 3.10: Phase portrait of 50 seconds' experimental data from *Experiment 3*, with walking speed (from left to right): 0.26m/s, 0.3m/s, 0.38m/s, 0.42m/s. Solid lines are for the desired values and dashed lines are for the actual measurements.

of the negative work.

While the energy efficiency Fig. 3.11 seems better than (Ambrose, W. Ma, C. Hubicki, et al., 2017), our measure shows experiment energy efficiency is ~ 10 times worse than simulation, and the efficiency on different surfaces does not vary as much as simulation data. This is not only caused by different external environments such as inconsistency of the lubricated treadmill and real-world uncertainties, but we posit the dominance of the $MCOT^+$ by nominal energy usage of the robot. That is, due to

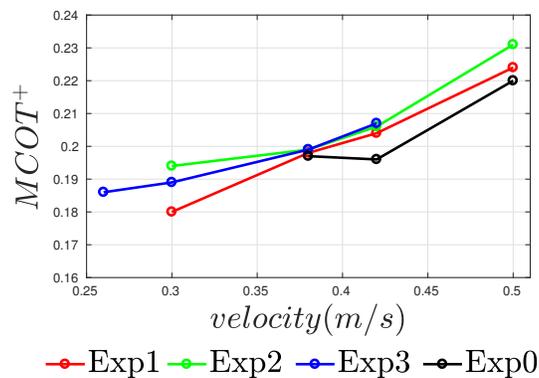


Figure 3.11: $MCOT^+$ of the experiments on AMEBR-3M. Those not included for certain speeds are failed experiments.

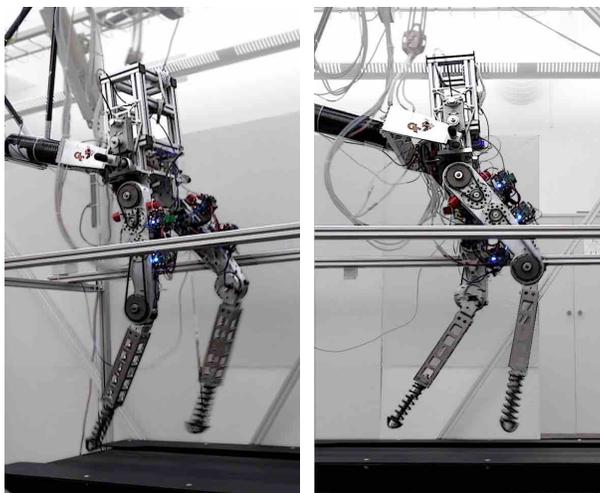


Figure 3.12: The spring-legged planar running biped, DURUS-2D, during take off (left) and while airborne (right).

the order of magnitude difference in the simulation and experimental $MCOT^+$, the comparatively small fluctuations in the $MCOT$ between different walking cannot be observed with the current experimental setup. Therefore, it is necessary to study differences in the $MCOT^+$ between slipping and nominal gaits wherein changes in energy usage can be isolated from nominal energy usage and the effects of the environment on the cost of transport.

3.4 Bipedal Running with Input to State Stability

The task of controlling the bipedal robots is often a precarious balance between maintaining formal stability guarantees and expanding control capabilities. This duality has been present since the genesis of bipedal control. Beginning in the

1960's, Zero Moment Point (Vukobratovic and Branislav Borovac, 2004) methods were the original foundation of formal biped control. Still, its validity required significant restrictions on the dynamics of the robot (fully-actuated flat-footed contact). In contrast, the Raibert hoppers (M. Raibert and Tello, 1986) exhibited agile bounces and flips that remain impressive today. But their control was built without the *a priori* confidence of formal methods. Research over the following decades has considerably narrowed this formality gap, with formal approaches rising to the challenge of underactuation (Manchester et al., 2011; H. Park et al., 2012; Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, 2013; Ayonga Hereid, Eric. Cousineau, et al., 2016) and highly dynamic robots incorporating formal analysis in their control (Bhounsule et al., 2014; Rezazadeh et al., 2015).

Bipedal robotic running, despite the decades that have passed since Raibert's hopper, remains an extremely difficult control problem. Very few control methodologies have been presented that lead to experimental success with prominent aerial phases (Tamada et al., 2014; Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, 2013). With an eye toward viewing bipedal running as a hybrid dynamical system: an alternating sequence of stance and flight domains with instantaneous impacts in between, the notion of hybrid zero dynamics (HZD) was used (J W Grizzle, Abba, and Plestan, 2001; A. Ames, 2014; A. E. Martin, Post, and Schmedeler, 2014a).

HZD, as previously introduced, operates on a principle of dimensional reduction, aimed at simplifying the numerous degrees of freedom present in legged machines while also allowing for underactuation. This framework was used to enable bipedal running on MABEL (Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, 2013), a pivotal demonstration showing the intersection of theory and experiment. However, on top of the HZD framework used on MABEL, there are also important expert-driven adjustments to the implementation, like the tuning of control loops, adding feedforward trajectories, and online parameter update routines. One way to interpret this is: the gap between the assumed model and the experimental testbed necessitated modifications in the control implementation needed to realize stable robotic running. We seek to reduce further the need for this expert adjustment with formal stabilizing controllers.

In this section, we will study this methodology on the DURUS-2D running robot. First, using the optimization algorithm mentioned in Sec. 3.3, we find some periodic solution to the closed-loop dynamics of running, which is stable in the sense of Lyapunov. However, the resulting gait that was built upon an ideal model and precise

sensing cannot guarantee experimental realization. Unlike theoretical simulation, where most variables are either measurable or exclusively solvable, real-world experiments suffer from a wide array of uncertainties. Indeed, uncertainties like unmodeled dynamics, nonlinear stiffness properties, damping effects and actuators, poor signal-to-noise ratio, and even deformations due to impacts are often observed. Therefore, we not only seek a fast optimization approach that yields feasible solutions under the assumed model but also a controller formally guaranteeing robustness under real-world constraints. In this section, we use the notion of *input to state stability* (ISS) that captures the practical limitations of the actuator inputs in an elegant manner. Specifically, we address the *phase*-based uncertainty that is typically a high deterrence in tracking parameterized functions. Similar problems involving inaccurate phase determinations were solved (Shishir Kolathaya, Ayonga Hereid, and Aaron D Ames, 2016), where pure time-based parameterizations were used. But this paper will construct time+state-based parameterizations to yield stronger stability conditions. Note that, in order to realize running, a variety of uncertainties need to be considered. So we will use the solutions from (Shishir Kolathaya and D. A. Ames, 2016; Angeli, 1999) to account for the remaining uncertainties.

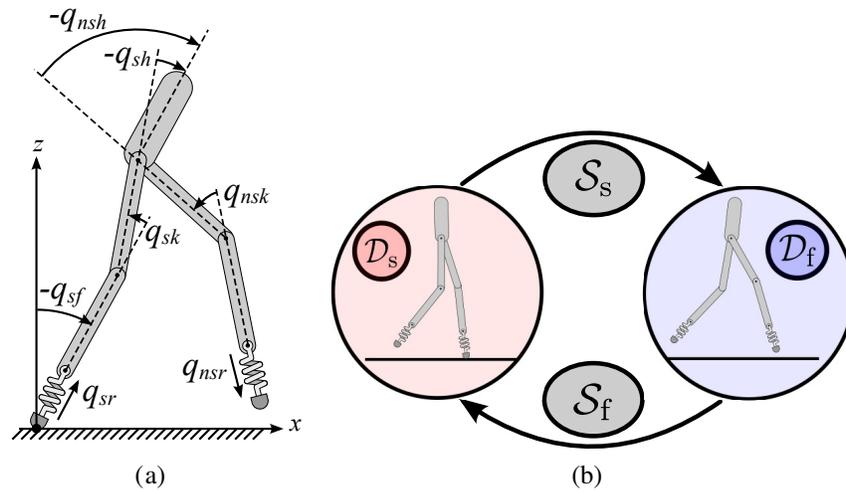


Figure 3.13: (a) The model of DURUS-2D with two linear springs; (b) the directed cycle structure of the multi-domain hybrid system model for DURUS-2D running.

Robot and Hybrid Dynamics. Similar to the setup in Sec. 3.3, we have a hybrid model for DURUS-2D running. As shown in Fig. 3.13a, the configuration space $Q \subset \mathbb{R}^n$, $n = 9$, of DURUS-2D is defined as

$$q = (s f_x, s f_z, \theta_{sf}, r_{sp}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, r_{nsp})^T \in Q,$$

where sf_x and sf_z are the positions of the end points of the stance foot along x and z directions, r_{sp}, r_{nsp} are the deflections of the springs on stance and non-stance legs, θ_{\square} are the joint angles of the stance foot, stance knee, stance hip, nonstance hip, and nonstance knee. In addition, the control inputs are defined as $u = (u_{sk}, u_{sh}, u_{nsh}, u_{nsk})^T \in \mathbb{R}^k$, $k = 4$, which represent the torque applied at knee and hip joints.

We have the same hybrid setting as those presented in (3.1) with a small modification of the continuous dynamics. The Equation of Motion (EOM) over a continuous domain \mathcal{D}_v , with $v \in \mathcal{V} \triangleq (s, f)$, is determined by the Euler-Lagrange equation and holonomic constraints

$$\begin{aligned} D(q)\ddot{q} + H(q, \dot{q}) &= Bu + J_v^T(q)F_v, \\ J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} &= 0, \end{aligned} \quad (3.46)$$

where $J_v(q) \in \mathbb{R}^{n \times m}$ is the Jacobian of the holonomic constraints $h_v(q)$, and $F_v \in \mathbb{R}^m$ is a *wrench* containing the constraint forces or moments, which can be explicitly solved as a function of system states and inputs. The holonomic constraints for each domain are defined as

$$\begin{aligned} h_s(q) &\triangleq (sf_x, sf_z, r_{nsp})^T \equiv 0, \\ h_f(q) &\triangleq (r_{sp}, r_{nsp})^T \equiv 0, \end{aligned} \quad (3.47)$$

meaning, the stance foot must remain on the ground during the stance domain, and stance and nonstance springs must be locked during the flight domain. Further, by defining the state vector $x = (q, \dot{q}) \in \mathbb{R}^{2n}$, the EOM can be converted to an affine control system, as given in (3.10).

We define the outputs (virtual constraint) of the system on a domain \mathcal{D}_v , $v \in \mathcal{V}$ as

$$y_v(q) = y^a(q) - y_v^d(\tau_v), \quad (3.48)$$

where the actual output is chosen as the four actuated joint angles:

$$y^a(q) \triangleq (\theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk})^T; \quad (3.49)$$

and y_v^d is the desired output represented by a set of fifth-order Bézier curves

$$y_v^d = \mathcal{B}(\alpha_v, \tau_v).$$

Therefore, the output has relative degree two with respect to inputs.

The *phase variable* τ_v is used to modulates the desired outputs y_v^d . Normally, in order to make the outputs purely state-based, we can have the phase variable $\tau_v : \mathcal{Q} \rightarrow \mathbb{R}_+$, purely a function of the robot configuration:

$$\tau_v(q) = \frac{\theta_{sf} - p_v^+}{p_v^- - p_v^+}, \quad (3.50)$$

with p_v^-, p_v^+ the (desired) initial and final position of θ_{sf} for \mathcal{D}_v . Although, it must be noted that state-based modulation has implementation difficulties due to noisy sensing of underactuated degrees of freedom of DURUS-2D. This motivates the use of a *time-based phase variable* $\tau_v : \mathbb{R}_+ \rightarrow \mathbb{R}_+$,

$$\tau_v(t) = \sum_{i=0}^5 p_i t^i, \quad (3.51)$$

where p_i is a set of power series polynomial coefficients obtained by a curve fitting from $\tau_v(q)$ w.r.t. time t . This has desirable stability properties under sensory perturbations, which will be discussed later.

The method of finding some optimal gait parameter α_v is then formulated as an optimization problem subject to the multi-domain hybrid system model. As was done in Sec. 3.3, we can produce some natural-looking running gaits for the bipedal robot, DURUS-2D (Fig. 3.12), in 43 seconds from a zero initial guess. See Fig. 3.17.

State-Based Feedback Controller and Zero Dynamics. To drive the virtual constraints (outputs) $y_v \rightarrow 0$ exponentially for each domain $\mathcal{D}_v, v \in \mathcal{V}$, we utilize the feedback linearization control law:

$$u_v = (L_g L_f y_v)^{-1} \left(-L_f^2 y_v + \mu_v \right), \quad (3.52)$$

with L the Lie derivative. Applying this control law yields the output dynamics $\ddot{y}_v = \mu_v$. Further, by picking μ_v as

$$\mu_v = -\frac{2}{\varepsilon} \dot{y}_v - \frac{1}{\varepsilon^2} y_v, \quad 0 < \varepsilon < 1, \quad (3.53)$$

the virtual constraints will converge to zero exponentially at the rate $1/\varepsilon > 0$. Since the number of virtual constraints is less than the degrees of freedom of the robot, the uncontrolled states evolve according to the *zero dynamics*. In other words, we have a set of states defined by the vector:

$$\eta_v \triangleq \begin{bmatrix} y_v \\ \dot{y}_v \end{bmatrix} \in \mathbb{R}^{2k}$$

that are controllable, and the set of states defined by z_v , that are uncontrollable and normal to η_v for each domain \mathcal{D}_v . We can then reformulate (3.46) to the following form:

$$\dot{\eta}_v = \underbrace{\begin{bmatrix} 0_{k \times k} & 1_{k \times k} \\ 0_{k \times k} & 0_{k \times k} \end{bmatrix}}_F \eta_v + \underbrace{\begin{bmatrix} 0_{k \times k} \\ 1_{k \times k} \end{bmatrix}}_G \mu_v$$

$$\dot{z}_v = \omega_v(\eta_v, z_v), \quad (3.54)$$

where ω_v is assumed Lipschitz continuous. The convergence of the outputs η_v can be shown in terms of Lyapunov functions:

$$V_\varepsilon(\eta_v) = \eta_v^\top P_\varepsilon \eta_v, \quad (3.55)$$

where P_ε is the solution to the *continuous time algebraic Riccati equation (CARE)*. See (A. Ames, Galloway, et al., 2014a, eq(23)). By choosing $\mu_v(\eta)$ from (3.53), we have

$$\dot{V}_\varepsilon \leq -\frac{\gamma}{\varepsilon} V_\varepsilon$$

with $\gamma > 0$ the constant obtained from the CARE.

Given the control law (3.53), the controllable states η_v are driven exponentially to zero. In other words, the control law (3.52) renders the *zero dynamics surface* exponentially stable and invariant over both continuous domains. However, due to the impact dynamics at the end of each domain, the zero dynamics invariance is not guaranteed. Therefore, the goal is to find a set of parameters $\alpha = \{\alpha_s, \alpha_f\}$, which defines the desired outputs (3.48), to ensure there exists a periodic orbit and the zero dynamics surface:

$$\mathcal{Z}_v = \{(q, \dot{q}) \in \mathcal{D}_v \mid y_v(q) = 0, \dot{y}_v(q, \dot{q}) = 0\}, \quad v \in \mathcal{V},$$

is invariant through impacts, i.e., hybrid invariant. Mathematically, hybrid invariance is represented as

$$\Delta(\mathcal{Z}_s \cap \mathcal{S}_s) \subset \mathcal{Z}_f, \quad \Delta(\mathcal{Z}_f \cap \mathcal{S}_f) \subset \mathcal{Z}_s. \quad (3.56)$$

Feedback Linearization for Time-Based Outputs

By using the feedback control law given by (3.24) and (3.53), it can be shown that with sufficiently small ε , the output dynamics are exponentially driven to

zero. In fact, (A. Ames, Galloway, et al., 2014a) shows that by picking a *rapidly exponentially stable control Lyapunov function* (RES-CLF), locally exponentially stable hybrid periodic orbits can be realized. However, in reality, due to the difficulty in estimating the phase variable (3.50) (which depends on the un-actuated degrees of freedom), a more robust controller is required that is less susceptible to the noisy state feedback. Motivated by the time-based implementation of the tracking controller in (Shishir Kolathaya, Ayonga Hereid, and Aaron D Ames, 2016; Jake Reher, Wen-Loong Ma, and Aaron D. Ames, 2019), the goal of this section is to construct a controller that uses a time-based instead of state-based desired trajectory for robotic running.

For the ease of notations, we will omit the domain representations (the subscripts v) in this section. If the state-based desired relative degree two outputs are functions of q , $y^d : \mathcal{Q} \rightarrow \mathbb{R}^k$, then the time-based desired outputs are functions of time $y^{t,d} : \mathbb{R}_+ \rightarrow \mathbb{R}^k$.

Following (3.51), we define the time-based output representation as follows:

$$y(t, q) = y^a(q) - y^{t,d}(\tau(t)). \quad (3.57)$$

Taking the derivative of (3.57) twice, we have

$$\ddot{y}(t, q) = L_f^2 y^a + L_g L_f y^a u - \ddot{y}^{t,d}, \quad (3.58)$$

which is different from (3.23). Similar to the construction of state-based controller (3.24), we would drive $y \rightarrow 0$ exponentially. Therefore, the feedback controller that linearizes the time-based output is given as

$$u_t = (L_g L_f y^a)^{-1} \left(-L_f^2 y^a + \ddot{y}^{t,d} + \mu_t \right), \quad (3.59)$$

where μ_t is the linear feedback applied after the feedback linearization. We can either pick μ_t via a simple PD law:

$$\mu_t = -\frac{2}{\varepsilon} \dot{y} - \frac{1}{\varepsilon^2} y, \quad (3.60)$$

for some $0 < \varepsilon < 1$, or via an optimal control law through *control Lyapunov functions* (CLFs). Nevertheless, using the time-based feedback linearizing controller (3.59) reduces the nonlinear system $\dot{x} = f(x) + g(x)u$ to the normal form:

$$\begin{aligned} \dot{\eta}_t &= \underbrace{\begin{bmatrix} 0_{k \times k} & 1_{k \times k} \\ 0_{k \times k} & 0_{k \times k} \end{bmatrix}}_F \eta_t + \underbrace{\begin{bmatrix} 0_{k \times k} \\ 1_{k \times k} \end{bmatrix}}_G \mu_t \\ \dot{z}_t &= \omega_t(\eta_t, z_t), \end{aligned} \quad (3.61)$$

which is similar to (3.54), but with the use of time-based outputs:

$$\eta_t \triangleq \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \in \mathbb{R}^{2k}.$$

Note that the zero dynamics coordinates, z_t , evolve based on time due to the dependency on η_t . Accordingly, if the time-based transverse dynamics η_t are 0, we have the zero dynamics $\dot{z}_t = \omega_t(0, z_t)$. Convergence of the time-based outputs can be ensured by picking an appropriate time-based control law (3.60). But this controller does not necessarily ensure the convergence of the state-based outputs. We are interested in the stability of the state-based transverse dynamics $\dot{\eta} = F\eta_t + G\mu_t$, given that the time-based control law is implemented on the robot. This implementation can be seen in Fig. 3.14, which is different from Fig. 3.2.

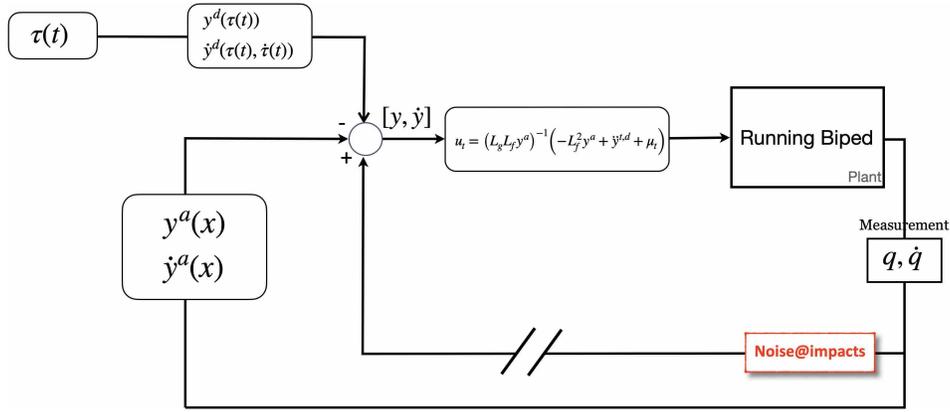


Figure 3.14: A block diagram showing the time-based feedback control structure.

State-Based vs. Time-Based Control Laws. Given the controller (3.60) that drives the time-based outputs $\eta_t \rightarrow 0$, we will study the evolution of the state-based outputs η in (3.54). By the assumption of Theorem 1 in (A. Ames, Galloway, et al., 2014a), the controller yields an exponentially stable periodic orbit for hybrid dynamics. Therefore, we will obtain conditions for the stability of this hybrid periodic orbit when a time-based control law is applied. Picking the input (3.59) on the dynamics of state-based output y , we have

$$\begin{aligned} \ddot{y} &= L_f^2 y + L_g L_f y u_t, \\ \ddot{y} &= \underbrace{L_f^2 y + L_g L_f y u}_{\triangleq \mu} + \underbrace{L_g L_f y (u_t - u)}_{\triangleq d}, \\ \ddot{y} &= \mu + L_g L_f y \left[(L_g L_f y^a)^{-1} (-L_f^2 y^a + \dot{y}_d^\top + \mu_t) - (L_g L_f y)^{-1} (-L_f^2 y + \mu) \right] \\ \ddot{y} &= \mu + d, \end{aligned} \tag{3.62}$$

where we can see that

$$d = L_g L_f y (L_g L_f y^a)^{-1} (-L_f^2 y^a + \ddot{y}^{t,d} + \mu_t) - \mu + L_f^2 y, \quad (3.63)$$

is obtained by substituting for u_t, u from (3.24) and (3.59). We can interpret (3.62) as that, the stabilizing control input $\mu(\eta)$ (which is state-based) should have been applied, but instead, the time-based input $\mu + d$ was applied to the state-based output dynamics of y . Applying a time-based feedback control law completely eliminated the dependency on the noisy phase variable $\tau(q)$, but the consequence is the appearance of the disturbance input d . The expression for d can be further simplified to

$$d(t, q, \dot{q}, \ddot{q}, \mu_t, \mu) = (\mu_t - \mu) + (\ddot{y}^{t,d} - \ddot{y}^d). \quad (3.64)$$

We know that, $y^d = y^d(\tau(q))$ (for bipedal robots), and it can be observed that d becomes small by minimizing the error $\ddot{y}^{t,d}(\tau(t)) - \ddot{y}^d(\tau(q))$. Therefore d can be termed *time-phase uncertainty*, or just *phase uncertainty*.

In the context of linear systems, it is important to have bounded state-based output dynamics if d is bounded. Of course, the time-based outputs $\eta_t \rightarrow 0$. Denoting the supremum of the uncertainty over time as $\|d\|_\infty$, we can easily establish that a bounded d results in bounded outputs y, \dot{y} (or just η), for the continuous dynamics. However, due to the impact dynamics that are not just nonlinear but also extremely destabilizing (the noisy impacts can be observed in the video (Wen-Loong Ma, 2016c)), output boundedness cannot be guaranteed for the hybrid dynamics. This motivates using the notion of input to state stability to establish boundedness on the state-based outputs for bipedal robotic running on DURUS-2D.

Going back to (3.62), we can substitute this formulation in (3.54), which results in the following representation:

$$\begin{cases} \dot{\eta} = F\eta + G\mu + Gd \\ \dot{z} = \omega(\eta, z) \end{cases}. \quad (3.65)$$

As mentioned before, we are free to pick $\mu(\eta)$ (say (3.53)), since the actual control input applied is time-based $\mu_t(\eta_t)$ (from (3.60)) which is implicit in d . From the point of view of the state-based outputs η , we have the following representation dynamics of the Lyapunov function, which is given in (3.55):

$$\dot{V}_\varepsilon = \eta^\top (F^\top P_\varepsilon + P_\varepsilon F)\eta + 2\eta^\top P_\varepsilon G\mu + 2\eta^\top P_\varepsilon Gd. \quad (3.66)$$

This is obtained by substituting (3.65) for η . Using the linear feedback law $\mu(\eta)$ from (3.53), the following is obtained:

$$\dot{V}_\varepsilon \leq -\frac{\gamma}{\varepsilon}V_\varepsilon + 2\eta^\top P_\varepsilon G d. \quad (3.67)$$

It should be noted that even though the time-based controller leads to convergence of time-based outputs $y \rightarrow 0$, (3.67) extends it to state-based outputs y that are driven exponentially to an ultimate bound; and this ultimate exponential bound is explicitly derived from d , which is established via the notion of *input to state stability* (ISS), which is given below.

Input-to-State Stability (ISS)

We will first introduce the basic definitions and results related to ISS for a *general nonlinear system* and then focus on the running dynamics. See (Eduardo D Sontag, 2008) for a detailed survey on ISS.

Assume we have a general nonlinear system, represented in the following fashion:

$$\dot{x} = f(x, d), \quad (3.68)$$

with x taking values in Euclidean space \mathbb{R}^n , the input $d \in \mathbb{R}^m$ for some positive integers n, m . The mapping $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is considered Lipschitz continuous and

$$\mathbf{0} = f(\mathbf{0}, \mathbf{0}).$$

It should be noted that in this system, we considered general dynamics f and general inputs d . In the context of the robotic problem in this thesis, the construction is such that a stabilizing controller $u(x)$ has been applied (such as (3.24)), which results in a closed-loop system f . Any deviation from this stabilizing controller can be viewed as $u(x) + d$, with d being a new disturbance input. In the example of the linearized system (3.65), a suitable stabilizing controller $\mu(\eta)$ is applied and the effect of the disturbance input d is analyzed. We assume that d takes values in the space of all Lebesgue measurable functions: $\|d\|_\infty = \text{ess. sup}_{t \geq 0} \|d(t)\| < \infty$, which can be denoted as $d \in \mathbb{L}^\infty$.

Class \mathcal{K}_∞ and \mathcal{KL} functions. A class \mathcal{K}_∞ function is a function $\alpha : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ which is continuous, strictly increasing, unbounded, and satisfies $\alpha(0) = 0$. And a class \mathcal{KL} function is a function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that $\beta(r, \cdot) \in \mathcal{K}_\infty$ for each t and $\beta(\cdot, t) \rightarrow 0$ as $t \rightarrow \infty$.

We can now define ISS for the system (3.68).

Definition 8. The system (3.68) is *input to state stable (ISS)* if there exists $\beta \in \mathcal{KL}$, $\iota \in \mathcal{K}_\infty$ such that

$$|x(t, x_0)| \leq \beta(|x_0|, t) + \iota(\|d\|_\infty), \quad \forall x_0, \forall t \geq 0, \quad (3.69)$$

and considered locally ISS, if this inequality is valid for an open ball of radius r , $x_0 \in \mathbb{B}_r(\mathbf{0})$.

Definition 9. The system (3.68) is *exponentially input to state stable (e-ISS)* if there exists $\beta \in \mathcal{KL}$, $\iota \in \mathcal{K}_\infty$ and a positive constant $\lambda > 0$ such that

$$|x(t, x_0)| \leq \beta(|x_0|, t)e^{-\lambda t} + \iota(\|d\|_\infty), \quad \forall x_0, \forall t \geq 0, \quad (3.70)$$

and considered locally e-ISS, if the inequality (3.70) is valid for an open ball of radius r , $x_0 \in B_r(0)$.

Definition 10. The system is said to hold the *asymptotic gain (AG)* property if there exists $\iota \in \mathcal{K}_\infty$ such that

$$\overline{\lim}_{t \rightarrow \infty} |x(t, x_0)| \leq \iota(\|d\|_\infty), \quad \forall x_0, d. \quad (3.71)$$

Definition 11. The system is said to be *zero stable* if there exists $\beta \in \mathcal{KL}$ such that:

$$|x(t, x_0)| \leq \beta(|x_0|, t), \quad \forall x_0, \forall t \geq 0. \quad (3.72)$$

ISS-Lyapunov functions. We can develop Lyapunov functions that satisfy the ISS conditions and achieve the stability property.

Definition 12. A smooth function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is an *ISS-Lyapunov function* for (3.68) if there exist functions $\underline{\alpha}$, $\bar{\alpha}$, α , $\iota \in \mathcal{K}_\infty$ such that

$$\begin{aligned} \underline{\alpha}(|x|) &\leq V(x) \leq \bar{\alpha}(|x|) \\ \dot{V}(x, d) &\leq -\alpha(|x|) \quad \text{for } |x| \geq \iota(\|d\|_\infty). \end{aligned} \quad (3.73)$$

The following lemma establishes the relationship between the ISS-Lyapunov function and the ISS of (3.68).

Lemma 1. The system (3.68) is ISS if and only if it admits a smooth ISS-Lyapunov function.

The proof of was given in (Eduardo D Sontag, 2008). In fact, the inequality can be made stricter by using the exponential estimate:

$$\dot{V}(x, d) \leq -cV(x) + \iota(\|d\|_\infty), \quad \forall x, d. \quad (3.74)$$

which is then called the *e-ISS Lyapunov function*.

Phase Uncertainty to State Stability (PSS)

Coming back to our discussion in (3.67), we now define the notion of *phase to state stability (PSS)*. Without loss of generality, we denote $(\eta, z) = (\eta_v, z_v)$, and the subscript v will be specified when a specific domain (s or f) is considered.

Definition 13. Assume a ball of radius r centered at the origin $\mathbf{0}$. The system given by (3.65) is locally *phase to η stable*, if there exists $\beta \in \mathcal{KL}$, $\iota \in \mathcal{K}_\infty$ such that

$$|\eta(t)| \leq \beta(|\eta(0)|, t) + \iota(\|d\|_\infty), \quad \forall \eta(0) \in \mathbb{B}_r(0), \forall t \geq 0, \quad (3.75)$$

and it is locally PSS if

$$|(\eta(t), z(t))| \leq \beta(|(\eta(0), z(0))|, t) + \iota(\|d\|_\infty), \quad \forall \eta(0) \in \mathbb{B}_r(0), \forall t \geq 0. \quad (3.76)$$

Based on the asymptotic gain and zero stability property of the system (3.65) w.r.t. the *phase uncertainty* d , we have the following lemma.

Lemma 2. *Given the controller $\mu(\eta)$ in (3.53), the system (3.65) is phase to η stable.*

Proof. Based on the constructions of the Lyapunov function V_ε in (3.66), we have the dynamics of the from (3.67):

$$\begin{aligned} \dot{V}_\varepsilon &\leq -\frac{\gamma}{\varepsilon} V_\varepsilon + 2\eta^\top P_\varepsilon G d \\ &\leq -\frac{\gamma}{\varepsilon} V_\varepsilon + 2|\eta| \|P_\varepsilon\|_2 \|d\|_\infty \\ &\leq -\frac{\gamma}{2\varepsilon} V_\varepsilon \quad \text{for } |\eta| \geq \frac{4c_2}{\gamma c_1 \varepsilon} \|d\|_\infty, \end{aligned} \quad (3.77)$$

which is thus an ISS-Lyapunov function, as defined in (3.73). \square

Time+state-based Control Law. We can also realize exponentially ultimate boundedness of the entire dynamics by appending a state-based linear feedback law to the time-based feedback controller in (3.60):

$$u_T = u_t + \bar{\mu}, \quad (3.78)$$

which results in the following output dynamics in the place of (3.62):

$$\ddot{y} = \mu + d + L_g L_f y \bar{\mu}. \quad (3.79)$$

$L_g L_f y$ can be explicitly computed as $L_g L_f y = \mathcal{J} D^{-1} \bar{B}$, where D and \bar{B} are obtained from the generalized manipulator dynamics in (2.18), and

$$\mathcal{J} \triangleq \frac{\partial}{\partial q} y(q)$$

is the Jacobian of the outputs. Since D is invertible, it can be easily shown that $\mathcal{J} D^{-1} \bar{B}$ is invertible. By applying the controller (3.78), the full-order system (3.65) will have an extra input $\bar{\mu}$ that yields:

$$\begin{cases} \dot{\eta} = F\eta + G\mu + Gd + G\mathcal{J}D^{-1}\bar{B}\bar{\mu} \\ \dot{z} = \omega(\eta, z) \end{cases}, \quad (3.80)$$

then (3.67) gets reformulated as

$$\dot{V}_\varepsilon \leq -\frac{\gamma}{\varepsilon} V_\varepsilon + 2\eta^\top P_\varepsilon Gd + 2\eta^\top P_\varepsilon G\mathcal{J}D^{-1}\bar{B}\bar{\mu}. \quad (3.81)$$

By picking a control law for the auxiliary input:

$$\bar{\mu} = -\frac{1}{2\bar{\varepsilon}} (\mathcal{J}D^{-1}\bar{B})^{-1} G^\top P_\varepsilon \eta, \quad (3.82)$$

we have the following simplification of (3.81):

$$\dot{V}_\varepsilon \leq -\frac{\gamma}{\varepsilon} V_\varepsilon + 2\eta^\top P_\varepsilon Gd - \frac{1}{\bar{\varepsilon}} \eta^\top P_\varepsilon G G^\top P_\varepsilon \eta. \quad (3.83)$$

Therefore, by defining the positive semi-definite function

$$\bar{V}_\varepsilon(\eta) = \eta^\top P_\varepsilon G G^\top P_\varepsilon \eta,$$

we can pick $\bar{\varepsilon}$ small enough to cancel the effect of *phase uncertainty* on the dynamics. Lemma 2 can now be redefined to obtain exponential ultimate boundedness for the new control input (3.78).

