

**DYNAMIC HUMANOID LOCOMOTION:
HYBRID ZERO DYNAMICS BASED GAIT
OPTIMIZATION VIA DIRECT COLLOCATION
METHODS**

A Thesis
Presented to
The Academic Faculty

by

Yongga A

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
The George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2016

Copyright © 2016 by Yongga A

**DYNAMIC HUMANOID LOCOMOTION:
HYBRID ZERO DYNAMICS BASED GAIT
OPTIMIZATION VIA DIRECT COLLOCATION
METHODS**

Approved by:

Professor Aaron Ames,
Committee Chair
School of Mechanical Engineering &
Electrical and Computer Engineering
Georgia Institute of Technology

Professor Aaron Ames, Advisor
School of Mechanical Engineering &
Electrical and Computer Engineering
Georgia Institute of Technology

Professor Jonathan Rogers
School of Mechanical Engineering
Georgia Institute of Technology

Professor Frank Hammond III
School of Mechanical Engineering &
Department of Biomedical Engineering
Georgia Institute of Technology

Professor Yorai Wardi
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Andrea Thomaz
Department Electrical and Computer
Engineering
University of Texas at Austin

Date Approved: 08/16/2016

To my parents, and my wife.

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dr. Aaron Ames, for his continuous support throughout my journey of pursuing this Ph.D. degree. His passion towards the theoretical understanding of a problem is inspiring and always guided me the right direction in my research. I would also like to thank the members of my committee: Dr. Andrea Thomaz, Dr. Jonathan Rogers, Dr. Frank Hammond III, and Dr. Yorai Wardi for their approvals of serving on my reading committee. This work would not have been possible without their help, insight, and guidance.

A special thanks to Christian Hubicki, who first introduced me to the wonderful world of trajectory optimization and who was always there to provide constructive feedback and suggestions. I would also like express my great gratitude to my friends and members of AMBER Lab, Matthew Powell, Huihua Zhao, Wenlong Ma, Shishir Yadukuma, Jake Reher, and Eric Ambrose, Ryan Sinnet, Eric Cousineau, for all enjoyable moments that we shared and valuable lessons that I have learned from you during the my graduate study. I would like to thank them for creating such a constructive and inspiring atmosphere in this lab.

Last but not least, I would like to thank my parents, Nimacangbu and Meedeg, and my little sister, Cholmon, for their endless support that always encourages me to pursue my goal. Particularly, I would like to thank my wife, Haiya, for accompanying me all the times and help me go through all my lows and highs. I would not achieve my goal without their loves.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF SYMBOLS	xiv
LIST OF ABBREVIATIONS	xvi
SUMMARY	xvii
I INTRODUCTION	1
1.1 The State-of-the-Art	2
1.1.1 Zero Moment Point	2
1.1.2 Spring-Loaded Inverted Pendulum	3
1.1.3 Whole Body Optimization Based Methods	4
1.1.4 Hybrid Zero Dynamics	5
1.2 Objectives of the Thesis	7
1.3 Organization of the Thesis	9
II HYBRID ZERO DYNAMICS FRAMEWORK	12
2.1 Multi-Domain Hybrid Systems	12
2.1.1 Directed Cycle	12
2.1.2 Hybrid System Model	14
2.1.3 Periodic Orbits and the Poincaré Return Map	15
2.2 Humanoid Locomotion as Hybrid Systems	17
2.2.1 Lagrangian Model	17
2.2.2 Constrained Dynamics	19
2.2.3 Domains and Guards	22
2.3 Feedback Control Law Design	25

2.3.1	Virtual Constraints	25
2.3.2	Input-Output Feedback Linearization	28
2.3.3	Hybrid Zero Dynamics	29
2.3.4	Control Lyapunov Functions	34
2.4	Summary	36
III DIRECT SHOOTING METHODS		37
3.1	Gait Optimization Problem	37
3.2	Single Shooting Optimization	39
3.2.1	Problem Formulation	39
3.2.2	Case Study: 5-link Planar Underactuated Biped	41
3.2.3	Issues	44
3.3	Multiple Shooting Optimization	45
3.3.1	Problem Formulation	46
3.3.2	Case Study: 7-Link Planar Biped with Compliant Legs	48
3.3.3	Issues	58
3.4	Summary	59
IV DIRECT COLLOCATION METHODS		60
4.1	Mathematical Background	60
4.1.1	Standard Collocation	62
4.1.2	Orthogonal Collocation	63
4.2	Constrained Dynamics	65
4.3	Local Direct Collocation Optimization	68
4.3.1	Collocation Constraints	68
4.3.2	Problem Formulation	71
4.3.3	Cost Function	78
4.4	Pseudospectral Optimization	79
4.4.1	Orthogonal Collocation Constraints	80
4.5	Structured NLP Construction	83

4.6	Control Lyapunov Function Synthesis	87
4.7	Summary	90
V	APPLICATIONS TO HUMANOID ROBOTS	92
5.1	DURUS – Efficient Humanoid Walking	93
5.1.1	Robot Model	93
5.1.2	3D Flat-Footed Walking	99
5.1.3	3D Multi-Contact Walking	114
5.1.4	3D Walking Efficiency	123
5.2	DRC-HUBO – Walks with Arm Swing	125
5.2.1	Hybrid System Model	127
5.2.2	Virtual Constraints	129
5.2.3	Optimization and Task Definitions	130
5.2.4	Simulation Results	132
5.2.5	Preliminary Hardware Experiments	133
5.3	PROXI – Online Optimal Gait Generation	135
5.3.1	Online Motion Planning	135
5.3.2	Hybrid System Model	137
5.3.3	Virtual Constraints	138
5.3.4	Online Gait Optimization	139
5.3.5	Experimental Results	140
5.4	Summary	147
VI	OPTIMIZATION PERFORMANCES	148
6.1	Solution Accuracy	149
6.1.1	Error Estimation	149
6.1.2	Numerical Results	151
6.1.3	Summary	154
6.2	Robustness Analysis of NLP Convergence	155
6.2.1	Robustness Measurement	155

6.2.2	Numerical Results	157
6.2.3	Summary	160
6.3	Computational Efficiency	161
6.3.1	Comparison versus the Single Shooting Optimization	161
6.3.2	Comparison versus Multiple Shooting Optimization	163
6.3.3	Comparison versus DIRCON	164
6.4	Performance Summary of Direct Collocation Methods	167
VII	CONCLUSION	171
7.1	Summary of Contributions	171
7.2	Perspectives on Future Work	174
APPENDIX A	— EQUATIONS OF MOTION OF ZERO DYNAM-	
	ICS	178
REFERENCES	181

LIST OF TABLES

5.1	Position-modulating outputs library for DURUS 3D walking.	104
5.2	A comparison of domain durations between human locomotion data versus the multi-contact walking gait on DURUS through optimization	119
5.3	Comparison of gait efficiency on various robot platforms	125
5.4	Performance comparison of online gait optimization with different methods of computing the Hessian (Total number of optimizations is: 198.)	144
6.1	Solution accuracy of local direct collocation optimization when using the CGL node distribution.	152
6.2	Solution accuracy of local direct collocation optimization when using the uniform node distribution.	152
6.3	Solution accuracy of the pseudospectral optimization.	154
6.4	Comparison results between the (local) direct collocation optimization vs the single shooting optimization.	162
6.5	Comparison results between the (local) direct collocation optimization vs the multiple shooting optimization.	164
6.6	Runtime and convergence test of the optimization.	168
6.7	Performance summary of the (local) direct collocation HZD optimization on different robot platforms.	170

LIST OF FIGURES

1.1	Experimental robotic platforms.	9
2.1	An example of <i>directed cycle</i> , i.e., discrete phases and transitions that represents the walking pattern of typical human walking pattern. . .	13
2.2	Illustration of the flat foot ground contact.	20
2.3	Illustration of the zero dynamics under the exponentially stable feedback controllers.	29
2.4	Illustration of Periodic Orbit	32
3.1	Illustration of the single shooting hybrid zero dynamics gait optimization.	40
3.2	Model configuration and outputs representation of a 5-link point feet planar robot.	42
3.3	The discretization of the periodic orbit on the hybrid zero dynamics manifold when using multiple shooting optimization methods.	46
3.4	DURUS-2D biped model.	49
3.5	The directed cycle of the hybrid control system model of DURUS-2D.	51
3.6	Periodic limit cycles of a stable periodic walking gait of DURUR-2D in simulation.	54
3.7	Desired outputs of a stable periodic walking gait of DURUR-2D in simulation.	55
3.8	Stable periodic orbits of walking gait of DURUR-2D	56
3.9	Joint angles tracking performance of DURUS-2D.	57
3.10	The walking gait snapshot comparison of the simulation and experimental results with DURUS-2D over one step.	57
4.1	Distributions of LG, LGR, and LGL points ($N = 7$).	64
4.2	Illustration of defect constraints and node distribution.	70
4.3	Demonstration of the global orthogonal collocation at Legendre-Gauss-Lobatto (LGL) points.	82
4.4	The illustration of sparse Jacobian matrix construction	86
5.1	Illustration of the process used to generate dynamic walking gaits.	93
5.2	The mechanical design of the humanoid DURUS. The design of linear springs at the end of legs is to absorb the energy at impact.	94

5.3	Iterative design process which facilitated a leg morphology satisfying hardware constraints in simulation.	95
5.4	The coordinates of DURUS robot	97
5.5	Illustration of the position of different contact points.	100
5.6	Domain graph of the two-domain 3D flat-footed walking model. . . .	101
5.7	3D flat-footed walking outputs defined in the sagittal plane.	103
5.8	Desired outputs of the 3D flat-footed walking gait	108
5.9	Convergence of outputs y_2 and \dot{y}_2 in 4 steps with input-output linearization controllers when simulating the flat-footed walking gait from rest.	109
5.10	Tracking of the velocity-modulating output using both IO linearization controllers and QP based CLF controllers in flat-footed walking simulation.	109
5.10a	Input-output linearization controllers	109
5.10b	QP based CLF controllers	109
5.11	Joint torque profiles when using two different feedback control laws in DURUS flat-footed walking simulation.	110
5.12	Phase portraits of representative joints in the 3D flat-footed walking simulation starting from the rest.	110
5.13	The setup of 3D DURUS walking experiments	112
5.14	Phase portrait plots of flat-footed walking with DURUS in 3D	113
5.15	Gait tiles of DURUS 3D flat-footed walking	113
5.16	Domains graph of 3D multi-contact walking. The red circles represent foot contact points.	114
5.17	Desired outputs of the 3D multi-contact walking gait	118
5.18	Convergence of outputs y_2 and \dot{y}_2 in 4 steps with input-output linearization controllers when simulating the multi-contact walking gait from rest.	119
5.19	Tracking of the velocity-modulating output using both IO linearization controllers and QP based CLF controllers in multi-contact walking simulation.	121
5.20	Joint torque profiles when using two different feedback control laws in DURUS multi-contact walking simulation.	121

5.21	Phase portraits of representative joints in the 3D multi-contact walking simulation starting from the rest.	122
5.22	Phase portrait plots of multi-contact walking with DURUS in 3D . . .	122
5.23	Gait tiles of DURUS 3D multi-contact walking	123
5.24	Electrical cost of transport (COT) for a 3D flat-footed walking experiment over 800 steps with a mean cost of transport of 1.33.	124
5.25	Electrical cost of transport (COT) for a 3D multi-contact walking experiment over 200 steps with a mean cost of transport of 1.02.	124
5.26	The DRC-HUBO humanoid.	126
5.27	Kinematic tree diagram of the DRC-HUBO platform.	128
5.28	Flow of domains for single-support walking with DRC-HUBO.	128
5.29	A stroboscopic image of full-body-optimized stable walking in simulation (DART environment).	132
5.30	Comparison of the center-of-mass position over time between the simulated gait and optimization.	133
5.31	A walking tiles figure comparing the optimized gait from the optimization and DART simulator.	134
5.32	Tiled images of a preliminary dynamic walking experiment with DRC-HUBO.	134
5.33	Robot model and hybrid system structure of the 5-link underactuated planar robot–RPOXI.	138
5.34	Experimental setup of the speed-regulated walking of a planar underactuated robot on a treadmill.	141
5.35	Snapshots of the nominal gait in simulation.	142
5.36	The histogram plots of total 198 gait optimizations.	142
5.37	The tracking of varied treadmill speed, \bar{v}_{tread}^d . The dashed line represents the speed, \bar{v}_{opt} , of the optimized gait.	143
5.38	The range of desired outputs over different walking speeds.	144
5.39	Snapshots of optimal gaits generated from the online optimizer at different walking speeds.	145
5.40	Phase portraits of optimal gaits at different walking speeds generated by the online optimizer.	145

6.1	Tiled still images from the simulation of Atlas multi-contact walking in 3D at $0.51m/s$	166
6.2	Periodic limit cycles of robot joints of the Atlas 3D multi-contact walking gait.	166
7.1	The competition space inhabited by both DURUS and STEPPR at the DARPA Robot Endurance Test.	173
7.2	Two-domain hybrid system model of dynamic running on DURUS.	175
7.3	Tiled still images the DURUS running at $2.0 m/s$	175
7.4	Two-step domain graph of the asymmetric amputee- prosthesis gait.	176
7.5	Phase portraits of the ankle and knee joints of both the amputee and prosthesis over 20 steps.	177

LIST OF SYMBOLS

\mathcal{C}	Indexing set of holonomic constraints
\mathcal{D}	A set of continuous domains
$\mathbf{E}(\cdot)$	Terminal cost of the cost function
E	A set of edges
f, g	Vector fields of dynamical systems
$h(q)$	Holonomic constraints
\mathcal{H}	Hybrid system
\mathcal{HC}	Hybrid control system
$\mathcal{J}(\cdot)$	Cost function
$\mathbf{L}(\cdot)$	Running cost of the cost function
L_f, L_g	Lie derivatives
\mathcal{O}	Indexing set of robot outputs
\mathcal{O}	Periodic orbit
$P(\cdot)$	Poincaré map
p_c^x, p_c^y, p_c^z	Cartesian positions of a point c
\mathcal{PZ}	Partial zero dynamics submanifold
\mathcal{Q}	Configuration space of extended coordinates
$\mathcal{R}(\cdot)$	Relabeling map of joint coordinates
\mathbb{R}^n	Eulicidean n -space
S	A set of switching surfaces
$SO(n)$	n -dimension special orthogonal group
$\text{sor}(\cdot)$	The source vertex of an edge
$\text{tar}(\cdot)$	The target vertex of an edge
$T_{I,v}$	Time to impact value

\mathcal{U}	Control inputs manifold
V	A set of vertices
\mathcal{X}	State space of the dynamical system
y^a, y^d	Actual and desired outputs
\mathcal{Z}	Zero dynamics submanifold
\square_{ds}	Notations defined on the 2D/3D double-support domain
\square_{ss}	Notations defined on the 2D/3D single-support domain
\square_{hl}	Notations defined on the 3D heel-lift domain
\square_{hs}	Notations defined on the 3D heel-strike domain
\square_{tl}	Notations defined on the 3D toe-lift domain
\square_{ts}	Notations defined on the 3D toe-strike domain
\square_e	Notations defined on a edge $e \in E$
$\square _{\square}$	Restriction of a map to a reduced dimensional space
\square_v	Notations defined on a vertex $v \in V$
$\square^{\mathcal{X}}$	Projection on to the state space \mathcal{X}
$\square^{\min}, \square^{\max}$	Minimum and maximum values
\square^*	Optimal solutions
$\square^{(i)}$	variables defined on node i
α	Parameters of virtual constraints
Δ	Reset map
η	Local coordinates of transverse dynamics
Γ	Direct graph or cycle
$\lambda(\cdot)$	Constraint wrenches
$\phi_c^x, \phi_c^y, \phi_c^z$	Orientations of a coordinate frame c
ξ	Local coordinates of zero dynamics

LIST OF ABBREVIATIONS

CGL	Chebyshev-Gauss-Lobatto points
CLF	Control Lyapunov function
COM	Center of mass
COT	Cost of transport
DOF	Degrees of freedom
HSS	Hermite-Simpson separated collocation scheme
HZD	Hybrid zero dynamics
IO	Input-output feedback linearization control
LGL	Legendre-Gauss-Lobatto points
LIPM	Linear inverted pendulum model
NLP	Nonlinear programming problem
PHZD	Partial hybrid zero dynamics
RES-CLF	Rapidly exponentially stabilizing control Lyapunov function
SLIP	Spring loaded inverted pendulum model
ZMP	Zero moment point

SUMMARY

Hybrid zero dynamics (HZD) has emerged as a popular framework for dynamic and underactuated bipedal walking, but has significant implementation difficulties when applied to the high degrees of freedom present in humanoid robots. The primary impediment is the process of gait design—it is difficult for optimizers to converge on a viable set of virtual constraints defining a gait. This dissertation presents a methodology that allows for the fast and reliable generation of efficient multi-domain robotic walking gaits through the framework of HZD, even in the presence of underactuation. To achieve this goal, we unify methods from trajectory optimization with the control framework of multi-domain hybrid zero dynamics. We present a novel optimization formulation in the context of direct collocation methods and HZD where we rigorously generate analytic Jacobians for the constraints. Two collocation methods, local collocation and pseudospectral (global) collocation, are developed within an unified framework, and their performance in different circumstances is comparatively studied. As a result, solving the resulting nonlinear program becomes tractable for large-scale NLP solvers, even for systems as high-dimensional as humanoid robots. We experimentally validate our methodology on multiple humanoid platforms, showing that the optimization approach yields dynamic and stable walking gaits for different walking configurations, including sustained 3D dynamic multi-contact walking.

CHAPTER I

INTRODUCTION

Humanoid robots have long held the promise of walking around in the human world the dynamic way that people walk. However, demonstrating dynamic and agile locomotion for humanoid robots that allows for navigation of terrain not approachable by wheeled robots is a challenging task. While humans and other biological bipeds can perform these motions with relative ease, translation of these capabilities to 3D humanoid systems is fraught with complexities in the form of nonlinearities and high degrees of freedom which must be coordinated. The faster and more nimble we demand these machines to be, the more the robot needs to reason about its full-order dynamics. It can be helpful tilt on the edges of its feet for toe-off and heel-strike maneuvers, exploit compliant linkages for impact reduction, and embrace the under-actuated dynamics of falling forward to the advantage of locomotion. These useful phenomena are frequently assumed away in more reduced-order frameworks, but are captured in more detailed full-order robot dynamics.

With the goal of bridging this gap in natural and efficient locomotion on bipedal robots, it becomes prudent for theorists to develop algorithmic approaches which are capable of exploiting the natural dynamics of the robot. However, to date, planning dynamic motion that reconciles the full body dynamics of the complex robot model has been primarily prevented by a particular computational bottleneck: gait synthesis. The focus of this thesis is on developing a systematic framework for implementing dynamic locomotion gaits for humanoids through unified hybrid zero dynamics (HZD) control framework and direct collocation based trajectory optimization methods.