Lemma 3. *Given the controllers $\mu(\eta)$ in (3.53), and $\bar{\mu}(\eta)$ in (3.82), the system (3.80) is exponentially phase to η stable w.r.t. the input disturbance $d \in \mathbb{L}^\infty$.*

Proof. We again pick the derivative of the Lyapunov function V_ε resulting in

$$\begin{aligned} \dot{V}_\varepsilon &\leq -\frac{\gamma}{\varepsilon} V_\varepsilon - \frac{1}{\bar{\varepsilon}} \eta^\top P_\varepsilon G G^\top P_\varepsilon \eta + 2\eta^\top P_\varepsilon Gd \\ &\leq -\frac{\gamma}{\varepsilon} V_\varepsilon \quad \text{for } |\eta| \geq \frac{2\bar{\varepsilon}c_2}{c_1^2\varepsilon^2} \|d\|_\infty, \end{aligned} \quad (3.84)$$

which satisfies the exponential estimate given by (3.74). \square

Periodic Orbit. Now Lemma 3 can be extended to include the uncontrolled states z given that they are stable on the zero dynamics manifold. Let $\mathcal{Y} \subset \mathbb{R}^{2k}$, $\mathcal{Z} \subset \mathbb{R}^{2(n-k)}$, and $\phi_t(\eta, z)$ be the flow of the dynamics in (3.80) with the initial condition:

$$(\eta, z) \in \mathcal{Y} \times \mathcal{Z};$$

And let the flow ϕ_t be periodic with the period $T_* > 0$, and a fixed point (η^*, z^*) if

$$\phi_{T_*}(\eta^*, z^*) = (\eta^*, z^*).$$

Associated with the periodic flow is the periodic orbit defined as

$$\mathcal{O} \triangleq \{\phi_t(\eta^*, z^*) \in \mathcal{Y} \times \mathcal{Z} \mid 0 \leq t \leq T_*\}. \quad (3.85)$$

Similarly, we denote the flow of the zero dynamics

$$\dot{z} = \omega(\mathbf{0}, z) \quad (3.86)$$

from (3.80) by $\phi_t|_z$, and for a periodic flow we denote the corresponding periodic orbit by $\mathcal{O}_z = \mathcal{O}|_z$. Due to the invariance of the zero dynamics (guaranteed by the HZD condition in Sec. 3.3), we have the mapping $\mathcal{O} = \iota_0(\mathcal{O}_z)$, where

$$\iota_0 : \mathcal{Z} \hookrightarrow \mathcal{Y} \times \mathcal{Z}$$

is the canonical embedding. For any (η, z) , we can denote the distance from \mathcal{O} as $\|(\eta, z)\|_{\mathcal{O}}$. We now have the following theorem to establish *phase-to-state stability* of the periodic orbit \mathcal{O} .

Theorem 1. *Assume that the periodic orbit $\mathcal{O}_z \subset \mathcal{Z}$ is exponentially stable in the zero dynamics. Given the controllers $\mu(\eta)$ in (3.53), $\bar{\mu}(\eta)$ in (3.82) applied on (3.80), that render the outputs exponential phase to η stable, then the periodic orbit \mathcal{O} obtained from the canonical embedding is exponentially phase to state stable.*

Proof. By the converse Lyapunov theorems, we can construct a quadratic Lyapunov function for the zero dynamics, $V_z(z)$ that satisfies the exponential inequality constraints:

$$\begin{aligned} r_1 \|z\|_{\mathcal{O}_z}^2 &\leq V_z \leq r_2 \|z\|_{\mathcal{O}_z}^2, \\ \dot{V}_z &\leq -r_3 V_z, \\ \left| \frac{\partial V_z}{\partial z} \right| &\leq r_4 \|z\|_{\mathcal{O}_z}, \end{aligned} \quad (3.87)$$

where $\|z\|_{\mathcal{O}_z} = \|(0, z)\|_{\mathcal{O}}$. We consider the following Lyapunov candidate for the full-order dynamics

$$V(\eta, z) = \sigma V_z(z) + V_\varepsilon(\eta). \quad (3.88)$$

This Lyapunov function is quadratic and satisfies the boundedness properties. Denote L_q as the Lipschitz constant of $\omega(\cdot, z)$. We can take the derivative

$$\begin{aligned} \dot{V} &= \sigma \frac{\partial V_z}{\partial z} \omega(0, z) + \dot{V}_\varepsilon \\ &= \sigma \frac{\partial V_z}{\partial z} \omega(0, z) + \sigma \frac{\partial V_z}{\partial z} (\omega(\eta, z) - \omega(0, z)) + \dot{V}_\varepsilon \\ &\leq -\sigma r_3 V_z + \sigma r_4 L_q \|z\|_{\mathcal{O}_z} \|\eta\| - \frac{\gamma}{\varepsilon} V_\varepsilon, \quad \text{for } |\eta| \geq \frac{2\bar{\varepsilon}c_2}{c_1^2\varepsilon^2} \|d\|_\infty. \end{aligned} \quad (3.89)$$

The rest of the derivation follows (A. Ames, Galloway, et al., 2014a, Appendix A.B), and the bounds on η are obtained from (3.84). By picking a suitable σ , we can render \dot{V}_ε negative definite, which satisfies Lemma 1. \square

This theorem has powerful implications due to the elimination of the noisy phase variable estimation. This elimination affects tracking, which yields lower errors than that for the noisy phase-based modulation. The time-based phase modulation is a smooth and better candidate to replicate the unknown actual phase of the robot. This methodology can be easily extended to all kinds of additive uncertainties observed in hybrid systems in general. See (Shishir Kolathaya and D. A. Ames, 2016) for the analysis on parameter uncertainty. Fig. 3.15 depicts the periodic orbit \mathcal{O} and the tube, which is defined by the bound δ_d . This theorem means that by using a time+state-based RES-CLF, any trajectory starting close to the tube will ultimately enter the tube defined by δ_d as long as $\|d\|_\infty < \delta_d$. This is also illustrated in Fig. 3.16.

Simulation and Experimental Realization of Bipedal Running

With the optimal running gait generated (introduced in Sec. 3.3) and time-dependent RES-CLF controller defined, we achieved sustainable robotic running. The goal of this section is to describe the experimental setup and the control methods adopted to realize stable running on DURUS-2D.

DURUS-2D Hardware. A popular approach for robotic running is to utilize the *spring-loaded inverted pendulum* (SLIP) model (R. Blickhan, n.d.; Rezazadeh et al., 2015), where the presence of springs allows for storing energy during high-speed

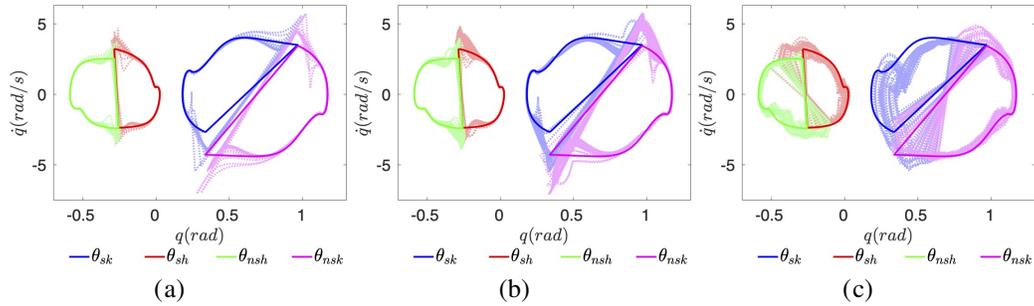


Figure 3.15: Limit cycles of (a) simulation where time-based IO + state-based PD controller was applied for 100 steps; (b) Simulation where white noise was added to $\tau_v(q)$; (c) Experimental data. Note the solid lines are the designated gait from optimization.

impacts, thereby improving energy efficiency and torque performance. The previous version of DURUS-2D (E. Cousineau and A. D. Ames, 2015), had rigid carbon fiber calves, unlike the current version, which has a linear spring at the end of each aluminum calf. The spring has a stiffness of 20000 N/m and damping constant of 100 N s/m. In addition, an 11.5 kg torso is installed to resemble the human weight distribution. The positions and velocities of the torso, knee, and hip joints are measured by the attached incremental encoders. Further, the actuated joints, knees and hips, are powered by BLDC motors via cycloidal gear reduction, which provides a maximum continuous torque of 200 N·m. With the new legs, DURUS-2D weighs 41.7 kg. EtherLAB software with MATLAB Simulink is built into DURUS-2D to guarantee a hard real-time environment. Other details about the electrical and software system can be found in (E. Cousineau and A. D. Ames, 2015).

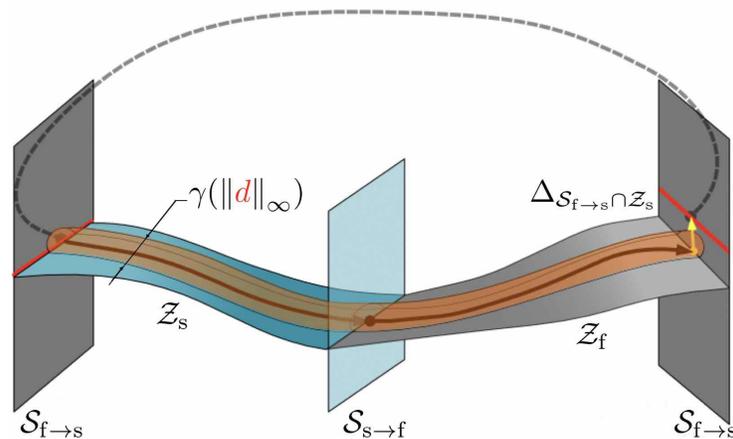


Figure 3.16: Bounded tube around the designed periodic orbit.

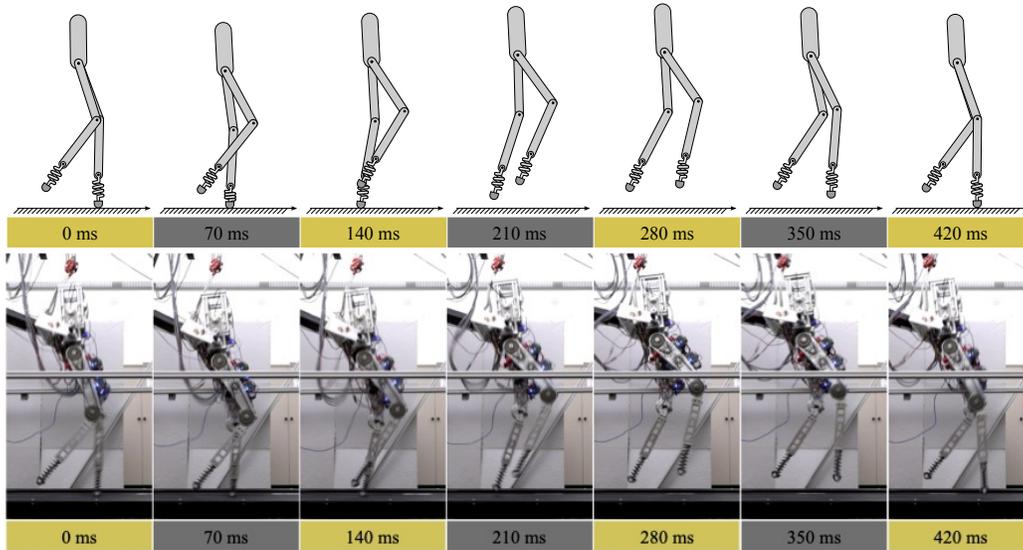


Figure 3.17: Running tiles of simulation (top) vs. experiment (bottom) for one step.

Experimental Setup. As shown in Fig. 3.18, DURUS-2D is mounted on a carbon fiber boom structure which is attached to a cage frame via a fixed one-dimensional track. This setup is used to isolate the lateral motions, leaving DURUS-2D to move freely in the sagittal plane. Moreover, the treadmill speed is measured by an encoder wheel and fed to the robot as environment feedback.

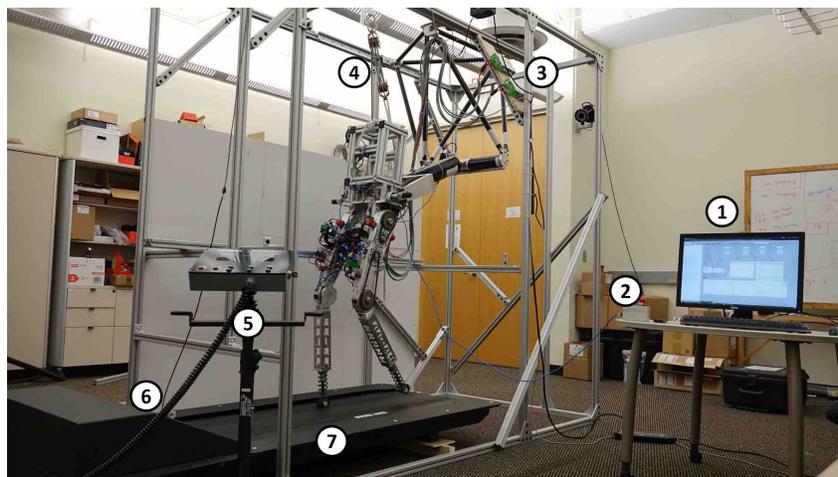


Figure 3.18: Experimental setup for DURUS-2D running: 1) control station computer, 2) emergency stop, 3) four Li-Po batteries, 4) tripping harness, 5) treadmill control panel, 6) encoder wheel to measure treadmill speed, 7) treadmill.

Switching logic. Guard condition is used to switch the controller to the subsequent domain (stance or flight). In simulation, the guard condition is triggered when non-stance spring returns to the neutral position for stance domain, i.e., $r_{sp} = 0$.

And when the nonstance foot lands on the ground, i.e., $nsf_z = 0$, the flight domain ends. However, due to a lack of an effective sensing mechanism, we developed a time+state-based switching logic for experiments. For a particular domain \mathcal{D}_v , the maximum value of time t_v^{max} and phase variable τ_v^{max} can be obtained from the gait design process. Then the guard condition is triggered when $t > 1.2t_v^{max}$. But if $t < 1.2t_v^{max}$, the guard will be triggered if $\tau_v(q) > \tau_v^{max}$. This way, the controller can respond to the feedback similarly to simulation while allowing for sensing noise of the phase variable.

Experimental Controller. Motivated by the results on ISS properties of PD controlled robotic systems in (Angeli, 1999), we can replace the time-based IO with a time-based PD control law, and claim that the resulting system still retains desirable stability properties. For a robot like DURUS-2D, the inertia of the motor (proportional to the square of the gear ratio) coupled with relatively light legs results in stronger ISS conditions for model-based uncertainty (see Shishir Kolathaya and D. A. Ames, 2016; Angeli, 1999). We therefore pick a time+state based PD control law as follows

$$u_E = -K_p^t y_v^t - K_d^t \dot{y}_v^t - K_p y_v - K_d \dot{y}_v, \quad (3.90)$$

where K_p^t, K_d^t, K_p, K_d are constant gain matrices with appropriately tuned values.

Results. We first validate the proposed control law in simulation. As explained in Sec. 3.3, a HZD running gait was first generated that meets all physical limitations, which assumed a feedback linearization controller (3.24). Then we utilized the time-based feedback linearization + state-based PD control law given by (3.78) in simulation; stable trajectory tracking is achieved that is ultimately bounded to the periodic orbit (see Fig. 3.21b for the evolution of virtual constraints, i.e., output errors, for 100 steps, and Fig. 3.15a for phase portrait that is also bounded around the desired gait) when the phase uncertainty is bounded (Fig. 3.21a).

However, in experiments, noisy sensing often occurs around impact dynamics. Therefore to simulate an unideal case, we added a noise signal with amplitude 0.1 to $\tau_s(q)$ before and after impacts (see Fig. 3.21c). By applying the same controller, ultimate boundness was also achieved (see Fig. 3.21d and Fig. 3.15b) and a stable bipedal running is accomplished. The running tiles are shown in Fig. 3.17. These simulated results, as a proof of concept, aligned with Theorem 1.

In reality, neither state-based phase measurements $\tau_v(q)$, nor time-based phase calculation $\tau_v(t)$ is capable of producing successful bipedal running (watch (Wen-Loong Ma, 2016c) for the failed running when the pure time-based controller was used). However, by applying a variant of time + state-based feedback

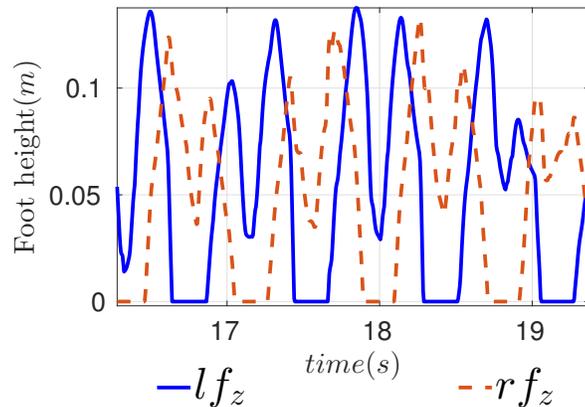


Figure 3.19: Left and right foot height (ground clearance).

as shown by (3.90), a sustainable running on DURUS-2D is immediately shown in real-world experiments. Multiple views in (Wen-Loong Ma, 2016a) show that the running is repeatable for over 150 steps. The phase portrait for 30 steps is shown in Fig. 3.15c, and the output errors are shown in Fig. 3.21f, both of which have shown ultimate boundedness. Further, the time-based and state-based phase variables are shown in Fig. 3.21e. Experimental running tiles are compared to simulation at Fig. 3.17. The most distinguishable feature of running, foot clearance, is shown in Fig. 3.19, with the maximum clearance 13 cm, and the flight domain takes 60% of one step. The average running speed is 1.75 m/s, and the measured average mechanical cost of transport (MCOT) for 100 steps is 0.5287.

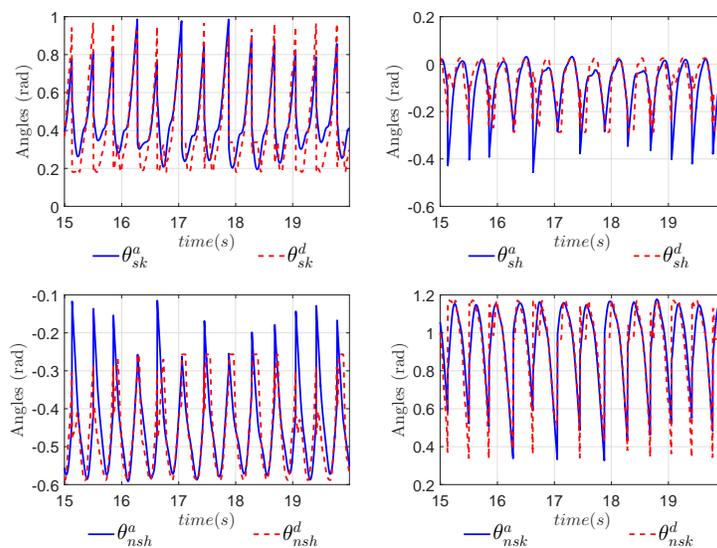


Figure 3.20: The actual vs. desired joint angles from the experiment.

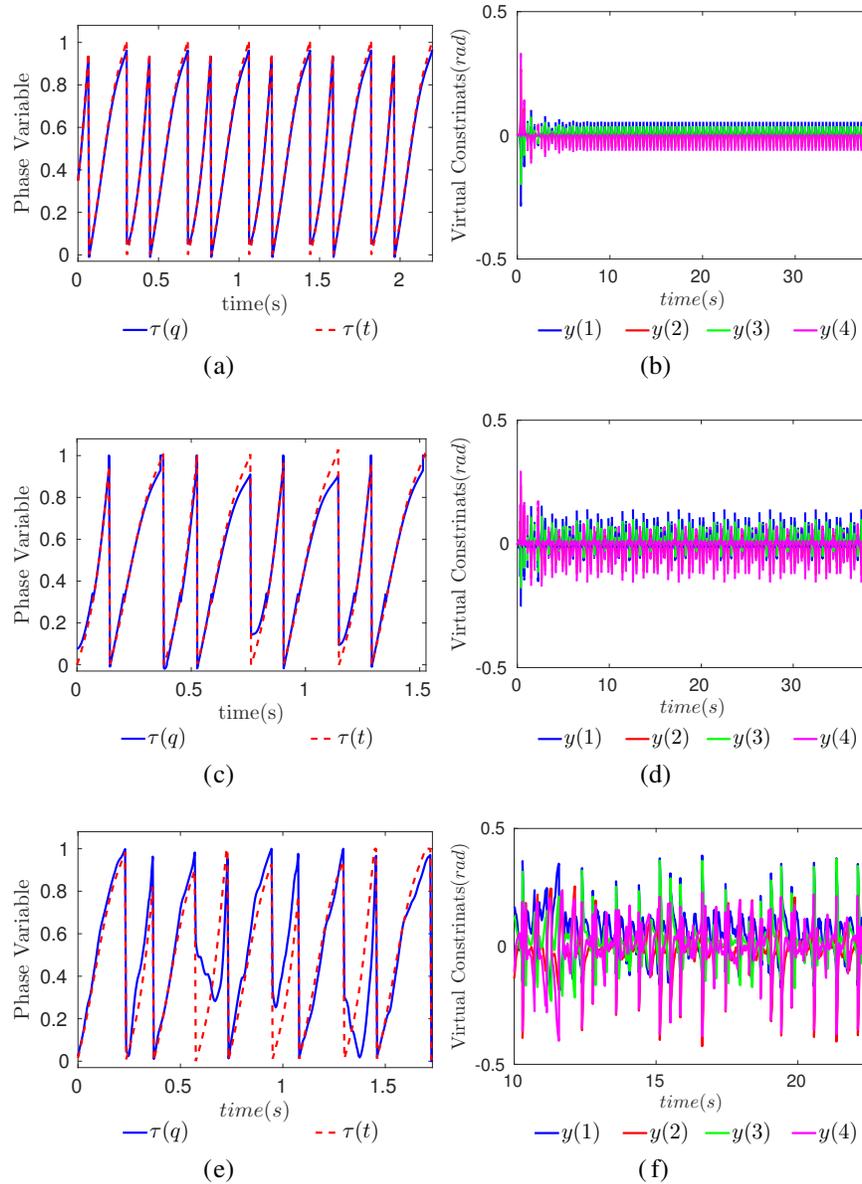


Figure 3.21: The left column is the phase variables which are used to calculate time- and state-based outputs: $y^{d,t}(t) = y_v^d(\tau_v(t))$, $y^d(q) = y_v^d(\tau_v(q))$; the right column is output errors (virtual constraints) showing ultimate boundedness. (a, b, c, d) are from two simulations with controller given by (3.78). And (c, d) has a sinusoidal disturbance with 10% amplitude added to $\tau_v(q)$; (e, f) are from experiments.

Chapter 4

COUPLED CONTROL SYSTEM AND QUADRUPEDAL APPLICATIONS

This chapter studies the control problem of quadrupedal locomotion. We first state our problem from the aforementioned HZD methodology in Chapter 3 to quadruped, enabling versatile gait generation and experimental realization. However, when it comes to real-world implementation, this methodology lacks implementation robustness and is thus not practical for field testing and iterative engineering. We then develop a framework, coupled control system (CCS), to design gaits and controllers for quadrupedal locomotion. The result is that we are able to apply this framework to two different quadrupedal robots (see Fig. 4.1) walking on various rough and sloped terrains.

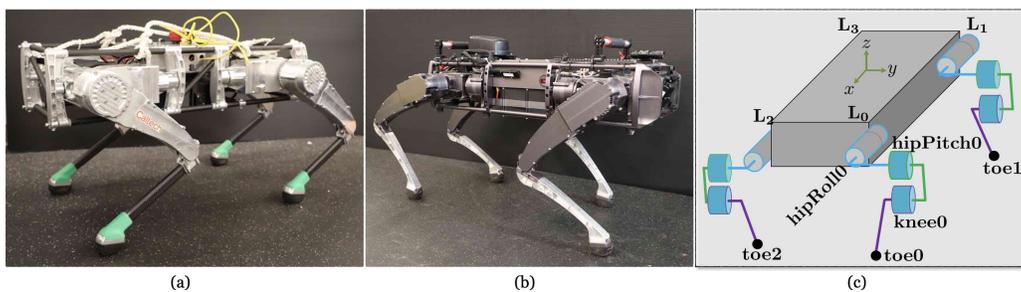


Figure 4.1: (a) Vision 60 V3.2, 26 kg, 0.45m tall; (b) Vision 60 V3.9, 44 kg, 0.55m tall; (c) Configuration coordinates of both quadrupedal robots. Both robots were manufactured by Ghost Robotics.

4.1 HZD, from Bipedal to Quadrupedal Robots

Quadrupedal locomotion has a long and rich history of outstanding agility and dynamic stability without formal analysis, thanks to the multi-support nature of such systems. Some famous quadrupedal examples include, but are not limited to, the BigDog (M. Raibert, Blankespoor, et al., 2008), Minitaur (De and Daniel E. Koditschek, 2018), ANYmal (M. Hutter et al., 2016), and Cheetah robot (Boussema et al., 2019a). State-of-the-art approaches for the controls and planning of quadrupeds mainly utilize model reduction to partly mitigate the computational complexity of the full-order techniques arising from nonlinearity and hybrid nature of models. For instance, massless legs, the linear inverted pendulum model (S. Kajita, Tani, and

Kobayashi, 1990; Hof, Gazendam, and Sinke, 2005) and planar motion planning are often utilized assumptions. While they do have many implementation advantages, one needs to design controllers that overcome the uncertainty induced by the difference between modeling and reality. For examples, the legs' total mass accounts for 44% of the mass of the whole body for the robot studied here in Fig. 4.15. Further, formal guarantees on stability are rarely studied in quadrupedal research.

On the other hand, the hybrid system approaches have become a powerful tool for modeling the dynamics of bipedal locomotion, in which, steady state locomotion corresponds to periodic solutions of these hybrid dynamical systems. One of the hybrid control approaches is the *hybrid zero dynamics* (HZD) framework (E. Westervelt et al., 2007). HZD is an extension of the notion of Byrnes-Isidori zero dynamics (Isidori, 1995) to hybrid models of locomotion for which the resultant zero dynamics manifolds are invariant under the continuous- and discrete-time dynamics. HZD has been successful for designing gaits for bipedal locomotion and provides experiment-level controllers, see e.g., (J.W. Grizzle, G. Abba, and F. Plestan, 2001; A. Ames, Galloway, et al., 2014a; C. Chevallereau, J.W. Grizzle, and Shih, 2009; Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, 2011; H.-W. Park, Ramezani, and J.W. Grizzle, 2013; Akbari Hamed, Buss, and J.W. Grizzle, 2016; R. Gregg and Righetti, 2013; A. E. Martin, Post, and Schmiedeler, 2014b; H. Dai and R. Tedrake, 2012; I. Poulakakis and J.W. Grizzle, 2009; Johnson, Burden, and D. E. Koditschek, 2016), but has not yet been applied to the control of quadrupeds. The challenge in computation and controls mainly arises from the increased degrees of freedom (DOF) and richer contact scenarios of quadrupeds over bipeds.

In this section, instead of building a controller based on the aforementioned empirically simplified models, we follow the HZD approach to design gaits for the full-order rigid-body dynamics of quadrupedal robots. In particular, we model their dynamics as a hybrid system, optimize trajectories via a HZD optimization algorithm, analyze the dynamic stability via the Poincaré return map and then validate the theoretically stable controller with experiments on a quadrupedal behavior, ambling on the Vision 60 robot (Fig. 4.1 (a)).

The Robot and Dynamics

Following the abstract construction of Sec. 2.1, we give more details of the configuration coordinates here. As shown in Fig. 4.1 (c), we model the quadruped as a 13-link system: one *body* link and four legs, each of which has three children

links —the *hip*, *upper*, and *lower* links. The configuration variables of Vision 60 are denoted by $q \in \mathcal{Q} \subset \mathbb{R}^n$ where $n = 18$ is the total number of degrees of freedom (DOF) without considering any contact constraints. Utilizing the floating base convention (Jessy W. Grizzle et al., 2014), we can have $q^\top = (\xi^\top, \theta^\top)$, in which $\xi \in \mathbb{R}^3 \times \text{SO}(3)$ represents the global Cartesian position and orientation of a frame attached to the *body* linkage, and the local coordinates $\theta \in \mathbb{R}^m$, with $m = 12$, denote the 12 joint variables: hip roll, hip pitch, and knee angles. These angles are denoted by $\theta_{hr_j}, \theta_{hp_j}, \theta_{k_j}$ for the j -th leg, all of which are actuated by Brushless DC electric (BLDC) motors. This yields the system's total DOF (without considering any constraints, see Sec. 2.2) to be 18 and control inputs $u \in \mathbb{R}^{12}$. With different scenarios of foot contacts with the ground, we have a mixture of overactuated, fully-actuated, and underactuated domains (i.e., phases) for the dynamics. We can define the state space $\mathcal{X} = T\mathcal{Q} \subseteq \mathbb{R}^{2n}$ with the state vector $x^\top = (q^\top, \dot{q}^\top)$.

We consider the nonlinear model of quadrupedal locomotion as a hybrid dynamical system, which is an alternating sequence of continuous- and discrete-time dynamics. The order of the sequence is dictated by contact events. In comparison with bipedal walking, the increased number of contact points of quadrupeds increased the complexity of the hybrid model substantially. In this section, we introduce a unified model for quadrupedal behaviors including walking, ambling, and trotting, based on which we design full-model-based optimal controllers as well as simulation validation.

The Continuous-Time Domain, Constrained Dynamics

Given the floating base coordinates, we can derive the unconstrained dynamics, i.e. without any contact constraints, by the Euler-Lagrange equations as:

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu$$

where the notations follows Sec. 2.3. Note that here the actuation matrix $B \in \mathbb{R}^{n \times m}$ only contains 0, 1 as its entries.

Because quadrupedal systems' numerous contact scenarios, we consider then $k \in \{1, 2, 3, 4\}$ feet standing on the ground, which means $\bar{k} = 4 - k$ feet are swinging in the air. This creates a variety of contact situations, which defines different types of quadrupedal behaviors (see Fig. 4.2). For each contact situation, we associate a *continuous domain*:

$$\mathcal{D}_v \triangleq \{(x, u) \mid h_v(q) = \dot{h}_v(q, \dot{q}) = 0, N_v \geq 0, h_{s,v}(q) \geq 0\} \quad (4.1)$$

where $v \in \{1, 2, 3, \dots\}$ is the domain index and $h_v(q) \in \mathbb{R}^k$ is the height of all standing feet with ground reaction force $N_v(x, u) \in \mathbb{R}^k$. The height of the other feet, referred as the swing feet, is given by $h_{s,v}(q) \in \mathbb{R}^{\bar{k}}$. We then have the constrained dynamics for \mathcal{D}_v as:

$$\begin{cases} D(q) \ddot{q} + H(q, \dot{q}) = B_v u + J_v^T(q) \lambda_v \\ J_v(q) \ddot{q} + \dot{J}_v(q, \dot{q}) \dot{q} = 0 \end{cases} \quad (4.2)$$

where $J_v(q) = \partial p_v(q) / \partial q$ represents the Jacobian matrix of the Cartesian position of the standing feet $p_v(q) \in \mathbb{R}^{3k}$, with the corresponding constraint wrench $\lambda_v \in \mathbb{R}^{3k}$. The first equation is considering ground forces as inputs and the second equation is treating the contacting as a holonomic constraint. Note that the actuation matrix B_v is domain-dependent. This is because the double and triple support phases (Fig. 4.2) create closed-chain structures that induce redundancy in control and constraints (internal overactuation). As detailed in Sec. 2.4, this is an underdetermined problem which often appears in multi-contact locomotion that could yield nonunique control value u . Therefore, we manually turn off the rear standing leg's hip pitch motor in the double support domain and turn off the diagonal standing legs' hip pitch motors and the other standing leg's hip roll motors for triple supporting phase. This implementation in return yields underactuated dynamics for the system. Now we can convert the constrained EOM (4.2) in domain v into the following form:

$$\dot{x} = f_v(x) + g_v(x)u. \quad (4.3)$$

To track a given set of time-based trajectories $\mathcal{B}_v(t)$, which will be detailed in later, we deployed an *input-output feedback linearization* controller:

$$u_{io}(x, t) = \mathcal{A}(x) (\mathcal{L}(x, t) - 2\varepsilon y(q, t) - \varepsilon^2 \dot{y}(q, \dot{q}, t)), \quad (4.4)$$

with the outputs $y(q, t) = y_a(q) - \mathcal{B}_v(t)$ and $\varepsilon > 0$. In this formulation, we chose the *actual outputs* $y_a(q)$ as all of the actuated joints. The notations follow directly from Sec. 3.2. As a result, (4.4) forces the system to converge to a desired gait exponentially, that is, $y_a(q) \rightarrow \mathcal{B}_v(t)$. The output dynamics become

$$\ddot{y} = -2\varepsilon \dot{y} - \varepsilon^2 y \quad (4.5)$$

for which $(y, \dot{y}) = (0, 0)$ is exponentially stable.

Prior to introducing the edges and the discrete dynamics, some assumptions are necessary to construct a feasible model both for computation and experiment:

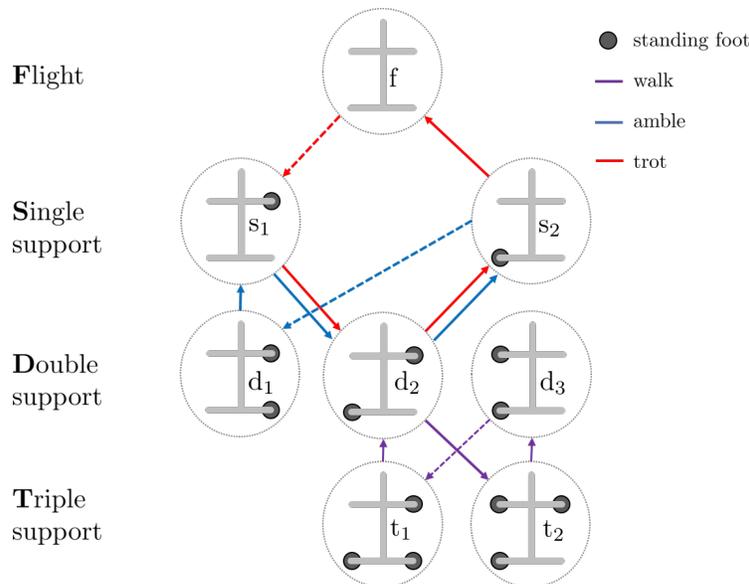


Figure 4.2: The cyclic directed graph for the multi-domain hybrid dynamics of walking, ambling, and trotting gaits. The dashed lines represent a *relabeling* map (as given in (3.32)) that flips the left and right legs’ contact attributes. We took these nomenclature from (Muybridge, n.d.).

- There is no ground slippage. This is partially guaranteed by enforcing a friction cone condition. However, it is worthwhile to mention that slipping locomotion has been observed on quadrupedal animals for energy efficiency, see (Gamus and Yizhar Or, 2013; W. Ma, Y. Or, and A D. Ames, 2019).
- The zoology studies (Muybridge, n.d.) have observed a pattern of 4×2 -domain¹ locomotion on quadrupedal animals. We refer interested readers to (Akbari Hamed, Wen-Loong. Ma, and Aaron. D. Ames, 2019) for the corresponding model. In this chapter, we assume the *stance leg transition domains*—*one leg strikes while another leg lifts*—are instantaneous and passive for walking and ambling. Hence s_1 and s_2 in Fig. 4.2 become edges, and we can have a 2×2 -domain behavior for *walk* and *amble*. But a 4×2 -domain model is still used for *trotting*.

The Discrete-Time Domain: Impact and Lift-Off

On the edge of \mathcal{D}_v in (4.1), one of the conditions reaches its bound. Thus we have two switching mechanisms:

¹The term $k \times 2$ denotes a gait with k -domain. See Fig. 4.2 for the directed graph of the gaits with k domains. The second half of the motion is directly a left-right mirror (i.e., symmetry) of the gait through the relabeling map. See Fig. 4.3 for the full motion with $m \times 2$ domains.

- *Lift off*: a standing foot of leg l_* lifts off from the ground, meaning $N_v^{l_*}(x, u) = 0$.
- *Impact*: a swing foot of leg l_* impacts the ground, meaning $h_{s,v}^{l_*}(q) = 0$, $\dot{h}_{s,v}^{l_*}(q, \dot{q}) < 0$.

For *lift off*, an identity map $x_{v+1}^+ = x_v^-$ is sufficient to represent the transition from current to the next domain, where x_v^- is the state at the end of the domain \mathcal{D}_v and x_{v+1}^+ is the state at the beginning of \mathcal{D}_{v+1} . However, the *impact* shall cause a jump in the velocity terms, the dynamics are given by

$$\begin{bmatrix} D & -J_{v+1}^T \\ J_{v+1} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_{v+1}^+ \\ \Lambda \end{bmatrix} = \begin{bmatrix} D\dot{q}_v^- \\ 0 \end{bmatrix} \quad (4.6)$$

which follows Sec. 2.7.

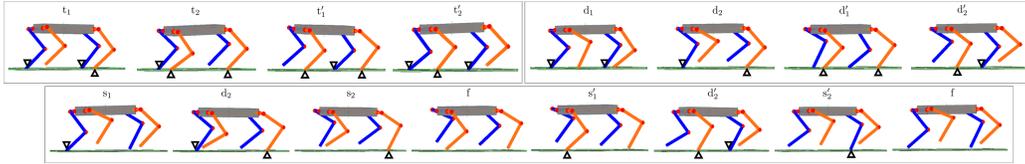


Figure 4.3: On the top left is the snapshot for a 2×2 -domain walking gait; on the top right is a 2×2 -domain ambling gait; and on the bottom is a 4×2 -domain trotting gait. The symbol \square' is a left-right mirror of the contact attributes for domain \square . The orange-colored legs are the left-side legs whose contact points are highlighted by triangles; the blue-colored legs are the right-side legs with contact points highlighted by inverted triangles.

Gait Design for Full-Order Dynamics

An alternating sequence of the continuous dynamics (2.19) and the discrete dynamics (2.31) composites a hybrid control system. Since this nonlinear hybrid model has captured abundant details of the dynamics, its complexity challenges the controller design and motion planning. We hereby employ an optimization toolbox FROST (Ayonga Hereid, C. M. Hubicki, et al., 2018) that parses hybrid system control problems into a nonlinear programming (NLP) based on direct collocation. As introduced in (3.44), to generate a feasible N -domain motion such as walking,

ambling, and trotting as shown in Fig. 4.2, the NLP is formulated as:

$$\begin{aligned} \min_{\alpha_v, x_i, \dot{x}_i, u_i} \quad & \sum_i \|u_i\|_2^2 \quad i = 1, 2, \dots \sum_{v=1}^N M_v \quad (4.7) \\ \text{s.t.} \quad & \mathbf{C1.} \text{ closed-loop dynamics} \\ & \mathbf{C2.} \text{ hybrid \& periodic continuity} \\ & \mathbf{C3.} \text{ physical feasibility} \end{aligned}$$

where M_v is the number of collocation points and α_v is the decision variable parameterizing the desired trajectory for domain \mathcal{D}_v . The cost function is to minimize the torque so that experiment implementation is achievable. The constraint **C1** is from (2.19) and (4.5), and the constraint **C2** is referring to the state continuity through each edge, which could be equipped with a discrete jump in states. **C3** enforces conditions including $\|u_i\|_\infty \leq 50$, $(q_i, \dot{q}_i) \in \mathcal{X}$, foot clearance and the friction pyramid conditions, so that the optimal solution is experimentally feasible.

Remark 3. Due to many types of aleatoric uncertainty in the model, not every solution of the NLP can be robust enough to lead to experimental success. Furthermore, some constraint setup can be ill-posed that the NLP converges poorly. Thus in experiments, “fine tuning” on the constraint setup are often needed. Using machine learning to automate the tuning process is presented in (). To present a complete and scientific report, the following heuristics were used:

1. Efficiently producing closed-loop controllers is the core innovation of FROST, but it also boosts the complexity of the problem. Thus using an open-loop setup (optimization without a feedback controller) to warm start a closed-loop problem is effective.
2. Tuning constraints on the acceleration (force) terms are normally more effective than tuning positions (“effective” is both referring to the converging speed and experimental robustness.) It is indeed intuitive to tune the constraints about the positional terms to change the appearance of a behavior, but it can often be too restrictive for the optimization to converge fast enough. A natural-looking gait should be the result of forces, not that of human-expertise refinement.

Optimal Gaits

Under the umbrella of the HZD framework (see Sec. 3.3), the sole difference among these behaviors in Fig. 4.2 is nothing but the ordered sequence of contact events,

which can be predefined by specifying the stance foot clearance as 0 and swing foot clearance as nonzero values. Therefore, in the HZD optimization (4.7), by changing the foot clearance constraints in **C3**, multiple quadrupedal behaviors such as *walk*, *amble*, and *trot* can be produced efficiently. We show the simulated behaviors in Fig. 4.3 and their phase portraits in Fig. 4.4. With some initial guesses supplied, the computation time for the presented gaits are 262.13 s for walking, 42.69 s for ambling, and 116.05 s for trotting on a Ubuntu 16.04 machine with Intel Core i7-6820 HQ CPU @ 2.7 GHz with 16 GB RAM. In a loose comparison, the reinforcement learning methods take hours to find a feasible solution (Tan et al., 2018).

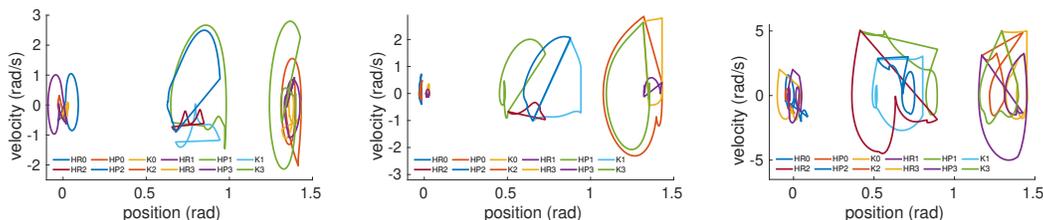


Figure 4.4: The periodic trajectories designed by NLP (4.7): walk (left), amble (middle), and trot (right).

Simulation and Experiment

With these stabilized quadrupedal dynamic gaits in simulation and optimization, we conduct an experiment with the ambling gait on Vision 60. The implemented controller is a PD approximation of the input-output linearizing controllers to track the time-based trajectories given by the optimization (4.7). That is, for a continuous domain \mathcal{D}_v , we have

$$u(q_a, \dot{q}_a, t) = -K_p(q_a - \mathcal{B}_v(t)) - K_d(\dot{q}_a - \dot{\mathcal{B}}_v(t)) \quad (4.8)$$

as the motor torque commands sent to each joint. This time-based PD implementation has been shown to have exceptional robustness for bipedal locomotion (S. Kolathaya, A. Hereid, and A. D. Ames, 2016). In addition, the domain switching method is also time-based with the event function given by the optimized trajectories.

The result is that Vision 60 ambles stably with the desired speed of 0.3 m/s. See (Wen-Loong Ma, 2019b) for Vision 60 ambling in an outdoor tennis court, and the snapshots are shown in Fig. 4.5. We logged 20 seconds of data and compared it with the simulated ambling controller in Fig. 4.6. Additionally, the average torque inputs (absolute value) are 7.73 N·m on the hip roll joints, 9.46 N·m on the hip pitch joints, and 16.17 N·m on the knee joints. It is worth mentioning that the consistent drifting

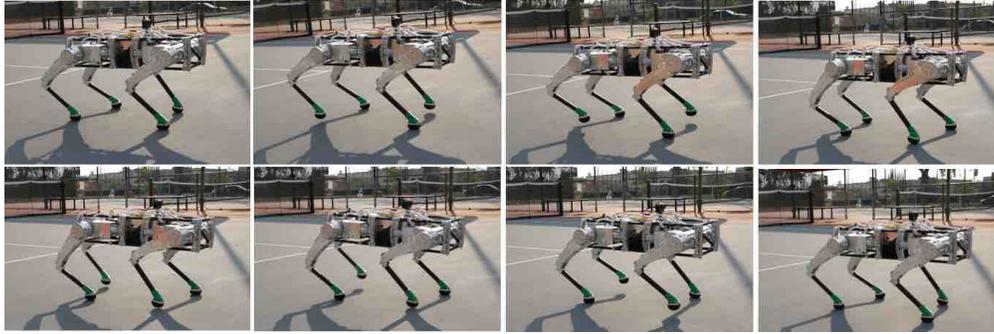


Figure 4.5: Snapshots of the Vision 60 ambling in an outdoor environment, showing a full step of 2×2 domains of the amble gait.

aside is expected, as there is no feedback information for the uneven terrain and that some manufacturing defects could cause asymmetric weight distribution. Some common solutions to avoid the drift is to use a joystick to manually offset the hip roll joints or *Raibert-type regulators* (see (Marc H. Raibert, Brown, and Chepponis, 1984)). The robot that keeps ambling without falling supports the feasibility of the full model-based HZD methods for quadrupedal locomotion.

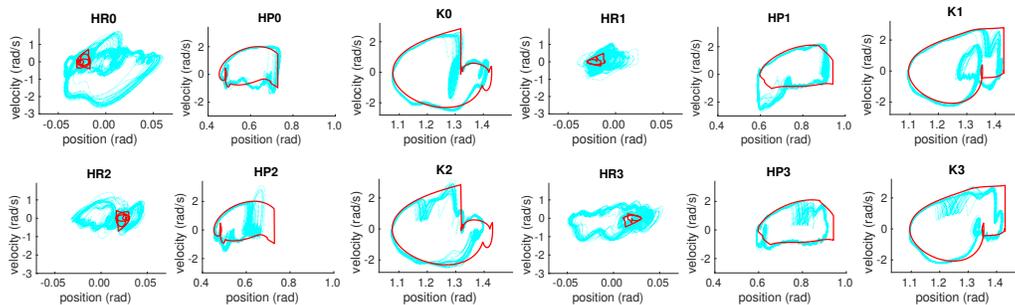


Figure 4.6: The designed gaits (in red) from optimization/simulation vs. the experimental data (in cyan) in the form of phase portrait for *amble*. HR is short for hip roll, HP is for hip pitch, and k is for knee.

Even though we can produce gaits and nominal controllers that lead to experimental success on hardware, the computational time around 2-4 minutes is not ideal for real-world implementation. The robustness against model uncertainty and rough terrain is also not practical compared to many simplified model-based approaches (Bledt et al., 2018; Boussema et al., 2019b). We next take the inspiration from Raibert's *virtual leg principle* (M. Raibert and Tello, 1986) and combine it with the HZD methodology. That is, the *coupled control system* (CCS) framework, where instead of empirically assuming a quadrupedal robot is equivalent to a low-DOF

bipedal robot, we consider it as a “collaboration” between two bipedal subsystems.

4.2 Coupled Control System and Its Solutions

To achieve dynamic walking on high-dimensional robotic systems, methods that assume simplified models have been applied, such as embedding the central pattern generators to multi-legged locomotion (Danner et al., 2017). Through another methodology, dimension reduction, hybrid zero dynamics (HZD) has proven to be an effective methodology. as a result of its ability to make theoretic guarantees (E. R. Westervelt, J. W. Grizzle, Chevallereau, et al., 2007) and yield walking for complex humanoids (Sreenath, 2011; Jacob Reher et al., 2016) without assuming model simplifications. The main idea behind this approach is that the full-order dynamics of the robot can be reduced to a lower-dimensional surface on which the system evolves. The system can then be studied via the low-dimensional dynamic representation and, importantly, guarantees made can be translated back to the full-order dynamics, i.e., periodic orbits (or walking gaits) in the low-dimensional system imply corresponding periodic orbits in the full-order system. The goal of this section is to capture this dimension reduction in a more general context—that of *coupled control systems*, which captures the ability to decompose a complex system into low-dimensional subsystems.

Another means of dimension reduction for robotic systems comes from isolating subsystems and coupling these subsystems at the level of reaction forces, i.e., Lagrange multipliers that enforce holonomic constraints. This is the idea underlying the highly efficient method for calculating the dynamics of robotic systems: spatial vector algebra (R. Featherstone, 2008). For example, a double pendulum can be decomposed into two single pendula connected via a holonomic constraint at the pivot joint (Ganesh, Aaron D Ames, and Bajcsy, 2007). More generally, for a robotic system, one can consider two equivalent ways of expressing the dynamics of this system (R. M. Murray et al., 1994; R. Featherstone, 2008):

$$\underbrace{D(q)\ddot{q} + H(q, \dot{q}) = u}_{\text{Full-Order Dynamics}} \Leftrightarrow \underbrace{\begin{cases} D_i(q_i)\ddot{q}_i + H_i(q_i, \dot{q}_i) = u_i + J_{h_i}^T \lambda \\ \text{s.t. } h(q) \equiv 0 \end{cases}}_{\text{Reduced-Order Coupled Dynamics}}$$

for $i = 1, 2$, where here h is a coupling (holonomic) constraint that is enforced via the Lagrange multiplier λ allowing for the higher-dimensional q to be decomposed into lower-dimensional components q_i , i.e., $q = (q_1, q_2)$. For example, a quadruped can be decomposed into two bipeds as in Fig. 4.7. Therefore, if one can make

guarantees on the reduced-order coupled systems, they can be translated into full-order dynamics. This idea has also, again, proven successful in robotic walking when dealing with what is referred to as the “pinned” and “unpinned” models—this success has been especially prevalent in the context of constructing efficient nonlinear constrained optimization problems, especially for bipedal walking robots that utilize hybrid zero dynamics (A. Hereid et al., 2018).

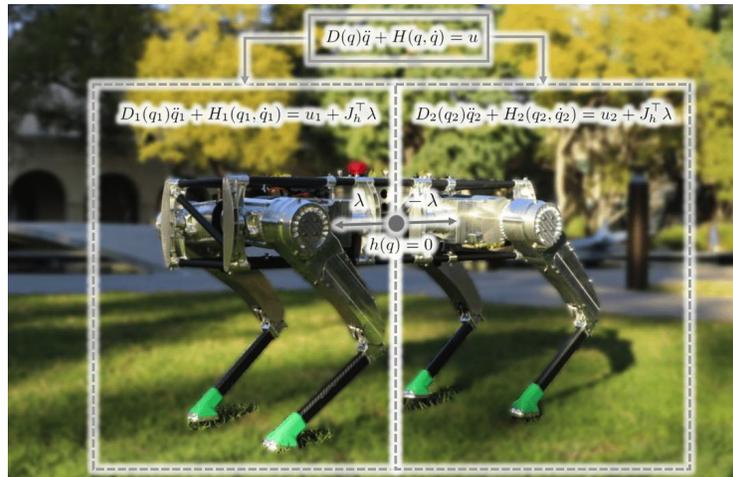


Figure 4.7: Conceptual illustration of the full-body dynamics decomposition, where the 3D quadruped — the Vision 60 — is decomposed into two constrained 3D bipedal robots.

The study of coupled dynamical and control systems has a long and rich history from which the method presented in this thesis has taken inspiration. First, from the computational perspective, the highly efficient method for calculating the dynamics of robotic systems — spatial vector algebra (R. Featherstone, 2008) — uses a similar concept: Lagrange multipliers that enforce holonomic constraints. Second, focusing on the coupled dynamics, the interconnected systems (Antonelli, 2013) have studied the synchronization of coupled oscillators (Dörfler and Bullo, 2014; Fujisaka and Yamada, 1983). Further, the passivity-based control (Hatanaka et al., 2015) has been proposed to design coupled controllers for multi-agent systems. The input-to-state stability analysis (Jiang, Mareels, and Wang, 1995) studied the Lyapunov stability of decoupled control laws. Third, in the control community, the most relevant examples are the multi-agent networks (Mesbahi and Egerstedt, 2010), the consensus problem (Ren, Beard, and Atkins, 2005) and the cooperative control problem (Mellinger et al., 2013; Chung and Slotine, 2009). These methods have been successfully demonstrated on a wide range of robotic applications, especially on drones. However, the problems considered in these frameworks are often coupled

on the control level — shared feedback information — but not the dynamics level, such as the general formulation considered in (Sloth, 2016). This allows the designer to utilize the built-in stabilizing controller of each subsystem to achieve some add-on optimality. In other words, each subsystem’s stability does not critically rely on the other subsystems’. In related work, the coordination of multiple quadrupedal robots via reaction forces has recently been studied (Kaveh Akbari Hamed, Kamidi, et al., 2020).

This section generalizes the aforementioned methods — zero dynamics and system decomposition through coupling constraints — and unifies them through a novel formulation: *coupled control systems*. We then utilize zero dynamics to reduce to a subsystem dependent on coupling constraints which are then eliminated via coupling relations to yield the final isolated subsystem. The main result of this paper is that solutions to the isolated subsystem are solutions to the full-order system. Thus periodic orbits on the subsystem yield periodic orbits on the full-order system. This result is leveraged to construct a nonlinear optimization problem utilizing collocation methods to generate these periodic solutions.

Coupled Control Systems. To define coupled control systems, we will consider a graph on which each node represents a control system, and each edge represents a coupling condition between these control systems. First, we briefly establish the notation related to graphs.

A **bidirected** graph (also termed a *symmetric directed graph*) is a tuple $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with $\mathbb{V} = \{1, 2, \dots, \ell\} \subset \mathbb{Z}$ a set of vertices (representing the indices of all subsystems) and $\mathcal{E} \subset \{(i, j) \mid (i, j) \in \mathcal{N} \times \mathbb{V}, i \neq j\}$ is a set of edges (these represent the connection relation between two subsystems, hence the bidirectionality—if a subsystem is connected to another subsystem, then the same holds in reverse). The bidirectionality of the graph implies that if $e = (i, j) \in \mathcal{E}$ then $\bar{e} = (j, i) \in \mathcal{E}$, i.e., every direct edge connects each vertex symmetrically. All of the edges of a sub-system i form a set, $\mathcal{G}_i = \{e \in \mathcal{E} \mid e = (i, j) \in \mathcal{E}\}$ with $|\mathcal{G}_i|$ the number of neighbor systems. By bidirectionality, if $e \in \mathcal{G}_i$ then $\bar{e} \in \mathcal{G}_j$.

We can now define the main object of interest:

Definition 14. A *coupled control system (CCS)* is defined to be a tuple:

$$\mathcal{C}_C = (\mathcal{G}, \mathcal{X}, \mathcal{Z}, \mathcal{U}, \Lambda, FG, PQ, C) \quad (4.9)$$

consisting of the following components:

- $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a *bidirected graph*,
- $\mathcal{X} = \{\mathcal{X}_i\}_{i \in \mathcal{N}}$, where $x_i \in \mathcal{X}_i$ is a set of *internal states*,
- $\mathcal{Z} = \{\mathcal{Z}_i\}_{i \in \mathcal{N}}$, where $z_i \in \mathcal{Z}_i$ is a set of *coupled states*,
- $\mathcal{U} = \{\mathcal{U}_i\}_{i \in \mathcal{N}}$, where $u_i \in \mathcal{U}_i$ is a set of *admissible inputs*,
- $\Lambda = \{\lambda_e\}_{e \in \mathcal{E}}$, is a set of *coupling inputs* that enforce dynamic coupling,
- $FG = \{(f_i, g_i, \check{g}_e)\}_{i \in \mathcal{N}, e \in \mathcal{G}_i}$ are the *internal state dynamics* where each systems' internal states evolve according to:

$$\dot{x}_i = f_i(x_i, z_i) + g_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{g}_e(x_i, z_i, z_j)\lambda_e, \quad (4.10)$$

- $PQ = \{(p_i, q_i, \check{q}_e)\}_{i \in \mathcal{N}, e \in \mathcal{G}_i}$ is the set of *coupled state dynamics* where the coupled states evolve according to:

$$\dot{z}_i = p_i(x_i, z_i) + q_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_j)\lambda_e, \quad (4.11)$$

- $C = \{c_e\}_{e \in \mathcal{E}}$ is a set of *coupling constraints*:²

$$c_e(z_i, z_j) \equiv 0, \quad \forall e = (i, j) \in \mathcal{E}. \quad (4.12)$$

Note that it is often convenient to write a CCS in a more compact form. More concretely, with the objects defined above in mind, we can denote a CCS in the following form of *semi-explicit* differential-algebraic equation (DAEs):

$$\mathcal{C}_C \triangleq \begin{cases} \dot{x}_i = f_i(x_i, z_i) + g_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{g}_e(x_i, z_i, z_j)\lambda_e \\ \dot{z}_i = p_i(x_i, z_i) + q_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_j)\lambda_e \\ \text{s.t. } c_e(z_i, z_j) \equiv 0 \quad \forall e = (i, j) \in \mathcal{G}_i \end{cases} \quad (4.13)$$

wherein the systems in (4.13) are defined for each $i \in \mathcal{N}$, and implicit in the definition for the corresponding bidirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and the fact that $x_i \in \mathcal{X}_i$, $z_i \in \mathcal{Z}_i$ and $u_i \in \mathcal{U}_i$ as defined above.

²As used in (2.8), the operator \equiv (“identical equality” of functions) is defined as, $f \equiv g$ for functions f, g meaning $f(x) = g(x)$ for all x defined. For example, $c_e(z_i, z_j) \equiv 0$ implies that $c_e(z_i(t), z_j(t)) = 0$ for all $t \in \mathbb{R}$ for which $z_i(t)$ and $z_j(t)$ are defined.

For a CCS \mathcal{C}_C we will denote the total internal state by $x = \{x_i\}_{i \in \mathcal{N}} \in \prod_{i \in \mathcal{N}} \mathcal{X}_i$ and the total coupled state by $z = \{z_i\}_{i \in \mathcal{N}} \in \prod_{i \in \mathcal{N}} \mathcal{Z}_i$. By slight abuse of notation, we may denote function evaluations on the total states such that the corresponding functions are evaluated system by system. For example, we can write $c(z) \triangleq \{c_e(z_i, z_j)\}_{e=(i,j) \in \mathcal{E}}$ for the coupling constraint.

Solutions. We can define solutions to coupled control systems by assuming the existence of a collection of feedback control laws: $u(x, z) \triangleq \{u_i(x_i, z)\}_{i \in \mathcal{N}}$. Applying these controllers to (4.13) yields a **coupled dynamical system (CDS)**:

$$\mathcal{D}_C \triangleq \begin{cases} \dot{x}_i = \underbrace{f_i(x_i, z_i) + g_i(x_i, z_i)u_i(x_i, z)}_{f_i^{\text{cl}}(x_i, z)} + \sum_{e \in \mathcal{G}_i} \check{g}_e(x_i, z_i, z_j) \lambda_e \\ \dot{z}_i = \underbrace{p_i(x_i, z_i) + q_i(x_i, z_i)u_i(x_i, z)}_{p_i^{\text{cl}}(x_i, z)} + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_j) \lambda_e \\ \text{s.t. } c_e(z_i, z_j) \equiv 0 \quad \forall e = (i, j) \in \mathcal{G}_i \end{cases} \quad (4.14)$$

again defined for all $i \in \mathcal{N}$. The **solution** to the coupled dynamic system, \mathcal{D}_C , is a set of solutions:

$$\{(x_i(t), z_i(t), \lambda_e(t))\}_{i \in \mathcal{N}, e \in \mathcal{G}_i} \quad \text{s.t.} \quad (4.14) \quad \forall t \in \mathbf{I} \subset \mathbb{R}$$

with initial condition: $\{(x_i(0), z_i(0), \lambda_e(0))\}_{i \in \mathcal{N}, e \in \mathcal{G}_i}$. Per the above notation, we will sometimes denote the solutions by $(x(t), z(t), \lambda(t))$ with initial condition $(x(0), z(0), \lambda(0))$. Note that solutions must satisfy the coupling constraints (4.12) at all time. Therefore,

$$\begin{aligned} \dot{c}(x, z) \equiv 0 &\Rightarrow \dot{c}_e(x, z_i, z_j, \dot{z}_i, \dot{z}_j) \equiv 0 \quad \forall e \in \mathcal{E} \\ &\Rightarrow \dot{c}_e(x, z, \dot{z}) = \underbrace{\frac{\partial c_e(z_i, z_j)}{\partial z_i}}_{\triangleq J_e^{(i,j)}(z)} \dot{z}_i + \underbrace{\frac{\partial c_e(z_i, z_j)}{\partial z_j}}_{\triangleq J_e^{(j,i)}(z)} \dot{z}_j \equiv 0. \end{aligned} \quad (4.15)$$

Since $\dot{z}_i(t)$ and $\dot{z}_j(t)$ both depend on $\lambda_e(t)$, for $e = (i, j)$, it is this derivative condition that $\lambda(t)$ must satisfy for all systems and all coupling constraints in order to be a solution. Concretely, and utilizing (4.15), this implies that the following condition must hold:

$$\begin{aligned} \dot{c}_e(x, z) &= J_e^{(i,j)}(z) \left(p_i^{\text{cl}}(x_i, z) + \sum_{e'=(i,k) \in \mathcal{G}_i} \check{q}_{e'}(x_i, z_i, z_k) \lambda_{e'} \right) \\ &+ J_e^{(j,i)}(z) \left(p_j^{\text{cl}}(x_j, z) + \sum_{e'=(j,l) \in \mathcal{G}_j} \check{q}_{e'}(x_j, z_j, z_l) \lambda_{e'} \right) \equiv 0. \end{aligned} \quad (4.16)$$

Therefore, to solve for coupling inputs λ_e that satisfy the coupling constraints, it is necessary to solve an equation that depends on the states of all systems connected to the i^{th} system via \mathcal{G}_i . To address this, we present a method for isolating a subsystem via conditions on the controllers of all other systems.

Isolating Control Subsystems

The main idea in approaching the analysis and design of controllers for CCSs is to isolate subsystems that encode the behavior of the overall CCS. This section outlines the procedure for isolating these subsystems through a two-step approach: restricting systems to the zero dynamics surface and leveraging this to calculate the coupling conditions explicitly. These can be combined to reduce the full-order CCS to a subsystem that no longer depends on the internal states of the other subsystems. We establish the main result of the paper encapsulating these constructions: solutions of the subsystem yield solutions of the full-order dynamics.

Zero Dynamics Manifolds. In addition to restricting the dynamics to the constraint manifold, we are also interested in understanding on controllers that restrict the system to invariant surfaces can be utilized to further isolate subsystems of the CCS in a way amenable to analysis.

Consider a CCS \mathcal{C}_C . The *zero dynamics manifold* is given for each subsystem, i.e., for each $i \in \mathcal{N}$, by:

$$\mathbf{Z}_i \triangleq \{(x, z) \in \mathcal{X} \times \mathcal{Z} \mid x_i \equiv 0\}. \quad (4.17)$$

where, by abuse of notation (although clear from context), $\mathcal{X} \triangleq \prod_{i \in \mathcal{N}} \mathcal{X}_i$ and $\mathcal{Z} \triangleq \prod_{i \in \mathcal{N}} \mathcal{Z}_i$. Thus, the zero dynamics manifold consists of internal states that are zero, i.e., the system evolves only according to the coupled states.

Definition 15. Consider a coupled control system \mathcal{C}_C . For $i \in \mathcal{N}$, a controller for the subsystem, $u_i^{\mathbf{Z}, \lambda}(x_i, z)$, *renders the zero dynamics manifold invariant* if it satisfies the following algebraic condition:

$$0 \equiv f_i(0, z_i) + g_i(0, z_i)u_i^{\mathbf{Z}, \lambda}(0, z) + \sum_{e \in \mathcal{G}_i} \check{g}_e(0, z_i, z_j)\lambda_e \quad (4.18)$$

where $u_i^{\mathbf{Z}, \lambda}$ implicitly depends on λ_e for $e \in \mathcal{G}_i$. We call $u_i^{\mathbf{Z}, \lambda}$ the nominal controller for the i^{th} subsystem.

Zero Dynamics. Consider the controller $u_i^{\mathbf{Z}}$ that renders the zero dynamics manifold \mathbf{Z}_i invariant. By applying this controller, the coupled dynamics evolve according to

the *zero dynamics*:

$$\begin{aligned} \dot{z}_i &= p_i(0, z) + q_i(0, z)u_i^{\mathbf{Z},\lambda}(0, z) + \sum_{e \in \mathcal{G}_i} \check{q}_e(0, z_i, z_j)\lambda_e \\ &\triangleq p_{i,e}^{\mathbf{Z},\lambda}(z). \end{aligned} \quad (4.19)$$

Note that $p_{i,e}^{\mathbf{Z},\lambda}$ depends on λ_e for $e \in \mathcal{G}_i$.

In many cases, one can explicitly solve for a controller that satisfies this condition (via a relative degree type condition)—often through the proper choice of internal states. This will be the case for mechanical systems, and especially those representing quadrupedal robots. Specifically, if for all $i \in \mathcal{N}$, the matrix $g_i(0, z)$ is invertible for all $z \in \mathcal{Z}$ then we can explicitly solve for the condition on $u_i^{\mathbf{Z}}$ that renders the zero dynamics invariant:

$$u_i^{\mathbf{Z},\lambda}(0, z) = -g_i(0, z_i)^{-1} \left(f_i(0, z_i) + \sum_{e \in \mathcal{G}_i} \check{g}_e(0, z_i, z_j)\lambda_e \right) \quad (4.20)$$

which now explicitly depends on λ_e for $e \in \mathcal{G}_i$.

λ -Coupled Subsystems. The key idea underlying the analysis of CCSs is to reduce the entire coupled system into the behavior of a single subsystem. This is achieved through the above constructions related to the zero dynamics.

For a CCS \mathcal{C}_C , assume that there exist $u_j^{\mathbf{Z}}$ that render the zero dynamics manifold \mathbf{Z}_j invariant for all $j \in \mathcal{N} \setminus \{i\}$ with $i \in \mathcal{N}$. Consider the following *λ -coupled control subsystem* (*λ -CCSub*):

$$\mathcal{C}_i^{\mathbf{Z},\lambda} \triangleq \begin{cases} \dot{x}_i = f_i(x_i, z_i) + g_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{g}_e(x_i, z_i, z_j)\lambda_e \\ \dot{z}_i = p_i(x_i, z_i) + q_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_j)\lambda_e \\ \dot{z}_j = p_{j,e}^{\mathbf{Z},\lambda}(z) \quad \forall j \in \mathcal{N} \setminus \{i\} \text{ s.t. } c_e(z_i, z_j) = 0 \quad \forall e \in \mathcal{E} \end{cases}, \quad (4.21)$$

Thus, the subsystem evolves according to the dynamics of the system i and the zero dynamics of all remaining systems—all of which are coupled via the coupling inputs $\lambda = \{\lambda_e\}_{e \in \mathcal{E}}$.

When a feedback controller $u_i(x_i, z)$ is applied to $\mathcal{C}_i^{\mathbf{Z},\lambda}$, the result is a dynamical subsystem, denoted by: $\mathcal{D}_i^{\mathbf{Z},\lambda}$. Solutions of this system are denoted by $(x_i(t), z(t), \lambda(t))$. The fact that the i^{th} subsystem yields behavior of the entire system is summarized in the following lemma. Before stating this result, we need some notation. Let

$(x_i, z) \in \mathcal{X}_i \times \mathcal{Z}$, and consider the canonical embedding $\iota : \mathcal{X}_i \times \mathcal{Z} \hookrightarrow \mathcal{X} \times \mathcal{Z}$ given by $\iota(x_i, z) = (x, z)$ where $x = \{x_j\}_{j \in \mathcal{N}}$ with $x_j = 0$ if $j \in \mathcal{N} \setminus \{i\}$ and $x_j = x_i$ if $j = i$.

Lemma 4. Let \mathcal{C}_C be a CCS and $\mathcal{C}_i^{\mathcal{Z}, \lambda}$ the corresponding λ -CCSub for some $i \in \mathcal{N}$ obtained by applying $u_j^{\mathcal{Z}, \lambda}$, for $j \in \mathcal{N} \setminus \{i\}$, that render the zero dynamics manifold \mathcal{Z}_j invariant. Let $u_i(x_i, z)$ be a feedback controller applied to $\mathcal{C}_i^{\mathcal{Z}, \lambda}$ resulting in the corresponding dynamical subsystem $\mathcal{D}_i^{\mathcal{Z}, \lambda}$ with solution $(x_i(t), z(t), \lambda(t))$. Then $(\iota(x_i(t), z(t)), \lambda(t))$ is a solution to \mathcal{D}_C —the CDS obtained by applying these controllers.

Proof. The proof follows trivially from the fact that the zero dynamics are invariant for all $j \in \mathcal{N} \setminus \{i\}$, i.e.,

$$\iota(x_i(t), z(t)) \in \bigcap_{j \in \mathcal{N} \setminus \{i\}} \mathcal{Z}_j \quad \forall t \in I \subset \mathbb{R}$$

for I the interval of existence of $(x_i(t), z(t))$. □

Explicit Coupling Conditions

The coupling between the control systems (4.13) is enforced via λ_e and the coupling constraints of the form (4.16). Similarly, even in the reduction to a subsystem (4.21), the coupling was still achieved through λ_e . We wish to generalize this to remove the coupling, i.e., isolate subsystems, while still preserving the overall behavior of the full system.