1.1 The State-of-the-Art

The study of robotic walking presents a wide range of mathematical and algorithmic challenges for which many solutions have been proposed. On the simpler side of the spectrum, many humanoid control methods will use a reduced-order model as a basis. Whereas formal methods for humanoid locomotion reason about a model of the machine, but only exist on a spectrum of the complexity of the model.

1.1.1 Zero Moment Point

Many existing methods for planning humanoid locomotion typically simplify the robot to a reduced-order model to mitigate the complexity of the gait planning operation. Since the origin of humanoid control, the zero moment point (ZMP) approaches have certified balance by ensuring that the center of pressure always rests within the support polygon of its feet [130, 131]. By ensuring ZMP condition, many approaches plan trajectories for a simple dynamical model, e.g., linear inverted pendulum model (LIPM), assumptions about the robot dynamics and then generate the whole body motion by conforming the robot to these analytically tractable dynamics [80]. Constraining the robot to these reduced-order dynamics, controllers can plan the position of the ZMP to ensure fall-free motions [81, 100, 114, 142], or quickly compute recovery steps [105, 114, 124]. The maturity and reliability of the LIPM-based planning approaches made it a prevalent component in control algorithms for humanoid walking at the DARPA Robotics Challenge [78, 85, 40].

These linear inverted pendulum based model simplification methods come at a cost; they limit the flexibility and capability of available control. These simplifications place stringent requirements on the design of the robot (e.g. all joints must be actuated with no significant compliance) and restrictions on the overall locomotion capabilities of the machine (e.g. the robot must always walk with a constant center-of-mass height [80]). This class of walking robots has very large feet to increase the

area of support polygon. The resulting walking motions are flat footed and quasi-static, which is a rather artificial behaviors when compared to energy-optimal bipedal trajectories [4, 123, 139].

Further, these methods cannot handle underactuation that presents in the system, a property that is very common in some of the most agile [106], robust [99], or efficient [20] bipedal robots. Both the planning framework and mechanics of these fully-actuated bipeds also demand near-zero leg-impact forces [70], which rules out the dynamic foot-strike behaviors documented in human locomotion [28] and other biological bipeds [21].

Finally, it is very difficult to provide *a priori* guarantees on whether any given reduced-order plan is feasible to execute given the full-order dynamics. Such methods typically employ inverse kinematics [79] or inverse dynamics [97] methods to compute the full-order control inputs at each instant. Solving such near-term inverse calculations does not imply that future inverse problems later in the trajectory will be feasible, which requires more computationally intensive planning [63, 148].

1.1.2 Spring-Loaded Inverted Pendulum

The Spring Loaded Inverted Pendulum (SLIP), based upon the concept of coupled compliant pendula, provides a low-dimensional representation of locomotion inspired by biological principles that has been shown to approximate animal walking and running behaviors in everything from cockroaches, to quail, to kangaroos [23], to humans [22]. As a result of this biological motivation, the ability to realize SLIP-like walking gaits on bipedal robots promises to result in natural, efficient and robust locomotion. This is evidenced by the classic work by Raibert on hopping robots [106], which has since motivated the study of walking and running in robotic systems with simple SLIP models [5, 49, 90, 113, 134].

Through very careful mechanical design, more dynamic bipeds have been built

to emulate the spring-mass like dynamical behaviors [6, 71, 76, 110, 112]. In this approach, the dynamics of the robot naturally follow a mathematical model which reproduces bipedal dynamics demonstrated by many animals. This model is attractive to the bipedal community as it allows control engineers to utilize an ideal model which does not suffer from the inherent nonlinearities, hybrid dynamics, and numerous degrees of freedom as considering a full-order biped.

However, if the dynamics of the robot do not closely follow the SLIP model, then more rigid mathematical guarantees must be made. Ultimately, the fundamental limitation in realizing the benefits of SLIP inspired locomotion is the low-dimensional nature of the SLIP model, and the difficulty of realizing this low-dimensional behavior on full-order high-dimensional walking robots.

1.1.3 Whole Body Optimization Based Methods

The rapid development of mechanical and actuation capabilities of modern robots has already made more dynamical locomotion on robot possible, however, current ZMP based criterion still requires the robot moves slowly and statically due to the lack of knowledge of the system dynamics. Adapted from the trajectory optimization community, some researchers began to utilize whole body dynamic gait optimization to generate walking motion, but still subjects to ZMP criterion as the stability condition [82]. A few others are able to demonstrate effective multi-contact locomotion by optimizing the whole robot actuation on a relatively small time duration [87]. The complexity and nonlinearity of the whole body dynamics, however, increase the difficulty of these optimization being converged a viable solution, and as a result, impede the applications of these optimization based approaches on actual humanoid platforms.

To mitigate the computational difficulty that arises from the full body dynamics of humanoids, roboticists have begun planning more general motions with versatile

contact conditions by using the machine’s center-of-mass (COM) dynamics and whole-body contact forces as a planning foundation [29, 84]. In those application, the whole body dynamics is represented by the six dimensional COM dynamics using the augmented linear and angular momentum at the COM due to the ground contact forces [124]. Instead of checking the stability via ZMP criterion, these optimization based methods rationale if the contact wrench sum (CWS) lies within the contact wrench cone (CWC) [37] as the stable balancing condition. The stability criterion based on the contact wrenches are checked as the contact wrench sum is equal to the rate of linear and angular momentum of the robot. This condition coupled with the COM dynamics, a nonlinear constrained optimization problem is formulated to generate more dynamic gaits for full-order humanoids [69, 85].

On the other end of the spectrum, it is also necessary to mention the trending fact that the application of advanced trajectory optimization methods, such as direct collocation, allows formulating the dynamic gait optimization in terms of the full body dynamics [103]. Using the open-loop trajectory from the direct collocation optimization, a classical linear quadratic regulator (LQR) based feedback controller can be constructed to stabilize the resulting open-loop trajectory for the contact-constrained dynamical system [102].

1.1.4 Hybrid Zero Dynamics

Hybrid zero dynamics (HZD) is a formal framework for synthesizing control for nonlinear hybrid systems that does not require simplifying dynamical assumptions [57, 135]. Since its inception, HZD has built a strong history of success in planar robot implementations for bipedal walking [7, 30, 31, 68, 89, 116, 120, 137] and running [88, 121]. Recent work has begun expanding the method into 3D applications [8, 26, 109]. HZD works by designing a set of *virtual constraints* that define the gait, which are enforced via feedback control of the actuated degrees of freedom. If these constraints

satisfy a hybrid invariant condition, i.e., the system is invariant through impact, all of the stability properties of the high-dimensional system are effectively captured in a lower-dimensional representation, termed as the *hybrid zero dynamics*. This key task of finding an appropriate set of virtual constraints (and parameters thereof) is typically relegated to a nonlinear optimization problem. When robots have as many linkages as humanoids, optimizing motions which meet HZD criteria becomes increasingly difficult for nonlinear programming (NLP) tools to solve. This has been a significant impediment toward applying HZD to full humanoid robots, which have far more degrees of freedom than the typical HZD testbed machines. In a broader view, this is where the HZD approach has had to pay the piper for its admissibility of highly dynamic gaits. While ZMP synthesis methods enjoy better computational tractability by restricting allowable gaits, HZD methods must confront the nonlinear gait optimization problem head on.

HZD optimizations are typically formulated by assigning one design variable to each virtual constraint parameter, and evaluate the objective and constraint violations of candidate solutions with a time-marching numerical integration method. This is highly analogous to a *direct single shooting* formulation in optimal control. Such “single shooting” is the most prevalent formulation for optimizing virtual constraints in the context of HZD [7, 56, 107]. This approach of optimizing only parameters and boundary state values reflects an instinctive desire to minimize the number of design variables for the optimization. Intuitively, one might assume that such minimization of the nonlinear programming problems dimensionality would be an advisable practice for maximizing an optimization’s speed and reliability. However, this formulation is prone to a number of issues, such as nonsmooth approximations of the constraint Jacobian [123] and the “tail wagging the dog” phenomenon [19], which can introduce pseudominima [129] or simply cause the algorithm to fail to find a solution. Given the nonlinearity of bipedal robot dynamics, it can be difficult to achieve reliable

convergence via this optimization and often relies upon expert users to seed it.

1.2 Objectives of the Thesis

Embracing planning and formal control that exploits the full body dynamics of the robot is a path toward unlocking the fully dynamic capabilities of the machine. We seek a formal framework that both synthesizes and stably controls dynamic gaits in a manner that reasons about the full-order nonlinear dynamics of a humanoid robot. With that being said, we employ the hybrid zero dynamics framework to realized dynamic humanoid locomotion.

The main goals of the thesis are twofold: (1) designing a computationally efficient optimization framework to overcomes the scalability issues of the hybrid zero dynamics gait synthesis; and (2) realizing dynamic and efficient walking behaviors on humanoid platforms by exploiting the natural dynamics of the machines.

To better solve HZD gait optimizations, we take some important lessons from the optimal control community. The *formulation* of the NLP can make an enormous difference: fewer design variables is not always better and not all design variables are equally complicated for an NLP solver to navigate. For common NLP-solving methods like interior point and sequential quadratic programming, convergence is more reliable when the objectives and constraints are smooth and relatively linear [108]. In a single shooting formulation, a change to a parameter or boundary state value can have a very nonlinear effect on the final state, after integrating nonlinear dynamics over non-trivial time scales. This type of nonlinearity can cause an optimizer to fail to converge or be uncertain if it has found a solution.

In this thesis, we systematically unify the hybrid zero dynamics framework with a technique from the trajectory optimization community: *direct collocation* methods [43, 62, 128]. A direct collocation formulation represents both the time-varying states

and inputs as parameterized curves, where system dynamics are enforced as equality constraints (called *defect constraints*), then transcribes the infinite optimal control problem into a finite nonlinear programming problem [19, 73]. This property has made direct collocation methods fast and reliable for trajectory optimization of dynamical systems [35, 48, 53, 138]. In particular, we consider two different collocation methods *local standard collocation* [62, 128] and *global orthogonal collocation* [43, 53]. In the local direct collocation, the time interval is divided into a finite set of subintervals, and the solution of the system is approximated with some specified functions—such as Hermite interpolation polynomials—within each subinterval. The *pseudospectral* method (a.k.a. global orthogonal collocation) employs particular orthogonal polynomial functions, e.g., Legendre polynomials, to approximate the solution of the system over an entire continuous domain. Satisfying collocation constraints at each subinterval (local method) or orthogonally collocated nodes (pseudospectral method) guarantees that the approximated continuous polynomials converge to the exact solution of the system. One main difference between these two approaches is that the degree of the polynomial is fixed and the number of subintervals is varied in a local collocation scheme, whereas the degree of the polynomial varies in a pseudospectral method [73].

As validation of this direct collocation based HZD optimization framework, we use the proposed gait optimization framework to generate dynamic walking gaits on three different robotic platforms (see Figure 1.1):

- DURUS: a 23-DOF underactuated humanoid with springy feet;
- DRC-HUBO: a 33-DOF fully-actuated humanoid with arms;
- PROXI: a planar point-feet underactuated biped.

We then experimentally implement the walking gaits on the robot hardware, demonstrate efficient and dynamic maneuvers for high DOFs humanoids. Through these

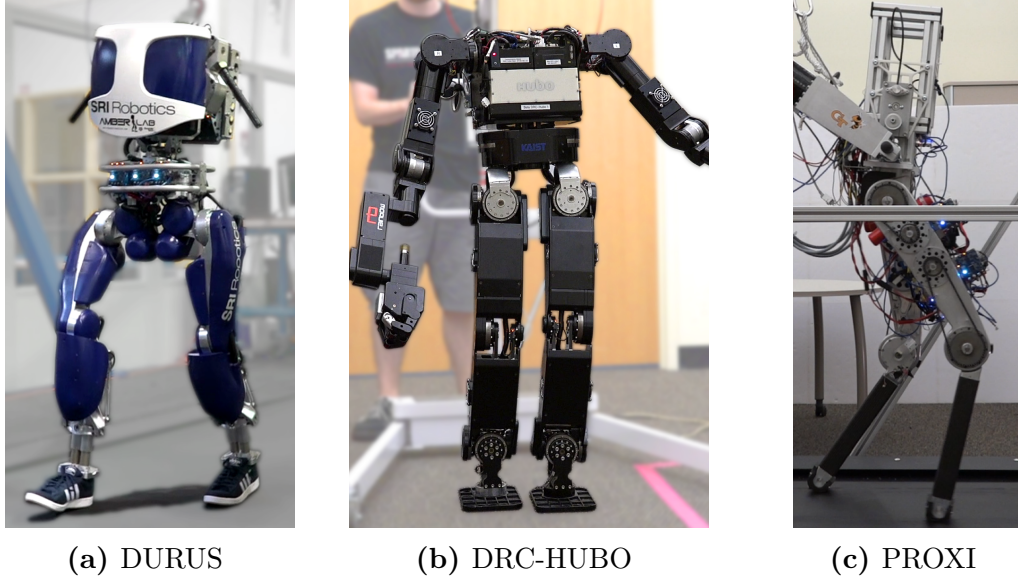


Figure 1.1: Experimental robotic platforms.

applications, we establish the scalability and effectiveness of the direct HZD optimization on a wide variety of robots. More importantly, we establish that the hybrid zero dynamics approach is capable of realizing efficient, robust and agile locomotion of humanoids by exploiting the natural dynamics, such as arm swings or compliant elements, of the robots.

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows.

Chapter 2 presents the formal construction of the hybrid zero dynamics control framework. We start with defining a general multi-domain hybrid system model with a predetermined directed cyclic structure. The construction of individual elements of the hybrid system model that is specific to bipedal locomotion is then introduced as a systematic methodology to model dynamic walking notion. With the model in hand, a class of feedback control laws based on virtual constraints, are designed to formulate the hybrid zero dynamics for bipedal walking gaits.

Chapter 3 introduces the main goal of gait optimization in the context of HZD. We

present necessary requirements of the HZD optimization and illustrate this process via the traditional single shooting method. Further, we introduce a direct multiple shooting formulation of HZD gait optimization as an alternative approach to the single shooting, and demonstrate its improved reliability with a case study of a planar underactuated robot with compliant mechanical designs. Finally, the pros and cons of direct shooting methods are discussed to show the limitations of these approaches.

Chapter 4 presents the main contribution of the thesis: an unified direct HZD optimization framework based on the collocation. Capitalizing the special formulation of collocation methods, we deform the reduced-dimensional hybrid zero dynamics into a dynamical system of constrained manifold with the dynamic equations given in an implicit DAEs system. Two different direct collocation optimization problems, local direct collocation and pseudospectral optimization, are formulated in terms of this constrained dynamical system. In addition, a sparsely structured nonlinear programming problem is designed based on defect variables and smart indexing, to improve the robustness and computational speed of the optimization.

Chapter 5 illustrates real-world applications of the direct HZD gait optimization on three different robotic platforms to demonstrate the flexibility, effectiveness and reliability of these novel hybrid zero dynamics based gait synthesizing methodologies. More importantly, we presents the experimental results of sustained 3D walking of full size humanoids, showing that the walking can be more efficient by exploiting the natural dynamics of the full order system. Further, an example of fast online gait generation is presented to illustrate the significant improvement of convergence speed of the HZD gait optimization, raising the possibility of designing gaits online, so that the robot can adjust its behavior to accommodate changes in environment.

In Chapter 6, we systematically evaluate the computational performances of the direct collocation optimization. The accuracy of the solution is validated by checking the discretization errors, and the robustness of convergence is tested via seeding the

optimizers with randomly generated initial guesses. Further, we compare the computational efficiency of the direct collocation optimization with direct shooting methods and other full body dynamic gait optimization approaches. Finally, we summarize the performance of the direct HZD optimization on different robotic platforms, and show the scalability of the approach with respect to the complexity of the system.

Chapter 7 concludes the important remarks and achievements of the thesis, and highlights the future works of the application.

Finally, Appendix A introduces a general procedure of deriving the reduced dimensional equations of motion of the partial zero dynamics for the multi-domain hybrid system model.

CHAPTER II

HYBRID ZERO DYNAMICS FRAMEWORK

In this chapter, we review the mathematical background and properties of hybrid zero dynamics framework for dynamical humanoid locomotion. The hybrid zero dynamics (HZD) methodology utilizes the hybrid nature of legged robots and realizes stable periodic behaviors by enforcing invariant reduced dimensional manifold via virtual constraints based feedback controllers.

2.1 Multi-Domain Hybrid Systems

Hybrid systems or systems with impulse effects are systems that exhibit both continuous dynamics and discrete dynamics, and thus have a wide range of applications to various types of physical systems [60]. One major application would be robotic legged locomotion, which consists of phases of continuous dynamics (e.g., when the leg swings forward) and discrete dynamics (e.g., when the foot strikes the ground); formally modeling this interplay of continuous and discrete dynamics results in a hybrid system model of robotic walking [7, 58]. In this section, the formal definition of hybrid systems is introduced.

2.1.1 Directed Cycle

A bipedal walking gait often consists of a collection of continuous phases, with discrete events triggering transitions between these different phases. The ordering structure of discrete domains can be represented via directed graphs.

Definition 2.1 (Directed Graph). A *directed graph* is a tuple $\Gamma = (V, E)$, where V is the set of vertices and $E \in V \times V$ is the set of edges. Let $\text{sor} : E \rightarrow V$ and

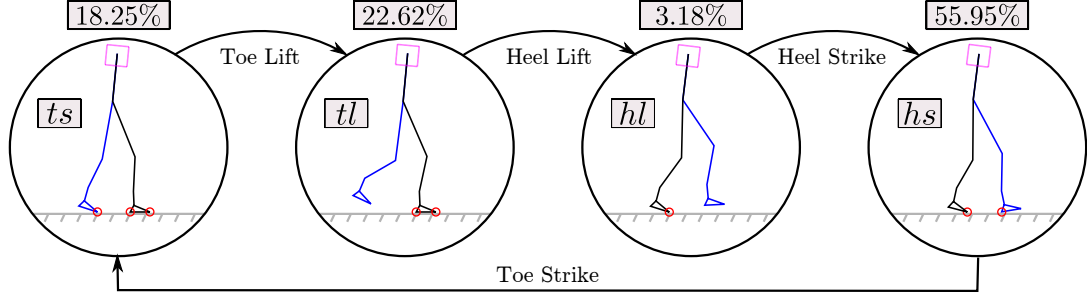


Figure 2.1: An example of *directed cycle*, i.e., discrete phases and transitions that represents the walking pattern of typical human walking pattern.