Coupling Relations. Consider (4.15) but now for the case of a λ -CCSub, as in (4.21), yielding:

$$\begin{aligned} \dot{c}_e^{\mathcal{Z}, \lambda}(x_i, z, u_i) &= J_e^{(i, j)}(z) \left(p_i(x_i, z_i) + q_i(x_i, z_i)u_i + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_k)\lambda_e \right) + \\ &J_e^{(j, i)}(z) \left(p_j(0, z_j) + q_j(0, z_j)u_j^{\mathcal{Z}, \lambda}(0, z) + \sum_{e' \in \mathcal{G}_j} \check{q}_{e'}(0, z_j, z_l)\lambda_{e'} \right) \\ &\equiv 0 \end{aligned} \tag{4.22}$$

Thus λ_e with $e = (i, j) \in \mathcal{G}_i$, implicitly depends on u_i —while the use of the controllers $u_j^{\mathcal{Z}, \lambda}$ eliminates the dependence on the controllers and internal states of the other systems. This motivates the following formulation:

Definition 16. For a λ -coupled control subsystem $C_i^{\mathbf{Z},\lambda}$ ($i \in \mathcal{N}$), a *coupling relation* is a functional relationship on the coupling inputs:

$$\lambda_e^{\mathbf{Z}}(x_i, z; u_i) = A_e^{\mathbf{Z}}(x_i, z)u_i + b_e^{\mathbf{Z}}(x_i, z), \quad (4.23)$$

for all $e = (i, j) \in \mathcal{E}$, that satisfies the coupling constraint (4.22).

Note that λ_e only depends on u_i and x_i for all $e \in \mathcal{E}$. Specifically, given another edge $e' = (j, l) \in \mathcal{E}$ we have $\lambda_{e'}^{\mathbf{Z}}(x_i, z; u_i) = A_{e'}^{\mathbf{Z}}(x_i, z)u_i + b_{e'}^{\mathbf{Z}}(x_i, z)$. In this case, it may happen that $A_{e'}^{\mathbf{Z}} = 0$, meaning that the relation is simply a constant relation for that edge.

Leveraging Coupling Relations. Given a coupling relation $\lambda^{\mathbf{Z}}$ we can remove the dependence on λ throughout $C_i^{\mathbf{Z},\lambda}$ and leave only a dependence on u_i .

Zero dynamics controllers: Recall that the controller $u^{\mathbf{Z},\lambda}$ that renders the zero dynamics surface invariant implicitly depends on λ_e via (4.18). Given a coupling relation, the dependence of λ_e is removed, and as a result we say that $u^{\mathbf{Z}}$ renders the zero dynamics manifold \mathbf{Z}_j invariant if:

$$0 \equiv f_j^{\mathbf{Z}}(0, z) + g_j^{\mathbf{Z}}(0, z)u_i + g_j(0, z_j) \left(u_j^{\mathbf{Z}}(0, z; u_i) - u_i \right) \quad (4.24)$$

where $u_j^{\mathbf{Z}}$ is now a function of u_i and

$$f_j^{\mathbf{Z}}(x_j, z) \triangleq f_j(x_j, z_j) + \sum_{e' \in \mathcal{G}_j} \check{g}_{e'}(x_j, z_j, z_l) b_{e'}^{\mathbf{Z}}(x_i, z), \quad (4.25)$$

$$g_j^{\mathbf{Z}}(x_j, z) \triangleq g_j(x_j, z_j) + \sum_{e' \in \mathcal{G}_j} \check{g}_{e'}(x_j, z_j, z_l) A_{e'}^{\mathbf{Z}}(x_i, z). \quad (4.26)$$

Coupling conditions: Returning to (4.22), given a coupling relation we can rewrite this coupling constraint as:

$$\begin{aligned} \dot{c}_e(x_i, z, \dot{z}) &= J_e^{(i,j)}(z) \left(p_i^{\mathbf{Z}}(x_i, z) + q_i^{\mathbf{Z}}(x_i, z)u_i \right) \\ &\quad + J_e^{(j,i)}(z) \left(p_j^{\mathbf{Z}}(x_i, z) + q_j^{\mathbf{Z}}(x_i, z)u_i \right) \equiv 0 \end{aligned} \quad (4.27)$$

where for $i \in \mathcal{N}$ the i associated with the subsystem $C_i^{\mathbf{Z},\lambda}$:

$$p_i^{\mathbf{Z}}(x_i, z) \triangleq p_i(x_i, z_i) + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_j) b_e^{\mathbf{Z}}(x_i, z) \quad (4.28)$$

$$q_i^{\mathbf{Z}}(x_i, z) \triangleq q_i(x_i, z_i) + \sum_{e \in \mathcal{G}_i} \check{q}_e(x_i, z_i, z_j) A_e^{\mathbf{Z}}(x_i, z) \quad (4.29)$$

and for $j \in \mathcal{N} \setminus \{i\}$, we have:

$$p_j^{\mathbf{Z}}(x_i, z) \triangleq p_j(0, z_j) + q_j(0, z_j)u_j^{\mathbf{Z}}(0, z) + \sum_{e' \in \mathcal{G}_j} \check{q}_{e'}(0, z_j, z_l)b_{e'}^{\mathbf{Z}}(x_i, z) \quad (4.30)$$

$$q_j^{\mathbf{Z}}(x_i, z) \triangleq \sum_{e' \in \mathcal{G}_j} \check{q}_{e'}(0, z_j, z_l)A_{e'}^{\mathbf{Z}}(x_i, z). \quad (4.31)$$

Isolating Subsystems

We now arrive at the key concept for which all of the previous constructions have built: reducing a CCS to a single subsystem that can be used to give guarantees about the entire CCS. This is achieved using the above coupling relation.

Definition 17. For a CCS C_C , for $i \in \mathcal{N}$, assume a coupling relation $\lambda^{\mathbf{Z}}$ that there exist $u_j^{\mathbf{Z}}$ that render the zero dynamics manifold \mathbf{Z}_j invariant for $j \in \mathcal{N} \setminus \{i\}$. Then the i^{th} *control subsystem (CSub)* associated with the CCS C_C is given by:

$$C_i^{\mathbf{Z}} \triangleq \begin{cases} \dot{x}_i = f_i^{\mathbf{Z}}(x_i, z) + g_i^{\mathbf{Z}}(x_i, z)u_i \\ \dot{z}_i = p_i^{\mathbf{Z}}(x_i, z) + q_i^{\mathbf{Z}}(x_i, z)u_i \\ \dot{z}_j = p_j^{\mathbf{Z}}(x_i, z) + q_j^{\mathbf{Z}}(x_i, z)u_i \quad \forall j \in \mathcal{N} \setminus \{i\} \end{cases} \quad (4.32)$$

where $f_i^{\mathbf{Z}}$ and $g_i^{\mathbf{Z}}$ are given in (4.25) and (4.26) and $p_i^{\mathbf{Z}}$, $q_i^{\mathbf{Z}}$, $p_j^{\mathbf{Z}}$, and $q_j^{\mathbf{Z}}$ are given in (4.28)—(4.31). For the sake of definiteness, we note that:

$$f_i^{\mathbf{Z}}(x_i, z) \triangleq f_i(x_i, z_i) + \sum_{e \in \mathcal{G}_i} \check{g}_e(x_i, z_i, z_j)b_e^{\mathbf{Z}}(x_i, z), \quad (4.33)$$

$$g_i^{\mathbf{Z}}(x_i, z) \triangleq g_i(x_i, z_i) + \sum_{e \in \mathcal{G}_i} \check{g}_e(x_i, z_i, z_j)A_e^{\mathbf{Z}}(x_i, z). \quad (4.34)$$

Given a feedback controller $u_i(x_i, z)$ the corresponding *dynamical subsystem* is denoted by $\mathcal{D}_i^{\mathbf{Z}}$.

Constrained Dynamics. Note that the coupling constraint (4.27) was not explicitly stated in the CSub $C_i^{\mathbf{Z}}$. This was because it was solved for via the coupling relation $\lambda^{\mathbf{Z}}$. That is, the system naturally evolves on the *constraint manifold*:

$$\mathbf{C} \triangleq \{(x, z) \in \mathcal{X} \times \mathcal{Z} : c_e(z) \equiv 0, \quad \forall e \in \mathcal{E}\}. \quad (4.35)$$

This is made formal in the following result which is the ultimate result of all the constructions presented in this paper. Additionally, it will be seen that solutions of the i^{th} subsystem can be used to construct solutions of the full-order CCS. In this context, recall that in Lemma 4 we utilized the embedded $\iota : \mathcal{X}_i \times \mathcal{Z} \rightarrow \mathcal{X} \times \mathcal{Z}$.

Theorem 2. Let \mathcal{C}_C be a CCS, and for $i \in \mathcal{N}$ assume that there exist $u_j^Z, j \in \mathcal{N} \setminus \{i\}$ that render the zero dynamics manifold \mathbf{Z}_j invariant. Let \mathcal{C}_i^Z be the corresponding i^{th} subsystem. Given a feedback controller $u_i(x_i, z)$ for the CSub with corresponding dynamical subsystem \mathcal{D}_i^Z with solution $(x_i(t), z(t))$ with $t \in I \subset \mathbb{R}$. If

$$\iota(x_i(0), z(0)) \in \mathbf{C} \quad \Rightarrow \quad \iota(x_i(t), z(t)) \in \mathbf{C} \quad \forall t \in I \subset \mathbb{R}$$

and $(\iota(x_i(t), z(t)), \lambda^Z(t))$, with

$$\lambda^Z(t) = \{\lambda_e^Z(x_i(t), z(t); u_i(x_i(t), z(t)))\}_{e \in \mathcal{E}},$$

is a solution to \mathcal{D}_C , the CDS obtained by applying u_i and $u_j^Z, j \in \mathcal{N} \setminus \{i\}$.

Proof. The condition that $(x(0), z(0)) \in \mathbf{C}$ is equivalent to $c_e(z(0)) = 0$. Concretely, $c_e(z_i(0), z_j(0)) = 0$. Since λ_e^Z is a coupling relation it satisfies (4.22) and more explicitly (4.27); therefore, and being explicit about the arguments, $\dot{c}_e(x(t), z(t)) = 0$ for all $t \in I$ and all $e \in \mathcal{E}$. It follows that $c_e(z(t)) = 0$ for all $t \in I$ and $e \in \mathcal{E}$.

The fact that $(\iota(x_i(t), z(t)), \lambda^Z(t))$ is a solution to \mathcal{D}_C assuming that $(x_i(t), z(t))$ is a solution to \mathcal{D}_i^Z follows Lemma 4 together with the construction of \mathcal{D}_i^Z . \square

Periodic Orbits. In the context of quadrupeds, we will be interested in generating periodic motions, i.e., walking. A solution to a CDS \mathcal{D}_C is **periodic** of period $T > 0$ if for some initial condition $(x(0), z(0), \lambda(0))$:

$$(x(t+T), z(t+T), \lambda(t+T)) = (x(t), z(t), \lambda(t))$$

with the resulting periodic orbit:

$$\mathcal{O} = \{(x(t), z(t)) \in \mathcal{X} \times \mathcal{Z} \mid 0 \leq t \leq T\}.$$

As a result of Theorem 2, periodic orbits in a subsystem correspond to periodic orbits in the full-order dynamics.

Corollary 1. Under the conditions of Theorem 2, assume that $(x_i(t), z(t))$ is a periodic solution of \mathcal{D}_i^Z with period $T > 0$ and corresponding orbit $\mathcal{O}_i = \{(x_i(t), z(t)) \in \mathcal{X}_i \times \mathcal{Z} \mid 0 \leq t \leq T\}$ then $(\iota(x(t), z(t)), \lambda^Z(t))$ is a periodic solution of the CDS with period $T > 0$ and corresponding periodic orbit $\mathcal{O} = \iota(\mathcal{O}_i)$.

4.3 CCS Optimization and Application to Quadrupedal Gaits

With the construction of coupled control systems in the graph in the previous section, we present a general optimization framework to solve for the solution to the i^{th} *control subsystem (CSub)* associated with the CCS C_C , while synthesising the corresponding nominal controllers that render forward invariance of the zero dynamics manifolds. The application of these ideas to periodic orbit generation will be discussed. The approach we will take is a *locally direct collocation* (Rao, 2009) based optimization method, which has been detailed in Sec. 2.9 and Sec. 3.3. In this section, we build on this method to solve for the solution and controllers to coupled control systems, while optimizing some metrics. The target is that through the proposed control decoupling framework, the computation of numerical solutions (including control synthesis) to a coupled control system can be achieved efficiently and robustly.

Coupled System Optimization

We first pose the previous formulations as a series of equality constraints to represent the controlled dynamics of C_i^Z for system $i \in \mathbb{V}$. In particular, assume a coupling relation as given in Def. 16 of the form: $\lambda_e^Z(x_i, z; u_i) = A_e^Z(x_i, z)u_i + b_e^Z(x_i, z)$, as in (4.23).

To define the decision variables, we first discretized time evenly $t \in [0, T]$, with grid index $\kappa = 0, 1, \dots, K$, i.e., $t^\kappa = \frac{\kappa}{K}T$. Consequently, we have the initial time $t^0 = 0$, final time $t^M = T$ and step size $\Delta t = T/K$. To formulate direct collocation constraints, we referred the even nodes (grid index that is an even number) as cardinal nodes and the other grids as interior nodes. In addition, we abbreviated the dependency on time t as $\square^\kappa \triangleq \square(t^\kappa)$ for notation simplicity.

Decision Variables. To introduce the other constructions, *decision variable* for optimization is denoted as

$$\mathbf{X} \triangleq \{\vartheta^\kappa\}_{\kappa=0,1,\dots,M}, \text{ with } \vartheta^\kappa \triangleq \{x_i^\kappa, \dot{x}_i^\kappa, z_i^\kappa, \dot{z}_i^\kappa, z_j^\kappa, \dot{z}_j^\kappa, u_i^\kappa, u_j^{\mathbf{Z},\kappa}\}_{j \in \mathcal{N} \setminus \{i\}}.$$

Thus we have decision variables associated with the i^{th} control subsystem C_i^Z . Note that we will use $z^\kappa = \{z_i^\kappa, z_j^\kappa\}_{j \in \mathcal{N} \setminus \{i\}}$ to denote all of the coupled coordinates.

Zero Dynamics Constraints. Recall that given a coupling relation, we had associated zero dynamics invariance conditions given by (4.58). We will enforce these conditions in the optimization to ensure that $u_j^{\mathbf{Z},\kappa}$ renders the zero dynamics \mathbf{Z}_j invariant.

Specifically, we have the constraint:

$$F_{\text{zero}}(\vartheta^k) \triangleq f_j^{\mathbf{Z}}(0, z^k) + g_j^{\mathbf{Z}}(0, z^k)u_i^k + g_j(0, z_j^k) \left(u_j^{\mathbf{Z},k} - u_i^k \right), \quad \forall j \in \mathcal{N} \setminus \{i\} \quad (\text{C.1})$$

where $f_j^{\mathbf{Z}}$ and $g_j^{\mathbf{Z}}$ are given as in (4.25) and (4.26), respectively.

Subsystem Dynamics Constraints. Following from the constructions in Sec. 4.2, we wish to define constraints corresponding to dynamics of the i^{th} control subsystem $C_i^{\mathbf{Z}}$ (as obtained from the coupling relation). By denoting $\chi^k = (x_i^k, z_i^k, z_j^k)$, and

$$F(\chi^k) \triangleq \begin{cases} f_i^{\mathbf{Z}}(x_i^k, z^k) + g_i^{\mathbf{Z}}(x_i^k, z^k)u_i^k \\ p_i^{\mathbf{Z}}(x_i^k, z^k) + q_i^{\mathbf{Z}}(x_i^k, z^k)u_i^k \\ p_j^{\mathbf{Z}}(x_i^k, z^k) + q_j^{\mathbf{Z}}(x_i, z)u_i^k \quad \forall j \in \mathcal{N} \setminus \{i\} \end{cases} ;$$

we hence obtain the dynamic constraints as:

$$F_{\text{dyn}}(\vartheta^k) \triangleq \dot{\chi}^k - F(\chi^k) = 0. \quad (\text{C.2})$$

Here these dynamics are given as in Def. 17, i.e., $f_i^{\mathbf{Z}}$ and $g_i^{\mathbf{Z}}$ are given in (4.25) and (4.26) and $p_i^{\mathbf{Z}}$, $q_i^{\mathbf{Z}}$, $p_j^{\mathbf{Z}}$, and $q_j^{\mathbf{Z}}$ are given in (4.28)-(4.31). The end result is an equality constraint imposed on the cardinal nodes to enforce all of the states and controllers satisfy the dynamic system (17).

Direct Collocation. To enforce that those solutions on cardinal nodes stay on the same vector flow, i.e., belong to one unique solution to the dynamic system (17), we employ an implicit stage-3 Runge-Kutta method for formulating this condition as equality constraints. We first use a piecewise Hermite interpolation (cubic) polynomial to compute the interpolated value of χ_c and its slope $\dot{\chi}_c$ at the center of the subinterval $[t^k, t^{k+1}]$, i.e., the *collocation point*, as:

$$\begin{aligned} \bar{\chi}_c &= (\chi^k + \chi^{k+1})/2 + \delta t(\dot{\chi}^k - \dot{\chi}^{k+1})/8 \\ \dot{\bar{\chi}}_c &= -3(\dot{\chi}^k - \dot{\chi}^{k+1})/2\delta t - (\dot{\chi}^k + \dot{\chi}^{k+1})/4. \end{aligned}$$

Then the *collocation* constraints is formed as:

$$d(\mathbf{X}) \triangleq \dot{\bar{\chi}}_c - F(\bar{\chi}_c) = 0. \quad (\text{C.3})$$

Physical Constraints. Additionally, the *path constraints*: $p(\cdot) \geq 0$, which are a set of nonequality constraints, are always used to enforce conditions along the time horizon. Examples include obstacle avoidance constraints for a robot arm

manipulation task, or some feasibility conditions on the solution to the dynamic equation, for which some real-world physics is represented. A set of equality constraints $b(\cdot) = 0$ that are imposed on the decision variables at $t = 0, T$ are often utilized to connect the initial and final condition:

$$b(\vartheta^\kappa) \triangleq \begin{bmatrix} \chi^0 - \chi^K \\ \dot{\chi}^0 - \dot{\chi}^K \end{bmatrix} = 0 \quad (4.36)$$

to force the optimization to only search for a solution to the dynamic system that is a periodic orbit.

Optimization problem. To find the solution to dynamical system in (4.32), we parse the decoupled control problem of the isolating subsystem $i \in \mathbb{V}$ as a mathematical programming:

$$\underset{\mathbf{X}}{\operatorname{argmin}} J(\mathbf{X}) \quad (\text{CCS-OPT})$$

$$\text{s.t. } F_{\text{zero}}(\vartheta^\kappa) = 0 \quad \kappa = 0, 1, \dots, K \quad (\text{C.1})$$

$$F_{\text{dyn}}(\vartheta^\kappa) = 0 \quad \kappa = 0, 2, 4, \dots \quad (\text{C.2})$$

$$d(\mathbf{X}) = 0 \quad (\text{C.3})$$

$$\vartheta^\kappa \in \mathcal{X} \times \mathcal{Z} \times \mathcal{U} \quad \kappa = 0, 1, \dots, K \quad (\text{C.4})$$

$$p(\vartheta^\kappa) \geq 0 \quad \kappa = 0, 1, \dots, K \quad (\text{C.5})$$

$$b(\vartheta^\kappa) = 0 \quad \kappa = 0, 1, \dots, K \quad (\text{C.6})$$

where $J(\cdot) \in \mathbb{R}$ is the cost function, often picked as the total energy consumption. Here (C.4) defines the upper and lower bounds of the decision variables, i.e., that they live in the admissible space of values. The other constraints are as stated above.

Solutions. As a result, the optimization (CCS-OPT) can simultaneously output trajectories (solutions) of the states $\{x_i(t), z(t)\}$, $u_j^{\mathbf{Z}}(t)$ that renders the zero dynamics manifold \mathbf{Z}_j invariant and the open-loop controller $u_i(t)$, with $t = [0, T]$ for which these solutions are defined. Note that one can also enforce the dynamics $\dot{x}_i^\kappa + \varepsilon x_i^\kappa = 0$ with $\varepsilon > 0$ to guarantee the converging attribute of the isolating subsystem i , in which case the controller $u_i(x_i, z)$ is equivalently an input-output feedback linearization controller. Per Theorem 2, given $u_j^{\mathbf{Z}}$ that renders invariant \mathbf{Z}_j and the feedback controller $u_i(x_i, z)$, we can compute $\lambda^{\mathbf{Z}}(t)$ using (4.23), hence $(\iota(x_i(t), z(t)), \lambda^{\mathbf{Z}}(t))$ is then a solution to the original CDS.

If we further impose a periodic condition on the solution's boundary such as (4.36), the optimization shall produce a periodic solution of period T to the CCS. Therefore,

according to Corollary 1, $(\iota(x_i(t)), z(t), \lambda^{\mathbf{Z}}(t))$ is a periodic solution to the CDS with period T .

Quadrupedal Locomotion as Symmetric Coupled Systems

A special case of CCSs that is of particular interest to our study of quadrupedal robots is CCSs that are *symmetric*, i.e., Symmetric Coupled Control Systems. In this case, we consider a CCS with only two systems that have identical (symmetric) coupling constraints—hence the use of the term “symmetric.” We will demonstrate this with an example consisting of a quadrupedal robot as two connected bipedal systems.

Definition 18. A *symmetric coupled control system (SCCS)* is given by a graph

$$\mathcal{G} = (\mathcal{N} = \{1, 2\}, \mathcal{E} = \{e = (1, 2), \bar{e} = (2, 1)\}), \quad (4.37)$$

together with a tuple represented by the following conditional expression:

$${}^{\dagger}\mathcal{C} \triangleq \begin{cases} \dot{x}_i = f_i(x_i, z_i) + g_i(x_i, z_i)u_i + \check{g}_e(x_i, z_i, z_j)\lambda_e \\ \dot{z}_i = p_i(x_i, z_i) + q_i(x_i, z_i)u_i + \check{q}_e(x_i, z_i, z_j)\lambda_e \\ \text{s.t. } c_e(z_i, z_j) = z_i - z_j \equiv 0 \\ \lambda_e = -\lambda_{\bar{e}} \end{cases} \quad (4.38)$$

for $i, j = 1, 2$ with $i \neq j$. Note that $x_i \in \mathcal{X}_i$ and $z_i \in \mathcal{Z}_i$ and we let $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ and $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2$, i.e., $x = (x_1, x_2)$ and $z = (z_1, z_2)$.

Solutions. We define solutions to coupled control systems by assuming the existence of feedback control laws: $u(x, z) \triangleq \{u_1(x_1, z), u_2(x_2, z)\}$. Applying these controllers to (4.38) yields a *symmetric coupled dynamical system* (${}^{\dagger}\mathcal{D}$):

$${}^{\dagger}\mathcal{D} \triangleq \begin{cases} \dot{x}_i = f_i^{\text{cl}}(x_i, z) + \check{g}_e(x_i, z)\lambda_e \\ \dot{z}_i = p_i^{\text{cl}}(x_i, z) + \check{q}_e(x_i, z)\lambda_e \\ \text{s.t. } c_e(z) \equiv 0, \quad \lambda_e = -\lambda_{\bar{e}} \end{cases} \quad (4.39)$$

where,

$$\begin{aligned} f_i^{\text{cl}} &\triangleq f_i(x_i, z_i) + g_i(x_i, z_i)u_i(x_i, z) \\ p_i^{\text{cl}} &\triangleq p_i(x_i, z_i) + q_i(x_i, z_i)u_i(x_i, z). \end{aligned}$$

Then the *solution* of the coupled dynamic system, ${}^{\dagger}\mathcal{D}$, is a set of solutions:

$$\left\{ (x_1(t), z_1(t), \lambda_e(t)), (x_2(t), z_2(t), \lambda_{\bar{e}}(t)) \right\} \text{ s.t. } (4.39) \quad \forall t \in \mathbf{I} \subset \mathbb{R}$$

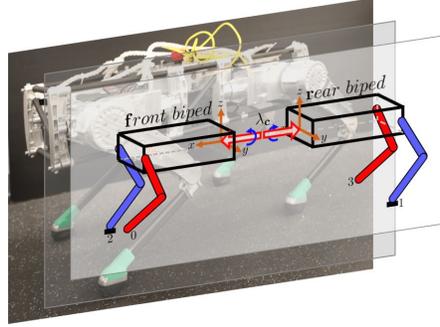


Figure 4.8: A conceptual illustration of the full-body dynamics decomposition, where the 3D quadruped — the Vision 60 — is decomposed into two constrained 3D bipedal robots.

with initial condition: $\{(x_1(0), z_1(0), \lambda_e(0)), (x_2(0), z_2(0), \lambda_{\bar{e}}(0))\}$, and $\mathbf{I} \subset \mathbb{R}$ is the time interval of their existence. Per the above notation, we will sometimes denote the solutions by $(x(t), z(t), \lambda(t))$ with initial condition $(x(0), z(0), \lambda(0))$.

Coupling constraints. Importantly, the solutions must satisfy the coupling constraints at all time. Therefore,

$$\begin{aligned}
 c_e(z) \equiv 0 &\Rightarrow \dot{c}_e(z, \dot{z}) \equiv 0 & (4.40) \\
 &\Rightarrow \underbrace{\frac{\partial c_e(z_i, z_j)}{\partial z_i}}_{\triangleq J_e^{(i,j)}(z)} \dot{z}_i + \underbrace{\frac{\partial c_e(z_i, z_j)}{\partial z_j}}_{\triangleq J_e^{(j,i)}(z)} \dot{z}_j \equiv 0 \\
 &\Rightarrow \dot{c}_e(x, z) = J_e^{(i,j)}(z) \left(p_i^{\text{cl}}(x_i, z) + \check{q}_e(x_i, z) \lambda_e \right) \\
 &\quad + J_e^{(j,i)}(z) \left(p_j^{\text{cl}}(x_j, z) + \check{q}_{\bar{e}}(x_j, z) \lambda_{\bar{e}} \right) \equiv 0. & (4.41)
 \end{aligned}$$

Hence, to solve for the coupling inputs λ_e that satisfy the coupling constraints, it is necessary to solve an equation that depends on the states of both subsystems. To address this, we present a method for isolating a subsystem via conditions on the controllers of the other systems in the next section. Before doing this, we utilize the following example to illustrate the concepts of coupled control systems.

Decompose a Quadruped into Bipedes

The motivating application considered here is to compute periodic solutions to the quadrupedal dynamics. As Fig. 4.1 shows, we decompose this quadruped into two bipedes, whose dynamics are on a CCS graph (according to Def. 18): $\mathcal{G} \triangleq (\mathcal{N} = \{f, r\}, \mathcal{E} = \{e = (f, r), \bar{e} = (r, f)\})$, where f, r label the *front* and *rear* bipedal systems, correspondingly. We pick the coordinates for these two subsystems

as $q_f = (\xi_f^\top, \theta_{L_2}^\top, \theta_{L_0}^\top)^\top$, $q_r = (\xi_r^\top, \theta_{L_1}^\top, \theta_{L_3}^\top)^\top$ with $\xi_i \in \mathbb{R}^3 \times \text{SO}(3)$ and the leg joints $\theta_{L_*} \in \mathbb{R}^3$. Since all leg joints are actuated, the inputs are $u_i \in \mathcal{U} \subset \mathbb{R}^6$. The continuous-time dynamics in Fig. 4.2, when toe1 and toe2 are on the ground, are modeled as constrained dynamics:

$$\begin{cases} D(q) \ddot{q} + H(q, \dot{q}) = Bu + J_f^\top(q) F_1 + J_r^\top(q) F_2 \\ J_f(q) \ddot{q} + \dot{J}_f(q, \dot{q}) \dot{q} = 0 \\ J_r(q) \ddot{q} + \dot{J}_r(q, \dot{q}) \dot{q} = 0 \end{cases}, \quad (4.42)$$

with the domain $\mathcal{D} \triangleq \{x \in \mathcal{X} : \dot{h}_1(q, \dot{q}) = \dot{h}_2(q, \dot{q}) = 0, h_{z_1}(q) = h_{z_2}(q) = 0\}$. In this formulation, we utilize the following notation: $D(q) \in \mathbb{R}^{n \times n}$ is the inertia-mass matrix; $H(q, \dot{q}) \in \mathbb{R}^n$ contains Coriolis forces and gravity terms; $h_1(q), h_2(q) \in \mathbb{R}^3$ are the Cartesian positions of toe1 and toe2, their Jacobians are $J_* = \partial h_*/\partial q$; $h_{z_1}(q), h_{z_2}(q)$ are these toes' height; $F_f, F_r \in \mathbb{R}^3$ are the ground reaction forces on toe1 and toe2; $B \in \mathbb{R}^{n \times m}$ is the actuation matrix. Essentially, we use a set of differential algebra equations (DAEs) to describe the dynamics of the quadrupedal robot that is subject to two holonomic constraints on toe1 and toe2.

We now decompose the quadrupedal full-body dynamics into two bipedal robots. First, as shown in Fig. 4.8, the open-loop dynamics can be equivalently written as:

$$\text{OL-Dyn} \triangleq \begin{cases} D_f \ddot{q}_f + H_f = J_f^\top F_f + B_f u_f - J_e^\top \lambda_e & (4.43) \\ J_f \ddot{q}_f + \dot{J}_f \dot{q}_f = 0 & (4.44) \\ D_r \ddot{q}_r + H_r = J_r^\top F_r + B_r u_r + J_e^\top \lambda_e & (4.45) \\ J_r \ddot{q}_r + \dot{J}_r \dot{q}_r = 0 & (4.46) \\ \ddot{\xi}_f - \ddot{\xi}_r = 0 & (4.47) \end{cases}$$

wherein we utilized the following notation: $D_f(q_f), D_r(q_r) \in \mathbb{R}^{12 \times 12}$ are the inertia-mass matrices of the front and rear bipedal robots; the Jacobians $J_f = \partial h_f / \partial q_f, J_r = \partial h_r / \partial q_r$ with the Cartesian positions of toe2 as $h_f(q_f)$ and toe1 as $h_r(q_r)$; the Jacobian matrix for the connection constraint (4.47) is $J_e = \partial(\xi_f - \xi_r) / \partial q_f$; $u_f^\top = (u_0^\top, u_2^\top)$ and $u_r^\top = (u_1^\top, u_3^\top)$. Note that the Cartesian position of toe2 only depends on q_f , which is due to the floating base coordinate convention.

Proposition 1. The dynamical system (OL-Dyn) is equivalent to the system (4.42).

Proof. We can write (4.43) and (4.45) as:

$$\begin{bmatrix} D_{b_f} & D_{b_0} & D_{b_2} \\ D_{b_0}^\top & D_0 & 0 \\ D_{b_2}^\top & 0 & D_2 \end{bmatrix} \begin{bmatrix} \ddot{\xi}_f \\ \ddot{\theta}_0 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} H_{b_f} \\ H_0 \\ H_2 \end{bmatrix} = B_f u_f + J_f^\top F_f - J_c^\top \lambda_e$$

$$\begin{bmatrix} D_{b_r} & D_{b_1} & D_{b_3} \\ D_{b_1}^\top & D_1 & 0 \\ D_{b_3}^\top & 0 & D_3 \end{bmatrix} \begin{bmatrix} \ddot{\xi}_r \\ \ddot{\theta}_1 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} H_{b_r} \\ H_1 \\ H_3 \end{bmatrix} = B_r u_r + J_r^\top F_r + J_e^\top \lambda_e$$

where each entry has a proper dimension to make the equations consistent. Expanding them yields:

$$\begin{bmatrix} D_{b_f} & D_{b_0} & 0 & D_{b_2} & 0 \\ D_{b_0}^\top & D_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ D_{b_2}^\top & 0 & 0 & D_2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\xi}_f \\ \ddot{\theta}_0 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} H_{b_f} \\ H_0 \\ 0 \\ H_2 \\ 0 \end{bmatrix} = \begin{bmatrix} -\lambda_e \\ u_0 \\ 0 \\ u_2 \\ 0 \end{bmatrix} + J_r^\top F_r,$$

$$\begin{bmatrix} D_{b_r} & 0 & D_{b_1} & 0 & D_{b_3} \\ 0 & 0 & 0 & 0 & 0 \\ D_{b_1}^\top & 0 & D_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ D_{b_3}^\top & 0 & 0 & 0 & D_3 \end{bmatrix} \begin{bmatrix} \ddot{\xi}_r \\ \ddot{\theta}_0 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} H_{b_r} \\ 0 \\ H_1 \\ 0 \\ H_3 \end{bmatrix} = \begin{bmatrix} \lambda_e \\ 0 \\ u_1 \\ 0 \\ u_3 \end{bmatrix} + J_f^\top F_f.$$

Combining these two equations, and using the fact that $\xi_f - \xi_r \equiv 0$ (holonomic constraint) yields the dynamics given in (2.7). It is worthwhile to note that all the terms that appear in these equations can be verified using traditional rigid body dynamics, and the corresponding details of the structure and necessary properties of the inertia-mass matrices can be found from the *branch-induced sparsity*, see (R. Featherstone, 2008). \square

Quadruped as CCS. Return to the symmetric CCS, we further abstract (OL-Dyn) into a set of Differential Algebraic Equations (DAEs):

$$\mathcal{R}_Q \triangleq \begin{cases} D_i \ddot{q}_i + H_i = J_i^\top F_i + B_i u_i + J_e^\top \lambda_e & (4.48) \\ J_i \ddot{q}_i + \dot{J}_i \dot{q}_i = 0 & (4.49) \\ \text{s.t. } c_e(\xi_i, \xi_j) = \xi_i - \xi_j \equiv 0 & (4.50) \\ \lambda_e = -\lambda_{\bar{e}} & (4.51) \end{cases}$$

with $D_i(q_i) \in \mathbb{R}^{n \times n}$ the mass-inertia matrix, $H_i(q_i, \dot{q}_i) \in \mathbb{R}^n$ the drift vector, and $B_i = \begin{bmatrix} \mathbf{0}_{6 \times 6} & I_{6 \times 6} \end{bmatrix}$ the actuation matrix. The contact (holonomic) constraint $h_i(q_i) \equiv 0$ is enforced via ground reaction forces $F_i \in \mathbb{R}^3$, whose second derivative is given in (4.49). More details of these notations can be found in (Wen-Loong Ma and Aaron D. Ames, 2020). Note that F_i can be eliminated by the solving (4.48)-(4.49) to have a shorter form: $D_i \ddot{q}_i + \bar{H}_i = \bar{B}_i u_i + \bar{J}_e^\top \lambda_e$. The derivation is similar to (2.17) hence omitted.

To obtain a CCS as in Def. 18, we pick “normal form” type coordinates (see Sastry, 1999), with the “output” that we wish to zero, given by

$$y_i(q_i, \alpha_i) = y^a(q_i) - y^d(\xi_i, \alpha_i), \quad (4.52)$$

where y^a , y^d are the actual and desired outputs, ξ_i represents a parameterization of time and $\alpha_i \in \mathbb{R}^{6 \times 6}$ are the coefficients for six fifth-order Bezier polynomials that are designed by the optimization algorithm in (CCS-OPT). Since our goal is to find a *symmetric ambling* gait for quadrupeds, we chose $\alpha_r = \mathcal{M} \alpha_f$, with the matrix \mathcal{M} representing a mirroring relation. We can then construct our internal states $x_i = (y_i^\top, \dot{y}_i^\top)^\top$, leaving the coupled states as $z_i = (\xi_i^\top, \dot{\xi}_i^\top)^\top$. The end result is a CCS of the form given in (4.38) for this mechanical system:

$$\begin{aligned} \dot{x}_i &= \underbrace{\begin{bmatrix} \dot{y}_i \\ J_{y_i} \dot{q}_i - J_{y_i} D_i^{-1} \bar{H}_i \end{bmatrix}}_{f_i(x_i, z_i)} + \underbrace{\begin{bmatrix} 0 \\ J_{y_i} D_i^{-1} \bar{B}_i \end{bmatrix}}_{g_i(x_i, z_i)} u_i + \underbrace{\begin{bmatrix} 0 \\ J_{y_i} D_i^{-1} \bar{J}_e^\top \end{bmatrix}}_{\check{g}_e(x_i, z_i, z_j)} \lambda_e \\ \dot{z}_i &= \underbrace{\begin{bmatrix} \dot{\xi}_i \\ -J_\xi D_i^{-1} \bar{H}_i \end{bmatrix}}_{p_i(x_i, z_i)} + \underbrace{\begin{bmatrix} 0 \\ J_\xi D_i^{-1} \bar{B}_i \end{bmatrix}}_{q_i(x_i, z_i)} u_i + \underbrace{\begin{bmatrix} 0 \\ J_\xi D_i^{-1} \bar{J}_e^\top \end{bmatrix}}_{\check{q}_e(x_i, z_i, z_j)} \lambda_e \\ \text{s.t. } & c_e(z_i, z_j) = z_i - z_j \equiv 0, \quad \lambda_e = -\lambda_{\bar{e}}, \end{aligned}$$

where $J_{y_i} = \partial y_i(q_i) / \partial q_i$, $J_\xi = \partial \xi / \partial q = \begin{bmatrix} I_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{bmatrix}$, and we suppressed the dependency on x_i, z_i for all entries.

λ -Coupled Subsystem. Given a CCS \mathcal{C}_C , we define the *zero dynamics manifold* for each subsystem $i \in \mathcal{N}$ as:

$$\mathbf{Z}_i \triangleq \{(x, z) \in \mathcal{X} \times \mathcal{Z} \mid x_i \equiv 0\}. \quad (4.53)$$

Thus, the zero dynamics manifold for i^{th} subsystem consists of the internal states, x_i , being zero, i.e., the system evolves only according to the coupled states z .

The key idea underlying the analysis of CCSs is to reduce the entire coupled system into the behavior of a single subsystem. This is achieved through the above constructions related to the zero dynamics. We start by designing controllers for the overall CCS on the zero dynamics of subsystem $j \in \mathcal{N}$. A controller $u_j^{\mathbf{Z},\lambda}(x_j, z)$ is said to **render the zero dynamics manifold \mathbf{Z}_j invariant** if it satisfies:

$$0 \equiv f_j(0, z_j) + g_j(0, z_j)u_j^{\mathbf{Z},\lambda}(0, z) + \check{g}_{\bar{e}}(0, z)\lambda_{\bar{e}} \quad (4.54)$$

where $u_j^{\mathbf{Z},\lambda}$ implicitly depends on $\lambda_{\bar{e}}$ for $\bar{e} = (j, i) \in \mathcal{E}$. By applying $u_j^{\mathbf{Z},\lambda}$, we obtain a **λ -coupled control subsystem (λ -CCSub)** for the i^{th} subsystem:

$$\dagger C_i^{\mathbf{Z},\lambda} \triangleq \begin{cases} \dot{x}_i = f_i(x_i, z_i) + g_i(x_i, z_i)u_i + \check{g}_e(x_i, z)\lambda_e \\ \dot{z}_i = p_i(x_i, z_i) + q_i(x_i, z_i)u_i + \check{q}_e(x_i, z)\lambda_e \\ \dot{z}_j = p_j(0, z_j) + q_j(0, z_j)u_j^{\mathbf{Z},\lambda}(0, z) + \check{q}_{\bar{e}}(0, z)\lambda_{\bar{e}} \\ \text{s.t. } c_e(z) \equiv 0, \quad \lambda_e = -\lambda_{\bar{e}} \end{cases} \quad (4.55)$$

Thus, the i^{th} subsystem evolves according to its own dynamics and the zero dynamics of all remaining systems—all of which are coupled via the coupling inputs $\lambda = \{\lambda_e, \lambda_{\bar{e}}\}$.

Explicit coupling conditions. The coupling between the control systems (4.38) is enforced via λ and the coupling constraints of the form (4.16). Similarly, even in the reduction to a subsystem (4.55), the coupling is still achieved through λ . We wish to generalize this so as to remove the coupling, i.e., isolate subsystems, while still preserving the overall behavior of the full system. We first define the **coupling relation** that allows the use of the controllers $u_j^{\mathbf{Z},\lambda}$ to eliminate the dependence on the controllers and internal states of the other subsystem.

Definition 19. For a λ -CCSub $C_i^{\mathbf{Z},\lambda}$ and $i \in \mathcal{N}$, a **coupling relation** is a functional relationship on the coupling inputs

$$\lambda_e^{\mathbf{Z}}(x_i, z; u_i) = A_e^{\mathbf{Z}}(x_i, z)u_i + b_e^{\mathbf{Z}}(x_i, z), \quad (4.56)$$

that satisfies the coupling constraint (4.15) for all $e = (i, j) \in \mathcal{E}$.

The coupling relation is then summarized in the following:

Lemma 5. For a CCS C_C , if we have

$$\check{Q}(x_i, z) \triangleq \begin{bmatrix} g_j(0, z_j) & -\check{g}_{\bar{e}}(0, z) \\ J_c^{(j,i)} q_j(0, z_j) & J_c^{(i,j)} \check{q}_e(x_i, z) - J_c^{(j,i)} \check{q}_{\bar{e}}(0, z) \end{bmatrix}$$

invertible, there exists a controller $u_j^{\mathbf{Z}}$ that renders \mathbf{Z}_j invariant and a coupling relation in (4.23), given by:

$$\begin{bmatrix} u_j^{\mathbf{Z}}(0, z; u_i) \\ \lambda_e^{\mathbf{Z}}(x_i, z; u_i) \end{bmatrix} = \check{Q}^{-1} \left(\begin{bmatrix} 0 \\ -J_c^{(i,j)} q_i(x_i, z_i) \end{bmatrix} u_i + \begin{bmatrix} -f_j(0, z_j) \\ -J_c^{(i,j)} p_i(x_i, z_i) - J_c^{(j,i)} p_j(0, z_j) \end{bmatrix} \right). \quad (4.57)$$

Proof. Evaluating (4.15) along the zero dynamics manifold \mathbf{Z}_j , i.e., $x_j \equiv 0$, yields:

$$\begin{aligned} J_e^{(i,j)}(z)(p_i(x_i, z) + q_i(x_i, z)u_i + \check{q}_e(x_i, z)\lambda_e) + \\ J_e^{(j,i)}(z)(p_j(0, z) + q_j(0, z)u_i + \check{q}_e(0, z)\lambda_e) = 0. \end{aligned}$$

Combining this with (4.18) and simultaneously solving for $u_j^{\mathbf{Z}}$ and $\lambda_e^{\mathbf{Z}}$ yields the desired result. \square

Recall that the controller $u_j^{\mathbf{Z}, \lambda}$ that renders the zero dynamics surface invariant implicitly depends on $\lambda_{\bar{e}}$ via (4.18). Now with a coupling relation, the dependence of $\lambda_{\bar{e}}$ is removed, and as a result we say that $u_j^{\mathbf{Z}}$ renders the zero dynamics manifold \mathbf{Z}_j invariant if:

$$0 \equiv f_j^{\mathbf{Z}}(0, z) + g_j^{\mathbf{Z}}(0, z)u_i + g_j(0, z_j) \left(u_j^{\mathbf{Z}}(0, z; u_i) - u_i \right) \quad (4.58)$$

where $u_j^{\mathbf{Z}}$ is now a function of u_i and

$$\begin{cases} f_j^{\mathbf{Z}}(x_j, z) & \triangleq f_j(x_j, z_j) - \check{g}_{\bar{e}}(x_j, z)b_e^{\mathbf{Z}}(x_i, z), \\ g_j^{\mathbf{Z}}(x_j, z) & \triangleq g_j(x_j, z_j) - \check{g}_{\bar{e}}(x_j, z)A_e^{\mathbf{Z}}(x_i, z). \end{cases} \quad (4.59)$$

Returning to (4.16), given a coupling relation we can rewrite this coupling constraint as:

$$\begin{aligned} \dot{c}_e(x_i, z) &= J_e^{(i,j)}(z) \left(p_i^{\mathbf{Z}}(x_i, z) + q_i^{\mathbf{Z}}(x_i, z)u_i \right) + J_e^{(j,i)}(z) \left(p_j^{\mathbf{Z}}(x_i, z) + q_j^{\mathbf{Z}}(x_i, z)u_i \right) \\ &= 0, \end{aligned} \quad (4.60)$$

where for the subsystem $C_i^{\mathbf{Z}, \lambda}$ we have

$$\begin{cases} p_i^{\mathbf{Z}}(x_i, z) & \triangleq p_i(x_i, z_i) + \check{q}_e(x_i, z)b_e^{\mathbf{Z}}(x_i, z) \\ q_i^{\mathbf{Z}}(x_i, z) & \triangleq q_i(x_i, z_i) + \check{q}_e(x_i, z)A_e^{\mathbf{Z}}(x_i, z) \\ p_j^{\mathbf{Z}}(x_i, z) & \triangleq p_j(0, z_j) + q_j(0, z_j)u_j^{\mathbf{Z}}(0, z) - \check{q}_{\bar{e}}(0, z)b_e^{\mathbf{Z}}(x_i, z) \\ q_j^{\mathbf{Z}}(x_i, z) & \triangleq -\check{q}_{\bar{e}}(0, z)A_e^{\mathbf{Z}}(x_i, z) \end{cases}. \quad (4.61)$$

Isolating subsystems. We now arrive at the key concept for which all of the previous constructions have built: reducing a CCS to a single subsystem that can be used to give guarantees about the entire CCS. This is based on the following definition.

Definition 20. For a CCS C_C , and $i \neq j \in \mathcal{N}$, assume a coupling relation λ_e^Z such that there exist u_j^Z rendering the zero dynamics manifold \mathbf{Z}_j invariant. Then the i^{th} control subsystem (CSub) associated with the CCS C_C is given by:

$$C_i^Z \triangleq \begin{cases} \dot{x}_i = f_i^Z(x_i, z) + g_i^Z(x_i, z)u_i \\ \dot{z}_i = p_i^Z(x_i, z) + q_i^Z(x_i, z)u_i \\ \dot{z}_j = p_j^Z(x_i, z) + q_j^Z(x_i, z)u_i \end{cases} \quad (4.62)$$

where $f_i^Z(x_i, z) \triangleq f_i(x_i, z_i) + \check{g}_e(x_i, z)b_e^Z(x_i, z)$, $g_i^Z(x_i, z) \triangleq g_i(x_i, z_i) + \check{g}_e(x_i, z)A_e^Z(x_i, z)$, and $p_i^Z, q_i^Z, p_j^Z, q_j^Z$ are given in (4.61). Furthermore, when a feedback controller $u_i(x_i, z)$ is applied to C_i^Z , the result is a dynamical system, denoted by \mathcal{D}_i^Z .

Apply to Quadrupedal Walking

For the quadrupedal dynamics \mathcal{R}_Q , since the output (4.52) has (vector) relative degree 2 with respect to u_i (see (E. R. Westervelt, J. W. Grizzle, Chevallereau, et al., 2007)), we can explicitly design the controller $u_j^{Z,\lambda}$ that renders \mathbf{Z}_j invariant:

$$u_j^{Z,\lambda} = (J_{y_i} D_j^{-1} \bar{B}_j)^{-1} (J_{y_j} D_j^{-1} \bar{H}_j - \dot{J}_{y_j} \dot{q}_j - J_{y_i} D_j^{-1} \bar{J}_e^T \lambda_e),$$

as given by Lemma 5. Hence, this controller satisfies (4.18) and renders a λ -coupled CSub, as in (4.55).

For robotic systems, we take these ideas one step further to obtain ‘‘bipeds’’ that are the isolated subsystems associated with quadrupeds, and include slack variables that are beneficial for gait generation. Operating on the invariant zero dynamics manifold \mathbf{Z}_j yields $y_j(q_j, \alpha_j) \equiv 0$, hence

$$\begin{aligned} &\Rightarrow \theta_a \equiv H_a^{-1} y^d(\xi_j, \alpha_j), \\ &\Rightarrow q_j^Z(\xi_j) \equiv (\xi_j^T, (H_a^{-1} y^d(\xi_j, \alpha_j))^T)^T \\ &\Rightarrow \ddot{q}_j^Z(\xi_j, \dot{\xi}_j, \ddot{\xi}_j) = J_z(\xi_j) \ddot{\xi}_j + \dot{J}_z(\xi_j, \dot{\xi}_j) \dot{\xi}_j, \end{aligned}$$

where $J_z = \partial q_j^Z(\xi_j) / \partial \xi_j$. In other words, if $u_j^{Z,\lambda}$ exists and is applied to j^{th} subsystem, the j^{th} bipedal dynamics given by in (4.48)-(4.49) are equivalent to:

$$\begin{cases} D_j \ddot{q}_j^Z(\xi_j, \dot{\xi}_j, \ddot{\xi}_j) + H_j = J_j^T F_j + B_j u_j^Z + J_e^T \lambda_e & (4.63) \\ J_j \ddot{q}_j^Z(\xi_j, \dot{\xi}_j, \ddot{\xi}_j) + \dot{J}_j \dot{q}_j^Z(\xi_j, \dot{\xi}_j) = 0 & (4.64) \end{cases}$$

where for simplicity we have suppressed the dependencies of $D_j(q_j(\xi_j))$, $J_j(q_j(\xi_j))$ and $H_j(q_j(\xi_j), \dot{q}_j(\xi_j, \dot{\xi}_j))$. We then leverage a specific structure of rigid-body dynamics when using the floating base convention: $B_j u_j + J_e^\top \lambda_e = (\lambda_e^\top, u_j^\top)^\top$. Utilizing this, (4.64) and the first 6 rows of (4.63) yield the following ‘‘bipedal’’ dynamics:

$$\mathcal{R}_B^{\mathbf{Z}_j} \triangleq \begin{cases} D_j^{\mathbf{Z}} \ddot{\xi}_j + H_j^{\mathbf{Z}} = \hat{J}_j^\top F_j + \lambda_e \\ J_j^{\mathbf{Z}} \ddot{\xi}_j + w_j^{\mathbf{Z}} = 0 \end{cases} \quad (4.65)$$

with $D_j^{\mathbf{Z}} = \hat{D}_j J_z$, $H_j^{\mathbf{Z}} = \hat{D}_j \dot{J}_z \dot{\xi}_j + \hat{H}_j$, $J_j^{\mathbf{Z}} = J_j J_z$, and $w_j^{\mathbf{Z}} = J_j \dot{J}_z \dot{\xi}_j + \dot{J}_j \dot{J}_z \dot{\xi}_i$. Here, we denote $\hat{\cdot}$ as the first 6 rows (block) of the a variable. Hence, $\mathcal{R}_B^{\mathbf{Z}_j}$ represents the dynamics of a subsystem j on \mathbf{Z}_j , i.e., (4.65) evolves according to (4.18) where F_j can be uniquely determined.

Coupling Relations for Robotic Systems. Following the previous construction, we can also have the explicit form of the controller $u_j^{\mathbf{Z}}$ that renders zero dynamics manifold \mathbf{Z}_j invariant and the coupling condition in (4.23) as

$$\begin{aligned} \begin{bmatrix} u_j^{\mathbf{Z}}(x_i, z; u_i; t) \\ \lambda_e^{\mathbf{Z}}(x_i, z; u_i; t) \end{bmatrix} &= \begin{bmatrix} J_{y_j} D_j^{-1} \bar{B}_j & -J_{y_j} D_j^{-1} \bar{J}_e^\top \\ J_{\xi_j} D_j^{-1} \bar{B}_j & -J_{\xi_j} D_j^{-1} \bar{J}_e^\top - J_{\xi_i} D_i^{-1} \bar{J}_e^\top \end{bmatrix}^{-1} \\ &\left(\begin{bmatrix} \ddot{y}^d - \dot{J}_{y_j} \dot{q}_j - J_{y_j} D_i^{-1} \bar{H}_j \\ -J_{\xi_i} D_i^{-1} \bar{H}_i + J_{\xi_j} D_j^{-1} \bar{H}_j \end{bmatrix} + \begin{bmatrix} 0 \\ J_{\xi_i} D_i^{-1} \bar{B}_i \end{bmatrix} u_i \right) \end{aligned} \quad (4.66)$$

where $J_{y_i} = \partial y_i / \partial q_i$, $J_{\xi_i} = \partial q_i / \partial \xi_i$. Note that we suppressed the the dependence on x_i, z, t for notation simplicity. From this, we can obtain the CSub $C_i^{\mathbf{Z}}$, as in (4.62).

Gait Generation for flat-ground ambling. Finally, we can apply the optimization in (CCS-OPT) to simultaneously produce trajectories (solutions) of the states $\{x_i(t), z(t)\}$, $u_j^{\mathbf{Z}}(t)$ that renders the zero dynamics manifold \mathbf{Z}_j invariant and the open-loop controller $u_i(t)$, $\forall t \in [0, T]$ for which these solutions are defined. Note that one can also enforce the dynamics $\dot{x}_i^\kappa + \varepsilon x_i^\kappa = 0$ with $\varepsilon > 0$ to guarantee the convergence attribute of the i^{th} isolating subsystem, in which case the controller $u_i(x_i, z)$ is equivalently an input-output feedback linearization controller.

When posing the control problem of quadrupeds, we leverage the subsystems representing the front and rear bipeds: $\mathcal{R}_B^{\mathbf{Z}_f}$ and $\mathcal{R}_B^{\mathbf{Z}_r}$, as given in (4.65). Note that these subsystems are still coupled through λ —while this could be explicitly solved for via Lemma 5, we keep it implicit due to the complexity of inverting the mass-inertia matrix for this particular robotic application. The i^{th} subsystem yield (C.1), (C.2), and (C.3) for (CCS-OPT). Specifically for all of the grid indices $\kappa = 0, 1, \dots, 5$, we

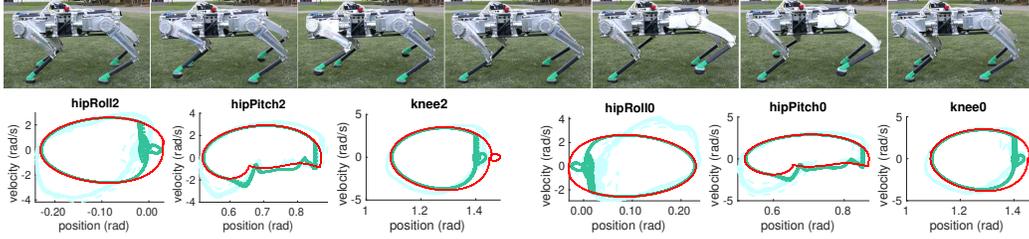


Figure 4.9: Top: Snapshots showing a full step of the ambling gait on an outdoor lawn. Bottom: The periodic trajectory produced by optimization (CCS-OPT) (in red) vs. the experimental tracking data (in cyan) vs. RaiSim simulation data (in green) in the form of phase portrait (limit cycle) using 18 seconds' data.

have the decision variables: $\vartheta^K = \{q_f^K, \dot{q}_f^K, \xi_r^K, \dot{\xi}_r^K, u_f^K, F_f^K, F_r^K, \alpha_f, \lambda_e^K\}$. Finally, the optimization converges to a periodic solution to the isolated bipedal system, which can then be composed to obtain the ambling motion of the quadruped (shown in Fig. 4.9) according to Theorem 1. We report that the optimization took 17.6s and 295 iterations of searching, which is over 58% faster than the full-model based approach in Sec. 4.1 (see (W. Ma, K. A. Hamed, and A. D. Ames, 2019)). The computational complexity is mitigated mainly due to the dimension reduction of the state space which is enabled by the representation of the quadrupedal dynamics as bipedal subsystems. For validation purposes, both simulations in the physics engine RaiSim and hardware experiments were conducted with a unified, time-based PD approximation of input-output linearizing controllers to track the the desired outputs (represented by $\alpha_f, \alpha_r = \mathcal{M}\alpha_f$):

$$u_i(q_i, \dot{q}_i, t) = -k_p(\dot{y}^a(q_i) - \dot{y}_t^d(t, \alpha_i)) - k_d(y^a(q_i) - y_t^d(t, \alpha_i)) \quad (4.67)$$

with k_p, k_d the PD gains. In addition, the switching detection and the event functions are also given by the optimized trajectories, meaning the walking controller will switch to next step when $t = T$. The result is successful ambling in simulation, and experimentally walking on flat and outdoor uneven terrains (see the video (Wen-Loong Ma, 2020c)). See Fig. 4.9 for walking tiles and the tracking performance. Remark that the averaged absolute torque inputs are 11.16 N·m, which are well within the hardware limits.

Time-based stepping in place. To benchmark the computational performance, we also generate stepping-in-place motion for the quadruped and demonstrate them on hardware. For this behavior, our desired output y_d will be a function of time instead

of states. We express the algorithm in a different way from (CCS-OPT) as:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{j=1}^{2N+1} \|\dot{\xi}_f\|_2^2 & (4.68) \\ \text{s.t.} \quad & \mathbf{C1.} \text{ isolating bipedal dynamics } \mathcal{R}_B^{\mathbf{Z}^j} \text{ in (4.65)} & j = 1, 3, \dots, 2N + 1 \\ & \mathbf{C2.} \text{ collocation constraints} & j = 2, 4, \dots, 2N \\ & \mathbf{C3.} \text{ impact dynamics in(4.6)} & j = 2N + 1 \\ & \mathbf{C4.} \text{ periodic continuity} & j = 1, 2N + 1 \\ & \mathbf{C5.} \text{ physical feasibility} & j = 1, 2, \dots, 2N + 1 \end{aligned}$$

with the following notation: $2N + 1 = 11$ is the total number of collocation grids; $\alpha \in \mathbb{R}^{36}$ are the coefficients for the Bezier polynomial that defines the desired trajectory $y^d(t)$; and \square^j is the corresponding quantities at time t_j with $t^{2N+1} = T$. The cost function is to minimize the body's vibration rate to achieve a more static torso movement. The constraints **C1-C3** solve the hybrid dynamics of bipedal robots subject to external forces. Details regarding the numerical optimization can be found in Sec. 2.9. Here, the periodic continuity constraint **C4** enforces state continuity through an edge, i.e., the post-impact states q^+, \dot{q}^+ , are equivalent to the initial states q^1, \dot{q}^1 . Therefore, the resultant trajectory is a periodic solution to the bipedal dynamics. **C5** imposed some feasibility conditions on the dynamics, including torque limits $\|u_i\|_\infty \leq 50$, joint feasible space $(q_i, \dot{q}_i) \in \mathcal{X}$, foot clearance, and the friction pyramid conditions. Note that we posed these constraints conservatively to reduce the difficulties implementing the optimized trajectories in experiments.

To validate that the CCS-based optimization's solutions are same as those of the full-order quadrupedal dynamics, we simulated an ambling step of the quadrupedal dynamics using the composed bipedal gaits. As shown in Fig. 4.10, we have the joint angles and constraint wrench (ground reaction force) on toe1 $F_{1,z}$, and toe2 $F_{2,z}$ of the quadruped matched with those corresponding external force to the bipedal dynamics.

The result of the methods presented is the ability to generate quadrupedal gaits rapidly. We benchmark the performance by considering computing speed for each of the quadrupedal locomotion patterns generated, as is shown in Table 4.1. In summary, with the objective tolerance and equality constraint tolerance configured as 10^{-8} and 10^{-5} respectively, we have the average computation time as 3.96 second, and time per iteration averages 0.039 second. For simulation, we first validated the dynamic stability of the gaits produced by the decomposition-based optimization

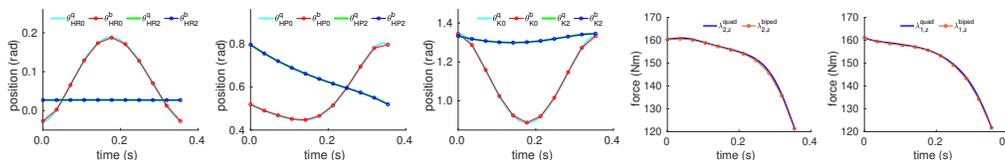


Figure 4.10: A comparison between the solution of bipedal walking dynamics obtained from the decomposition-based optimization and a simulated step of the full-order quadrupedal dynamics using the composed bipedal gaits; here MATLAB ODE45 was used.

problem using a third party physics engine, MuJoCo. These gaits include a diagonally symmetric ambling and four stepping-in-place behaviors. Then we conducted experiments, walking on a outdoor tennis court, using the same control law as that in simulation in outdoor environments. We report that for all given optimal gaits, the PD gains are picked as $k_p = 230, 230, 300, k_d = 5$ for the hip roll, hip pitch, and knee joints, respectively. The averaged absolute joint torque inputs are logged in Table 4.2, all of which are well within the hardware limitations. See (Wen-Loong Ma, 2020a) for the video of Vision 60 stepping in place in both simulation and experiments.

Table 4.1: Computing performance of gait generation. This is performed on a Linux machine with an i7-6820HQ CPU @2.70 GHz and 16 GB RAM.

Behaviors	gait1	gait2	gait3	gait4	amble
frequency (Hz)	2.5	2.3	2.2	2.6	2.83
clearance (cm)	11	12	15	13	13
# of iterations	96	122	98	46	147
time of IPOPT (s)	1.60	2.10	1.62	0.81	2.59
time of evaluation (s)	1.94	3.24	2.10	0.94	2.86
NLP time(s)	3.54	5.34	3.72	1.75	5.45

Table 4.2: Average torque inputs in experiments and simulations.

Experiments	gait1	gait2	gait3	gait4	amble
$\bar{u}_{HR}(\text{N}\cdot\text{m})$	5.04,	4.83	4.16	5.14	7.11
$\bar{u}_{HP}(\text{N}\cdot\text{m})$	3.65	5.24	5.26	3.77	6.28
$\bar{u}_K(\text{N}\cdot\text{m})$	16.45	16.50	16.86	16.95	18.36
MuJoCo					
$\bar{u}_{HR}(\text{N}\cdot\text{m})$	7.80	9.23	10.27	8.68	8.06
$\bar{u}_{HP}(\text{N}\cdot\text{m})$	6.78	9.14	10.71	6.64	7.27
$\bar{u}_K(\text{N}\cdot\text{m})$	18.49	18.38	18.45	18.61	19.03



Figure 4.11: Comparison between MuJoCo simulation (animated) and experiments. The upper two are for stepping in place, gait4; the lower two are logged for a full step of the ambling gait.

Walking on Sloped Terrains

We now make a few changes to the dynamics and the optimization algorithm to generate walking motion on sloped terrains.

Guard condition and discrete dynamics on slopes. For the symmetric ambling motion, the diagonal toes of the quadrupedal robot stay on the ground while the other two toes are swing in the air. This means each bipedal robot has one toe-foot in contact with the ground. Concretely, the Cartesian position of the stance toe remains zero, i.e. $h_s(q_i) \equiv 0$. This contact constraint is enforced by a ground reaction force $F_i \in \mathbb{R}^3$, where $i \in \{f, r\}$. We additionally denote the Cartesian position of the nonstance toe along x, z directions as $h_{ns}^x(q_f), h_{ns}^z(q_f) \in \mathbb{R}$. The nonstance foot's height on the slope is then given by

$$\hat{h}_{ns}(q_i) = h_{ns}^z(q_i) \sin \phi - h_{ns}^x(q_i) \cos \phi.$$

Hence we can define the single-support domain for bipedal dynamics (4.65) as:

$$\mathcal{D}_i \triangleq \{(q_i, \dot{q}_i) \in \mathcal{Q}_i \times T\mathcal{Q}_i \mid h_s(q_i) = 0, \hat{h}_{ns}(q_i) \geq 0\}.$$

The guard is then defined on the edge of the domain \mathcal{D}_i as $\mathcal{S}_i \triangleq \{(q_i, \dot{q}_i) \in \mathcal{Q}_i \times T\mathcal{Q}_i \mid h_s(q_i) = \hat{h}_{ns}(q_i) = 0, \hat{h}_{ns}(q_i, \dot{q}_i) < 0\}$, on which we define the discrete dynamics that represents plastic impacts (see equation (20) in (Wen-Loong Ma and Aaron D. Ames, 2020)) at the event that nonstance toe touches down on the sloped terrain.

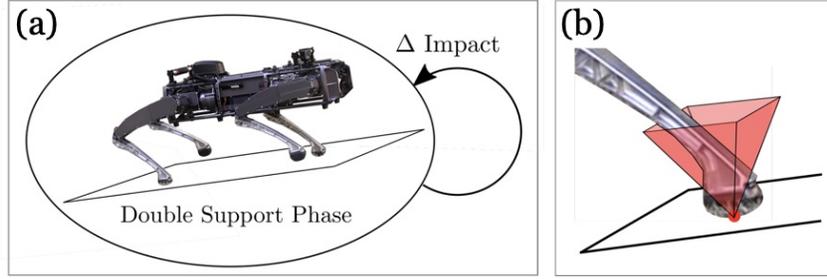


Figure 4.12: (a) The cyclic directed graph for the multi-domain hybrid dynamics on sloped terrain; (b) friction cone (pyramid) condition on some sloped terrain.

The friction cone condition on slopes. The feasibility conditions $\mathcal{A}_i(F_i) \geq 0$ are enforced to guarantee the solutions to the dynamics is physically realizable. In particular, we have the friction pyramid condition on sloped terrains as:

$$\mathcal{A}_i(F_i) \triangleq \begin{cases} F_i^z \cos \phi - F_i^x \sin \phi \\ \mu(F_i^z \sin \phi + F_i^x \cos \phi) - |F_i^z \cos \phi - F_i^x \sin \phi| \\ \mu(F_i^z \sin \phi + F_i^x \cos \phi) - |F_i^y| \end{cases} \quad (4.69)$$

where μ is the coefficient of dynamic friction of the ground. It is worth noting the feasibility formulation (4.69) is more restrictive for walking on sloped terrains than walking on stairs or level ground.

Optimization constraints. In practice, path constraints (inequality constraints) are often used to “fine-tune” the optimal results according to human intuition and physical limitations. Evaluating the optimality of an optimization solution based on experimental performance is rather empirical, and intensive constraint tuning is often needed for field testing. The ultimate target of this paper is to present a method that can be seamlessly used to produce periodic gaits for hardware experiments on sloped terrains. Hence, we explicitly list our path constraints as follows:

- Joint angles do not exceed physical limits;
- Absolute joint velocities below 4 rad/s;
- Absolute acceleration less than 120 rad/s²;
- Absolute joint torque less than 50 N·m;
- Stepping period $t^K \in [0.29, 0.37]$;

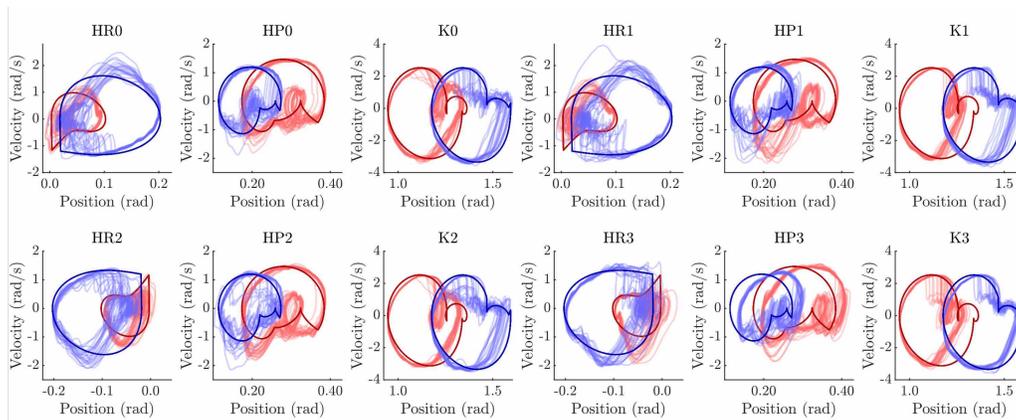


Figure 4.13: Phase portraits of the designed gaits (solid lines) vs. experimental data (transparent overlay) for quadrupedal walking on 13° (red) and $20^\circ \sim 25^\circ$ (blue) slopes. HR, HP, K are short for hip roll, hip pitch, and knee, respectively.

- Nonstance toe's height $\hat{h}_i(q_i^\kappa) \geq 0.1$ at $\kappa = K/2$;
- Nonstance toe's absolute velocities (parallel to the sloped ground) slower than 1.3 m/s;
- Ground impact velocity $\dot{\hat{h}}_{ns}(q_i, \dot{q}_i) \in [-2, -0.2]$.

Remark that these constraints setup are not modified throughout the optimization for all of the gaits that are experimentally tested.