$\text{tar} : E \rightarrow V$ be the maps that determine the source and target vertices of an edge, respectively. In other words, an edge $e \in E$ can be given as $e = \{\text{sor}(e) \rightarrow \text{tar}(e)\}$.

When using a directed graph to model the ordering structure of a locomotion behavior, each vertex represents a admissible continuous phase (or domain) and each edge represents a possible discrete transition between two continuous phases. In this study, we interest in a periodic steady-state walking gait, in which the transitions between the continuous phases become predetermined and periodic; this motivates the use of a multi-domain hybrid system with a predetermined ordering of phases (or domains) as represented by a *directed cycle*, i.e., a cyclic directed graph.

Definition 2.2 (Directed Cycle). A *directed cycle* is a directed graph $\Gamma = (V, E)$, with $V = \{v_1, v_2, \dots, v_{n_p}\}$ a set of vertices and $E = \{e_1 = (v_1 \rightarrow v_2), e_2 = (v_2 \rightarrow v_3), \dots, e_{n_p} = (v_{n_p} \rightarrow v_1)\}$ a set of edges. For a directed cycle, the maps sor and tar are one-to-one and onto. Hence, their inverse maps $\text{sor}^{-1} : V \rightarrow E$ and $\text{tar}^{-1} : V \rightarrow E$ exist and are well-defined.

Example 2.1. In Figure 2.1, a directed cycle with four discrete domains illustrates the domain structure of a typical human walking gait pattern consists of four different discrete phases depends on the different contact conditions. This gait pattern is obtained by recording kinematic trajectories from walking of multiple human subjects [127]. This directed cycle is given by a graph $\Gamma = (V, E)$ consisting of four vertices

and four edges:

$$V = \{ts, tl, hl, hs\},$$

$$E = \{ts \rightarrow tl, tl \rightarrow hl, hl \rightarrow hs, hs \rightarrow ts\}.$$

2.1.2 Hybrid System Model

With the notation of the directed cycle in hand, we introduce the formal definition of the multi-domain hybrid control system as a tuple that is consistent with the literature on hybrid systems (see [7, 58, 117]).

Definition 2.3 (Multi-Domain Hybrid Control System). A *multi-domain hybrid control system* is a tuple,

$$\mathcal{HC} = (\Gamma, \mathcal{D}, S, \Delta, FG), \quad (2.1)$$

where

- $\Gamma = \{V, E\}$ is a *directed cycle*,
- $\mathcal{D} = \{\mathcal{D}_v\}_{v \in V}$ is a set of admissible *domains* with $\mathcal{D}_v \subseteq \mathcal{X}_v \times \mathcal{U}_v$ a open connected smooth submanifold of the state space $\mathcal{X}_v \in \mathbb{R}^{n_v}$ and the control space $\mathcal{U}_v \in \mathbb{R}^{m_v}$,
- $S = \{S_e\}_{e \in E}$ is a set of *guards* or *switching surfaces* with $S_e = \{(x, u) \in \mathcal{D}_{\text{sor}(e)} \mid H_e(x, u) = 0, \dot{H}_e(x, u) \neq 0\}$ a nonempty embedded submanifold of co-dimension one of the domain $\mathcal{D}_{\text{sor}(e)}$,
- $\Delta = \{\Delta_e\}_{e \in E}$ is a set of *reset maps* with $\Delta_e : S_e^{\mathcal{X}} \cap \mathcal{X}_{\text{sor}(e)} \rightarrow \mathcal{X}_{\text{tar}(e)}$ a smooth map, where $S_e^{\mathcal{X}} := S_e|_{\mathcal{X}}$ is a canonical projection of S_e onto the state space \mathcal{X} ,
- $FG = \{FG_v\}_{v \in V}$ is a set of continuous affine *control systems* defined on \mathcal{D}_v , i.e., $\dot{x} = f_v(x) + g_v(x)u$, where $x \in \mathcal{X}_v$ and $u \in \mathcal{U}_v$.

Consequently, a *multi-domain hybrid system* is a multi-domain hybrid control system with $\mathcal{U}_v = \{\emptyset\}$ for all $v \in V$, e.g., if any state-based feedback controller has been applied to make the system closed-loop.

Definition 2.4 (Multi-Domain Hybrid System). The formal definition of a *multi-domain hybrid system* is given as a tuple,

$$\mathcal{H} = (\Gamma, \mathcal{X}, S^{\mathcal{X}}, \Delta, F), \quad (2.2)$$

where $\Gamma = \{V, E\}$ is a directed cycle, $\mathcal{X} = \{\mathcal{X}_v\}_{v \in V}$ is a set of admissible domains, $S^{\mathcal{X}} = \{S_e^{\mathcal{X}}\}_{e \in E}$ is a set of *guards* with $S_e^{\mathcal{X}} \subset \mathcal{X}_{\text{sor}(e)}$, $\Delta = \{\Delta_e\}_{e \in E}$ is a set of *reset maps* and $F = \{F_v\}_{v \in V}$ is a set of autonomous dynamical systems defined on \mathcal{X}_v , i.e., $\dot{x} = f_v(x)$ where $x \in X_v$.

2.1.3 Periodic Orbits and the Poincaré Return Map

Periodic behaviors of robotic legged locomotion can be represented by periodic orbits of the multi-domain hybrid systems defined in (2.2). Let $\mathcal{X} = \mathcal{X}_{v_1} \cup \mathcal{X}_{v_2} \cdots \cup \mathcal{X}_{v_{n_p}}$ be an open connected submanifold, a solution $\phi(t)$ of an autonomous multi-domain hybrid system \mathcal{H} in (2.2) is *periodic* if there exists a finite $T > 0$ such that $\phi(t + T) = \phi(t)$ for all $t \in [t_0, \infty)$ with an initial condition $x_0 \in \mathcal{X}$ at time t_0 , i.e., $\phi(t_0) = x_0$. A set $\mathcal{O} \subset \mathcal{X}$ is a periodic orbit of \mathcal{H} if $\mathcal{O} = \{\phi(t) | t \geq t_0\}$ for some periodic solution $\phi(t)$. If a periodic solution has discrete transitions, then the corresponding periodic orbit \mathcal{O} is not closed (see [57, 94]). We denote $\bar{\mathcal{O}}$ as the set closure of \mathcal{O} .

A periodic orbit \mathcal{O} is *transversal* to $S_e^{\mathcal{X}}$ if for all $x^* = \bar{\mathcal{O}} \cap S_e^{\mathcal{X}}$ the vector field f_v is not tangent to $S_e^{\mathcal{X}}$ at the point x^* , i.e., $\frac{\partial H_e(x)}{\partial x} f_{\text{sor}(e)}(x) \neq 0$. A periodic orbit \mathcal{O} is *transversal* if it is transversal to S_e for all $e \in E$, and is *period-one* if its closure intersects $S_e^{\mathcal{X}}$ at exactly one point for all $e \in E$. The orbital stability in the context of *Lyapunov* can be determined by the *Poincaré* return map of multi-domain hybrid systems followed from the general development in [58] and analogous to those of single domain hybrid systems defined in [95]. Let $\phi_v(t, x_0)$ be the maximal solution

of (2.2) with initial condition $x_0 \in \mathcal{X}_v$ at time $t = 0$. The *time-to-impact* function, $T_{I,v} : \mathcal{X}_v \rightarrow \mathbb{R}$, is the time from the initialization of the system on a given domain v to the first intersection with the subsequent guard $S_e^{\mathcal{X}}$ and is given as a partial map:

$$T_{I,v}(x_0) := \inf\{t \geq 0 \mid \phi_v(t, x_0) \in S_e^{\mathcal{X}}\}, \quad (2.3)$$

if $\exists t$ such that $\phi_v(t, x_0) \in S_e^{\mathcal{X}}$. Moreover, we denote that

$$x_e^- = \lim_{t \nearrow T_{I,v}} \phi_{\text{sor}(e)}(t), \quad (2.4)$$

$$x_e^+ = \lim_{t \searrow T_{I,v}} \phi_{\text{tar}(e)}(t) \quad (2.5)$$

be the left and right limits of a solution $\phi(t)$ at the guard $S_e^{\mathcal{X}}$ respectively. By definition, $x_e^- \in S_e^{\mathcal{X}} \cap \mathcal{X}_{\text{sor}(e)}$ and $x_e^+ = \Delta_e(x_e^-) \in \mathcal{X}_{\text{tar}(e)}$. The generalized Poincaré map $P_v : S_{\text{tar}^{-1}(v)}^{\mathcal{X}} \rightarrow S_{\text{sor}^{-1}(v)}^{\mathcal{X}}$ for a domain \mathcal{D}_v from its preceding guard $S_{\text{tar}^{-1}(v)}^{\mathcal{X}}$ to the subsequent guard $S_{\text{tar}^{-1}(v)}^{\mathcal{X}}$ is then defined as a partial map:

$$P_v(x_{\text{tar}^{-1}(v)}^-) := \phi_v(T_{I,v} \circ \Delta_{\text{tar}^{-1}(v)}(x_{\text{tar}^{-1}(v)}^-), \Delta_{\text{tar}^{-1}(v)}(x_{\text{tar}^{-1}(v)}^-)), \quad (2.6)$$

so that $x_{\text{sor}^{-1}(v)}^- = P_v(x_{\text{tar}^{-1}(v)}^-)$. For a multi-domain hybrid system in (2.2) with Γ , the Poincaré return map can be defined as the composition of generalized Poincaré maps of each domain v , starting at any point in the cycle Γ . For example, if we select $S_{e_1}^{\mathcal{X}}$ as the Poincaré section of the system (2.2), the Poincaré return map $P : S_{e_1} \rightarrow S_{e_1}$ is given as:

$$P := P_{v_1} \circ P_{v_{n_p}} \circ \dots \circ P_{v_2}. \quad (2.7)$$

If an orbit \mathcal{O} is period-one, then it has a fixed point $x^* = P(x^*)$, where $x^* = \bar{\mathcal{O}} \cap S_{e_1}^{\mathcal{X}}$. According to the Proposition 4. in [58], P is also the Poincaré return map for the single domain hybrid system:

$$\bar{\mathcal{H}} = \{\bar{\mathcal{X}}, \bar{S}^{\mathcal{X}}, \bar{\Delta}, \bar{F}\}, \quad (2.8)$$

where $\bar{\mathcal{X}} = \mathcal{X}_{e_1}$, $\bar{S}^{\mathcal{X}} = S_{e_1}^{\mathcal{X}}$, $\bar{F} = f_{v_1}$, and $\bar{\Delta} = \Delta_{e_{n_p}} \circ P_{v_{n_p}} \circ \dots \circ P_{v_2}$. By connecting multi-domain models to single-domain models, the stability of a periodic orbit \mathcal{O} of (2.2) can be determined by the stability of the equivalent single-domain model. According to the Theorem. 0 in [95], the stability of the periodic orbit can be determined by the stability of the fixed point x^* given that the periodic orbit is transversal to $\bar{S}^{\mathcal{X}}$. Results in [135] show that if $\bar{\mathcal{H}}$ in (2.8) is C^1 , then the partial map P is well-defined and differentiable in a neighborhood of a fixed point x^* , and hence, the stability property of the fixed point can be checked by evaluating the spectral radius of the Jacobian of P at x^* . More specifically, if all eigenvalues lie within the unit circle, i.e., have magnitude less than 1, then the periodic orbit is stable.

2.2 *Humanoid Locomotion as Hybrid Systems*

As discussed in previous section, dynamic locomotion of legged robots can be modeled as a hybrid control system \mathcal{HC} in (2.1). In this section, the construction of each element of the hybrid control system model for dynamical humanoid locomotion is developed based on the Lagrangian and kinematic constraints of a generalized robot model. The continuous dynamics is determined by the Lagrangian model of the mechanical system and contact points with the external environment, e.g. the ground. The transitions between different domains occur at the changes of contact points. Utilizing the formal definition of multi-domain hybrid systems, there is the necessary framework to discuss how to use the Lagrangian and contact constraints of the mechanical system of a bipedal robot to determine each element of the corresponding multi-domain hybrid system model.

2.2.1 *Lagrangian Model*

Motivated by the desire to consider robot models in a generalized position, i.e., not impose assumptions on contact constraints, we consider the floating base coordinates of the robot (see [58]). Assuming R_0 is a fixed inertial frame and R_b is a reference

frame rigidly attached to the base link on the robot, then the Cartesian position $p_b \in \mathbb{R}^3$ and the orientation $\phi_b \in SO(3)$ of R_b with respect to R_0 , respectively, compose the floating base coordinates of the robot, i.e., $q_b = (p_b, \phi_b)$. The multi-body system of a robot is often modeled as a kinematic tree of rigid links. Let $q_r \in \mathcal{Q}_r$ be the body coordinates of the robot, where \mathcal{Q}_r is the configuration space of the body coordinates, then

$$q = (q_b, q_r) \in \mathcal{Q} = \mathbb{R}^3 \times SO(3) \times \mathcal{Q}_r \quad (2.9)$$

is a set of generalized coordinates for the robot.

Remark 2.1. If the interested behavior of bipedal locomotion is planar, i.e., the motion of the robot is confined within the sagittal plane only, the base coordinates that are not on the sagittal plane will be constrained to zero. These coordinates includes the Cartesian position in y -axis and orientations of the base link along x -axis and z -axis. As a result, the base coordinates of the robot becomes

$$q_b = (p_x, p_z, \phi_y), \quad (2.10)$$

and correspondingly the generalized coordinates of the system is

$$q = (q_b, q_r) \in \mathcal{Q} = \mathbb{R}^2 \times SO(2) \times \mathcal{Q}_r. \quad (2.11)$$

Let $T\mathcal{Q}$ be the tangent bundle of \mathcal{Q} . The Lagrangian of the robot, $\mathcal{L} : T\mathcal{Q} \rightarrow \mathbb{R}$, is given as the difference between the kinetic energy, $T : T\mathcal{Q} \rightarrow \mathbb{R}$, and the potential energy, $V : \mathcal{Q} \rightarrow \mathbb{R}$, as:

$$\mathcal{L}(q, \dot{q}) := T(q, \dot{q}) - V(q). \quad (2.12)$$

The equations of motion of the multi-body system can be determined from the Lagrangian using the classic Euler-Lagrange equations [96]

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \Upsilon(q, \dot{q}, u), \quad (2.13)$$

where $\Upsilon(q, \dot{q})$ is the vector of external forces, such as ground reaction forces, actuator torques, and joint friction forces (if applicable).

2.2.2 Constrained Dynamics

We assume that the robot and the ground are modeled as rigid bodies. That is, the following assumptions should hold: the bodies cannot exert a force when there is no contact and the bodies that constitute a contact cannot interpenetrate, e.g., the distance between the ground and the feet cannot be negative. Therefore, these contacts characterize a set of *unilateral constraints* in the form given as [74]:

$$\lambda^T h(q) = 0, \quad h(q) \geq 0, \quad \lambda \geq 0, \quad (2.14)$$

where $h(q)$ is a vector represents a set of possible contact conditions of the feet and the ground; λ is a vector of normal ground reaction forces, which are sometimes referred as *Lagrangian multiplier* [86]. The orthogonality condition $\lambda^T h(q) = 0$ shows that for each possible contact i , if $h_i(q) > 0$, then there is no such contact and the corresponding normal force λ_i should be zero; on the other hand, if there exists a non-zero contact force, i.e., $\lambda_i > 0$, then a contact has been established, $h_i(q) = 0$. These unilateral constraints are described as “complementarity conditions”, and consequently the system can be modeled as a complementarity dynamical system [74, 93]. Such complementarity Lagrangian systems are very useful for “real-world” simulation of rigid body dynamics with contact constraints [15, 125], or trajectory optimization of constrained rigid body systems without *a priori* defined sequence of contact types and events [103]. In the context of multi-domain hybrid systems, however, the ordering of contact conditions is predetermined as a directed cycle. Hence, we use another commonly used method in the robotic community—*holonomic constraints*—to model the contact condition [58, 59]. The unilateral constraints in (2.14) describe necessary conditions for the control law design and trajectory optimization, and the condition at which either a contact distance and a force crosses zero determines each switching surface of the multi-domain hybrid system model.

Generally speaking, a holonomic constraint, $h_c : \mathcal{Q} \rightarrow \mathbb{R}$, is a function of the robot

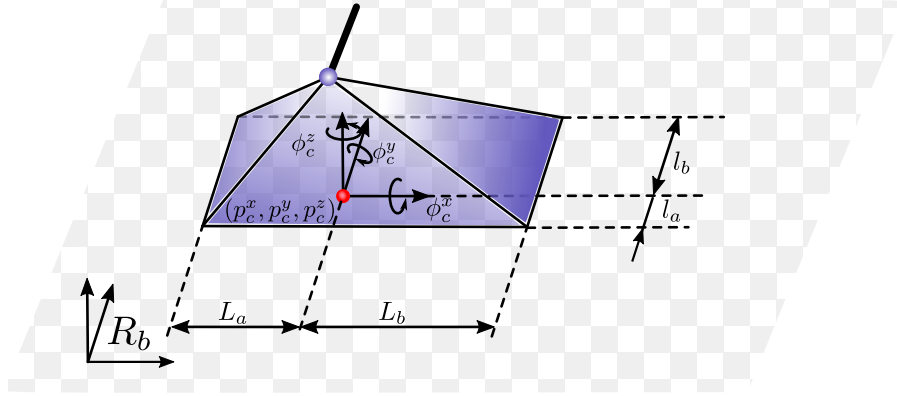


Figure 2.2: Illustration of the flat foot ground contact.

configuration that describes the kinematic constraints resulting from a contact that must be satisfied throughout the period of a given domain. For example, if the stance foot of a robot is assumed to be flat on the ground, then the associated holonomic constraints are defined as the Cartesian positions of a point on the stance foot link and the orientations of the stance foot link. To construct holonomic constraints of a rigid link that may have contacts with the external environment, we consider a reference frame, R_c , attached to the rigid link on which the contact point c is located, as shown in Figure 2.2. Let $[p_c^x, p_c^y, p_c^z]^T : \mathcal{Q} \rightarrow \mathbb{R}^3$ be the 3-dimensional Cartesian position of the point, $[\phi_c^x, \phi_c^y, \phi_c^z]^T : \mathcal{Q} \rightarrow SO(3)$ be the orientations¹ of R_c with respect to the inertial frame, R_0 , respectively. At most, we have six possible holonomic constraints associated with a given rigid link of the robot. In [58], Grizzle et al. presented thorough discussions of different types of foot contacts, including flat foot contact, heel or toe only contact, and point foot contact. Corresponding constraints on the ground reaction wrenches are also discussed in [58].