Results. By only changing the slope angles to $\phi = 0^\circ, 13^\circ, 15^\circ, 20^\circ, 25^\circ$ in the optimization problem (4.68), we were able to generate periodic solutions efficiently. These solutions are further used to obtain quadrupedal gaits for experiments. We hereby report the average computational time for all five gaits are 9.7 seconds and 271 searching iterations on a Ubuntu 16.04 machine with Intel Core i7-6820 HQ CPU @ 2.7 GHz with 16 GB RAM. The phase portrait of the gaits on the 13° and $20^\circ \sim 25^\circ$ sloped terrains are shown in solid lines in Fig. 4.13. Before directly enabling these sloped walking gaits optimized by the CCS optimization framework on the actual hardware — a 44 kg, 56 cm-wide robot (Fig. 1.1) — we first validate their physical feasibility and dynamic stability under a feedback control law in a physics engine, RaiSim (Hwangbo, J. Lee, and Marco Hutter, 2018). The quadrupedal robot was first tested on a consistently graded 13° grassy slope with minimal surface variation, which was replicated in RaiSim by creating a plane of constant incline. In order to truly test this methodology, the quadruped was next asked to traverse a grassy slope just after it had rained with inclination ranging from 20° to 25° . In order to emulate

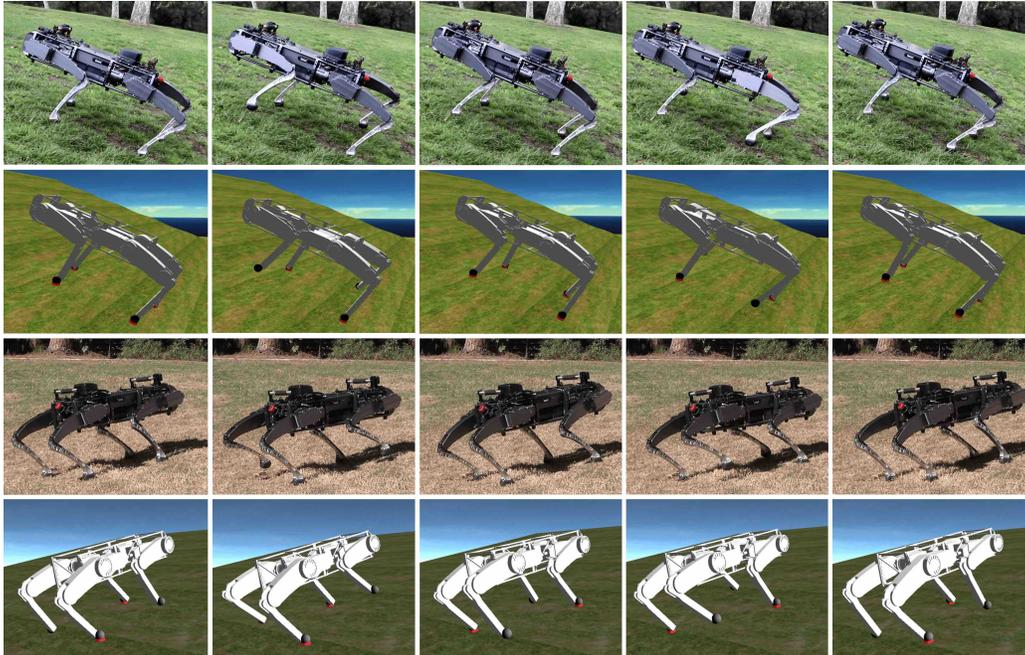


Figure 4.14: Full steps of gait tiles for the Vision 60 ambling in outdoor grasslands. The top two is comparison between simulation and experiments for a 13° slope. The bottom two are for walking on a ramp with varying slopes of $20^\circ \sim 25^\circ$.

the varying slopes in RaiSim, a terrain was created with a sinusoidally oscillating height varying between 20° and 25° with a frequency that approximated that of the outdoor environment. As a baseline, both a level-ground gait-based controller and the stock controller were tested on the $20^\circ \sim 25^\circ$ slope. We report that neither were able to navigate on the sloped terrain successfully. As shown in the gait tiles in Fig. 4.14, our proposed method allows the robot to successfully amble across both the 13° slope and the 20° and 25° slope despite the unmodeled variation in slope and lowered friction effects. See Wen-Loong Ma, 2020b for a video demonstration of the experimental validation of the quadruped, in which we also showed all five gaits walking on slopes of 0° , 13° , 15° , 20° , 25° in RaiSim. This result demonstrates the necessity of designing optimal trajectories and controllers based on the specific terrain types using full-body dynamics. Additionally, we remark that the PD gains and ground friction coefficient are the same across all simulation and experimental implementations.

In addition, we logged 20 seconds of experimental data and compared them with the desired ambling gait designed by the optimization, as seen in Fig. 4.13 with phase portraits. Note the difference in the desired behavior for the two terrains. This diversity of behavior further motivates the use of unique gaits conditioned on

the terrain to ensure the stability of the robotic system. An important metric to quantify control performance for locomotion is the mechanical cost of transport (MCOT), which was calculated using equation (16) of (W. Ma, Y. Or, and A. D. Ames, 2019). Unlike the traditional formulation where the distance term only accounts for the horizontal displacement (Collins et al., 2005a), we used the averaged three-dimensional velocity on the sloped terrain, which also considers the vertical displacement. We report that in simulation, the MCOT for 13° and $20^\circ \sim 25^\circ$ sloped ambulating are 2.01 and 2.86, accordingly.

4.4 Coupled Control Lyapunov Functions

After introducing the coupled control system-based optimization for generating physically feasible gaits for the high-dimensional quadrupedal dynamics, we still need to formally define the control law for driving the local subsystem to the designated stable behaviors.

In this section, we will accomplish two targets. First, through the formulation of coupled control Lyapunov functions, we formally define the stability criteria of each subsystem while they are dynamically coupled with the rest of the system due to the shared zero dynamics. We can then utilize these Lyapunov functions to synthesize local optimal control laws for each subsystem to guarantee the overall coupled control system's stability, hence the full-order dynamics. Second, when applying to rigid-body dynamics, we incorporate quadratic programming formulations with two types of Lyapunov functions for controller design. First, feedback linearization-based Control Lyapunov Functions (CLFs) are synthesized and demonstrated on a cart-pole example showing stability. Second, PD-inspired Lyapunov functions are used to synthesize model-free CLFs for experimental robustness. These CLFs are applied to stabilizing the continuous dynamics of quadrupedal locomotion. This stabilization is demonstrated in simulation with regard to hopping and running. Finally, we present this framework on hardware, specifically the Vision 60 robot v3.9 (see Fig. 4.15). We empirically show that it is able to walk stably and robustly in outdoor environments.

Recall some CLF definitions from Sec. 3.4,

Definition 21. A smooth function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is an *exponentially stabilizing control Lyapunov function (ES-CLF)* for

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

if there exists constants $c_1, c_2, c_3 > 0$ such that

$$\begin{aligned} c_1 |x|^2 &\leq V(x) \leq c_2 |x|^2 \\ \inf_u (L_f V(x) + L_g V(x)u + c_3 V(x)) &\leq 0 \end{aligned} \quad (4.70)$$

for all $x \in \mathcal{X}$. If there further exists a constant $\varepsilon \in (0, 1)$ such that

$$\begin{aligned} c_1 |x|^2 &\leq V(x) \leq \frac{c_2}{\varepsilon^2} |x|^2 \\ \inf_u \left(L_f V(x) + L_g V(x)u + \frac{c_3}{\varepsilon} V(x) \right) &\leq 0 \end{aligned} \quad (4.71)$$

for all $x \in \mathcal{X}$, then $V(x)$ is a **rapidly exponentially stabilizing control Lyapunov function (RES-CLF)**.

Definition 22. A smooth function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is an **input-to-state stabilizing control Lyapunov function (ISS-CLF)** for

$$\dot{x} = f(x) + g(x)(u + d),$$

if there exist functions $\underline{\alpha}, \bar{\alpha}, \alpha, \iota \in \mathcal{K}_\infty$ as such that $\forall x, d$

$$\begin{aligned} \underline{\alpha}(|x|) &\leq V(x) \leq \bar{\alpha}(|x|) \\ \dot{V}(x, d) &\leq -\alpha(|x|) + \iota(\|d\|_\infty), \end{aligned}$$

and the exponential estimate is:

$$\dot{V}(x, d) \leq -cV(x) + \iota(\|d\|_\infty).$$

CLF from the Viewpoint of ISS

We first consider an affine control system in the form of

$$\begin{aligned} \mathcal{C} &\triangleq \begin{cases} \dot{x}_1 = f_1(x_1, x_2) + g_1(x_1, x_2)(u_1 + d_1) \\ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)(u_2 + d_2) \end{cases} \\ \Leftrightarrow \dot{x} &= \underbrace{\begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} g_1(x) & 0 \\ 0 & f_2(x) \end{bmatrix}}_{g(x)}(u + d) \end{aligned} \quad (4.72)$$

where $u = (u_1^\top, u_2^\top)^\top \in \mathcal{U} \subset \mathbb{R}^m$ are the inputs, and $d = (d_1^\top, d_2^\top)^\top \in \mathbb{R}^m$ are termed as the *disturbance inputs* with d_i the subsystem disturbance. Equivalently, we can write the i^{th} subsystem \mathcal{C}_i as

$$\dot{x}_i = f_i(x) + g_i(x)(u_i + d_i), \quad \forall i \in \mathcal{N} \quad (4.73)$$

for a set of nodes $\mathcal{N} = \{1, 2\}$, where $x \triangleq (x_1^\top, x_2^\top)^\top$, $x_i \in \mathcal{X}_i \subset \mathbb{R}^{n_i}$, $u_i \in \mathcal{U}_i \subset \mathbb{R}^{m_i}$.

Using the same notations from Sec. 4.2, we first present the following definitions that are the foundation of the main theorem in this section.

Definition 23. A smooth function $V : \mathbb{R}^{n_i} \rightarrow \mathbb{R}_+$ is a *rapidly exponential input-to-state stabilizing control Lyapunov function (Re-ISS-CLF)* for

$$\dot{x}_i = f(x, z) + g(x, z)(u_i + d_e)$$

with $x_i \in \mathbb{R}^{n_i}$, if there exists constants $c_1, c_2, c_3 > 0$, $\varepsilon \in (0, 1)$, $\bar{\varepsilon} > 0$ such that $\forall x, z, d$,

$$c_1 |x_i|^2 \leq V_i(x_i) \leq \frac{c_2}{\varepsilon^2} |x_i|^2 \quad (4.74)$$

$$\inf_{u_i \in \mathcal{U}_i} \left(L_f V_i(x, z) + L_g V_i(x, z) u_i + \frac{c_3}{\varepsilon} V_i(x_i) + \frac{1}{\bar{\varepsilon}} |L_g V_i|^2 \right) \leq 0.$$

The construction of Def. 23 is motivated by the *rapidly exponentially stabilizing control Lyapunov function (RES-CLF)* from (A. Ames, Galloway, et al., 2014b). Based on Def. 23, we can form a class of control laws directly:

$$K_i(x, z) \triangleq \{u_i \in \mathcal{U}_i : L_{f_i} V_i + L_{g_i} V_i u_i + \frac{c_{3,i}}{\varepsilon_i} V_i + \frac{1}{\bar{\varepsilon}_i} |L_{g_i} V_i|^2 \leq 0\}, \quad (4.75)$$

which yields the set of control values which satisfy the desired convergence property for each subsystem $i \in \mathcal{N}$. The (constant) parameters $c_{1,i}, c_{2,i}, c_{3,i}, \varepsilon_i, \bar{\varepsilon}_i$ are associated with each subsystem with $i \in \mathcal{N}$.

We then prove that the combination of these stabilizing controllers also yields stability for the full-order system in the following theorem.

Theorem 3. Assume there exists a ISS-stabilizing controllers $u_i(x) \in K_i(x)$ for each subsystem $i \in \mathcal{N}$ given by (4.73). Given a collection of such controllers $\{u_i(x)\}_{i \in \mathcal{N}}$, the solution to the full-order dynamics in (4.72) is e-ISSable.

Proof. We consider an Re-ISS-CLF candidate for the full-order system:

$$V(x) = V_1(x_1) + V_2(x_2). \quad (4.76)$$

It can be seen that

$$\begin{aligned} \min_i (c_{1,i}) |x|^2 &= \min_i (c_{1,i}) (|x_1|^2 + |x_2|^2) \leq V(x) \\ V(x) &\leq \max_i \left(\frac{c_{2,i}}{\varepsilon_i^2} \right) (|x_1|^2 + |x_2|^2) = \max_i \left(\frac{c_{2,i}}{\varepsilon_i^2} \right) |x|^2. \end{aligned}$$

Note that we use the notation $\|d\|_\infty \triangleq \sup_{t \geq 0} \{|d_i|\}$, which yields some properties that will be used later:

$$\begin{aligned} |d| &\geq \max_i(|d_i|) \geq |d_i| \\ \|d\|_\infty &= \sup_{t \geq 0} \{|d|\} \geq \sup_{t \geq 0} \{|d_i|\} = \|d_i\|_\infty. \end{aligned}$$

Then,

$$\begin{aligned} \dot{V}(x, u, d) &= \dot{V}_1(\cdot) + \dot{V}_2(\cdot) \\ &= \sum L_{f_i} V_i + L_{g_i} V_i u_i + L_{g_i} V_i d_i \\ &\leq \sum -\frac{c_{3,i}}{\varepsilon_i} V_i - \frac{1}{\bar{\varepsilon}_i} L_{g_i} V_i L_{g_i} V_i^\top + L_{g_i} V_i d_i \\ &\leq \sum -\frac{c_{3,i}}{\varepsilon_i} V_i - \frac{1}{\bar{\varepsilon}_i} L_{g_i} V_i L_{g_i} V_i^\top + |L_{g_i} V_i| |d_i| \\ &\leq \sum -\frac{c_{3,i}}{\varepsilon_i} V_i - \frac{1}{\bar{\varepsilon}_i} |L_{g_i} V_i|^2 + |L_{g_i} V_i| |d| \\ &\leq \sum -\frac{c_{3,i}}{\varepsilon_i} V_i - \left(\frac{1}{\sqrt{\bar{\varepsilon}_i}} |L_{g_i} V_i| - \sqrt{\bar{\varepsilon}_i} \frac{\|d\|_\infty}{2} \right)^2 + \bar{\varepsilon}_i \frac{\|d\|_\infty^2}{4} \\ &\leq \sum -\frac{c_{3,i}}{\varepsilon_i} V_i(x_i) + \bar{\varepsilon}_i \frac{\|d\|_\infty^2}{4} \\ &\leq -2 \min_i \left(\frac{c_{3,i}}{\varepsilon_i} \right) V(x) + \frac{\max_i(\bar{\varepsilon}_i)}{2} \|d\|_\infty^2, \end{aligned}$$

which satisfies Def. 22, in which case $\iota(\|d\|_\infty) = \frac{\max_i(\bar{\varepsilon}_i)}{2} \|d\|_\infty^2$. \square

Remark that an effective way to reduce the effect of *overall disturbance* $\|d\|_\infty$ is to decrease all $\bar{\varepsilon}_i$.

Exponential stability. Note that the Lyapunov function given by Def. 23 is also a RES-CLF for the system:

$$\dot{x}_i = f_i(x) + g_i(x)u_i \quad \forall i \in \mathcal{N}$$

with f_i, g_i the same as (4.72). Concretely, there exists constants $c_1, c_2, c_3 > 0$ and $0 < \varepsilon < 1$ such that

$$\begin{aligned} c_{1,i} |x_i|^2 &\leq V_i(x_i) \leq \frac{c_{2,i}}{\varepsilon_i^2} |x_i|^2 \\ \inf_u \left(L_{f_i} V_i + L_{g_i} V_i u_i + \frac{c_{3,i}}{\varepsilon_i} V_i \right) &\leq 0. \end{aligned}$$

Under certain conditions, we can establish exponential stability for system (4.72), which are summarized as follows.

Corollary 2. In addition to the conditions given by Theorem 3, if we further have

$$|d(t)| \leq c_4 |x(t)| \quad \text{and} \quad |L_g V(x)| \leq c_5 |x| \quad \forall x. \quad (4.77)$$

The solution to the full-order dynamics in (4.72) is exponentially stable provided that $2 \min_i (\frac{c_{3,i}}{\varepsilon_i}) > c_4 c_5 \min_i (c_{1,i})$.

Proof. We first have c_1, c_2 given by the proof of Thm. 3. Then

$$\begin{aligned} \dot{V} &= \sum \dot{V}_i = \sum L_{f_i} V_i + L_{g_i} V_i (u_i + d_i) \\ &\leq \sum -\frac{c_{3,i}}{\varepsilon_i} V_i + L_{g_i} V_i d_i \\ &\leq -2 \min_i \left(\frac{c_{3,i}}{\varepsilon_i} \right) V + \sum L_{g_i} V_i \cdot d_i \\ &\leq -2 \min_i \left(\frac{c_{3,i}}{\varepsilon_i} \right) V + L_g V \cdot d \\ &= -2 \min_i \left(\frac{c_{3,i}}{\varepsilon_i} \right) V + |L_g V| |d| \\ &\leq -2 \min_i \left(\frac{c_{3,i}}{\varepsilon_i} \right) V + c_4 c_5 |x|^2 \\ &\leq -2 \min_i \left(\frac{c_{3,i}}{\varepsilon_i} \right) V + c_4 c_5 \min_i (c_{1,i}) V. \end{aligned}$$

Therefore, if $2 \min_i (\frac{c_{3,i}}{\varepsilon_i}) > c_4 c_5 \min_i (c_{1,i})$, \dot{V} is negative definite which implies exponential stability of the full-order system, according to Def. 21. \square

Note that for a general situation when the disturbance does not completely vanish on the zero dynamics surface, i.e., $|d(t)| \leq c_4 |\eta| + c_6 |z| + c_7$, the system exponentially converges to a ultimate bound for robotic dynamics.

Shared zero dynamics. For interconnected systems, we often need to consider the *shared dynamics* among subsystems. We consider such systems in normal form as:

$$C^z \triangleq \begin{cases} \dot{x}_1 = f_1(x, z) + g_1(x, z)(u_1 + d_1) \\ \dot{x}_2 = f_2(x, z) + g_2(x, z)(u_2 + d_2) \\ \dot{z} = \omega(x, z) \end{cases} \quad (4.78)$$

where, the z -dynamics $z = \omega(x, z)$ are regarded as the *passive dynamics*, and we call $\dot{z} = \omega(0, z)$ the *zero dynamics*. We assume $\omega(x, z)$ is locally Lipschitz. Built on Theorem 3, we have the following theorem to characterize the full-order system's stability.

We now present the main theorem of this section that guarantees the stability of the full-order system by taking values from $K_i(x, z)$, $\forall i \in \mathcal{N}$.

Theorem 4. For a dynamical system given by (4.78), let O_z be an exponentially stable periodic orbit of the zero dynamics $\dot{z} = \omega(\mathbf{0}, z)$. If there exists an ISS-CLF $V_i(x_i)$ for each subsystem $i \in \mathcal{N}$, then for all locally Lipschitz continuous feedbacks $u_i(x) \in K_i(x)$ given by (4.75), the full-order periodic orbit $O \triangleq \iota(O_z)$ is ultimately bounded, with the bounds tending to zero as $|d_1|, |d_2| \rightarrow 0$.

Proof. First, we use the converse Lyapunov theorem from (Hauser and Chung Choo Chung, 1994) to construct the following Lyapunov function for the zero dynamics. Given O_z is an exponentially stable periodic orbit of \mathcal{Z} , there exists a Lyapunov function $V_z : \mathcal{Z} \rightarrow \mathbb{R}_+$ such that in a neighborhood $B_\delta(O_z)$ of O_z ,

$$\begin{aligned} r_1 \|z\|_{O_z}^2 &\leq V_z(z) \leq r_2 \|z\|_{O_z}^2, \\ \dot{V}_z(z) &\leq -r_3 \|z\|_{O_z}^2, \\ \left| \frac{\partial V_z}{\partial z} \right| &\leq r_4 \|z\|_{O_z}. \end{aligned}$$

Next we have the following Lyapunov function candidate for the full-order system:

$$V(x, z) = \sum_i V_i(x_i) + \sigma V_z(z).$$

It is clear that $V(x, z)$ satisfies the first inequality in Def. 23 (the definition of Lyapunov functions for an invariant set, such as periodic orbits, can be found in (E. D. Sontag and Yuan Wang, 1996).) We first take the derivative of the subsystems' Lyapunov functions to get:

$$\begin{aligned} \sum_i \dot{V}_i &= \sum_i L_{f_i} V_i + L_{g_i} V_i u_i + L_{g_i} V_i d_i \\ &\leq \sum_i -\frac{c_{3,i}}{\varepsilon_i} V_i - \frac{1}{\bar{\varepsilon}_i} |L_{g_i} V_i|^2 + |L_{g_i} V_i| |d_i| \\ &\leq \sum_i -\frac{c_{3,i}}{\varepsilon_i} V_i - \left(\frac{1}{\sqrt{\bar{\varepsilon}_i}} |L_{g_i} V_i| - \sqrt{\bar{\varepsilon}_i} \frac{\|d\|_\infty}{2} \right)^2 + \bar{\varepsilon}_i \frac{\|d\|_\infty^2}{4} \\ &\leq -\min_i \left(\frac{c_{3,i}}{\varepsilon_i} c_{1,i} \right) |x|^2 + \frac{\max_i(\bar{\varepsilon}_i)}{2} \|d\|_\infty^2. \end{aligned}$$

Then the total derivative of the Lyapunov function $V(x, z)$ becomes:

$$\begin{aligned}
\dot{V} &= \sigma \frac{\partial V_z}{\partial z} w(0, z) + \sigma \frac{\partial V_z}{\partial z} (w(x, z) - w(0, z)) + \sum_i \dot{V}_i \\
&\leq -\sigma r_3 \|z\|_{O_z}^2 + \sigma r_4 \|z\|_{O_z} |w(x, z) - w(0, z)| + \sum_i \dot{V}_i \\
&\leq -\sigma r_3 \|z\|_{O_z}^2 + \sigma r_4 \|z\|_{O_z} L_z |x| + \sum_i \dot{V}_i \\
&\leq -\sigma r_3 \|z\|_{O_z}^2 + \sigma r_4 L_z \|z\|_{O_z} |x| - \underbrace{\min_i \left(\frac{c_{3,i}}{\varepsilon_i} c_{1,i} \right)}_{r_6} |x|^2 + \frac{\max_i(\bar{\varepsilon}_i)}{2} \|d\|_\infty^2, \\
&= - \begin{bmatrix} \|z\|_{O_z} & |x| \end{bmatrix} A \begin{bmatrix} \|z\|_{O_z} \\ |x| \end{bmatrix} + \frac{\max_i(\bar{\varepsilon}_i)}{2} \|d\|_\infty^2,
\end{aligned}$$

with L_z the Lipschitz constant for $\omega(x, z)$ and

$$A = \begin{bmatrix} \sigma r_3 & -\frac{1}{2}\sigma r_4 L_z \\ -\frac{1}{2}\sigma r_4 L_z & r_6 \end{bmatrix},$$

which satisfies Def. 22. We then can pick σ such that A is positive definite, i.e., V is a Lyapunov function for the periodic orbit $O = \iota(O_z)$. \square

The proof is inspired by the construction of (A. Ames, Galloway, et al., 2014b, Appx.B). We note that an effective way to reduce the effect of the disturbance is to decrease $\bar{\varepsilon}_i$. Further, since ISS-CLF is one robust type of CLFs, we will continue to use the terminology CLFs for clarity. Note that for the class of robotic systems of interest (such as the quadruped in Fig. 4.18), it can be shown that any CLF qualifies as an ISS-CLF (see (Angeli, 1999)). In other words, the set given by (4.75) needs not have $L_{g_i} V_i$.

Stability condition. As the theorem suggested, we can thus construct the local control Lyapunov functions for a coupled control system. Following the construction of (4.75), we have a class of controllers using a linear constraint of the input:

$$\rho_i(\eta, z) + \psi_i(\eta, z) u_i + \frac{1}{\varepsilon_i} |\psi_i(\eta, z)|^2 \leq 0 \quad (4.79)$$

where $\rho_i(\eta, z) = L_{f_i} V_i(\eta, z) + \frac{\gamma}{\varepsilon_i} V_i(\eta_i)$ and $\psi_i(\eta, z) = L_{g_i} V_i$. A practical control law that satisfies (4.79) is a minimum-norm in $K_i(\eta, z)$, given by

$$m_i(\eta, z) = \operatorname{argmin}\{ |u_i|^2 : u_i \in K_i(\eta, z) \}, \quad (4.80)$$

which can be solved by some quadratic programming (QP) algorithm.

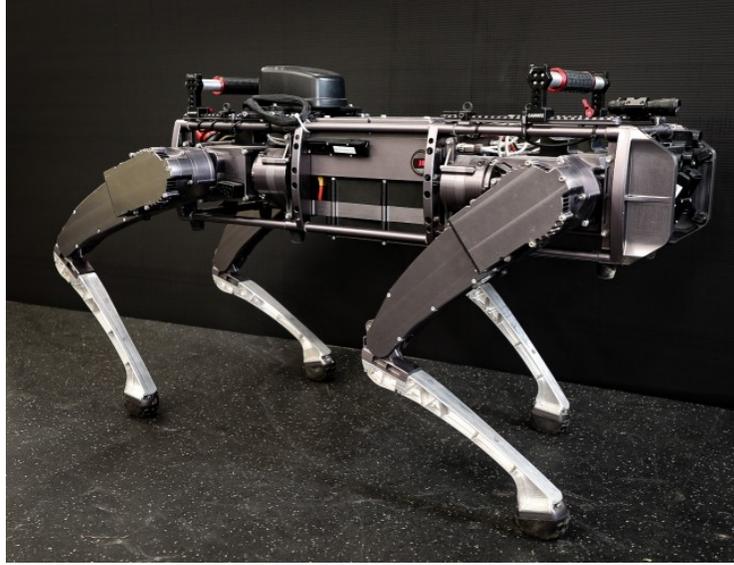


Figure 4.15: The quadrupedal robot, Vision 60 v3.9.

4.5 Coupled Mechanical Systems

In this section, we apply the coupled control Lyapunov function to rigid-order dynamics. Given a system that is composed of multiple interconnected rigid-bodies, the equations of motion (EOMs) of the *full-order dynamics* (also referred to as full-order dynamics) can be obtained through Euler–Lagrange equations:

$$D(q)\ddot{q} + H(q, \dot{q}) = B(q)u, \quad (4.81)$$

The notation is consistent with Sec. 2.3. In this section, we are interested in the dynamical systems that can be considered as a collection of **two** subsystems with index $i \in \{1, 2\} \triangleq \mathcal{N}$. We first define subsystem configurations as $q_i \in \mathcal{Q}_i \subset \mathbb{R}^{n_i}$ such that $\cup_{i \in \mathcal{N}} \iota_i(\mathcal{Q}_i) = \mathcal{Q}$ with $\iota_i : \mathcal{Q}_i \rightarrow \mathbb{R}^n$ as a canonical embedding. Since the goal is to control each subsystem individually, the subsystem inputs are defined as components of the full-system inputs $u^\top = (u_1^\top, u_2^\top)$ with $u_i \in \mathcal{U}_i \in \mathbb{R}^{m_i}$ and $\sum_{i \in \mathcal{N}} m_i = m$. We also define a set of edges $\mathcal{E} \triangleq \{(1, 2), (2, 1)\}$ representing the subsystems' connection.

For a dynamical system that is composed of two subsystems (coupled via constraints), such as the *coupled mechanical systems* considered in Fig. 4.16, we have

$$\mathcal{R}_Q \triangleq \begin{cases} D_1(q_1)\ddot{q}_1 + H_1(q_1, \dot{q}_1) = B_1(q_1)u_1 + J_e^\top(q_1, q_2)\lambda_e \\ D_2(q_2)\ddot{q}_2 + H_2(q_2, \dot{q}_2) = B_2(q_2)u_2 + J_{\bar{e}}^\top(q_1, q_2)\lambda_{\bar{e}} \\ \text{s.t. } c_{e,q}(q_1, q_2) \equiv 0, \quad \lambda_e + \lambda_{\bar{e}} = 0 \end{cases} \quad (4.82)$$

where $\lambda_e, \lambda_{\bar{e}} \in \Lambda_i \in \mathbb{R}^{l_i}$ are the coupling forces and $c_{e,q}$ is the coupling constraint. We can solve the connection force explicitly to reach the form in (4.81) using

$$\begin{aligned} \lambda_e = -\lambda_{\bar{e}} = & (J_e D_1^{-1} J_e - J_{\bar{e}} D_2^{-1} J_{\bar{e}})^{-1} [J_e D_1^{-1} (H_1 - B_1 u_1) \\ & + J_{\bar{e}} D_2^{-1} (H_2 - B_2 u_2) - \dot{J}_e \dot{q}_1 - \dot{J}_{\bar{e}} \dot{q}_2] \end{aligned} \quad (4.83)$$

where $J_e(q_1, q_2) = \partial c_{e,q} / \partial q_i$ and $e \triangleq (i, j), \bar{e} \triangleq (j, i) \in \mathcal{E}$.

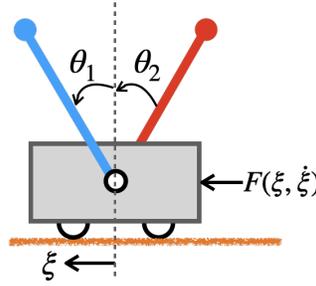


Figure 4.16: A cart-pole system with two inverted pendula, each is directly actuated by a motor. The mass of the cart and pendula are $2M$ and m , correspondingly. The length of both pendulum is l . We have the configuration coordinates as $(\xi, \theta_1, \theta_2)^\top$.

Subsystem Dynamics

After defining the subsystem with an index set \mathcal{N} , we can pick the outputs (the features that we are interested in controlling) of each i^{th} subsystem as

$$y_i(q_i) = y_i^d(q_i) - y_i^a(q_i) \quad (4.84)$$

where $y_i^d, y_i^a \in \mathbb{R}^{m_i}$ are the *desired* outputs and the *actual* outputs, respectively. Since y_i is a function of the “positional states” q_i , it has a relative degree two with respect to the control inputs. We then have the i^{th} subsystem dynamics in output coordinates as

$$\ddot{y}_i = \mathcal{L}_i(q, \dot{q}) + \mathcal{A}_i(q) u_i + \mathcal{A}_{ji}(q) u_j \quad (4.85)$$

for all $i \in \mathcal{N}$. Note that $\mathcal{A}_{ji}(q) \in \mathbb{R}^{m_i \times m_j}$ maps u_j with $j \neq i$ to the configuration space of the i^{th} subsystem.

Depending on the given EOMs, there are different ways to obtain the expressions efficiently in (4.85). One direct method from (4.81) is given as

$$\begin{aligned} \begin{bmatrix} \mathcal{L}_1(q, \dot{q}) \\ \mathcal{L}_2(q, \dot{q}) \end{bmatrix} &= \dot{J}_y \dot{q}^2 - J_y D^{-1} H \triangleq \mathcal{L}(q, \dot{q}) \\ \begin{bmatrix} \mathcal{A}_1(q) & \mathcal{A}_{21}(q) \\ \mathcal{A}_{12}(q) & \mathcal{A}_2(q) \end{bmatrix} &= J_y D^{-1} B \triangleq \mathcal{A}(q), \end{aligned} \quad (4.86)$$

where $J_y = \partial y / \partial q$ with the full-system outputs are denoted as $y = (y_1^\top, y_2^\top)^\top$.

For underactuated systems where $m < n$, zero dynamics will show up in the transition to output coordinates (see (Isidori, 2011)). As discussed in Sec. 3.2, there exists a change of coordinates via a diffeomorphism:

$$\begin{bmatrix} q_1 \\ \dot{q}_1 \\ q_2 \\ \dot{q}_2 \end{bmatrix} \mapsto \begin{bmatrix} \eta_1 \\ \eta_2 \\ z \end{bmatrix}$$

which yields a set of dynamic equations representing the *coupled control system*:

$$\mathcal{C}_C \triangleq \begin{cases} \ddot{y}_1 = \mathcal{L}_1(\eta, z) + \mathcal{A}_1(\eta, z)u_1 + \mathcal{A}_{21}(\eta, z)u_2 \\ \ddot{y}_2 = \mathcal{L}_2(\eta, z) + \mathcal{A}_2(\eta, z)u_2 + \mathcal{A}_{12}(\eta, z)u_1 \\ \dot{z} = \omega(\eta, z) \end{cases} \quad (4.87)$$

where $\eta = (\eta_1^\top, \eta_2^\top)^\top \in \mathcal{X}$ are the “controlled states,” and $\eta_i = (y_i^\top, \dot{y}_i^\top)^\top$. Note that both \mathcal{L} and \mathcal{A} now depend on the new coordinates η, z . The z -dynamics, $\dot{z} = \omega(\eta, z)$, are regarded as the *internal dynamics* with $z \in \mathcal{Z}$, and we call $\dot{z} = \omega(\mathbf{0}, z)$ the *zero dynamics*, i.e., the dynamics on the zero dynamics manifold:

$$\mathcal{Z} = \{(\eta, z) \in \mathcal{X} \times \mathcal{Z} : \eta_i = 0, \forall i \in \mathcal{N}\}. \quad (4.88)$$

We assume $\omega(\eta, z)$ is locally Lipschitz in η . Note that we can also convert the formulation given by W. Ma, Csomay-Shanklin, and A. D. Ames, 2021, Eq.1 into the form of (4.87) again by using (4.83). In this form, not only are the dynamics of each subsystem coupled through the shared zero dynamics coordinates, but the inputs are also coupled, i.e., u_j ($i \neq j$) appears in the i^{th} subsystem dynamics.

Disturbed Subsystem.

To design local controllers for the i^{th} subsystem that are independent of the “disturbance” caused by the other subsystems’ inputs, we first introduce the *nominal inputs* that are built using the zero dynamics. We then introduce the main result of this paper, a theorem that leads to the synthesis of the networked control architecture. It is this controller that later enables us to control each sub-bipedal system individually with stability guarantees.

Before designing the control law $u(x) \triangleq \{u_i(x)\}_{i \in \mathcal{N}}$, we first give the concept of a *nominal control input* in the following definition.

Definition 24. The control input that renders the zero dynamics surface $\mathcal{Z} \triangleq \{(q, \dot{q}) : y_i = \dot{y}_i = 0, \forall i \in \mathcal{N}\}$ forward-invariant is the *nominal input* for a coupled control system (4.87), i.e.,

$$\mathbf{0} = \mathcal{L}_i(\mathbf{0}, z) + \mathcal{A}_i(\mathbf{0}, z)u_i^{\mathbf{Z}} + \mathcal{A}_{ji}(\mathbf{0}, z)u_j^{\mathbf{Z}} \quad (4.89)$$

for all $i \in \mathcal{N}$. We further define $u^{\mathbf{Z}}(z) \triangleq \{u_i^{\mathbf{Z}}(z)\}_{i \in \mathcal{N}}$.