Let \mathcal{C}_v be an indexing set of all holonomic constraints defined on \mathcal{D}_v , we state the

¹Euler angles calculated from the transformation matrix can be used to express the orientation of the link to which the reference frame rigidly attached [96].

holonomic constraints of the domain as a vector contains all constraints:

$$h_v(q) = \{h_c(q)\}_{c \in \mathcal{C}_v} \equiv \text{constant}. \quad (2.15)$$

Let k_v is the number of total holonomic constraints, $h_v(q) \in \mathbb{R}^{k_v}$ constrains the positions and/or orientations of the contact links to be fixed, as a consequence, the associated velocities (or angular velocities) and accelerations (or angular accelerations) should be zero. Hence, the kinematic constraints are defined as

$$J_v(q)\dot{q} = 0, \quad (2.16)$$

where $J_v(q) = \frac{\partial h_v(q)}{\partial q}$ is the Jacobian of $h_v(q)$.

Given the mass, inertia and length properties of each link of a robot model, the equations of motion for a domain \mathcal{D}_v are determined by the classical Euler-Lagrange equation in (2.13):

$$D(q)\ddot{q} + H(q, \dot{q}) = B_v u + J_v^T(q)\lambda_v, \quad (2.17)$$

where $D(q)$ is the inertia matrix, $H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q)$ is the vector containing the Coriolis and gravity term, B_v is the actuator distribution matrix, and $\lambda_v : T\mathcal{Q} \times \mathcal{U}_v \rightarrow \mathbb{R}^{k_v}$ is a vector of contact *wrenches* containing the constraints forces and/or moments. To enforce the holonomic constraints, the differentiation of (2.16) should be zero, i.e.,

$$J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} = 0. \quad (2.18)$$

In addition, the wrenches $\lambda_v(q, \dot{q}, u)$ can be determined by solving (2.17) and (2.18) simultaneously (see [117]):

$$\lambda_v(q, \dot{q}, u) = -\Xi_v(q) \left(\dot{J}_v(q, \dot{q})\dot{q} + J_v(q)D^{-1}(q) (B_v(q)u - H(q, \dot{q})) \right), \quad (2.19)$$

where, for simplicity, $\Xi_v(q) = (J_v(q)D^{-1}(q)J_v^T(q))^{-1}$. Substituting the closed form solution of $\lambda_v(q, \dot{q}, u)$ into (2.17) yields the affine control system of the continuous dynamics,

$$\dot{x} = f_v(x) + g_v(x)u, \quad (2.20)$$

with $x = (q, \dot{q}) \in T\mathcal{Q}$ being the states of the system. The vector fields of the dynamical system, $f_v(x)$ and $g_v(x)$, are determined by:

$$f_v(x) = \begin{bmatrix} \dot{q} \\ D^{-1}(q) \left((J_v^T(q)\Xi_v(q)J_v(q)D^{-1}(q) - I) H(q, \dot{q}) - J_v^T(q)\Xi_v(q)\dot{J}_v^T(q, \dot{q})\dot{q} \right) \end{bmatrix}, \quad (2.21)$$

$$g_v(x) = \begin{bmatrix} 0 \\ D^{-1}(q) (I - J_v^T(q)\Xi_v(q)J_v(q)D^{-1}(q)) B_v(q) \end{bmatrix}, \quad (2.22)$$

respectively.

2.2.3 Domains and Guards

As discussed in the previous section, the foot contact with the ground is unilateral in essence, therefore, a certain set of conditions should be enforced on constraints forces and moments in order to satisfy the holonomic constraints assumption. Specifically, we state these conditions for a given domain \mathcal{D}_v in the form of inequalities:

$$\nu_v(q)\lambda_v(q, \dot{q}, u) \geq 0, \quad (2.23)$$

where $\nu_v(q)$ depends on the physical parameters of the system, such as the geometry of the foot and the friction coefficient with the ground. Furthermore, the admissible domain configuration should also consider other *unilateral constraints* related to the robot postures. For example, the non-stance foot should always be above the ground during the swing phase. In other words, the height of the non-stance foot should always be positive. We denote these unilateral constraints as $v_v(q) \geq 0$. Combining (2.23) and unilateral constraints (if present) together yields the domain of admissibility for each $v \in V$:

$$\mathcal{D}_v = \{(q, \dot{q}, u) \in T\mathcal{Q} \times \mathcal{U}_v | A_v(q, \dot{q}, u) \geq 0\}, \quad (2.24)$$

where

$$A_v(q, \dot{q}, u) = \begin{bmatrix} \nu_v(q)\lambda_v(q, \dot{q}, u) \\ \nu_v(q) \end{bmatrix}, \quad (2.25)$$

defines the boundary of the domain manifold.

Consequently, a guard S_e is a proper subset of the boundary of the domain, $\mathcal{D}_{\text{sor}(e)}$, determined by an edge condition associated with the transition from $\mathcal{D}_{\text{sor}(e)}$ to the subsequent domain, $\mathcal{D}_{\text{tar}(e)}$. Let $H_e(q, \dot{q}, u)$ be an appropriate element from the vector in (2.25) corresponding to a transition, then the guard is defined as

$$S_e = \{(q, \dot{q}, u) \in T\mathcal{Q} \times \mathcal{U}_v | H_e(q, \dot{q}, u) = 0, \dot{H}_e(q, \dot{q}, u) \neq 0\}. \quad (2.26)$$

When a guard is reached, it indicates a change in the contact. For bipedal locomotion, it could be the establishment of new contacts, e.g., the heel or sole of the swing foot hits the ground, or the break of existing contacts, e.g., the lift-off events of the swing foot. As a consequence, the states of the robot will undergo a discrete change. This discrete dynamics of the system can be encapsulated as a reset map $\Delta_e : S_e^{\mathcal{X}} \cap \mathcal{X}_{\text{sor}(e)} \rightarrow \mathcal{X}_{\text{tar}(e)}$ that maps the current states of the system on a guard to the subsequent domain. Given pre-impact states (q_e^-, \dot{q}_e^-) on $S_e^{\mathcal{X}} \cap \mathcal{X}_{\text{sor}(e)}$, the post-impact states (q_e^+, \dot{q}_e^+) of $\mathcal{X}_{\text{tar}(e)}$ are computed using a reset map Δ_e by assuming a perfectly plastic impact (if an impact occurs) [52, 75] for the reason that both robot and ground are modeled as rigid bodies. Following the presentation in [58], the robot configuration is invariant through impact, i.e., $q_e^+ = q_e^-$. Because the impact occurs instantaneously by the rigid body assumption, and due to the conservation of generalized momentum we have

$$D(q_e^+)(\dot{q}_e^+ - \dot{q}_e^-) = J_{\text{tar}(e)}^T(q_e^+) \delta F_{\text{tar}(e)}, \quad (2.27)$$

where $\delta F_{\text{tar}(e)} : T\mathcal{Q} \rightarrow \mathbb{R}^{k_{\text{tar}(e)}}$ is a vector of the intensity of impulsive contact wrenches over the infinitesimal impact event. In addition, the kinematic constraints of the

subsequent domain should be satisfied after the impact, i.e.,

$$J_{\text{tar}(e)}(q_e^+) \dot{q}_e^+ = 0. \quad (2.28)$$

Combining (2.27) and (2.28) together, we obtain the direct relationship between the pre- and post-impact velocities:

$$\dot{q}_e^+ = \underbrace{(I - D^{-1}(q_e^+) J_{\text{tar}(e)}^T(q_e^+) \Xi_{\text{tar}(e)}(q_e^+) J_{\text{tar}(e)}(q_e^+))}_{\Delta_{\dot{q}}(q)} \dot{q}_e^-. \quad (2.29)$$

Therefore, the reset map of a given guard can be written as

$$(q_e^+, \dot{q}_e^+) = \Delta_e(q_e^-, \dot{q}_e^-) := \begin{bmatrix} \Delta_q(q_e^-) \\ \Delta_{\dot{q}}(q) \dot{q}_e^- \end{bmatrix}, \quad (2.30)$$

where $\Delta_q(q_e^-)$ represents the change in the robot configuration, which is often an identity map (see Remark 2.2) and $\Delta_{\dot{q}}(q)$ is determined from (2.29).

Remark 2.2 (Coordinate Relabeling). In the study of symmetric walking gaits, a bipedal robot is often modeled in terms of “stance” and “non-stance” leg angles instead of physical “left” and “right” leg angles to reduce the number of discrete domains. In these cases, the robot configuration needs to be relabeled if there is a change in the “stance” and “non-stance” leg, e.g., when the “non-stance” leg becomes the “stance” leg. As a result, $\Delta_q(q_e^-)$ is no longer an identity map. This relabeling process can be denoted as:

$$\Delta_q(q_e^-) := \mathcal{R}(q_e^-), \quad (2.31)$$

where $\frac{\partial \mathcal{R}(q)}{\partial q}$ has full rank. It is important to note that this map is a linear map in many application [7, 146].

With the definition of each element of the multi-domain robotic locomotion, we can construct the corresponding multi-domain hybrid control system model \mathcal{HC} of the dynamic robotic locomotion as in (2.1), where \mathcal{D} is given in (2.24), S is given in (2.26), Δ is given in (2.30), and the affine control systems are defined in (2.20), respectively.

2.3 Feedback Control Law Design

In this section, virtual constraint is introduced as a mean to synthesis feedback controllers that realize stable and robust multi-domain robotic walking on humanoid robots. By formulating virtual constraints in a way such that the multi-domain hybrid system is hybrid invariant, the result is a feedback controller that drives the full-order model of the robot to a reduced-dimensional space—termed the *zero dynamics*—wherein the evolution of the system is dictated by the low dimensional hybrid dynamical system. The behavior on this reduced-order space can be used to encode the full-order behavior of the bipedal humanoid robot.

2.3.1 Virtual Constraints

Analogous to holonomic constraints, virtual constraints (also termed *outputs* in the control literature [7]) are defined as holonomic functions that modulate the robot states in order to achieve a certain desired behavior via stated-based feedback controllers [135]. The term “virtual” comes from the fact that these constraints are enforced via actuators instead of mechanical constraints. Given a set of functions of the mechanical configuration of a robot $y^a(q)$, and corresponding stated-based desired trajectories $y^d(q)$, virtual constraints is defined as the difference between actual and desired outputs:

$$y(q) := y^a(q) - y^d(q), \quad (2.32)$$

where $y^a(q)$ is termed as actual outputs and $y^d(q)$ is termed as desired output. In particular, desired outputs are often given in terms of particular parameterized smooth curves. The following definition provides a systematical construction of virtual constraints for the multi-domain hybrid control system of bipedal locomotion.

Definition 2.5. Given $v \in V$, $y_v^a = (y_{1,v}^a, y_{2,v}^a)$ is an admissible combination of robot outputs consisting of velocity-modulating outputs, $y_{1,v}^a : \mathcal{Q} \rightarrow \mathbb{R}^{n_{1,v}}$, and position-modulating outputs, $y_{2,v}^a : \mathcal{Q} \rightarrow \mathbb{R}^{n_{2,v}}$. Given m_v be the total number of admissible

controls and k_v be the total number of holonomic constraints, the total number of position-modulating outputs, $n_{2,v}$, is determined by

$$n_{2,v} = \begin{cases} m_v - n_{1,v}, & \text{if } m_v \leq N - k_v, \\ N - k_v - n_{1,v}, & \text{if } m_v > N - k_v. \end{cases} \quad (2.33)$$

Let \mathcal{O}_v be an indexing set for $y_{2,v}^a$ whereby $y_{2,v}^a(q) = [y_{2,o}^a(q)]_{o \in \mathcal{O}_v}$. A output combination is *independent* if the Jacobian of $y_v^a(q)$ has a full rank.

Remark 2.3. The idea of the velocity modulating output is originated from the study of human-inspired control. By analyzing the human locomotion data, Ames et. al. proposed that the forward velocity of the hip renders almost constant [13]. Hence, if there are enough admissible actuators present in the bipedal robot, the forward velocity can be controlled via feedback controllers as a velocity-modulating output. The admissible condition is determined by if the forward velocity of the hip is fully controllable. For example, we define the forward velocity as the velocity-modulating output if the stance foot is flat on the ground and both ankle and knee joints are actuated. In the case of point-feet bipedal robots [30, 136] or if the stance foot is in contact with the ground at the heel or toe only [144], the forward velocity cannot be controlled directly. Consequently, we will not define a velocity-modulating output for these cases.

Remark 2.4. The idea of virtual constraints is based on the method of computed torques [135], the total number of output should not exceed the total degrees of actuation and total unconstrained degrees of freedom, whichever is smaller. If the number of outputs is more than the total unconstrained degrees of freedom—which is the total degrees of freedom minus the number of holonomic constraints—the system becomes an over-constrained system. Such over-constrained systems should be avoided in the control law design.

Despite the desired output can be given in various forms, we typically define desired outputs as follows:

- the desired velocity-modulating output (if present) is assumed to be a *constant* [7] and denoted as \bar{v}_v , and
- the desired position-modulating outputs are given in term of Bézier polynomials:

$$y_{2,v}^d(\tau(q)) = [y_2^d(\tau(q), \alpha_o)]_{o \in \mathcal{O}_v},$$

dictated by a set of parameters $\{\alpha_o\}_{o \in \mathcal{O}_v}$ and a state-based parameterization of time $\tau(q)$.

In particular, the Bézier polynomial of degree M , determined by $M + 1$ coefficients, is given as:

$$y_2^d(\tau(q), \alpha_o) := \sum_{k=0}^M \alpha_o[k] \frac{M!}{k!(M-k)!} \tau(q)^k (1 - \tau(q))^{M-k}, \quad (2.34)$$

for each output $o \in \mathcal{O}_v$. The introduction of state-based parameterization of time, $\tau(q)$, is motivated by the desire to create an autonomous controller, considering the fact that autonomous systems are more robust than non-autonomous systems [135]. To have a well-defined continuous reference trajectory, $\tau(q)$ must be monotonic over a given time interval. This time interval could be the duration of a continuous domain or the entire gait cycle of one step. For the forward motion of humanoid robots, one possible candidate could be the linearized hip position of the robot [8, 38, 104]. Let $\theta(q)$ is a strictly monotonic (strictly increasing or decreasing) function of q , the parameterization of time is given as the normalization of the θ over the time duration,

$$\tau(q) := \frac{\theta(q) - \theta(q_0)}{\theta(q_f) - \theta(q_0)} \in [0, 1], \quad (2.35)$$

where q_0 and q_f are the configurations of the robot at the beginning and end of the particular time duration.

With the definition of outputs in hand, the *virtual constraints* of locomotion at a given domain are defined as the difference between the actual and desired outputs of the robot:

$$y_{1,v}(q, \dot{q}, \bar{v}_v) = \dot{y}_{1,v}^a(q, \dot{q}) - \bar{v}_v, \quad (2.36)$$

$$y_{2,v}(q, \alpha_v) = y_{2,v}^a(q) - y_{2,v}^d(\tau(q), \alpha_v), \quad (2.37)$$

for $v \in V$, where $(y_{1,v}, y_{2,v})$ is a set of admissible outputs as defined in Definition 2.5, and are relative degree 1 and (vector) relative degree 2 by definition (see [115] for the definition of relative degree), respectively, due to the assumption of the admissible output combinations are only depends on q . In particular, we combine the parameters of virtual constraints for a domain so that $\alpha_v = \bar{v}_v \cup \{\alpha_o\}_{o \in \mathcal{O}_v} \in \mathbb{R}^{p_v}$.

2.3.2 Input-Output Feedback Linearization

In this section, we introduce a state-based feedback controller to enforce the virtual constraints on the robot. With this objective in mind, differentiating the relative degree 1 output once and differentiating the relative degree 2 outputs twice yields:

$$\begin{bmatrix} \dot{y}_{1,v} \\ \ddot{y}_{2,v} \end{bmatrix} = \underbrace{\begin{bmatrix} L_{f_v} y_{1,v}(q, \dot{q}) \\ L_{f_v}^2 y_{2,v}(q, \dot{q}) \end{bmatrix}}_{L_{f_v}} + \underbrace{\begin{bmatrix} L_{g_v} y_{1,v}(q, \dot{q}) \\ L_{g_v} L_{f_v} y_{2,v}(q, \dot{q}) \end{bmatrix}}_{\mathcal{A}_v} u, \quad (2.38)$$

where L_{f_v} and L_{g_v} are the Lie derivatives with respect to the vector fields f_v and g_v , \mathcal{A}_v is the *decoupling matrix* which is invertible due to the specific choice of virtual constraints. To drive the virtual constraints $y_v = (y_{1,v}, y_{2,v}) \rightarrow 0$ exponentially, we consider the input-output feedback linearization control law,

$$u^{\alpha_v} = \mathcal{A}_v(-L_{f_v} + \mu_v), \quad (2.39)$$

for some $\mu_v \in \mathbb{R}^{n_{1,v} + n_{2,v}}$ yields linear output dynamics of the form:

$$\begin{bmatrix} \dot{y}_{1,v} \\ \ddot{y}_{2,v} \end{bmatrix} = \mu_v \quad (2.40)$$

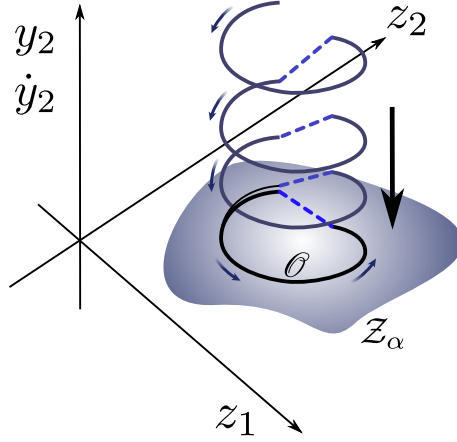


Figure 2.3: Illustration of the zero dynamics under the exponentially stable feedback controllers.

An example choice of μ_v is given by [11]:

$$\mu_v = \begin{bmatrix} -\varepsilon y_{1,v} \\ -2\varepsilon \dot{y}_{2,v} - \varepsilon^2 y_{2,v} \end{bmatrix} \quad (2.41)$$

for some $\varepsilon > 0$. In particular, it results in the output dynamics:

$$\begin{bmatrix} \dot{y}_{1,v} \\ \dot{y}_{2,v} \\ \ddot{y}_{2,v} \end{bmatrix} = \underbrace{\begin{bmatrix} -\varepsilon & 0 & 0 \\ 0 & 0 & I \\ 0 & -\varepsilon^2 I & -2\varepsilon I \end{bmatrix}}_{F_v^{cl}} \begin{bmatrix} y_{1,v} \\ y_{2,v} \\ \dot{y}_{2,v} \end{bmatrix}. \quad (2.42)$$

Since F_v^{cl} is Hurwitz by definition, the resulting linear dynamics is exponentially stable. The convergence rate of (2.42) is determined by ε due to the fact that all eigenvalues of F_v^{cl} are located at $-\varepsilon$. By choosing ε large enough, the feedback controller yields the output dynamics being rapidly attractive (see Figure 2.3).

2.3.3 Hybrid Zero Dynamics

Applying the feedback controllers given in (2.39) in each domain of the hybrid control system (2.3) yields a hybrid system model \mathcal{H}^α given as,

$$\mathcal{H}^\alpha = (\Gamma, \mathcal{X}, S^\mathcal{X}, \Delta, F^\alpha), \quad (2.43)$$

where F^α is a set of dynamical systems defined on \mathcal{X} , i.e., $\dot{x} = f_v^{\alpha v}(x) = f_v(x) + g_v(x)u^{\alpha v}$ with $x \in \mathcal{X}_v$. Moreover, the control law in (2.39) renders the *zero dynamics* submanifold \mathcal{Z}_v^α of \mathcal{X}_v , given in

$$\mathcal{Z}_v^\alpha = \{(q, \dot{q}) \in \mathcal{X}_v | y_{1,v} = 0, y_{2,v} = 0, \dot{y}_{2,v} = 0\}, \quad (2.44)$$

forward invariant and rapidly attractive over all continuous domain $v \in V$. That is, any solution that starts in \mathcal{Z}_v^α remains on \mathcal{Z}_v^α until it reaches a guard. In particular, we explicitly show that the zero dynamics is dependent on the parameter set α . For the simplicity of the notation, we drop the superscript α in the remainder of the thesis. As a result, the full-dimensional dynamical system is restricted to a low-dimensional invariant embedded submanifold of \mathcal{X}^v . However, the controller not necessarily guarantees that the reduced dimensional manifold being invariant through the impact dynamics due to the discrete change of joint velocities. Hence, we define that at a given edge $e \in E$, the reduced dimensional system is impact invariant if and only if $\mathcal{Z}_{\text{sor}(e)} \cap S_e^\mathcal{X} \neq \emptyset$ and

$$\Delta_e(\mathcal{Z}_{\text{sor}(e)} \cap S_e^\mathcal{X}) \subset \mathcal{Z}_{\text{tar}(e)}. \quad (2.45)$$

A manifold $\mathcal{Z} = \bigcup_{v \in V} \mathcal{Z}_v$ is called hybrid invariant if it is invariant over all domains of continuous dynamics and impact invariant through all discrete impacts, i.e., any solution that starts in \mathcal{Z} still remains in \mathcal{Z} even after impulse effects. If a feedback control law renders \mathcal{Z} hybrid invariant, then we say that the multi-domain hybrid control system has a *hybrid zero dynamics* (HZD). The hybrid invariance of the hybrid zero dynamics also reflects on the generalized partial map P_v in (2.6), i.e.,

$$P_v(S_{\text{tar}^{-1}(v)}^\mathcal{X} \cap \mathcal{Z}_v) \subset S_{\text{sor}^{-1}(v)}^\mathcal{X} \cap \mathcal{Z}_v. \quad (2.46)$$

By enforcing hybrid zero dynamics, the full order behavior of the hybrid system is restricted to a reduced-dimensional dynamical system that is independent of the control inputs [94]. The advantage of studying the hybrid zero dynamics manifold is

that the evaluation of orbital stability of the full order system can be performed on the reduced dimensional restricted system. With the far less degrees of freedom, the numerical computation time on the hybrid zero dynamics manifold could be reduced significantly. In some special cases, the solutions of restricted Poincaré return map could be obtained analytically [8, 135, 137].

To construct the restricted hybrid system mode, let $\xi_v \in \mathcal{Z}_v \subset \mathbb{R}^{\ell_v}$ be the local coordinates of the \mathcal{Z}_v and $\eta_v \in \mathcal{X}_v^a \subset \mathbb{R}^{n_v - \ell_v}$ be the vector of controlled normal states. In addition, suppose that there exist local coordinates transformations $\Phi_v^z : \mathcal{X}_v \rightarrow \mathcal{Z}_v$ and $\Phi_v^\eta : \mathcal{X}_v \rightarrow \mathcal{X}_v^a$, so that $(\xi_v, \eta_v) = (\Phi_v^z(x), \Phi_v^\eta(x)) := \Phi_v(x)$. With this coordinate transformation, the affine control system in (2.20) can be represented in the zero dynamics normal form as [115]:

$$\dot{\eta}_v = b_v(\eta_v, \xi_v) + a_v(\eta_v, \xi_v)u \quad (2.47)$$

$$\dot{\xi}_v = z_v(\eta_v, \xi_v) \quad (2.48)$$

where the vector fields b_v , a_v , and z_v are assumed to be locally Lipschitz continuous. The affine control system in (2.47) is termed as the output dynamics, and sometimes also called as transverse dynamics; the autonomous system in (2.48) is termed as the zero dynamics. In addition, under the feedback control law in (2.39), η_v vanishes on the zero dynamics manifold, i.e., $b_v(\mathbf{0}, \xi_v) = \mathbf{0}$. Hence, the dynamic equation of the restricted manifold \mathcal{Z}_v can be given as

$$\dot{\xi}_v = z_v(\mathbf{0}, \xi_v). \quad (2.49)$$

Using the discussions in Section 2.1.3, we denote the solution of (2.49) by $\phi^z(t, \xi_0)$ with $\xi_0 \in \mathcal{Z}$, and correspondingly, the restricted periodic orbit of the zero dynamics is denoted as $\mathcal{O}_{\mathcal{Z}} \subset \mathcal{Z}$. On the basis of (2.46), we define the *restricted Poincaré map*, $\rho_v : S_{\text{tar}^{-1}(v)}^{\mathcal{X}} \cap \mathcal{Z}_v \rightarrow S_{\text{sor}^{-1}(v)}^{\mathcal{X}} \cap \mathcal{Z}_v$, as $\rho_v = P_v|_{\mathcal{Z}}$. Suppose that $\Delta_e^{\mathcal{Z}} := \Delta_e|_{S_e \cap \mathcal{Z}_{\text{sor}(e)}}$ is the restricted reset map for the zero dynamics and $T_{\rho,v}(\xi_0) := T_{I,v}(x_0|_{\mathcal{Z}_v})$ is the

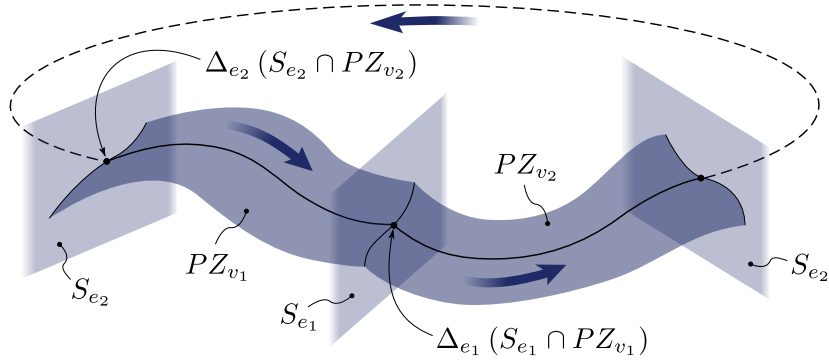


Figure 2.4: Illustration of the restricted hybrid invariant periodic orbit of a two-domain hybrid system.

restricted time-to-impact function, then the restricted Poincaré map for the domain \mathcal{D}_v can be explicitly expressed as:

$$\rho_v(\xi_{\text{tar}^{-1}(v)}^-) := \phi_v^z(T_{\rho,v} \circ \Delta_{\text{tar}^{-1}(v)}^z(\xi_{\text{tar}^{-1}(v)}^-), \Delta_{\text{tar}^{-1}(v)}^z(\xi_{\text{tar}^{-1}(v)}^-)). \quad (2.50)$$

Particularly, the Poincaré return map for the multi-domain hybrid zero dynamics can be defined similarly as:

$$\rho := \rho_{v_1} \circ \rho_{v_{np}} \circ \cdots \circ \rho_{v_2}, \quad (2.51)$$

therefore, ρ is also the restricted Poincaré map of the single domain hybrid system:

$$\mathcal{H}_Z^{(\alpha,\varepsilon)} = \{\bar{Z}, \bar{S}_Z, \bar{\Delta}_Z, \bar{F}_Z\}, \quad (2.52)$$

where $\bar{Z} = Z_{v_1}$, $\bar{S}_Z = S_{e_1}^X \cap Z_{v_1}$, $\bar{\Delta}_Z = \Delta_{e_{np}}^Z \circ \rho_{v_{np}} \circ \cdots \circ \rho_{v_2}$, and $\bar{F}_Z = z_{v_1}$, respectively. Hence, the invariance of the multi-domain hybrid zero dynamics which is guaranteed by the satisfaction of the multi-domain HZD constraints in (2.45) is equivalent to the invariance of the hybrid zero dynamics of the single-domain hybrid system in (2.52).

The stability of hybrid systems, particularly those exhibiting periodic behaviors, is determined by the existence and stability properties of periodic orbits. It is shown in [9, 95] that if there exists a feedback control law that ensures the invariant manifold is sufficiently rapidly attractive, then the existence and stability of the periodic orbit

\mathcal{O} of the full order system can be determined from those of the reduced-dimensional zero dynamics periodic orbit, $\mathcal{O}_{\mathcal{Z}} \in \mathcal{Z}$, through the canonical embedding $\iota : \mathcal{Z} \rightarrow \mathcal{X}$. In other words, if the system has HZD and there exists a hybrid periodic orbit, $\mathcal{O}_{\mathcal{Z}}$, of $\mathcal{H}_{\mathcal{Z}}^{\alpha}$, as shown in Figure 2.4, then the full order system also has a hybrid periodic orbit \mathcal{O} obtained by $\mathcal{O} = \iota(\mathcal{O}_{\mathcal{Z}})$.

Theorem 2.1 (Periodic orbits of the multi-domain system [146]). *Suppose that $\mathcal{O}_{\mathcal{Z}}$ be an exponentially stable transverse periodic orbit of the corresponding hybrid zero dynamics system (2.52), and there exists $\varepsilon > 0$ such that the feedback control law defined in (2.39) renders \mathcal{Z}_v exponentially attractive for all $v \in V$, then $\mathcal{O} = \iota(\mathcal{O}_{\mathcal{Z}})$ is an exponentially stable periodic orbit for the full order dynamics of the multi-domain hybrid system (2.43).*

Proof. The proof comes directly from the *Proposition 4* in [58] along with the proof in *Theorem 2* in [10] with the reconstructed single-domain hybrid zero dynamics system (2.52). From discussion in [10], we know the feedback control law as defined in (2.39) is *Lipschitz continuous* and belong to the *rapidly exponential stable-control Lyapunov function* (RES-CLF) based control set K_{ε} as defined in [10], which therefore illustrates the proof of this theorem. \square

Remark 2.5 (Partial Hybrid Zero Dynamics). In fact, if the reset map involves a plastic impact, it is quite difficult to guarantee the velocity-modulating output to be a constant due to the discrete changes in joint velocities caused by impact. Hence, we enforce hybrid zero dynamics condition only on virtual constraints that related to the relative degree 2 outputs: $y_{2,v}$. We term the resulting embedded submanifold the *partial zero dynamics* manifold (see [7]), given by:

$$\mathcal{PZ}_v = \{(q, \dot{q}) \in \mathcal{X}_v | y_{2,v} = 0, \dot{y}_{2,v} = 0\}. \quad (2.53)$$

The hybrid invariance of the partial zero dynamics can be defined similarly as above. If there exists a set of parameters $\alpha = \{\alpha_v\}_{v \in V}$ so that for any edge $e \in E$, the

submanifold $\mathcal{PZ}_{\text{sor}(e)}$ is impact invariant if, and only if

$$\Delta_e(S_e^{\mathcal{X}} \cap \mathcal{PZ}_{\text{sor}(e)}) \subset \mathcal{PZ}_{\text{tar}(e)}. \quad (2.54)$$

A manifold $\mathcal{PZ} = \{\mathcal{PZ}_v\}_{v \in V}$ is called hybrid invariant if it is invariant over all domains of continuous dynamics and impact invariant through all discrete dynamics, i.e., solutions that start in \mathcal{PZ} remain in \mathcal{PZ} , even after impulse effects. If a feedback control law renders \mathcal{PZ} hybrid invariant, then we say that the multi-domain hybrid control system has a *partial hybrid zero dynamics* (PHZD). Note that, the reason we term this dynamics as *partial zero dynamics* is because the actual zero dynamics also contains the relative degree one output which is in fact *controlled*. For a domain that only has relative degree two outputs, the \mathcal{PZ}_v surface is actually the *full zero dynamic surface*, i.e., $\mathcal{PZ}_v = \mathcal{Z}_v$.

For the partial zero dynamics surface, we could explicitly define the relative degree one output as part of ξ_v satisfying (2.48) with applying a pre-feedback controller. That is to say, because of the full control authority, we could carefully shape the dynamics of relative degree one outputs to the form of (2.48), which can be reasonably viewed as a “controllable” zero dynamics. This discussion is important because it allows the general construction of the zero dynamics as in (2.47) and (2.48) to suit for all domains with different types of actuation and outputs defined. Hence the discussion developed for the hybrid zero dynamics case can also be applied directly for the partial hybrid zero dynamics. The partial hybrid zero dynamics (PHZD) can be seen as a more general formulation of hybrid zero dynamics, therefore, we use the partial hybrid zero dynamics in the remainder of the thesis in all cases.

2.3.4 Control Lyapunov Functions

In the previous section, the transverse dynamics in (2.47) is stabilized with the input-output feedback linearization controllers in (2.39). In fact, there are many other feedback controllers can be utilized to stabilize the output dynamics for a given set of

virtual constraints parameters. In this section, we briefly survey a family of rapidly exponentially stabilizing control Lyapunov functions (CLFs) in the context of hybrid system based on the classical notations of CLF in control literature [16, 118, 119].

Definition 2.6 (RES-CLF [9, 10, 14]). A continuously differentiable function $V_v^\varepsilon : \mathcal{X}_v^a \rightarrow \mathbb{R}$ is a **rapidly exponentially stabilizing control Lyapunov function (RES-CLF)** if there exist positive constants $c_1, c_2, c_3 > 0$ such that for all $0 < \varepsilon < 1$

$$c_1 \|x\|^2 \leq V_v^\varepsilon(\eta_v) \leq \frac{1}{\varepsilon^2} c_2 \|x\|^2 \quad (2.55)$$

$$\inf_{u \in \mathcal{U}_v} \left[L_{a_v} V_v^\varepsilon(\eta_v, \xi_v) + L_{b_v} V_v^\varepsilon(\eta_v, \xi_v) u + \frac{1}{\varepsilon} c_3 V_v^\varepsilon(\eta_v) \right] \leq 0. \quad (2.56)$$

Motivated by [44], we can consider a set

$$K_v^\varepsilon(\eta_v, \xi_v) = \{u \in \mathcal{U}_v : L_{a_v} V_v^\varepsilon(\eta_v, \xi_v) + L_{b_v} V_v^\varepsilon(\eta_v, \xi_v) u + \frac{1}{\varepsilon} c_3 V_v^\varepsilon(\eta_v) \leq 0\} \quad (2.57)$$

that consists of a family of control inputs that yields $\dot{V}_v^\varepsilon(\eta_v, \xi_v, u) \leq -\frac{c_3}{\varepsilon} V_v^\varepsilon(\eta_v)$ and exponentially stabilizes the output dynamics in (2.47) at a rate of ε . This can be verified from the definition of the RES-CLF [9]. More importantly, this results in a family of optimal control inputs, given as

$$m_v^\varepsilon(\eta_v, \xi_v) = \operatorname{argmin}\{\|u\| : u \in K_v^\varepsilon(\eta_v, \xi_v)\} \quad (2.58)$$

$$= \operatorname{argmin}\{\|u\| : \psi_{0,v}^\varepsilon(\eta_v, \xi_v) + \psi_{1,v}^\varepsilon(\eta_v, \xi_v)^T u \leq 0\}, \quad (2.59)$$

where

$$\psi_{0,v}^\varepsilon(\eta_v, \xi_v) = L_{a_v} V_v^\varepsilon(\eta_v, \xi_v) + \frac{1}{\varepsilon} c_3 V_v^\varepsilon(\eta_v) \quad (2.60)$$

$$\psi_{1,v}^\varepsilon(\eta_v, \xi_v) = L_{b_v} V_v^\varepsilon(\eta_v, \xi_v)^T, \quad (2.61)$$

which can be solved by a quadratic program (QP) that minimizes the control efforts:

$$m_v^\varepsilon(\eta_v, \xi_v) = \operatorname{argmin}_{u \in \mathcal{U}_v} u^T u \quad (2.62)$$

$$\text{s.t.} \quad \psi_{0,v}^\varepsilon(\eta_v, \xi_v) + \psi_{1,v}^\varepsilon(\eta_v, \xi_v)^T u \leq 0.$$

The end result is a family of optimal controllers solved via CLF based quadratic program to stabilize the transverse dynamics. This has been successfully applied to locomotion and manipulation in bipedal robots [11, 65], and has been implemented experimentally on planar bipedal walking [45].

2.4 Summary

In this chapter, we present the mathematical background of the hybrid zero dynamics framework for dynamic locomotion of humanoid robots. In particular, the generalized multi-domain hybrid system model with a predetermined directed cycle is introduced to model the behavior of humanoid locomotion. By properly designing a set of virtual constraints to enforce via feedback control, hybrid invariance can be achieved, assuring stable periodic locomotion despite periodic leg impacts and contact changes. The hybrid zero dynamics framework provides a general approach to model, design, and control legged robotic locomotion, even in the presence of underactuation.

With the mathematical model of the humanoid locomotion defined, the next task is to determine a feasible gait trajectory for the hybrid dynamical system so that the proposed control can be applied to realize dynamic locomotion on the humanoid robots. This process is typically relegated to a nonlinear trajectory optimization problem. In the next two chapters, we introduce several different approaches to formulate the gait optimization problem, particularly within the context of the hybrid zero dynamics framework.

CHAPTER III

DIRECT SHOOTING METHODS

In this chapter, we present how to use nonlinear optimization techniques to generate feasible locomotion gaits for the legged robots. We start with the general statement of the gait optimization problem, and then introduce the commonly used direct shooting method as a mean to solve the nonlinear optimization problem. Two different approaches are discussed in this chapter, and illustrated via specific examples.