For the rigid-body dynamics of interest, the decoupling matrix $\mathcal{A}(\mathbf{0}, z)$ is assumed to be invertible. Hence, the unique controller that satisfies (4.89) would be as follows:

$$u^{\mathbf{Z}}(z) = -\mathcal{A}^{-1}(\mathbf{0}, z)\mathcal{L}(\mathbf{0}, z) \triangleq \begin{bmatrix} u_1^{\mathbf{Z}} \\ u_2^{\mathbf{Z}} \end{bmatrix}. \quad (4.90)$$

By considering the nominal control input $u_j^{\mathbf{Z}}$ of the j^{th} subsystem ($j \neq i$), we can reformulate the subsystem dynamics (4.85) to remove the dependence on the other interconnected (coupled) subsystem's control input. Concretely, we have

$$\begin{aligned} \ddot{y}_i &= \mathcal{L}_i(\eta, z) + \mathcal{A}_i(\eta, z)u_i + \mathcal{A}_{ji}(\eta, z)(u_j + u_j^{\mathbf{Z}}(z) - u_j^{\mathbf{Z}}(z)) \\ &= \mathcal{L}_i(\eta, z) + \mathcal{A}_{ji}(\eta, z)u_j^{\mathbf{Z}}(z) + \underbrace{\mathcal{A}_i(\eta, z)u_i + \mathcal{A}_{ji}(\eta, z)(u_j - u_j^{\mathbf{Z}}(z))}_{\triangleq d_e} \end{aligned} \quad (4.91)$$

where we denote

$$d_e(\eta, z, u_j) \triangleq \mathcal{A}_{ji}(\eta, z)(u_j - u_j^{\mathbf{Z}}(z)), \quad e \triangleq (j, i) \quad (4.92)$$

as the *disturbance* induced by the j^{th} subsystem's inputs to the i^{th} subsystem. Having established the disturbed subsystem dynamics as in (4.91), the coupled control system in (4.87) becomes a *disturbed coupled control system*, as:

$$C_C^d \triangleq \begin{cases} \ddot{y}_1 = \mathcal{L}_1(\eta, z) + \mathcal{A}_{21}(\eta, z)u_2^{\mathbf{Z}}(z) + \mathcal{A}_1(\eta, z)u_1 + d_e \\ \ddot{y}_2 = \mathcal{L}_2(\eta, z) + \mathcal{A}_{12}(\eta, z)u_1^{\mathbf{Z}}(z) + \mathcal{A}_2(\eta, z)u_2 + d_{\bar{e}} \\ \dot{z} = \omega(\eta, z) \end{cases} \quad (4.93)$$

where each subsystem is only subject to local controller and a disturbance term. This is where we can utilize input-to-state stabilizing control Lyapunov functions to reject the disturbance while stabilizing each subsystem.

Remark 4. Note that the overall disturbance vanishes on the invariant zero dynamics manifold \mathbf{Z} , i.e., $d(\mathbf{0}, z, u_i^{\mathbf{Z}}) = \{d_e(\mathbf{0}, z, u_i^{\mathbf{Z}})\}_{\forall e \in \mathcal{E}} = \mathbf{0}$. This can be seen by plugging $u_i(0, z) = u_i^{\mathbf{Z}}$ from (4.89) into (4.91), whereby

$$0 = \mathcal{L}_i(0, z) + \mathcal{A}_{ji}(0, z)u_j^{\mathbf{Z}} + \mathcal{A}_i(0, z)u_i^{\mathbf{Z}} + d_e = 0 + d_e,$$

which yields $d_e = 0$ and further $d = 0$.

After posing the dynamics of regular rigid-body dynamics (4.81) into the coupled mechanical system form in (4.93), we are ready to design local controllers to stabilize each subsystem, further stabilizing the full-order system. To do so, we explain the controller design procedures with two concrete examples: a cart-pole system with two invert pendula and a quadrupedal robot that is decoupled into two bipedal systems.

4.6 Example 1. Feedback Linearization-Based CLFs

For the cart-pole system shown in Fig. 4.16. We use Lagrangian method to get:

$$\begin{aligned}
 L &= M\dot{\xi}^2 + \frac{1}{2}mv_1^2 + \frac{1}{2}mv_2^2 - mgl \cos \theta_1 - mgl \cos \theta_2 \\
 v_i^2 &= (\dot{\xi} - l \cos \theta_i \dot{\theta}_i)^2 + (l \sin \theta_i \dot{\theta}_i)^2 \\
 \Rightarrow D &= \begin{bmatrix} 2M + 2m & -ml \cos \theta_1 & -ml \cos \theta_2 \\ -ml \cos \theta_1 & ml^2 & 0 \\ -ml \cos \theta_2 & 0 & ml^2 \end{bmatrix}, \\
 H &= \begin{bmatrix} ml\dot{\theta}_1 \sin \theta_1 + ml\dot{\theta}_2 \sin \theta_2 \\ -mgl \sin \theta_1 \\ -mgl \sin \theta_2 \end{bmatrix} + \begin{bmatrix} -F(\xi, \dot{\xi}) \\ 0 \\ 0 \end{bmatrix}.
 \end{aligned}$$

The forcing function $F(\xi, \dot{\xi})$ can be one the following:

$$2(m + M)(-\xi + \dot{\xi}(1 - \xi^2 - \dot{\xi}^2)) \quad (\text{F1})$$

$$2(m + M)(-\xi + \dot{\xi}(1 - \xi^2)) \quad (\text{F2})$$

$$2(m + M)(-100\xi - 10\dot{\xi}) \quad (\text{F3})$$

which will yield (a variation of) the von der Pol oscillation, or an exponentially stable equilibrium point for the zero dynamics. (Powell and Aaron D Ames, 2014) has used a similar setup. We will study the case with a globally stable orbit on zero dynamics, which is given by (F1), denoted by \mathcal{O}_z .

Separated by the dashed line as Fig. 4.16, we can view each pendulum-cart system as a subsystem with index $i \in \mathcal{N} \triangleq \{1, 2\}$. Let the subsystem configuration be $q_i = (\xi, \theta_i)$. With a target to control the outputs (a.k.a. the virtual constraint (Jessy W. Grizzle et al., 2014)), we define the subsystem output (virtual constraint) as

$$y_i(q_i) = \theta_i$$

for the i^{th} subsystem, i.e. the goal is to drive both pendula upright as $t \rightarrow \infty$. With the output Jacobian obtained by

$$y(q) = \begin{bmatrix} y_1(q_1) \\ y_2(q_2) \end{bmatrix} \Rightarrow J_y \triangleq \frac{\partial y}{\partial q} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

we can use (4.86) to obtain the dynamics in the form of coupled control systems, as in (4.87). Note that the zero dynamics $\dot{z} = \omega(\mathbf{0}, z)$ — when $\theta_i = \dot{\theta}_i = 0$, $\forall i \in \mathcal{N}$ — become a Van der Pol oscillator due to the force function: $\ddot{\xi} = F(\xi, \dot{\xi})/(2m + 2M)$.

For rigid-body dynamics with invertible decoupling matrices \mathcal{A}_\square , we can apply an input-output feedback-linearization:

$$u_i(\eta, z) = \mathcal{A}_i^{-1}(\eta, z) \left(-\mathcal{L}_i(\eta, z) - \mathcal{A}_{ji}(\eta, z)u_j^Z(z) + \mu_i \right) \quad (4.94)$$

with μ_i the *auxiliary input* for each subsystem $i \in \mathcal{N}$. The nominal control input u_j^Z with $j \neq i$ is then given by (4.90). The subsystem output dynamics now become:

$$\ddot{y}_i = \mu_i + d_e. \quad (4.95)$$

If we define $\eta_i = (y_i^\top, \dot{y}_i^\top)^\top$, we can obtain the linearized subsystem dynamics as

$$\dot{\eta}_i = \underbrace{\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}}_F \eta_i + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_G (\mu_i + d_e), \quad (4.96)$$

which is in the form of (4.78). Therefore, we can define the coupled control Lyapunov functions according to Theorem 4. Concretely, for each subsystem i , we have

$$V_i(\eta_i) = \eta_i^\top P_i \eta_i, \text{ with } P_i \triangleq \begin{bmatrix} \frac{1}{\varepsilon_i} I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\varepsilon_i} I & 0 \\ 0 & I \end{bmatrix}. \quad (4.97)$$

with $\varepsilon_i \in (0, 1)$ a constant and $P \in \mathbb{R}^{2 \times 2}$ the solution to the *continuous time algebraic Riccati equation* (CARE). More details can be found in A. Ames, Galloway, et al., 2014b, Sec.3.

Remark 5. Based on the CLF chosen, an appropriate control can be constructed that yields control robustness. For example, using the feedback linearization of the form (4.94) we can choose the control law as:

$$u_i = \mathcal{A}_i^{-1} \left(-\mathcal{L}_i - \mathcal{A}_{ji}u_j^Z - \frac{1}{\varepsilon_i^2} K_p y_i - \frac{1}{\varepsilon_i} K_d \dot{y}_i - \frac{1}{\varepsilon_i} L_G V_i^\top \right) \quad (4.98)$$

where $K_p, K_d \geq 0$. This controller, inspired by (Eduardo D. Sontag, 1999), is a specific example that belongs to the set $K_i(\eta, z)$.

CLF-QP. We now present the QP formulation that calculates control values using the chosen CLFs. Note that μ_i is only an auxiliary input instead of the actual system-level input. We will replace it with u_i for better numerical conditioning for the optimal control problem. Based on (4.94) we have μ_i as a function of u_i :

$$\mu_i = \mathcal{L}_i + \mathcal{A}_{ji}u_j^{\mathbf{Z}} + \mathcal{A}_i u_i.$$

Then the stability condition (4.79) can be re-written as:

$$\rho_i + \psi_i(\mathcal{A}_i u_i + \mathcal{L}_i + \mathcal{A}_{ji}u_j^{\mathbf{Z}}) + \frac{1}{\bar{\varepsilon}_i} |\psi_i|^2 \leq 0, \quad (4.99)$$

$$\text{with } \rho_i = L_F V_i + \frac{\gamma}{\varepsilon_i} V_i = \eta_i^\top (F^\top P_i + P_i F) \eta_i + \frac{\gamma}{\varepsilon_i} V_i,$$

$$\psi_i = L_G V_i = 2\eta_i^\top P_i G.$$

Finally, we have the following QP formulation that encodes the CLF for subsystem $i \in \mathcal{N}$:

$$\begin{aligned} u_i^* = \operatorname{argmin}_{u_i \in \mathcal{U}_i} & \left| \mathcal{L}_i + \mathcal{A}_{ji}u_j^{\mathbf{Z}} + \mathcal{A}_i u_i \right|^2 & (4.100) \\ \text{s.t. (C1)} & \rho_i + \psi_i(\mathcal{A}_i u_i + \mathcal{L}_i + \mathcal{A}_{ji}u_j^{\mathbf{Z}}) + \frac{1}{\bar{\varepsilon}_i} |\psi_i|^2 \leq 0 \\ \text{(C2)} & -u_{\max} \leq u_i \leq u_{\max} \end{aligned}$$

where, (C1) is the stability constraint, and (C2) is added according to the actual torque bounds from the physical actuators to guarantee the realizability. We regarded (4.100) the *CLF-QP for coupled mechanical systems*.

Remark 6. With both subsystems taking values from $K_i(\eta, z)$, we have the disturbance as

$$d_e(\eta, z) = \mathcal{A}_{ji}(\eta, z)(u_j(\eta, z) - u_j^{\mathbf{Z}}(z)).$$

Assuming $d(\eta, z)$ to be locally Lipschitz in η yields

$$|d(\eta, z) - d(\mathbf{0}, z)| \leq c_4 \|\eta - \mathbf{0}\| \Rightarrow |d| \leq c_4 \|\eta\|$$

with c_4 the Lipschitz constant. An effective way to reduce c_4 is to form an optimization problem inside a tube around the given desired trajectory \mathcal{O} , i.e. $\min \max_{(\eta, z) \in \text{tube near } \mathcal{O}} |d(\eta, z)| / \|\eta\|$. Additionally, we have

$$\left| \sum L_G V_i \right| = \left| 2 \sum \eta_i^\top P_i G \right| \leq 2 \|\eta\| \sum \|P_i G\|_2.$$

Hence, if we pick $c_5 = 2 \sum \|P_i G\|_2$, we can obtain exponential stability for the periodic solution to the full-order system according to Corollary 2. This is also reflected in the simulation data that will be shown next.

Simulation. We present two simulation results (see Fig. 4.17 and video (Wen-Loong Ma, 2021)) to demonstrate this stability result. As shown in Fig. 4.16, we pick the model as $l = 0.5, M = 15, m = 5$. Given an initial condition $x(0) = (0, 0.1, -0.1, 0.1, 0, 0)^\top$, we first simulate the specific control law given by (4.98) with $K_p = 5, K_d = 0.1, \varepsilon_i = 0.5, \bar{\varepsilon}_i = 0.5$. Then we simulate the decentralized optimal controller given by (4.100) with $\varepsilon_i = 0.5, \bar{\varepsilon}_i = 0.5$. The data is shown in Fig. 4.17. As Corollary 2 suggests, both simulations show exponential stability, and the disturbance vanishes on the zero dynamics surface.

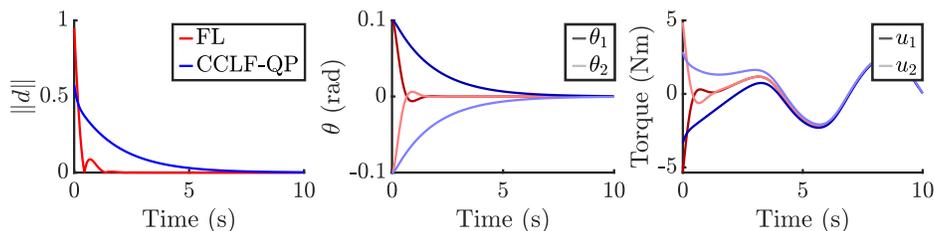


Figure 4.17: The cart-pole simulation. data labeled FL (red) used the feedback linearization controller (4.98); data labeled CLF-QP (blue) used the controller (4.100); dark and light variants of the colors are used to distinguish between the first and second pendula. Both simulations show stability.

4.7 Example 2. Quadrupedal Walking with Model-Free CLFs

We can also use local CLFs to stabilize the overall system for more complicated robots, such as quadrupedal locomotion. In this section, we will apply the local control laws to an 18-DOF quadrupedal robot (see Fig. 4.15) by viewing it as two connected bipedal robots (see Fig. 4.18). The advantage is that we simultaneously consider each subsystem's stability through the local CLFs and the feasibility conditions such as the motor torque saturation.

Note that for the gaits of interest, *walking*, *running*, and *hopping*, we have a diagonal double-support phase, and a flight phase. In the flight phase, where none of the toes touch the ground, the holonomic constraints are not required. The detailed model of multi-domain behaviors for quadrupedal robots is discussed in Sec. 4.1, and we omitted domain index for the ease of notations in this section. Note that in the hybrid system setting, it was previously shown that RES-CLFs provably stabilize

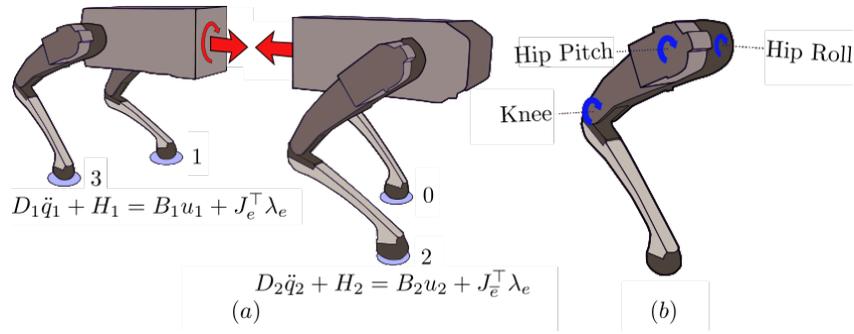


Figure 4.18: (a) The decomposition of a quadrupedal robot into two bipedal systems with toe indices labeled and (b) the configuration of the quadruped; each leg has a point contact toe and three rotational joints with motors.

the continuous dynamics in such a way that the hybrid dynamics are also stabilized under the assumption of HZD (A. Ames, Galloway, et al., 2014b, Thm. 2). This result has been extended to the ISS-CLFs in (S. Kolathaya, J. Reher, et al., 2018).

Recall the quadruped dynamics \mathcal{R}_Q that were posed as coupled bipedal dynamics in (4.48)—(4.51). We then write the coordinates for these two subsystems as $q_i = (\xi^\top, \theta_{i,st}^\top, \theta_{i,nst}^\top)^\top \in Q_i$, where the subscript st marks the joints of the leg that are in contact with the ground, and nst for the swing legs. We then define outputs for each subsystem, the bipeds, as

$$y_i(t, q_i) = y_i^a(t) - y_i^d(q_i) \quad i \in \mathcal{N} \quad (4.101)$$

where the desired outputs (trajectory) $y_i^d(t) \in \mathbb{R}^6$ are given by a set of Bézier polynomials generated by the CCS optimization in Sec. 4.3. The actual outputs are picked as

$$y_i^a(q_i) = Y_i q_i = \begin{bmatrix} \theta_{st,hr} \\ \theta_{st, hp} - \frac{\theta_{st,k}}{2} \\ \theta_{st,k} \\ \theta_{nst,hr} \\ \theta_{nst, hp} - \frac{\theta_{nst,k}}{2} \\ \theta_{nst,k} \end{bmatrix}, \quad (4.102)$$

where the subscript hr, hp, k are short for the *hip-roll*, *hip-pitch*, and *knee* joints. This output structure represents the roll angle, pitch angle, and leg length of the *virtual leg*, which is the virtual linkage connecting the hip and the toe. Note that if the quadrupedal robot has nonidentical legs for the front and rear subsystems, we will have a different output structure, i.e., $Y_f \neq Y_r$. This will be an interesting future

direction for understanding how to cooperate asymmetric quadrupeds. Next, given the full-system output Jacobian $J_y = \partial y / \partial q$, we can use (4.86) to obtain the CCS dynamics as in (4.85).

On the zero dynamics surface where both subsystems' output coordinates remain zero, we have the configuration coordinates and their velocity terms satisfying:

$$(q^Z, \dot{q}^Z) = \{(q, \dot{q}) \mid y_i(q) = \dot{y}_i(q, \dot{q}) = 0, \quad \forall i \in \mathcal{N}\}.$$

The nominal inputs that satisfy the invariant condition (4.89) can therefore be obtained by

$$u^Z(q^Z, \dot{q}^Z, t) = -\mathcal{A}(q^Z)^{-1} \mathcal{L}(q^Z, \dot{q}^Z) \triangleq \begin{bmatrix} u_1^Z \\ u_2^Z \end{bmatrix}. \quad (4.103)$$

We then can have the disturbed subsystem dynamics as given in (4.93), after which we can control each bipedal system using the coupled control Lyapunov functions.

Model-free CLF-QP. For the quadruped chosen in this study, any CLF qualifies as an ISS-CLF (Angeli, 1999). Hence, we can choose a specific form of the CLF, which is motivated by the Proportional-Derivative control law-inspired Lyapunov function S. Kolathaya and Veer, 2019, eq(24), and use it for the underactuated bipedal systems. The advantage with this class of CLFs is that the corresponding *stability constraint* can be expressed in a model-free fashion. Therefore, an improved experimental robustness against model uncertainty can be obtained. Formally, we have the following model-free stability constraint:

$$(\alpha_i(y_i)y_i^\top + \dot{y}_i^\top)(K_p y_i + K_d \dot{y}_i) + (\alpha_i(y_i)y_i^\top + \dot{y}_i^\top)J_{y_i^\Lambda}^{-\top} u_i \leq 0 \quad (4.104)$$

for subsystem $i \in \mathcal{N}$, where, $\alpha_i(y_i) = \frac{k_0}{1+|y_i|}$ with a constant $k_0 > 0$. Concretely, in comparison with (4.75), $(\alpha_i(y_i)y_i^\top + \dot{y}_i^\top)J_{y_i}^{-1}$ is in the place of $L_{g_i}V_i$ terms, and the remaining terms are in place of $L_{f_i}V_i$ terms. $K_p, K_d \geq 0$ are the diagonal matrices that form the PD gains. The Jacobian matrix of the actual output with respect to the actuated joints for the i^{th} subsystem is given by $J_{y_i^\Lambda} = \partial y_i / \partial q_i^\Lambda$, where q_i^Λ are the actuated joints of the i^{th} bipedal system. We then have the QP formulation utilizing the model-free CLFs as:

$$\begin{aligned} \underset{u_i \in \mathbb{R}^6, \delta \in \mathbb{R}}{\operatorname{argmin}} \quad & |u_i - u_i^{\text{ref}}|^2 + 1000\delta^2 \\ \text{s.t.} \quad & (\alpha_i(y_i)y_i^\top + \dot{y}_i^\top) \left[(K_p y_i + K_d \dot{y}_i) + J_{y_i^\Lambda}^{-\top} u_i \right] \leq \delta \\ & -u_{\max} \leq u_i \leq u_{\max} \end{aligned} \quad (4.105)$$

for the i^{th} subsystem, where $\delta \geq 0$ is a relaxation for better numerical stability given a high penalty weight of 1000. To formulate a model-free QP problem, we also modify the nominal inputs u_i^Z to a output-feedback PD control law,

$$u_i^{\text{ref}} = -J_{y_i^A}^\top (K_p y_i + K_d \dot{y}_i).$$

Simulation. Before enabling the proposed method on hardware, we first validated the CCLF-QP in simulation using the physics engine RaiSim (Hwangbo, J. Lee, and Marco Hutter, 2018). In particular, we wish to control this quadrupedal robot as two connected bipedal robots performing quadrupedal behaviors such as walking, hopping, and running. These behaviors can be generated as single-domain or multi-domain periodic solutions to the coupled control system using the optimization method introduced in Sec. 4.2. The specific controller we put to the test to achieve stable tracking of the desinated periodic gaits is given in (4.105). The PD gains $K_p, K_d \geq 0$ are diagonal matrices and are picked as the same value across all three simulation tests. As a result, the local controller utilizing CCLFs renders stabilization of the given periodic gaits for walking, hopping, and running on Vision 60 in RaiSim. An animation is provided in (Wen-Loong Ma, 2021). We show the gait tiles in Fig. 4.19, and the phase portraits of the simulation data in Fig. 4.20.



Figure 4.19: Top: Snapshots showing a two full steps of the walking gait on an outdoor lawn. Bottom: Running gait in the RaiSim simulation environment.

Hardware. The robot we studied in this paper is the Vision 60 v3.9 quadrupedal robot from Ghost Robotics. As show in Fig. 4.15, this robot is 44 kg, 54 cm wide and 50~60 cm tall. It uses a hierarchical computation structure to perform various tasks. In our experiments, we implement the optimal controller with a QP solver *OSQP* on the onboard Jetson AGX Xavier computer from NVIDIA. Furthermore, a 1kHz hard real-time operating system enforces the communication between the mainboard and motor drivers to realize the torque commands from the control algorithm (4.105).

Experiments and Data Analysis. As a first step towards controlling complex systems such as quadrupedal robots to achieve various dynamical behaviors using the

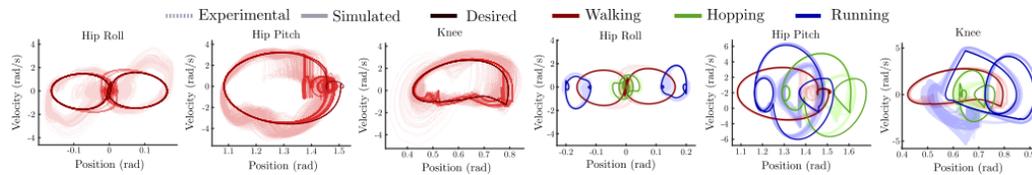


Figure 4.20: Left three: Experimental (dashed transparent) and simulated (solid transparent) phase portraits for walking plotted against the desired values (solid). Right three: Simulated (transparent) vs. desired (solid) phase portraits for walking (red), hopping (green), and running (blue) behaviors.

local control laws, we conducted some walking experiments on the Vision 60 robot. To avoid robustness challenges caused by model uncertainties, especially unpredictable uncertainties introduced by the terrain dynamics, we applied the model-free QPs in (4.105). As the supplementary video (Wen-Loong Ma, 2021) shows, we are able to achieve robust walking with the Vision 60 on rough outdoor terrains with moderate slope variation and surface roots. We show the gait tiles of the walking experiments in Fig. 4.19. We also provide a comparison between experimental data, simulation data, and the desired trajectory in Fig. 4.20. We note that the tracking is ultimately bounded by a tube around the desired trajectory in the continuous domains, which provides empirical evidence for future works formally establishing hybrid stability.

CONCLUSION AND FUTURE WORK

A complete procedure of modeling, optimizing, controller design, and experiments of complex legged robots was shown in this work. The methodology was generalized around the concept of Lyapunov stability. To conclude, we will summarize the main contributions and discuss some interesting future directions.

5.1 Summary

Trajectory optimization for complex locomotion. Trajectory optimization was originally widely used in computational fluid dynamics and often was used with inverse kinematics and searching algorithms when it was first introduced to robotics. From a dynamics perspective, off-line trajectory optimization is used to find solutions to dynamical systems. In this dissertation, we first formally define the solution to a hybrid system, which represents the dynamics of the legged systems ranging from bipedal to quadrupedal systems. We then formulated the solution-finding problem into open-loop and closed-loop trajectory optimization problems. In various applications, these methods have been scaled from simple motions to multiple dynamic locomotions on over ten robotic systems, including monopedal, bipedal, and quadrupedal systems, and we have shown high-performance behaviors such as bipedal running, rough terrain locomotion, and high-energy efficiency walking. The applicability has been successfully validated.

Robust high-performance locomotion. In a perfect world, the off-line planning with a predefined feedback controller should drive the system to the desired behavior since we possess the principal information of the dynamics. But this is not true. Even with a slight difference in the measurement or the modeling, a carefully crafted controller could fail. While there are many types of uncertainty, and each has a different worst-case scenario to destroy stability, we used experimental research to find the most critical uncertainty, phase-time uncertainty, to bipedal running. Building on the top of the hybrid zero dynamics framework, we used the concept of input-to-state stability to describe the phase uncertainty to state stability for bipedal locomotion. This new characterization of stability has provided some slack space for parameter tuning while guaranteeing dynamic stability formally. Eventually, the synthesized

control laws yield bipedal running on a human-size robot, and it is repeatable and robust.

Coupled Control System. Because of bipedal systems' highly dynamic nature and instability from underactuation, understanding and controlling the full-order dynamics appears necessary. When we carry this philosophy from bipedal systems to quadrupedal robots, the computational cost has increased due to the more complicated contact scenario and hybrid system structure. We considered a quadruped as two connected bipedal systems, and design motion and controller using those methods applied to the bipedal system to address this problem. This intuitive approach was then formally characterized as the framework of the coupled control system. In this methodology, we can design trajectory and controller for each subsystem while also formally guaranteeing the original system's stability and preserving its solutions. This new way of controlling robotic systems drove two quadrupedal robots to walk on multiple outdoor terrains.

5.2 Future Work

The development of this thesis originates from the methodology:

A_1 offline prediction/planning/optimization + A_2 online/real-time control/tracking, where A_1 searches for a plan that is globally or locally optimal, and A_2 executes the tracking problem in implementation and rejects unplanned problems. For example, we put all of the details (motor, compliance) of the running robot, DURUS, into our model, which has made the dynamical system high-dimensional, stiff, and hard to solve. But because of this, we can confidently derive a plan from it that can represent "most" of the reality. The HZD framework and CCS methods both are to reduce the stability analysis for this complex problem so that it is feasible for field implementation. Then we can use the control Lyapunov functions based QP to reject small perturbation with the smallest necessary control effort.

An important future direction is to merge these two components. This can be from a model predictive control (MPC) perspective. We can use more advanced computational tools to accelerate trajectory optimization in both algorithmic software and hardware. The optimal control component can "look further" into the future with a more precise model. We can also rigorously synthesize a computationally cheap model for a well-defined task (the SLIP model for lightweight legs on flat ground is a good example). Besides, many theorems in nonlinear control assume a given Lyapunov function. While this is a restrictive assumption, many unexplored Lyapunov

candidates in existing robotic literature and computational Lyapunov functions are also compelling. Understanding these methods of control Lyapunov functions for locomotion will be an important future direction. Additionally, the coupled control system framework looks at the distributed/decentralized/parallel/networked computation/control problem from a dynamics perspective. New directions include expanding these methods to more physically connected agents collaboration and incorporating more biped-related control methods in subsystem design. Lastly, collaboratively carrying a payload that is more than each subsystem's capacity is becoming necessary for disaster response and extreme environment autonomy. Further development of CCS can bridge this gap between high-performance single robot maneuver and multi-agent collaboration.

BIBLIOGRAPHY

- Aëšström, Karl J. and P.R. Kumar (2014). “Control: A perspective”. In: *Automatica* 50.1, pp. 3–43. ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2013.10.012>. URL: <http://www.sciencedirect.com/science/article/pii/S0005109813005037>.
- Ahmadi, Mojtaba and Martin Buehler (2006). “Controlled Passive Dynamic Running Experiments With the ARL-Monopod II”. In: *IEEE Trans. on Robotics* 22.5, pp. 974–986.
- Akbari Hamed, K., B.G. Buss, and J.W. Grizzle (2016). “Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations”. In: *The International Journal of Robotics Research* 35.8, pp. 977–999.
- Akbari Hamed, Kaveh., Wen-Loong. Ma, and Aaron. D. Ames (July 2019). “Dynamically Stable 3D Quadrupedal Walking with Multi-Domain Hybrid System Models and Virtual Constraint Controllers”. In: *2019 American Control Conference (ACC)*, pp. 4588–4595. DOI: 10.23919/ACC.2019.8815085.
- Ambrose, E., W. Ma, and A. D. Ames (2021). “Towards the Unification of System Design and Motion Synthesis for High-Performance Hopping Robots”. In: *International Conference on Robotics and Automation (ICRA)*.
- Ambrose, E., W. Ma, C. Hubicki, et al. (2017). “Toward benchmarking locomotion economy across design configurations on the modular robot: AMBER-3M”. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1270–1276. DOI: 10.1109/CCTA.2017.8062633.
- Ames, A. D. (May 2014). “Human-Inspired Control of Bipedal Walking Robots”. In: *IEEE Transactions on Automatic Control* 59.5, pp. 1115–1130. ISSN: 0018-9286. DOI: 10.1109/TAC.2014.2299342.
- (May 2014). “Human-Inspired Control of Bipedal Walking Robots”. In: *Automatic Control, IEEE Transactions on* 59.5, pp. 1115–1130. ISSN: 0018-9286. DOI: 10.1109/TAC.2014.2299342.
- Ames, A.D., K. Galloway, et al. (Apr. 2014a). “Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics”. In: *Automatic Control, IEEE Transactions on*. ISSN: 0018-9286. DOI: 10.1109/TAC.2014.2299335.
- (2014b). “Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics”. In: *Automatic Control, IEEE Transactions on* 59.4, pp. 876–891. ISSN: 0018-9286. DOI: 10.1109/TAC.2014.2299335.

- Ames, Aaron D. et al. (2017). “First steps toward formal controller synthesis for bipedal robots with experimental implementation”. In: *Nonlinear Analysis: Hybrid Systems* 25, pp. 155–173. ISSN: 1751-570X. DOI: <https://doi.org/10.1016/j.nahs.2017.01.002>. URL: <http://www.sciencedirect.com/science/article/pii/S1751570X1730002X>.
- Angeli, David (1999). “Input-to-state stability of PD-controlled robotic systems”. In: *Automatica* 35.7, pp. 1285–1290. ISSN: 0005-1098. DOI: [http://dx.doi.org/10.1016/S0005-1098\(99\)00037-0](http://dx.doi.org/10.1016/S0005-1098(99)00037-0).
- Antonelli, Gianluca (2013). “Interconnected dynamic systems: An overview on distributed control”. In: *IEEE Control Systems Magazine* 33.1, pp. 76–88.
- Arnold, V.I. (1989). *Mathematical Methods of Classical Mechanics*. 2nd ed. 60. Springer-Verlag New York.
- Ascher, Uri M. and Linda R. Petzold (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. 1st. USA: Society for Industrial and Applied Mathematics. ISBN: 0898714125.
- Barton, Paul I. and Cha Kun Lee (Oct. 2002). “Modeling, Simulation, Sensitivity Analysis, and Optimization of Hybrid Systems”. In: *ACM Trans. Model. Comput. Simul.* 12.4, pp. 256–289. ISSN: 1049-3301. DOI: [10.1145/643120.643122](https://doi.org/10.1145/643120.643122). URL: <http://doi.acm.org/10.1145/643120.643122>.
- Becerra, V. M. (2010). “Solving complex optimal control problems at no cost with PSOPT”. In: *2010 IEEE International Symposium on Computer-Aided Control System Design*, pp. 1391–1396. DOI: [10.1109/CACSD.2010.5612676](https://doi.org/10.1109/CACSD.2010.5612676).
- Betts, John T. (1998). “Survey of Numerical Methods for Trajectory Optimization”. In: *Journal of Guidance, Control, and Dynamics* 21.2, pp. 193–207. DOI: [10.2514/2.4231](https://doi.org/10.2514/2.4231). eprint: <https://doi.org/10.2514/2.4231>. URL: <https://doi.org/10.2514/2.4231>.
- Bhounsule, Pranav A. et al. (2014). “Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge”. In: *The International Journal of Robotics Research* 33.10, pp. 1305–1321. DOI: [10.1177/0278364914527485](https://doi.org/10.1177/0278364914527485). URL: <http://ijr.sagepub.com/content/33/10/1305.short>.
- Big Dog Walking* (2010). Boston Dynamics. URL: <https://youtu.be/cNZPRsrwumQ>.
- Bledt, G. et al. (Oct. 2018). “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252. DOI: [10.1109/IROS.2018.8593885](https://doi.org/10.1109/IROS.2018.8593885).
- Blickhan, R. (n.d.). “The spring–mass model for running and hopping”. In: *Journal of Biomechanics* 22.11 (), pp. 1217–1227.
- Blickhan, Reinhard and RJ Full (1993). “Similarity in multilegged locomotion: bouncing like a monopode”. In: *Journal of Comparative Physiology A* 173.5, pp. 509–517.