3.1 Gait Optimization Problem

The goal of gait generation of the legged robotic locomotion in essence is to determine a trajectory of robot states, $x(t)$, of the multi-domain hybrid system model defined in (2.1). This problem can be formulated as a infinite-dimensional *trajectory optimization* or *optimal control problem*, in which the state trajectory $x(t)$ and control inputs $u(t)$ are determined by optimizing a certain cost function while satisfying both path and boundary constraints. This problem can be formulated as a *optimal control* (or *trajectory optimization*) problem as follows: determine the a solution or trajectory, $x(t)$, of the system defined in Definition 2.3 under a control profile $u(t)$,

$$(x^*(t), u^*(t)) = \operatorname{argmin} \mathcal{J}(x(t), u(t)) \quad (3.1)$$

that minimize the cost function given in the Bolza form as

$$\mathcal{J} := \sum_{v \in V} \left(\mathbf{E}_v(x(t_v^0), u(t_v^0), x(t_v^f), u(t_v^f)) + \int_{t_v^0}^{t_v^f} \mathbf{L}_v(x(t), u(t)) dt \right), \quad (3.2)$$

subject to the dynamic equations as in (2.20)

$$\dot{x} = f_v(x(t)) + g_v(x(t))u(t), \quad (3.3)$$

the *path* constraints

$$\mathbf{C}_v^{\min} \leq \mathbf{C}_v(x(t), u(t)) \leq \mathbf{C}_v^{\max}, \quad (3.4)$$

and the *boundary* constraints

$$\mathbf{B}_v^{\min} \leq \mathbf{B}_v(x(t_v^0), u(t_v^0), x(t_v^f), u(t_v^f)) \leq \mathbf{B}_v^{\max}, \quad (3.5)$$

for all $v \in V$. In particular, \mathbf{E}_v is the terminal cost and \mathbf{L}_v is the running cost, and t_v^0 and t_v^f is the initial and final time correspond to the solution on the domain \mathcal{D}_v , respectively. It is obvious to note that this formulation corresponds to the definition of the multi-domain hybrid system. In particular, the admissible domains \mathcal{D} determine the path constraints in (3.4), and the guard S and reset maps Δ are encoded in the boundary constraints in (3.5).

In the classic trajectory optimization problem, the control $u(t)$ is considered as an “open-loop” controller and is optimized alongside the state trajectory as an input profile. In the framework of hybrid zero dynamics, however, the control inputs $u(t)$ are determined by the virtual constraints based feedback controllers as in (2.39). Hence, the task of the hybrid zero dynamics based gait optimization problem is to determine a set of parameters $\alpha = \{\alpha_{v_1}, \alpha_{v_2}, \dots, \alpha_{v_{n_p}}\}$ so that the system has a hybrid invariant periodic orbit, $\mathcal{O}_{\mathcal{Z}}$, on the reduced dimensional hybrid zero dynamics submanifold.

The most intuitive approach to formulating such an optimal control problem is arguably the *direct shooting* method. In this chapter, we present how to formulate the hybrid zero dynamics based gait optimization problem using direct shooting methods. In particular, two different approaches are introduced: the *direct single shooting* method and the *direct multiple shooting* method. In a detailed survey paper, [108] makes a clear conceptual distinction between indirect methods (such as indirect shooting, indirect multiple shooting) and direct methods (e.g. the direct shooting and direct multiple shooting methods addressed in this chapter). In practice, direct methods are significantly more common in legged robot control than indirect methods. So

for ease of communication, we both drop the “direct” adjective and rename “direct single shooting” to “single shooting” [83] to verbally distinguish it from “multiple shooting.”.

3.2 *Single Shooting Optimization*

A subset of direct optimization methods [108], the direct single shooting method (heretofore referred to as *single shooting* method) simulates the dynamics via a single time-marching numerical integration, and then evaluates the objective and all constraints along that solution. Such “single shooting” is the most prevalent formulation for optimizing virtual constraints in the context of HZD.

3.2.1 Problem Formulation

Let

$$\alpha = \{\alpha_{v_1}, \alpha_{v_2}, \dots, \alpha_{v_{n_p}}\} \in \mathbb{R}^p \quad (3.6)$$

be a set of virtual constraints parameters, where $p = \sum_{v \in V} \dim(\alpha_v)$, and $\xi^- \in \bar{S}_{\mathcal{Z}}$ be a point on the guard of the connected single domain hybrid restricted system given in (2.52), the HZD based gait optimization problem can be formulated as:

$$(\alpha^*, \xi^{-*}) = \underset{(\alpha, \xi^-)}{\operatorname{argmin}} \sum_{v \in V} \left(\mathbf{E}_v^{\mathcal{Z}}(\xi_{v_i}(t_v^0), \xi_v(t_v^f), \alpha_v) + \int_{t_v^0}^{t_v^f} \mathbf{L}_v^{\mathcal{Z}}(\xi_v(t), \alpha_v) dt \right) \quad (3.7)$$

$$\text{s.t. } \Delta_e^{\mathcal{Z}}(\mathcal{Z}_{\text{sor}(e)} \cap S_e^{\mathcal{X}}) \subset \mathcal{Z}_{\text{tar}(e)}, \quad \forall e \in E, \quad (3.8)$$

$$\xi^- = \rho(\xi^-), \quad (3.9)$$

$$\mathbf{C}_v^{\min} \leq \mathbf{C}_v^{\mathcal{Z}}(\xi_v(t), \alpha_v) \leq \mathbf{C}_v^{\max}, \quad (3.10)$$

$$\mathbf{B}_v^{\min} \leq \mathbf{B}_v^{\mathcal{Z}}(\xi_v(t_v^0), \xi_v(t_v^f), \alpha_v) \leq \mathbf{B}_v^{\max}. \quad (3.11)$$

where $\xi_v(t)$ is the solution of zero dynamics subject to the ODE given in (2.49), and $\mathbf{E}_v^{\mathcal{Z}}$, $\mathbf{L}_v^{\mathcal{Z}}$, $\mathbf{C}_v^{\mathcal{Z}}$, and $\mathbf{B}_v^{\mathcal{Z}}$ are the terminal costs, running costs, path constraints and terminal constraints computed on the zero dynamics manifold, respectively. The

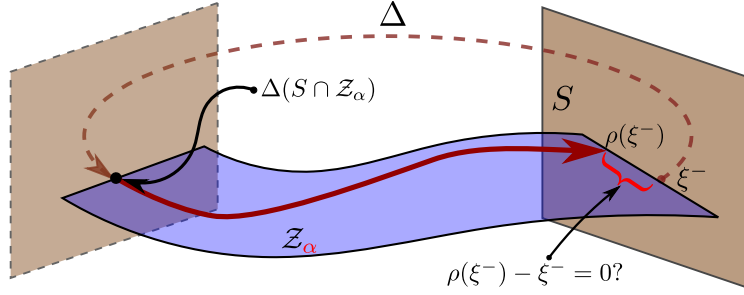


Figure 3.1: Illustration of the single shooting hybrid zero dynamics gait optimization. The optimizer picks a α and ξ^- at every iteration, where α determine the reduced dimensional surface \mathcal{Z}_α and ξ^- determine the initial point of the restricted trajectory, then evaluate the zero dynamics forward to check if all constraints are satisfied. If neither constraints are satisfied nor the cost function has the optimal value, the optimizer search for the next pair of parameters and boundary values (i.e., α and ξ^-). The iteration repeats until the optimizer reaches an optimal solution.

full-order states and the feedback control inputs are calculated from the canonical embedding ι and the input-output feedback linearization control law defined in (2.39). The evaluation the cost function and constraints is performed by simulating the hybrid zero dynamics via a single time-marching numerical integration.

In particular, the constraint (3.8) ensures that the system is hybrid invariant under the parameters set α^* and the constraint (3.9) guarantees that there exist a periodic orbit \mathcal{O}_Z on the hybrid zero dynamics manifold. Moreover, ξ^{-*} is the fixed point of the restricted periodic orbit by the constraints (3.9), i.e., $\xi^{-*} = \rho(\xi^{-*})$. Figure 3.1 illustrates the procedure of this single shooting optimization. In general, it is very difficult to formulate the stability condition of the fixed point in the optimization, therefore, the stability of the fixed point is verified *a posteriori*. In some special cases, however, the Jacobian of the Poincaré return map ρ can be obtained analytically, so that it could be evaluated directly in the optimization to make sure the resulting periodic orbit is stable [8, 135].

The single shooting formulation has been widely used in the hybrid zero dynamics literature [7, 120, 136, 137] due to its simple and straightforward description of the

problem. Particularly, when the system has only one discrete domain and the zero dynamics has dimension of only two, the hybrid invariant and periodic constraints can be constructed explicitly. This feature has been successfully employed in the single shooting optimization, as illustrated in the following example.

3.2.2 Case Study: 5-link Planar Underactuated Biped

Arguably the most common application of the single shooting method for the hybrid zero dynamics gait optimization is the single-domain walking of 5-link planar point feet biped [7, 135]. The robot has only one degree of underactuation.

Robot Model. As shown in Figure 3.2a, the generalized coordinates of such a robot is given by

$$q = (p_x, p_z, q_{sf}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk}),$$

where p_x and p_z are the positions of the stance foot with respect to the world frame, q_{sf} , q_{sk} , and q_{sh} represents stance foot, knee and hip angles, respectively, and q_{nsk} and q_{nsh} represents non-stance knee and hip angles respectively, in which the last four joints are actuated.

Due to the plastic impact of the rigid links, the double support phase is regarded as instantaneous [58]. Thus, the walking of such a biped has only one continuous domain as the non-stance leg swings in the air and one discrete event when the non-stance foot hits the ground. We model it as a *single domain hybrid control system* as a special case of (2.1). The holonomic constraints of \mathcal{D} are the positions of the stance foot (p_x, p_z) and the guard condition is the height of the non-stance foot, $h_{nsh}(q)$. The reset map of the system is then given by (2.30), with a coordinate relabeling occurring at impact to switch the stance and non-stance leg joints [7].

Zero Dynamics. Since the robot has four actuated joints, the following four virtual constraints are considered, with the actual outputs defined as

$$y_2^a(q) = \begin{bmatrix} q_{sk} & q_{nsk} & q_{tor} & \delta m_{nsl} \end{bmatrix}^T, \quad (3.12)$$

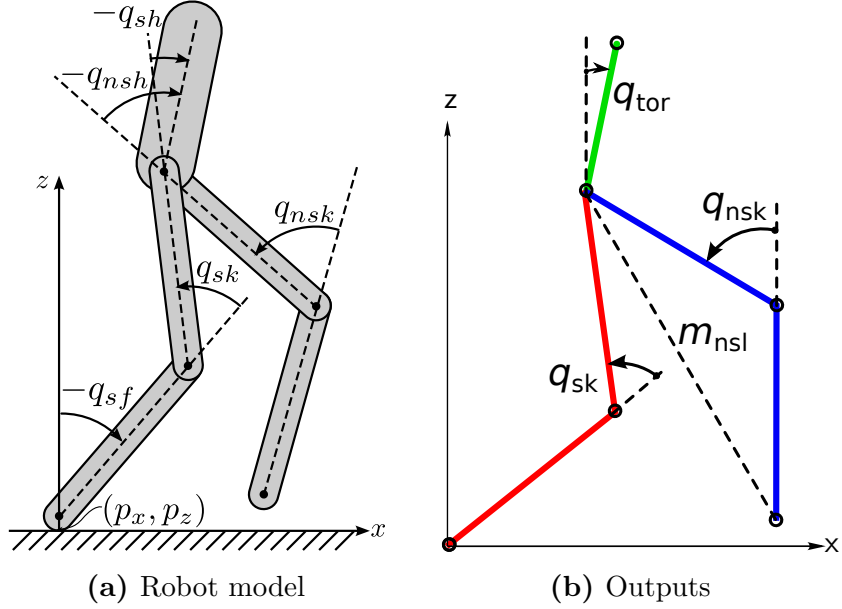


Figure 3.2: Model configuration and outputs representation of a 5-link point feet planar robot.

where δm_{nsl} is the linearized non-stance slope [7], as shown in Figure 3.2b. The desired outputs, $y_2^d(\tau(q), \alpha)$, are given in terms of the parameters set α and the parameterized time given in (2.35). In particular, we pick parameterized time $\theta(q)$ as

$$\theta(q) := q_{sf} \quad (3.13)$$

considering that the stance foot angle is monotonically decreasing.

With the definition of virtual constraints given in (3.12), the resulting zero dynamics \mathcal{Z} has only dimension of two due to the fact that there is only one underactuated degree of freedom. We begin by picking the zero dynamics coordinates as

$$\xi_1 = \theta(q), \quad (3.14)$$

$$\xi_2 = \gamma_0(q)\dot{q} \quad (3.15)$$

based on the discussion in Appendix A. In fact, γ_0 is the row in the inertia matrix $D(q)$ that corresponds to the underactuated degree of freedom in this case [135]. Moreover, the zero dynamics equation can be given in a way so that both $\dot{\xi}_1$ and $\dot{\xi}_2$ can be written as functions of ξ_1 . For more details of the construction of the zero

dynamics equation, we refer to Chapter 5 of [135]. An advantage of this formulation is the restricted Poincaré map $\rho : S \cap \mathcal{Z} \rightarrow S \cap \mathcal{Z}$ is a one dimensional partial map that is only a function of ξ_2 . Thus, $S \cap \mathcal{Z}$ is an one-dimensional embedded smooth submanifold of \mathcal{D} , and as a result, there exist a unique point $q^- \in \mathcal{Q}$ such that

$$\begin{bmatrix} y_2(q^-, \alpha) \\ h_{nsf}(q^-) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}. \quad (3.16)$$

That is, given a parameter set α , $q^-(\alpha) \in S \cap \mathcal{Z}$ can be explicitly solved [135].

This fact has been exploited in [7], so that the HZD constraint in (3.8) and (3.9) of the hybrid zero dynamics gait optimization can be re-formulated in terms of parameters α only. Let $\vartheta(\alpha) = q$ be the solution to

$$\begin{bmatrix} y_2(\Delta_q q) \\ h_{nsf}(q) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \quad (3.17)$$

and $\dot{\vartheta}(\alpha)$ is then given by

$$\dot{\vartheta}(\alpha) := \begin{bmatrix} \frac{\partial y_2}{\partial q}(\vartheta(\alpha)) \\ \gamma_0(\vartheta(\alpha)) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}. \quad (3.18)$$

Further, assume that $\Delta_{\mathcal{Z}}(q^-)$ be the restricted reset map of the zero dynamics, i.e., $\xi_2^+ = \Delta_{\mathcal{Z}}(q^-)\xi_2^-$, and $V_{\mathcal{Z}}(\xi_1)$ be the potential energy of the zero dynamics, respectively. Based on Theorem 3 in [7], if a set of parameters α that satisfies the following constraints:

$$y_2(\vartheta(\alpha)) = \mathbf{0}, \quad (3.19)$$

$$dy_2(\Delta_q \vartheta(\alpha)) \Delta_{\dot{q}}(\vartheta(\alpha)) \dot{\vartheta}(\alpha) = \mathbf{0}, \quad (3.20)$$

$$dh_{nsf}(\vartheta(\alpha)) \dot{\vartheta}(\alpha) < 0, \quad (3.21)$$

$$\frac{\Delta_{\mathcal{Z}}(\vartheta(\alpha))^2}{1 - \Delta_{\mathcal{Z}}(\vartheta(\alpha))^2} V_{\mathcal{Z}}(\vartheta(\alpha)) + V_{\mathcal{Z}}^{\max} < 0, \quad (3.22)$$

$$0 < \Delta_{\mathcal{Z}}(\vartheta(\alpha)) < 1, \quad (3.23)$$

then it yields hybrid zero dynamics $\Delta(S \cap \mathcal{Z}) \subset \mathcal{Z}$. Moreover, there exist an $\tilde{\varepsilon} > 0$ such that for all $\varepsilon > \tilde{\varepsilon}$ the hybrid system has a stable periodic orbit $\mathcal{O}_{\mathcal{Z}}$ with a fixed point $(q^-, \dot{q}^-) \in S \cap \mathcal{Z}$ can be computed explicitly from the parameters.

This property of explicitly determining the stable periodic orbit only in terms of the parameters set greatly reduced the difficulty of the gait optimization. However, it is benefited from the fact that there is only one degree of freedom in the system. The same method could not extend to general cases, especially for robots that have multiple degrees of underactuation.

3.2.3 Issues

The single shooting approach of optimizing only parameters and boundary state values reflects an instinctive desire to reduce the number of design variables for the optimization. Intuitively, one might assume that such minimization of the nonlinear programming problem's (NLP's) dimensionality would be an advisable practice for maximizing an optimization's speed and reliability. In practice, this approach encounters several issues that impede the convergence of the optimization, especially when the nonlinearity of the system increases.

The major concern is the sensitivity of the optimization variables. Any change in the parameters or the initial (or boundary) values propagates along the evaluation of the trajectory. The consequence is that the constraints behave very nonlinearly with respect to NLP variables, thereby making the optimization difficult to converge.

Secondly, the computational cost of calculating the gradient information of the cost and constraints also causes the evaluation of the optimization very slow. Very often, the cost function and constraints can only be computed numerically due to their complicated dependence on the optimization variables. Hence, the only way to calculate the gradients is using numerical finite difference approximations. In order to calculate the gradient information, it is necessary to integrate the dynamics for

the perturbation in the direction of each optimization variables during each iteration. If the second order Hessian is required for the optimizer, then the number of such evaluation increases even more, so does the computation time. Further, the finite difference approximations only provide estimated values, which might be not necessarily accurate enough. Despite many optimization solver does not require very accurate estimation of the gradient information, an inaccurate information could results in a slow convergence to feasible solutions.

Finally, it is often difficult to solve unstable systems when using the single-shooting method. Considering that the restricted Poincaré map is a partial map, that is, not any initial states can be guaranteed to have a complete solution. More specifically, the robot might not be able to complete one step starting from any initial condition, or the time-to-impact might be infinity. An exception would be the cases in which the zero dynamics has a dimension of two, as illustrated the previous example. Hence the initial condition can be solved explicitly from the parameters to make sure the zero dynamics has a complete solution, the single shooting optimization does not suffer from the partial map issue. On the other hand, if the dimension of the zero dynamics is higher than two, there is no such explicit solution exists. As a result, the single shooting method is not appropriate to solve such a system.

3.3 Multiple Shooting Optimization

It is noted in the trajectory optimization community, not all design variables are equally complicated for an NLP solver to navigate, a fact which *multiple shooting* methods exploit [18]. In a single shooting formulation, a change to a parameter or boundary state value can have a very nonlinear effect on the final state, after integrating nonlinear dynamics over non-trivial time scales. This type of nonlinearity can cause an optimizer to fail to converge or be uncertain if it has found a solution.

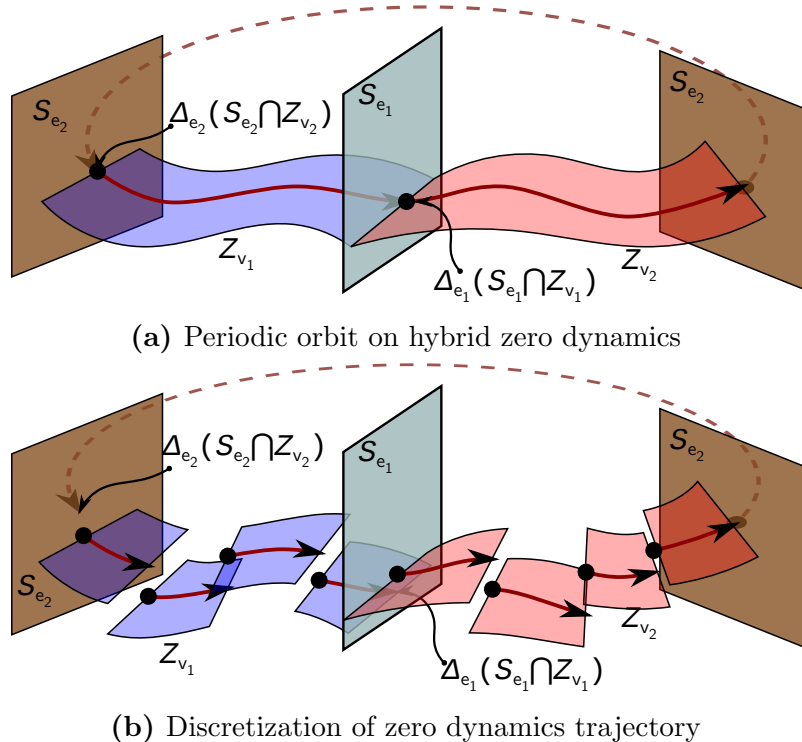


Figure 3.3: The discretization of the periodic orbit on the hybrid zero dynamics manifold when using multiple shooting optimization methods.

Multiple shooting is designed to ameliorate this manner of nonlinearity [24]; specifically it splits each discretization segment of the input tape into its own small trajectory optimization problem, each with its own set of control parameters and of initial state conditions, but a shorter horizon for integration. With finer discretization, this integration horizon approaches zero, rendering the relationship between control inputs and the post-integration state increasingly linear. This property has made multiple shooting methods fast and reliable for planning gaits for walking robots [41, 64, 82].

3.3.1 Problem Formulation

We start with uniformly dividing the continuous domain time interval into N_v shooting grids,

$$t_v^0 = t_0 < t_1 < t_2 \cdots < t_{N_v} = t_v^f, \quad N_v \geq 1. \quad (3.24)$$

with an abstract example shown in Figure 3.3. For each shooting grid defined on $t \in [t_i, t_{i+1}] \subset \mathbb{R}$, we define the initial condition ξ_v^i and control parameters α_v^i of the zero dynamics as the augmented optimization variables. Then solve the zero dynamics ODE on the shooting grid independently from an artificial initial values ξ_v^i :

$$\dot{\xi}_v(t) = z_v(\xi_v(t), \alpha_v^i), \quad t \in [t_i, t_{i+1}], \quad (3.25)$$

$$\xi_v(t_i) = \xi_v^i, \quad (3.26)$$

where $i \in \{0, 1, \dots, N_v - 1\}$. By integrating (3.25) forward till the end of the time interval, we obtain the trajectory pieces of the zero dynamics, denoted by $\xi_v(t_{i+1}; \xi_v^i, \alpha_v^i)$, where the extra arguments after the semicolon represent the dependence on the initial values and parameters. In order to ensure the continuity of states and the consistency of parameters between two neighboring grids, we impose equality constraints on each grid, given by

$$\xi_v(t_{i+1}; \xi_v^i, \alpha_v^i) - \xi_v^{i+1} = 0, \quad (3.27)$$

$$\alpha_v^i - \alpha_v^{i+1} = 0, \quad (3.28)$$

for all $i \in \{0, 1, \dots, N_v - 1\}$. In addition, the initial states of each shooting grid ξ_v^i must be on the zero dynamics manifold determined by α_v , and the final states of the domain ξ_v^n must be on the corresponding switching surface. These constraints can be formulated as

$$\xi_v^i \in \mathcal{Z}_v, \quad \xi_v^{N_v} \in \mathcal{Z}_v \cap S_{sor-1(v)}^{\mathcal{X}}. \quad (3.29)$$

The periodic orbit requirement is imposed through the restricted reset map:

$$\Delta_e^{\mathcal{Z}}(\xi_{sor(e)}^{N_{sor(e)}}) - \xi_{tar(e)}^0 = 0, \quad (3.30)$$

for all $e \in E$. Let \mathbf{w} is the combination of optimization variables defined on each domain, i.e., $\mathbf{w} = \{\mathbf{w}_v\}_{v \in V}$, where

$$\mathbf{w}_v = [\xi_v^0, \alpha_v^0, \dots, \xi_v^{N_v}, \alpha_v^{N_v}, T_v]^T \quad (3.31)$$

with $T_v = t_v^f - t_v^0$, the hybrid zero dynamics based *multiple shooting* optimization problem for a multi-domain hybrid system is constructed as:

$$\begin{aligned}
\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} & \sum_{v \in V} \left(\mathbf{E}_v^Z(\xi_{vi}(t_v^0), \xi_v(t_v^f), \alpha_v) + \sum_{i=0}^{n-1} \left(\int_{t_v^i}^{t_v^{i+1}} \mathbf{L}_v^Z(\xi_v(t), \alpha_v) dt \right) \right) \quad (3.32) \\
\text{s.t.} & \quad \xi_v(t_{i+1}; \xi_v^i, \alpha_v^i) - \xi_v^{i+1} = 0, \\
& \quad \alpha_v^i - \alpha_v^{i+1} = 0, \\
& \quad \xi_v^i \in \mathcal{Z}_v, \\
& \quad \xi_v^{N_v} \in \mathcal{Z}_v \cap S_{\text{sor}^{-1}(v)}^{\mathcal{X}}, \\
& \quad \Delta_e^Z(\xi_{\text{sor}(e)}^{N_{\text{sor}(e)}}) - \xi_{\text{tar}(e)}^0 = 0, \\
& \quad \mathbf{C}_v^{\min} \leq \mathbf{C}_v^Z(\xi_v^i, \alpha_v^i) \leq \mathbf{C}_v^{\max}, \\
& \quad \mathbf{B}_v^{\min} \leq \mathbf{B}_v^Z(\xi_v^0, \xi_v^n, \alpha_v^0) \leq \mathbf{B}_v^{\max},
\end{aligned}$$

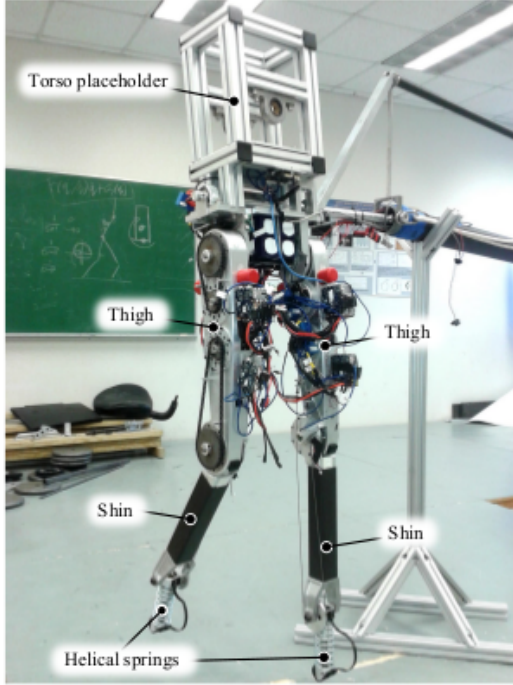
for all $v \in V$ and $e \in E$. Note that, the running cost is calculated as the sum of integrals on all shooting grids, and the path and terminal constraints are evaluated at each point of the discretization. Moreover, the constraints (3.29) and (3.30) together guarantee that the system is hybrid invariant at least on these particular points $\xi_v^{N_v}$ and the resulting reduced dimensional manifold has a periodic orbit.

3.3.2 Case Study: 7-Link Planar Biped with Compliant Legs

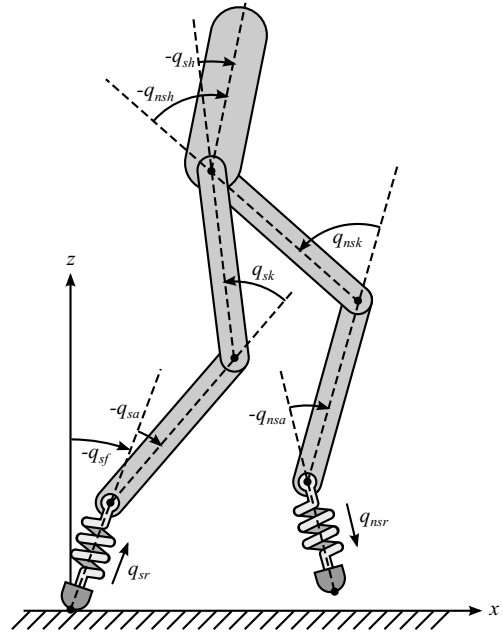
In this example, we apply the direct multiple shooting formulation of the hybrid zero dynamics gait optimization approach to generate dynamic walking gaits for a underactuated 7-link planar biped, DURUS-2D.

3.3.2.1 Robot Model

DURUS-2D is a prototype design of the humanoid robot DURUS, which will be discussed in Section 5.1, built by SRI International. One key improvement present in DURUS-2D which differs from typical 5-link planar point-feet biped is a linear spring at the end of each leg, as shown in Figure 3.4a. The compliance provided by



(a) Mechanical design.



(b) Kinematic configuration.

Figure 3.4: DURUS-2D biped designed and built by SRI International. Each leg of the robot has an actuated hip and knee joint, a constrained ankle joint, and a passive linear spring. The ankle joints are constrained to rotate in the reverse direction of the knee joints via a mechanical structure that consists of two pulleys and a string. The rotation angle of the ankle joint is half of the knee joint, and the calf and the thigh are designed to have the same length. These mechanical structures are designed to approximately align the passive linear springs along the line that connects the feet and the hip point all the time, so that the walking of the robot emulates the spring loaded inverted pendulum (SLIP) walking.

the linear springs is designed to absorb energy at impact and reduce the energy loss so that more efficient locomotion can be achieved. The design of the compliant legs is inspired by the Spring-Loaded Inverted Pendulum model (SLIP). The SLIP model is widely used as a low-dimensional representation of bipedal locomotion to generate efficient behaviors due to the presence of compliance [22, 32].

While the mechanical design of compliant elements provides better energy efficiency, the increased underactuated degrees of freedom cause designing a HZD gait for such a highly underactuated robot becomes very challenging problem. Especially with the typical single shooting approach, the gait optimization problem encounters

convergence issues due to the increased dimensions of the hybrid zero dynamics manifold. Even with good initial guesses, we found that the single shooting optimization often fails to converge to a feasible solution. In this example, we demonstrate that the multiple shooting approach is capable of improving the reliability of the HZD gait optimization problem.

Robot Configuration. The configuration

$$q = (p_x, p_z, q_{sf}, q_{sr}, q_{sa}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk}, q_{nsa}, q_{nsr}) \in \mathcal{Q} \subset \mathbb{R}^{11} \quad (3.33)$$

of the robot is shown in Figure 3.4b, where q_{sr} and q_{nsr} represent the deflections of the springs, and p_x and p_z represent the position of the stance foot. In particular, we model the linear springs as prismatic joints with forces that equal to the actual spring forces. Hence, the continuous dynamics of the robot can be obtained via (2.17) augmented with the spring forces, which can be achieved by modifying the vector $H(q, \dot{q})$ in (2.17) as

$$H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) + B_s(kq + b\dot{q}), \quad (3.34)$$

where where k is the spring stiffness and b is the damping coefficient, and B_s is the distribution matrix for the spring forces. Because the springs cannot exert impulsive forces at impact, the discrete dynamics in (2.27) still applies to DURUS-2D model.

Hybrid System Model. Due to the existence of the springs, the stance foot will not leave the ground immediately after the non-stance foot hits the ground. Therefore, the walking of DURUS-2D has two continuous domains: the *double support* domain \mathcal{D}_{ds} , where both feet are on the ground, and the *single support* domain \mathcal{D}_{ss} , where the non-stance leg swings in the air, as shown in Figure 3.5. In addition to the contact constraints, we have to consider the kinematic constraints of the pulley system. Based on the description of the mechanical design in Figure 5.34, these constraints can be written as:

$$h_{sa}(q) := q_{sa} + \frac{q_{sk}}{2}, \quad h_{nsa}(q) := q_{nsa} + \frac{q_{nsk}}{2}. \quad (3.35)$$

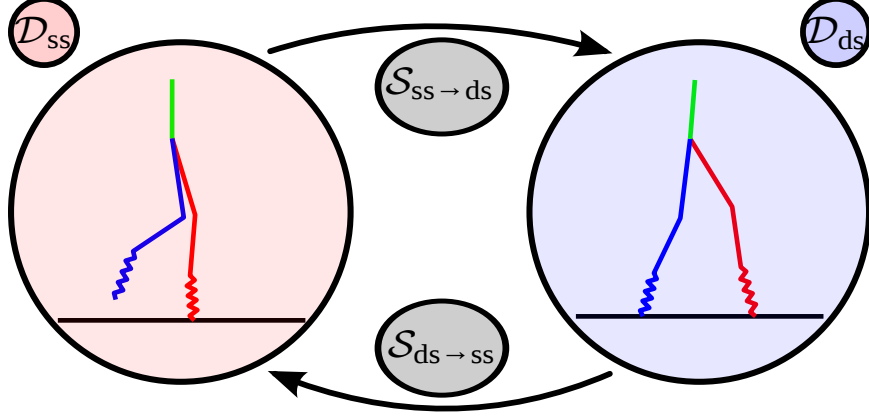


Figure 3.5: The directed cycle of the hybrid control system model of DURUS-2D.

Based on the contact conditions and kinematic constraints, the corresponding two-domain hybrid control system model can be constructed as in Definition 2.3. The holonomic constraints of each domain are given by

$$h_{ds}(q) = (p_x, p_z, h_{sa}(q), h_{nsa}(q)) \in \mathbb{R}^4, \quad (3.36)$$

$$h_{ss}(q) = (p_x, p_z, h_{sa}(q), h_{nsa}(q), p_{nsf}^x(q), p_{nsf}^z(q)) \in \mathbb{R}^6, \quad (3.37)$$

where $p_{nsf}(q)$ is the Cartesian position of the non-stance foot. Let λ_{sf}^x , λ_{sf}^z , λ_{nsf}^x , and λ_{nsf}^z be the ground constraint forces on the stance and non-stance foot respectively, the domains of admissibility are given as

$$\mathcal{D}_{ds} := \{(q, \dot{q}, u) \in T\mathcal{Q} \times U \mid \lambda_{sf}^z \geq 0, \lambda_{nsf}^z \geq 0, |\lambda_{sf}^x| \leq \mu\lambda_{sf}^z, |\lambda_{nsf}^x| \leq \mu\lambda_{nsf}^z\}, \quad (3.38)$$

$$\mathcal{D}_{ss} := \{(q, \dot{q}, u) \in T\mathcal{Q} \times U \mid \lambda_{sf}^z \geq 0, |\lambda_{sf}^x| \leq \mu\lambda_{sf}^z, p_{nsf}^z(q) \geq 0\}. \quad (3.39)$$

The transition $\mathcal{D}_{ss} \rightarrow \mathcal{D}_{ds}$ occurs when the non-stance foot hits the ground, and the transition $\mathcal{D}_{ds} \rightarrow \mathcal{D}_{ss}$ occurs when the normal force acting on the non-stance foot becomes zero. Hence the guards are defined as

$$S_{ds \rightarrow ss} = \{(q, \dot{q}, u) \in \mathcal{D}_{ds} \mid \lambda_{nsf}^z(q, \dot{q}, u) = 0\}, \quad (3.40)$$

$$S_{ss \rightarrow ds} = \{(q, \dot{q}, u) \in \mathcal{D}_{ss} \mid p_{nsf}^z(q) = 0, \dot{p}_{nsf}^z(q, \dot{q}) \leq 0\}. \quad (3.41)$$

Because there is no impact occurs at the transition from the double-support to single-support domain, therefore, $\Delta_{ds \rightarrow ss} = \mathbf{I}$. The reset map from the single-support to

double-support domain comply the impact equation in (2.29). Note that there will be a coordinate change after the foot hits the ground, which can be determined by the relabeling matrix

$$\mathcal{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.42)$$

Hence, the reset map, $\Delta_{ss \rightarrow ds}$, is determined by (2.30).

3.3.2.2 Multiple-Shooting Virtual Constraint Optimization

In this section, we use the direct multiple shooting optimization in (3.32) to generate optimal walking gaits for DURUS-2D.

Virtual Constraints. Considering that the robot has underactuated point feet, we only define position-modulating outputs. Because the stance foot angle, q_{sf} , is monotonically decreasing within a step duration, we pick the phase variable $\theta(q)$ in (2.35) as the stance foot angle, i.e., $\theta(q) := q_{sf}$. The choice of virtual constraints for DURUS-2D is inspired by the work in [13]. Thus, we define the following outputs for each domain:

$$y_{2,ss}^a(q) = \begin{bmatrix} q_{sk} & q_{nsk} & q_{tor} & \delta m_{nsl} \end{bmatrix}^T, \quad (3.43)$$

$$y_{2,\text{ds}}^a(q) = \begin{bmatrix} q_{sk} & q_{nsk} & q_{tor} \end{bmatrix}^T, \quad (3.44)$$

where $q_{tor} = -q_{sf} - q_{sa} - q_{sk} - q_{sh}$ is the torso angle and $\delta m_{nsl} = -q_{sf} - q_{sa} - q_{sk} - q_{sh} + q_{nsh} + \frac{1}{2}q_{nsk}$ is the non-stance slope. It can be noted that the non-stance slope, δm_{nsl} , is omitted during the *double support* domain due to the fact that both legs are constrained on the ground so that the output δm_{nsl} —which represents the forward motion of swing leg—becomes redundant. With the definition of virtual constraints and holonomic constraints of each domain, we can formulate the zero dynamics equation based on the discussion in Appendix A.

Remark 3.1. To have a square system, i.e., the inputs and the outputs have the same dimension, we assume that there are only three actuated joints, $(q_{sk}, q_{sh}, q_{nsk})$, in the system during the double-support domain of DURUS-2D walking by considering one of the originally actuated joints as passive joint. We pick the non-stance hip joint as the passive joint, since all outputs defined on the double-support domain are not directly related to this joint.