- Boussema, C. et al. (Apr. 2019a). “Online Gait Transitions and Disturbance Recovery for Legged Robots via the Feasible Impulse Set”. In: *IEEE Robotics and Automation Letters* 4.2, pp. 1611–1618. ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2896723.
- (Apr. 2019b). “Online Gait Transitions and Disturbance Recovery for Legged Robots via the Feasible Impulse Set”. In: *IEEE Robotics and Automation Letters* 4.2, pp. 1611–1618. ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2896723.
- Box, George E. P. (1976). “Science and Statistics”. In: *Journal of the American Statistical Association* 71.356, pp. 791–799. DOI: 10.1080/01621459.1976.10480949. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1976.10480949>. URL: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1976.10480949>.
- Champneys, Alan R and Péter L Várkonyi (2016). “The Painlevé paradox in contact mechanics”. In: *IMA Journal of Applied Mathematics* 81.3, pp. 538–588.
- Chang, A. H. et al. (2017). “Learning to jump in granular media: Unifying optimal control synthesis with Gaussian process-based regression”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2154–2160. DOI: 10.1109/ICRA.2017.7989248.
- Chang, Tai-Heng and Yildirim Hurmuzlu (1993). “Sliding control without reaching phase and its application to bipedal locomotion”. In: *Journal of Dynamic Systems, Measurement, and Control* 115.3, pp. 447–455.
- Chevallereau, C., J.W. Grizzle, and C.-L. Shih (Feb. 2009). “Asymptotically Stable Walking of a Five-Link Underactuated 3-D Bipedal Robot”. In: *Robotics, IEEE Transactions on* 25.1, pp. 37–50. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2010366.
- Chung, Soon-Jo and Jean-Jacques E Slotine (2009). “Cooperative robot control and concurrent synchronization of Lagrangian systems”. In: *IEEE transactions on Robotics* 25.3, pp. 686–700.
- Clark, Andrew J and Timothy E Higham (2011). “Slipping, sliding and stability: locomotor strategies for overcoming low-friction surfaces”. In: *Journal of Experimental Biology* 214.8, pp. 1369–1378.
- Collins, S. et al. (2005a). “Efficient Bipedal Robots Based on Passive-Dynamic Walkers”. In: *Science* 307.5712, pp. 1082–1085. DOI: 10.1126/science.1107799. eprint: <http://www.sciencemag.org/content/307/5712/1082.full.pdf>.
- (2005b). “Efficient bipedal robots based on passive-dynamic walkers”. In: *Science* 307.5712, pp. 1082–1085.

- Cousineau, E. and A. D. Ames (May 2015). “Realizing underactuated bipedal walking with torque controllers via the ideal model resolved motion method”. In: *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 5747–5753. DOI: 10.1109/ICRA.2015.7140004.
- Da, X. et al. (2016). “From 2D Design of Underactuated Bipedal Gaits to 3D Implementation: Walking With Speed Tracking”. In: *IEEE Access* 4, pp. 3469–3478. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2582731.
- Dai, H. and R. Tedrake (Dec. 2012). “Optimizing robust limit cycles for legged locomotion on unknown terrain”. In: *Decision and Control, IEEE 51st Annual Conference on*, pp. 1207–1213. DOI: 10.1109/CDC.2012.6425971.
- Dai, Hongkai, Andres Valenzuela, and Russ Tedrake (2014). “Whole-body Motion Planning with Simple Dynamics and Full Kinematics”. In: *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. ISBN: 9781479971732. DOI: 10.1109/HUMANOIDS.2014.7041375.
- Dalibard, Sébastien et al. (2013). “Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach”. In: *The International Journal of Robotics Research* 32.9-10, pp. 1089–1103. DOI: 10.1177/0278364913481250. eprint: <https://doi.org/10.1177/0278364913481250>. URL: <https://doi.org/10.1177/0278364913481250>.
- Danner, Simon M et al. (Nov. 2017). “Computational modeling of spinal circuits controlling limb coordination and gaits in quadrupeds”. In: *eLife* 6. Ed. by Ronald L Calabrese, e31050. ISSN: 2050-084X. DOI: 10.7554/eLife.31050. URL: <https://doi.org/10.7554/eLife.31050>.
- De, Avik and Daniel E. Koditschek (2018). “Vertical hopper compositions for preflexive and feedback-stabilized quadrupedal bounding, pacing, pronking, and trotting”. In: *The International Journal of Robotics Research* 37.7, pp. 743–778. DOI: 10.1177/0278364918779874. eprint: <https://doi.org/10.1177/0278364918779874>. URL: <https://doi.org/10.1177/0278364918779874>.
- Di Carlo, J. et al. (Oct. 2018). “Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9. DOI: 10.1109/IROS.2018.8594448.
- Dörfler, Florian and Francesco Bullo (2014). “Synchronization in complex networks of phase oscillators: A survey”. In: *Automatica* 50.6, pp. 1539–1564. ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2014.04.012>. URL: <http://www.sciencedirect.com/science/article/pii/S0005109814001423>.
- Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Kluwer international series in engineering and computer science: Robotics. Springer. ISBN: 9780387743158. URL: <http://books.google.com/books?id=UjWbvqWaf6gC>.

- Featherstone, Roy (2010). “Exploiting Sparsity in Operational-space Dynamics”. In: *The International Journal of Robotics Research* 29.10, pp. 1353–1368. DOI: 10.1177/0278364909357644. eprint: <https://doi.org/10.1177/0278364909357644>. URL: <https://doi.org/10.1177/0278364909357644>.
- Fujisaka, Hirokazu and Tomoji Yamada (1983). “Stability theory of synchronized motion in coupled-oscillator systems”. In: *Progress of theoretical physics* 69.1, pp. 32–47.
- Gamus, Benny and Yizhar Or (2013). “Analysis of dynamic bipedal robot walking with stick-slip transitions.” In: *ICRA*, pp. 3348–3355.
- (2015). “Dynamic bipedal walking under stick-slip transitions”. In: *SIAM Journal on Applied Dynamical Systems* 14.2, pp. 609–642.
- Ganesh, Sumitra, Aaron D Ames, and Ruzena Bajcsy (2007). “Composition of dynamical systems for estimation of human body dynamics”. In: *International Workshop on Hybrid Systems: Computation and Control*. Springer, pp. 702–705.
- Geijtenbeek, Thomas, Michiel van de Panne, and a. Frank van der Stappen (Nov. 2013). “Flexible muscle-based locomotion for bipedal creatures”. In: *ACM Transactions on Graphics* 32.6, pp. 1–11. ISSN: 07300301. DOI: 10.1145/2508363.2508399.
- Gill, Philip E., Walter Murray, and Michael A. Saunders (2005). “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization”. In: *SIAM Review* 47.1, pp. 99–131. DOI: 10.1137/S0036144504446096. eprint: <https://doi.org/10.1137/S0036144504446096>. URL: <https://doi.org/10.1137/S0036144504446096>.
- Gregg, R.D. and L. Righetti (Oct. 2013). “Controlled Reduction With Unactuated Cyclic Variables: Application to 3D Bipedal Walking With Passive Yaw Rotation”. In: *Automatic Control, IEEE Transactions on* 58.10, pp. 2679–2685. ISSN: 0018-9286. DOI: 10.1109/TAC.2013.2256011.
- Griewank, Andreas, David Juedes, and Jean Utke (June 1996). “Algorithm 755: ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++”. In: *ACM Trans. Math. Softw.* 22.2, pp. 131–167. ISSN: 0098-3500. DOI: 10.1145/229473.229474. URL: <https://doi.org/10.1145/229473.229474>.
- Grizzle, J W, G Abba, and F Plestan (Jan. 2001). “Asymptotically Stable Walking for Biped Robots: Analysis via Systems with Impulse Effects”. In: *IEEE Trans. on Automatic Control* 46.1, pp. 51–64.
- (Jan. 2001). “Asymptotically stable walking for biped robots: Analysis via systems with impulse effects”. In: *Automatic Control, IEEE Transactions on* 46.1, pp. 51–64. ISSN: 0018-9286. DOI: 10.1109/9.898695.

- Grizzle, Jessy W. et al. (2014). “Models, feedback control, and open problems of 3D bipedal robotic walking”. In: *Automatica* 50.8, pp. 1955–1988. ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2014.04.021>. URL: <http://www.sciencedirect.com/science/article/pii/S0005109814001654>.
- Gruhn, Matthias, Lyuba Zehl, and Ansgar Büschges (2009). “Straight walking and turning on a slippery surface”. In: *Journal of Experimental Biology* 212.2, pp. 194–209.
- Hager, William W. et al. (Sept. 2019). “Convergence Rate for a Radau Hp Collocation Method Applied to Constrained Optimal Control”. In: *Comput. Optim. Appl.* 74.1, pp. 275–314. ISSN: 0926-6003. DOI: 10.1007/s10589-019-00100-1. URL: <https://doi.org/10.1007/s10589-019-00100-1>.
- Hamed, Kaveh Akbari and Jessy W. Grizzle (June 2013). “Robust event-based stabilization of periodic orbits for hybrid systems: Application to an underactuated 3D bipedal robot”. In: *American Control Conference (ACC), 2013*, pp. 6206–6212. DOI: 10.1109/ACC.2013.6580811.
- Hamed, Kaveh Akbari, Vinay R. Kamidi, et al. (2020). “Distributed Feedback Controllers for Stable Cooperative Locomotion of Quadrupedal Robots: A Virtual Constraint Approach”. In: *American Control Conference*. arXiv: 1902.03690 [math.OC].
- Hatanaka, Takeshi et al. (2015). *Passivity-Based Control and Estimation in Networked Robotics*. Springer International Publishing.
- Hauser, John and Chung Choo Chung (1994). “Converse Lyapunov functions for exponentially stable periodic orbits”. In: *Systems & Control Letters* 23.1, pp. 27–34. ISSN: 0167-6911. DOI: [https://doi.org/10.1016/0167-6911\(94\)90078-7](https://doi.org/10.1016/0167-6911(94)90078-7).
- Hereid, A. et al. (2018). “Dynamic Humanoid Locomotion: A Scalable Formulation for HZD Gait Optimization”. In: *IEEE Transactions on Robotics*, pp. 1–18. ISSN: 1552-3098. DOI: 10.1109/TR0.2017.2783371.
- Hereid, Ayonga and Aaron D. Ames (Sept. 2017). “FROST: Fast Robot Optimization and Simulation Toolkit”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ. Vancouver, BC, Canada.
- Hereid, Ayonga, Eric. Cousineau, et al. (2016). “3D Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics”. In: *IEEE International Conference on Robotics and Automation*.
- Hereid, Ayonga, Christian M Hubicki, et al. (2018). “Dynamic Humanoid Locomotion: A Scalable Formulation for HZD Gait Optimization”. In: *IEEE Transactions on Robotics*.

- Hereid, Ayonga, Shishir Kolathaya, et al. (2014). “Dynamic Multi-domain Bipedal Walking with Atrias Through SLIP Based Human-inspired Control”. In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*. HSCC '14. Berlin, Germany: ACM, pp. 263–272. ISBN: 978-1-4503-2732-9. DOI: [10.1145/2562059.2562143](https://doi.org/10.1145/2562059.2562143). URL: <http://doi.acm.org/10.1145/2562059.2562143>.
- Hof, A.L., M.G.J. Gazendam, and W.E. Sinke (2005). “The condition for dynamic stability”. In: *Journal of Biomechanics* 38.1, pp. 1–8. ISSN: 0021-9290. DOI: <https://doi.org/10.1016/j.jbiomech.2004.03.025>. URL: <http://www.sciencedirect.com/science/article/pii/S0021929004001642>.
- Hutter, M. et al. (Oct. 2016). “ANYmal - a highly mobile and dynamic quadrupedal robot”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. DOI: [10.1109/IROS.2016.7758092](https://doi.org/10.1109/IROS.2016.7758092).
- Hwangbo, Jemin, Joonho Lee, and Marco Hutter (2018). “Per-contact iteration method for solving contact dynamics”. In: *IEEE Robotics and Automation Letters* 3.2, pp. 895–902.
- Isidori, A. (1995). *Nonlinear Control Systems*. Springer; 3rd edition.
- (2011). “The zero dynamics of a nonlinear system: From The Origin To the latest progresses of a long successful story”. In: *Proceedings of the 30th Chinese Control Conference*, pp. 18–25.
- Jiang, Zhong-Ping, Iven Mareels, and Yuan Wang (1995). “A Lyapunov Formulation of Nonlinear Small Gain Theorem for Interconnected Systems”. In: *IFAC Proceedings Volumes* 28.14. 3rd IFAC Symposium on Nonlinear Control Systems Design 1995, Tahoe City, CA, USA, 25-28 June 1995, pp. 625–630. ISSN: 1474-6670. DOI: [https://doi.org/10.1016/S1474-6670\(17\)46898-4](https://doi.org/10.1016/S1474-6670(17)46898-4). URL: <http://www.sciencedirect.com/science/article/pii/S1474667017468984>.
- Johansen, Tor A. and Thor I. Fossen (2013). “Control allocation—A survey”. In: *Automatica* 49.5, pp. 1087–1103. ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2013.01.035>.
- Johnson, A. M., S. A. Burden, and D. E. Koditschek (2016). “A hybrid systems model for simple manipulation and self-manipulation systems”. In: *The International Journal of Robotics Research* 35.11, pp. 1354–1392. DOI: [10.1177/0278364916639380](https://doi.org/10.1177/0278364916639380). eprint: <https://doi.org/10.1177/0278364916639380>. URL: <https://doi.org/10.1177/0278364916639380>.
- Kajita, S., F. Kanehiro, et al. (2001). “The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation”. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. Vol. 1, 239–246 vol.1.

- Kajita, S., K. Tani, and A. Kobayashi (July 1990). “Dynamic walk control of a biped robot along the potential energy conserving orbit”. In: *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, 789–794 vol.2. DOI: 10.1109/IROS.1990.262497.
- Kajita, Shuuji et al. (2004). “Biped walking on a low friction floor”. In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 4. IEEE, pp. 3546–3552.
- Kaneko, Kenji et al. (2005). “Slip observer for walking on a low friction floor”. In: *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, pp. 634–640.
- Kelly, M. (2017). “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation”. In: *SIAM Review* 59.4, pp. 849–904. DOI: 10.1137/16M1062569. eprint: <https://doi.org/10.1137/16M1062569>. URL: <https://doi.org/10.1137/16M1062569>.
- Khalil, Hassan K (2002). *Nonlinear systems*. Vol. 3. Prentice hall Upper Saddle River.
- Kiehl, M. (1994). “Parallel multiple shooting for the solution of initial value problems”. In: *Parallel Computing* 20.3, pp. 275–295. ISSN: 0167-8191. DOI: [https://doi.org/10.1016/S0167-8191\(06\)80013-X](https://doi.org/10.1016/S0167-8191(06)80013-X). URL: <https://www.sciencedirect.com/science/article/pii/S016781910680013X>.
- Kolathaya, S., A. Hereid, and A. D. Ames (July 2016). “Time dependent control Lyapunov functions and hybrid zero dynamics for stable robotic locomotion”. In: *2016 American Control Conference (ACC)*, pp. 3916–3921. DOI: 10.1109/ACC.2016.7525524.
- Kolathaya, S., J. Reher, et al. (June 2018). “Input to State Stabilizing Control Lyapunov Functions for Robust Bipedal Robotic Locomotion”. In: *2018 Annual American Control Conference (ACC)*, pp. 2224–2230. DOI: 10.23919/ACC.2018.8430946.
- Kolathaya, S. and S. Veer (2019). “PD based Robust Quadratic Programs for Robotic Systems”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6834–6841.
- Kolathaya, Shishir and D Aaron Ames (2016). “Parameter to State Stability of Control Lyapunov Functions for Hybrid System Models of Robots”. In: *Nonlinear Analysis Hybrid Systems*. Elsevier.
- Kolathaya, Shishir, Ayonga Hereid, and Aaron D Ames (2016). “Time Dependent Control Lyapunov Functions and Hybrid Zero Dynamics for Stable Robotic Locomotion”. In: *American Control Conference (ACC)*. IEEE.
- Kolathaya, Shishir, Wen-Loong Ma, and Aaron D Ames (2015). “Composing Dynamical Systems to Realize Dynamic Robotic Dancing”. In: *Algorithmic Foundations of Robotics XI*. Springer, Cham, pp. 425–442.

- Koolen, T. et al. (2012). “Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models”. In: *The International Journal of Robotics Research*. DOI: 10.1177/0278364912452673. eprint: <http://ijr.sagepub.com/content/early/2012/06/29/0278364912452673.full.pdf+html>. URL: <http://ijr.sagepub.com/content/early/2012/06/29/0278364912452673.abstract>.
- Lanczos, C. (1986). *The Variational Principles of Mechanics*. Dover Books On Physics. Dover Publications. ISBN: 9780486650678.
- Li, Chen, Tingnan Zhang, and Daniel I. Goldman (2013). “A Terradynamics of Legged Locomotion on Granular Media”. In: *Science* 339.6126, pp. 1408–1412. ISSN: 0036-8075. DOI: 10.1126/science.1229163. eprint: <https://science.sciencemag.org/content/339/6126/1408.full.pdf>. URL: <https://science.sciencemag.org/content/339/6126/1408>.
- Ma, W., N. Csomay-Shanklin, and A. D. Ames (2021). “Coupled Control Systems: Periodic Orbit Generation With Application to Quadrupedal Locomotion”. In: *IEEE Control Systems Letters* 5.3, pp. 935–940. DOI: 10.1109/LCSYS.2020.3006963.
- Ma, W., K. A. Hamed, and A. D. Ames (2019). “First Steps Towards Full Model Based Motion Planning and Control of Quadrupeds: A Hybrid Zero Dynamics Approach”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5498–5503. DOI: 10.1109/IROS40897.2019.8968189.
- Ma, W., Y. Or, and A. D. Ames (2019). “Dynamic Walking on Slippery Surfaces : Demonstrating Stable Bipedal Gaits with Planned Ground Slippage*”. In: *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3705–3711. DOI: 10.1109/ICRA.2019.8793761.
- Ma, W. -L., N. Csomay-Shanklin, and A. D. Ames (2020). “Quadrupedal Robotic Walking on Sloped Terrains via Exact Decomposition into Coupled Bipedal Robots”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4006–4011. DOI: 10.1109/IROS45743.2020.9341181.
- Ma, W. -L., N. Csomay-Shanklin, S. Kolathaya, et al. (2021). “Coupled Control Lyapunov Functions for Interconnected Systems, With Application to Quadrupedal Locomotion”. In: *IEEE Robotics and Automation Letters* 6.2, pp. 3761–3768. DOI: 10.1109/LRA.2021.3065174.
- Ma, Wen-Loong (2016a). *Bipedal robotic running of DURUS-2D*. Caltech. URL: https://youtu.be/6k9_tf4ctk4.
- (2016b). *DURUS running*. Caltech. URL: <https://youtu.be/7uQ0Y3reavo>.
- (2016c). *Failed time-based running*. Caltech. URL: <https://youtu.be/RrqeVDjB7ec>.
- (2019a). *Bipedal walking on slippery surfaces*. Caltech. URL: <https://youtu.be/G9dhcgycyI>.

- Ma, Wen-Loong (2019b). *Quadrupedal Robot Ambling Experiment*. Caltech. URL: <https://youtu.be/MQCPOKCop8Q>.
- (2020a). *Experimental video of Vision 60 using the composed stepping gaits*. Caltech. URL: <https://youtu.be/dn2k103VX1Y>.
 - (2020b). *Experimental video of Vision 60 walking on slopes*. Caltech. URL: <https://youtu.be/uJedboyzDjc>.
 - (2020c). *Generating gaits using CCS-OPT*. Caltech. URL: <https://youtu.be/GlpgSXMinoU>.
 - (2021). *CCLF-QP controlled studies*. Caltech. URL: <https://youtu.be/D0LWc7VFx1M>.
- Ma, Wen-Loong and Aaron D. Ames (2020). “From Bipedal Walking to Quadrupedal Locomotion: Full-Body Dynamics Decomposition for Rapid Gait Generation”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4491–4497. DOI: 10.1109/ICRA40945.2020.9196841.
- Ma, Wen-Loong, Ayonga Hereid, et al. (2016). “Efficient HZD gait generation for three-dimensional underactuated humanoid running”. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, pp. 5819–5825.
- Ma, Wen-Loong, Shishir Kolathaya, et al. (2017). “Bipedal Robotic Running with DURUS-2D: Bridging the Gap Between Theory and Experiment”. In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. HSCC '17. Pittsburgh, Pennsylvania, USA: ACM, pp. 265–274. ISBN: 978-1-4503-4590-3. DOI: 10.1145/3049797.3049823. URL: <http://doi.acm.org/10.1145/3049797.3049823>.
- Manchester, I. R. et al. (Jan. 2011). “Stable dynamic walking over uneven terrain”. In: *The International Journal of Robotics Research* 30.3, pp. 265–279. ISSN: 0278-3649. DOI: 10.1177/0278364910395339. URL: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364910395339>.
- Martin, A. E., D. C. Post, and J. P. Schmiedeler (2014a). “Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet”. In: *The International Journal of Robotics Research* 33.7, pp. 988–1005. ISSN: 0278-3649. DOI: 10.1177/0278364914522141. URL: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364914522141>.
- (2014b). “The effects of foot geometric properties on the gait of planar bipeds walking under HZD-based control”. In: *The International Journal of Robotics Research* 33.12, pp. 1530–1543. DOI: 10.1177/0278364914532391.
- Martin, William C, Albert Wu, and Hartmut Geyer (2015). “Robust Spring Mass Model Running for a Physical Bipedal Robot”. In: *International Conference on Robotics and Automation*, pp. 6307–6312.

- Mellinger, Daniel et al. (2013). “Cooperative Grasping and Transport Using Multiple Quadrotors”. In: *Distributed Autonomous Robotic Systems: The 10th International Symposium*. Ed. by Alcherio Martinoli et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 545–558. ISBN: 978-3-642-32723-0. DOI: 10.1007/978-3-642-32723-0_39.
- Mesbahi, Mehran and Magnus Egerstedt (2010). *Graph theoretic methods in multi-agent networks*. Vol. 33. Princeton University Press.
- Mombaur, Katja (2009). “Using optimization to create self-stable human-like running”. In: *Robotica* 27.03, pp. 321–330.
- Morris, B. and J. W. Grizzle (Dec. 2005). “A Restricted Poincaré Map for Determining Exponentially Stable Periodic Orbits in Systems with Impulse Effects: Application to Bipedal Robots”. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 4199–4206. DOI: 10.1109/CDC.2005.1582821.
- Moyer, BE et al. (2006). “Gait parameters as predictors of slip severity in younger and older adults”. In: *Ergonomics* 49.4, pp. 329–343.
- Murray, Richard M et al. (1994). *A mathematical introduction to robotic manipulation*. CRC press.
- Muybridge, Eadweard (n.d.). *Animals in Motion*. London:Chapman and Hall, LD. URL: <https://ecommons.cornell.edu/handle/1813/58718>.
- Or, Yizhar (2014). “Painlevé’s paradox and dynamic jamming in simple models of passive dynamic walking”. In: *Regular and Chaotic Dynamics* 19.1, pp. 64–80.
- Or, Yizhar and Moti Moravia (2016). “Analysis of foot slippage effects on an actuated spring-mass model of dynamic legged locomotion”. In: *International Journal of Advanced Robotic Systems* 13.2, p. 69.
- Orin, David E., Ambarish Goswami, and Sung-Hee Lee (2013). *Centroidal dynamics of a humanoid robot*. Vol. 35. Springer, pp. 161–176.
- Park, H.-W., A. Ramezani, and J.W. Grizzle (Apr. 2013). “A Finite-State Machine for Accommodating Unexpected Large Ground-Height Variations in Bipedal Robot Walking”. In: *Robotics, IEEE Transactions on* 29.2, pp. 331–345. ISSN: 1552-3098. DOI: 10.1109/TR0.2012.2230992.
- Park, HW et al. (May 2012). “Switching control design for accommodating large step-down disturbances in bipedal robot walking”. In: *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*. Ieee, pp. 45–50. ISBN: 978-1-4673-1405-3. DOI: 10.1109/ICRA.2012.6225056. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6225056>
http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=6225056.

- Poulakakis, I. and J.W. Grizzle (Aug. 2009). “The Spring Loaded Inverted Pendulum as the Hybrid Zero Dynamics of an Asymmetric Hopper”. In: *Automatic Control, IEEE Transactions on* 54.8, pp. 1779–1793. ISSN: 0018-9286. DOI: 10.1109/TAC.2009.2024565.
- Poulakakis, Ioannis and Jessy W Grizzle (2009). “The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper”. In: *Automatic Control, IEEE Transactions on* 54.8, pp. 1779–1793.
- Poulakakis, Ioannis and JW Grizzle (2007). “Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper”. In: *Proc. of the European Control Conference*, pp. 1–8.
- Powell, Matthew J and Aaron D Ames (2014). “Hierarchical control of series elastic actuators through control Lyapunov functions”. In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, pp. 2986–2992.
- Raibert, M., K. Blankespoor, et al. (2008). “BigDog, the Rough-Terrain Quadruped Robot”. In: *IFAC Proceedings Volumes* 41.2. 17th IFAC World Congress, pp. 10822–10825. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20080706-5-KR-1001.01833>. URL: <http://www.sciencedirect.com/science/article/pii/S1474667016407020>.
- Raibert, M. and E. R. Tello (Nov. 1986). “Legged Robots That Balance”. In: *IEEE Expert* 1.4, pp. 89–89. ISSN: 0885-9000. DOI: 10.1109/MEX.1986.4307016.
- Raibert, M. H. et al. (n.d.). “Dynamically Stable Legged Locomotion (September 1985-September 1989)”. In: (), pp. 49–77.
- Raibert, Marc H., H. Benjamin Brown, and Michael Chepponis (1984). “Experiments in Balance with a 3D One-Legged Hopping Machine”. In: *The International Journal of Robotics Research* 3.2, pp. 75–92. DOI: 10.1177/027836498400300207. eprint: <http://ijr.sagepub.com/content/3/2/75.full.pdf+html>. URL: <http://ijr.sagepub.com/content/3/2/75.abstract>.
- Rao, A. V. (2009). “A survey of numerical methods for optimal control”. In: *Advances in the Astronautical Sciences*.
- Reher, J. et al. (May 2016). “Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1794–1801. DOI: 10.1109/ICRA.2016.7487325.
- Reher, J. P. et al. (2016). “Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS”. In: *Twelfth International Workshop on Algorithmic Foundations on Robotics*.
- Reher, Jacob et al. (2016). “Realizing Dynamic and Efficient Bipedal Locomotion on the Humanoid Robot DURUS”. In: *IEEE International Conference on Robotics and Automation (ICRA)*.

- Reher, Jake, Wen-Loong Ma, and Aaron D. Ames (2019). “Dynamic Walking with Compliance on a Cassie Bipedal Robot”. In: *2019 18th European Control Conference (ECC)*, pp. 2589–2595. doi: 10.23919/ECC.2019.8796090.
- Ren, Wei, Randal W Beard, and Ella M Atkins (2005). “A survey of consensus problems in multi-agent coordination”. In: *Proceedings of the 2005, American Control Conference, 2005*. IEEE, pp. 1859–1864.
- Rezazadeh, Siavash et al. (2015). “Spring-mass Walking with ATRIAS in 3D: Robust Gait Control Spanning Zero to 4.3 KPH on a Heavily Underactuated Bipedal Robot”. In: *Proceedings of the ASME 2015 Dynamic Systems and Control Conference*.
- Sakagami, Yoshiaki et al. (2002). “The intelligent ASIMO: System overview and integration”. In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 3. IEEE, pp. 2478–2483.
- Sastry, Shankar (1999). *Nonlinear systems: analysis, stability, and control*. Vol. 10. Springer New York.
- Sloth, C. (2016). “On the computation of Lyapunov functions for interconnected systems”. In: *2016 IEEE Conference on Computer Aided Control System Design (CACSD)*, pp. 850–855. doi: 10.1109/CACSD.2016.7602545.
- Sontag, E. D. and Yuan Wang (1996). “New characterizations of input-to-state stability”. In: *IEEE Transactions on Automatic Control* 41.9, pp. 1283–1294.
- Sontag, Eduardo D (2008). “Input to state stability: Basic concepts and results”. In: *Nonlinear and optimal control theory*. Springer, pp. 163–220.
- (1999). “Control-Lyapunov functions”. In: *Open Problems in Mathematical Systems and Control Theory*. Ed. by Vincent Blondel et al. London: Springer London, pp. 211–216. ISBN: 978-1-4471-0807-8. doi: 10.1007/978-1-4471-0807-8_40. URL: https://doi.org/10.1007/978-1-4471-0807-8_40.
- Spence, A. et al. (2007). “Effects of substrate properties on equine locomotion”. In: *Comparative Biochemistry and Physiology a-Molecular & Integrative Physiology* 146.4. Suppl. S, S109–S109.
- Spong, Mark (1998). “Underactuated mechanical systems”. In: *Control Problems in Robotics and Automation*. Ed. by Bruno Siciliano and Kimon Valavanis. Vol. 230. Lecture Notes in Control and Information Sciences. Springer Berlin / Heidelberg, pp. 135–150. URL: <http://dx.doi.org/10.1007/BFb0015081>.
- Spong, Mark W. (1989). *Robot Dynamics and Control*. 1st. USA: John Wiley & Sons, Inc. ISBN: 047161243X.
- Spotmini Walking* (2016). Boston Dynamics. URL: <https://youtu.be/tf7IEVTDjng>.
- Sreenath, K. (Aug. 2011). “A Compliant Hybrid Zero Dynamics Controller for Stable, Efficient and Fast Bipedal Walking on MABEL”. In: *The International Journal of Robotics Research* 30.9, pp. 1170–1193. doi: 10.1177/0278364910379882.

- Sreenath, K., H.-W. Park, I. Poulakakis, and J. Grizzle (Mar. 2013). “Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL”. In: *The International Journal of Robotics Research* 32.3, pp. 324–345. ISSN: 0278-3649. DOI: 10.1177/0278364912473344. URL: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364912473344>.
- Sreenath, K., H.-W. Park, I. Poulakakis, and J. W. Grizzle (Aug. 2011). “Compliant Hybrid Zero Dynamics Controller for achieving Stable, Efficient and Fast Bipedal Walking on MABEL”. In: *The International Journal of Robotics Research* 30.9, pp. 1170–1193.
- (2013). “Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL”. In: *The International Journal of Robotics Research* 32.3, pp. 324–345. DOI: 10.1177/0278364912473344.
- Srinivasan, Manoj and Andy Ruina (2006). “Computer optimization of a minimal biped model discovers walking and running”. In: *Nature* 439.7072, pp. 72–75.
- Strandberg, Lennart and Håkan Lanshammar (1981). “The dynamics of slipping accidents”. In: *Journal of occupational accidents* 3.3, pp. 153–162.
- Tabuada, P. et al. (2017). “Data-driven control for feedback linearizable single-input systems”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 6265–6270. DOI: 10.1109/CDC.2017.8264603.
- Tamada, T. et al. (Nov. 2014). “High-speed bipedal robot running using high-speed visual feedback”. In: *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 140–145. DOI: 10.1109/HUMANOIDS.2014.7041350.
- Tan, Jie et al. (2018). “Sim-to-Real: Learning Agile Locomotion For Quadruped Robots”. In: *CoRR* abs/1804.10332. arXiv: 1804.10332. URL: <http://arxiv.org/abs/1804.10332>.
- Tedrake, R. et al. (Nov. 2015). “A closed-form solution for real-time ZMP gait generation and feedback stabilization”. In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 936–940. DOI: 10.1109/HUMANOIDS.2015.7363473.
- Tinetti, Mary E and Christianna S Williams (1997). “Falls, injuries due to falls, and the risk of admission to a nursing home”. In: *New England journal of medicine* 337.18, pp. 1279–1284.
- Tsai, Lung-Wen (1999). *Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators*. 1st. USA: John Wiley & Sons, Inc. ISBN: 0471325937.
- Vázquez, JA and M Velasco-Villa (2013). “Experimental estimation of slipping in the supporting point of a biped robot”. In: *Journal of applied research and technology* 11.3, pp. 348–359.
- Vejdani, H. R. et al. (May 2015). “Touch-down angle control for spring-mass walking”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5101–5106. DOI: 10.1109/ICRA.2015.7139909.

- Villarreal, Dario J and Robert D Gregg (2014). “A survey of phase variable candidates of human locomotion”. In: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, pp. 4017–4021.
- Vukobratovic, Miomir and Branislav Borovac (2004). “Zero-moment point? thirty five years of its life”. In: *International Journal of Humanoid Robotics* 1.1, pp. 157–173.
- Vulobratocić, M. and B. Borovac (2004). “ZERO-MOMENT POINT — THIRTY FIVE YEARS OF ITS LIFE”. In: *International Journal of Humanoid Robotics* 01.01, pp. 157–173. DOI: 10.1142/S0219843604000083. eprint: <http://www.worldscientific.com/doi/pdf/10.1142/S0219843604000083>. URL: <http://www.worldscientific.com/doi/abs/10.1142/S0219843604000083>.
- Wächter, A. and L. T. Biegler (2006). “On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming”. In: *Mathematical Programming* 106.1, pp. 25–57.
- Wensing, Patrick M and David Orin (2014). “3D-SLIP steering for high-speed humanoid turns”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4008–4013.
- Westervelt, E. R., J. W. Grizzle, C Chevallereau, et al. (June 2007). *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation. Boca Raton: CRC Press.
- (2007). *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton.
- (2007). *Feedback Control of Dynamic Bipedal Robot Locomotion*. Taylor & Francis/CRC.
- Xi, Weitao, Yevgeniy Yesilevskiy, and C David Remy (2015). “Selecting gaits for economical locomotion of legged robots”. In: *The International Journal of Robotics Research*, pp. 0278–3649. DOI: 10.1177/0278364915612572. URL: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364915612572>.
- Zhao, Hui-Hua et al. (2014). “Human-inspired multi-contact locomotion with AMBER2”. In: *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*. Best Paper Award Finalist of ICCPS 2014. IEEE, pp. 199–210. DOI: 10.1109/ICCPS.2014.6843723.
- Zhao, Huihua et al. (Feb. 2016). “Multi-contact bipedal robotic locomotion”. In: *Robotica* FirstView, pp. 1–35. ISSN: 1469-8668. DOI: 10.1017/S0263574715000995. URL: http://journals.cambridge.org/article_S0263574715000995.