Cost Function and Constraints. To seek an efficient walking gait, we define the cost function as the *mechanical cost of transport* of the gait. To use the Bolza form of the cost function, we define the running cost as

$$\mathbf{L}_v(\xi_v^i, \alpha_v^i) = \frac{1}{mgd} W_v(\xi_v^i, \alpha_v^i), \quad (3.45)$$

where m is the total mass of the robot, g is the acceleration due to gravity, d is the distance traveled during one step, and W_v is the sum of the work done at each shooting grid with

$$W_v(\xi_v^i, \alpha_v^i) = \sum_{j=1}^{m_v} \|u_j^a(\xi_v(t), \alpha_v^i) \cdot \dot{q}_j^a(\xi_v(t), \alpha_v^i)\|, \quad (3.46)$$

where m_v is the number of actuated joints of the domain $v \in V$, and \dot{q}_j^a and u_j^a are the velocity and the actuator torque of the actuated joint j , computed from the zero

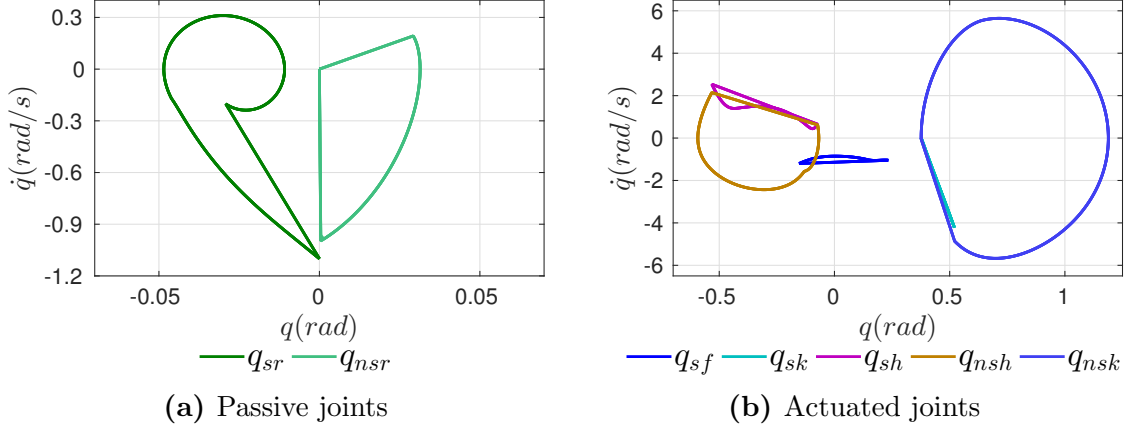


Figure 3.6: Periodic limit cycles of a stable periodic walking gait of DURUR-2D in simulation.

dynamics states, $\xi_v(t)$, and virtual constraints parameters, α_v . There is no terminal cost defined in this case.

The multiple shooting virtual constraint gait optimization is then formulated as in (3.32). Since the time duration of \mathcal{D}_{ds} is much shorter than that of \mathcal{D}_{ss} , we set the shooting grids for the two domains to be 15 and 25, respectively. In particular, following path constraints are considered:

- actuator torque limits,
- joint velocity limits,
- joint angle ranges,
- non-stance foot height clearance, and
- torso angle range.

Because the stability condition of the periodic orbit cannot be explicitly computed for a high dimensional zero dynamics model. Thus, the stability of the resulting hybrid zero dynamics is checked *a posteriori* by numerically solving for the fixed point of the Poincaré map of the zero dynamics.

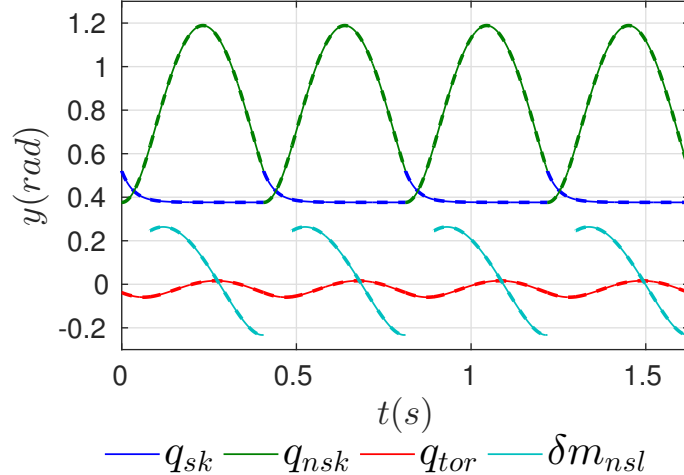


Figure 3.7: Desired outputs of a stable periodic walking gait of DURUR-2D in simulation.

3.3.2.3 Simulation and Experimental Results

A simulation of a stable, periodic walking gait for DURUS-R is performed in which the robot starts from the fixed point on the guard and is controlled by the feedback linearization controller defined in (2.39). The virtual constraints parameters are obtained from the HZD based multiple shooting optimization and we set $\varepsilon = 10$ as the control gain. The phase portraits of the robot joints in Figure 3.6 show the resulting stable periodic orbit of the underactuated degrees of freedom, i.e., the reduced dimensional zero dynamics. The maximum magnitude of eigenvalues of the full order system Poincaré map is 0.6461, which further proves the stability of the walking gait. Figure 3.7 shows the tracking performance of the proposed controller.

The resulting walking gait is implemented experimentally on the hardware. The robot is supported by a freely-rotating four-bar linkage boom that restricts the motion of the robot to the sagittal plane while keeping the robot level with the ground. This boom design ensures that the boom neither holds the robot upward nor adds weight to the robot. The experiment was conducted by performing several trials under the same conditions. For each trial, the robot walks approximately 300 steps with no sign of failing before being stopped by the experimenter. Figure 3.8 shows the phase portraits

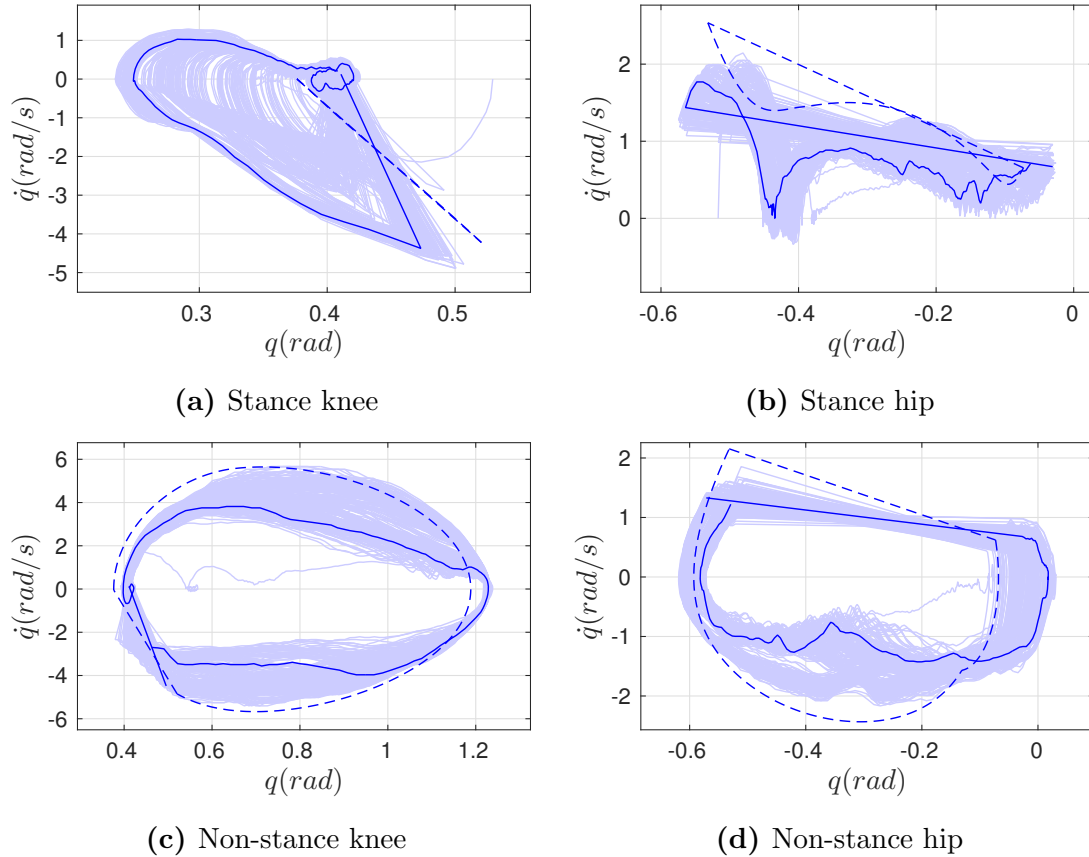


Figure 3.8: Stable periodic orbits of walking gait of DURUR-2D. The dashed lines represent the phase portraits joints from simulation, the solid lines represent the phase portraits of actuated joints during one step in experiment, and the shaded areas represent the history of the actuated joints phase portraits over approximately 300 steps.

of the robot joints from the experimental data. The limit cycles deviate slightly from the corresponding limit cycles of the simulated gait, which are shown by dotted lines. The tracking of the actuated joints from the experiment is shown in Figure 3.9. The maximum tracking error appears on the stance knee joint. A possible reason for this to happen could be the velocity changes of the robot at impact during the experiment are greater than those in simulation, due to the mismatch of the model information that used in the optimization, especially the spring constants. In addition, Figure 3.10 shows a stroboscopic comparison of the walking gait in simulation and experiment during one step. The average cost of electrical transport for this gait is 0.99.

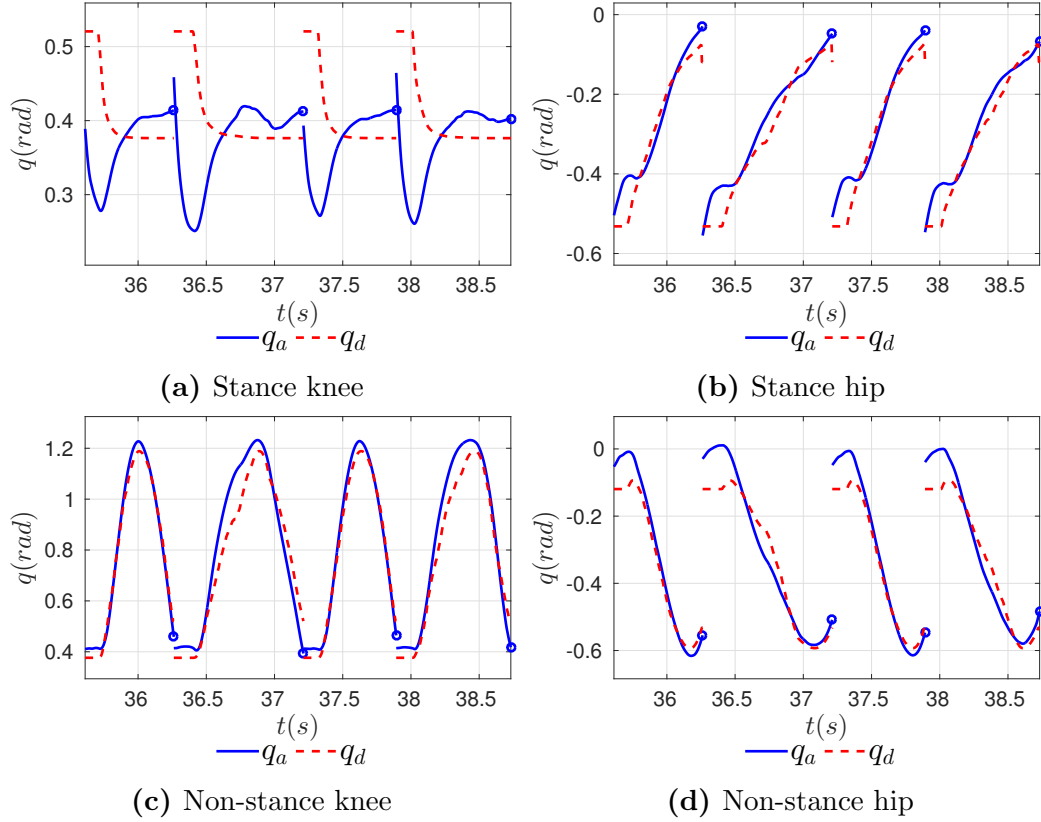


Figure 3.9: Joint angles tracking performance of DURUS-2D.

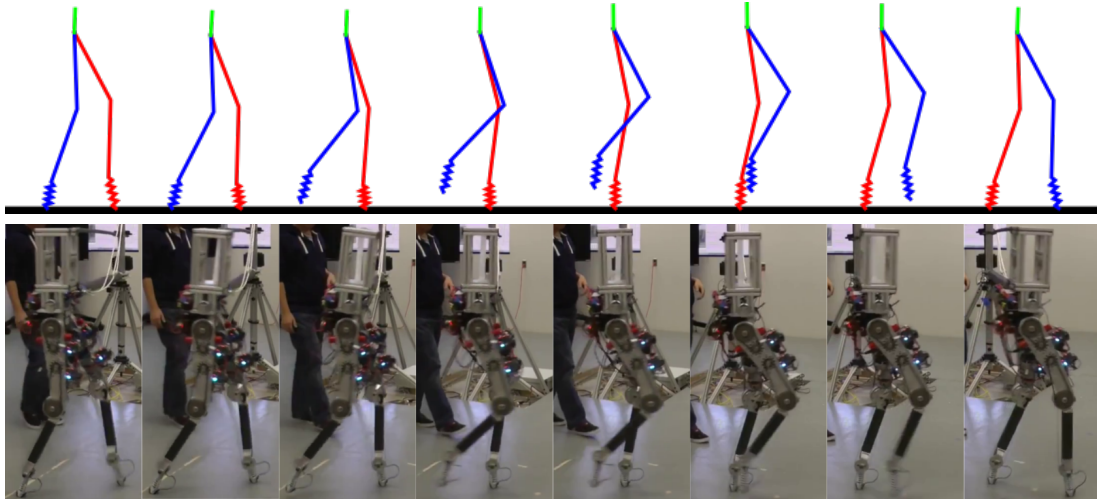


Figure 3.10: The walking gait snapshot comparison of the simulation and experimental results with DURUS-2D over one step.

3.3.2.4 Optimization Performance

Since the mechanical structure of DURUS-2D is inspired by the SLIP model, the initial guess of the parameter set is determined from the stable walking gait of the

SLIP model. A similar approach can be found in [68]. Even with these raw fitted data as the initial guesses, the proposed optimization converges reliably. Despite DURUS-2D has a higher dimension of zero dynamics manifolds, the gait optimization converges to an optimal solution as fast as the same gait optimization of a simple 5-link planar underactuated biped.

In summary, the hybrid zero dynamics based multiple shooting method converges reliably and rapidly even with bad initial guesses. Moreover, this method outperforms the single shooting method in the case of compliant DURUS-2D biped. In many cases, the single shooting optimization fails to converge to a feasible solution due to the high underactuated degrees of freedom of the robot. The end result is a stable, efficient gait for the robot.

3.3.3 Issues

The multiple shooting method improves the robustness of the HZD optimization problem, particularly of the robot that has compliant elements. By discretizing the continuous time intervals to shorter shooting grids, it increasingly reduces the non-linearity of constraints. Further, previous knowledge of state trajectory can be utilized in the initialization of the optimization variables, so that the negative effects of the partial maps are reduced noticeably. As a result, the multiple shooting method is more robust to unstable system than the single shooting method. Since it still uses explicit integration for the system dynamics, the multiple shooting method is still unable to completely overcome issues risen from unstable systems and partial maps.

Despite that the reliability of the optimization is enhanced via the multiple shooting approach, the performance of the convergence speed does not have significant improvement over the single shooting method. This is mainly due to the numerical evaluation of the Jacobian and gradient information. As in the single shooting method, it also uses the finite difference approximations to compute the gradients.

Moreover, the number of optimization variables is increased significantly compared the single shooting method, the computational effort for calculating the numerical Jacobian increase even more. However, the Jacobian is a block sparse matrix due to the discretization, which could be effectively utilized in the calculation of the Jacobian.

3.4 Summary

In this chapter, we begin with formulating the basic gait trajectory optimization problem for the multi-domain hybrid systems. Two prevalent shooting methods are introduced and illustrated by examples. Despite being simple and straightforward, these two methods both have a few issues that prevent their applications on high degrees of freedom and highly underactuated humanoids. These optimization approaches are often very slow to converge and extremely sensitive to initial guesses seeded to the solver. With regard to the reliability and computational issues of the direct shooting methods, in the next chapter we propose a novel HZD optimization framework unified with the direct collocation methods—an effective approach has been successfully applied in many trajectory optimization problems [35, 48, 53, 138].

CHAPTER IV

DIRECT COLLOCATION METHODS

As we discussed in the previous chapter, a straightforward and traditional approach to transcribing the hybrid zero dynamics gait design optimization is via *direct shooting* methods. However, it is extremely difficult to extend this approach to general cases, especially to higher dimensional humanoid robots or highly underactuated robots with compliant joints. The forward integration of the zero dynamics is required if the closed-form solution cannot be obtained analytically. As robots are built with more actuated linkages or have higher degrees of underactuation, such gait optimization problems become increasingly intractable for direct shooting methods. Despite that it improves the reliability of the hybrid zero dynamics gait optimization via discretization, the multiple shooting approach also runs into scalability issues with increasing degrees of freedom, e.g. when applying it to humanoids [64].

In this chapter, we present the optimization formulation of multi-domain bipedal locomotion using the *direct collocation* methods. Motivated by the desire to provide a generalized gait optimization tool for HZD type bipedal locomotion, the discussion in this chapter is based on the general multi-domain hybrid system model and hybrid zero dynamics framework presented in Chapter 2.

4.1 Mathematical Background

We start with review the mathematical background of the collocation methods. Historically, collocation methods are widely used in solving numerical solution of differential algebraic equations (DAEs) or ordinary differential equations (ODEs). Consider

an ODE given in

$$\dot{y}(t) = f(y(t)). \quad (4.1)$$

Suppose we consider an approximation of the solution of the (4.1) by a polynomial of degree K over a time interval $[t_i, t_{i+1}]$:

$$\tilde{y}^{(i)}(t) = \sum_{k=0}^K a_k^{(i)}(t - t_i)^k, \quad t \in [t_i, t_{i+1}] \quad (4.2)$$

with coefficients $a^{(i)} = (a_0^{(i)}, a_1^{(i)}, \dots, a_K^{(i)})$. To ensure that $\tilde{y}^{(i)}(t)$ is indeed an accurate approximation of the actual solution, these coefficients must be chosen such that the approximation matches the solution at the beginning of the interval, that is,

$$\tilde{y}^{(i)}(t_i) = y(t_i), \quad (4.3)$$

and the derivatives match at the points $\tau_j = t_i + (t_{i+1} - t_i)\rho_j$ with

$$0 \leq \rho_1 \leq \rho_2 \leq \dots \leq \rho_K \leq 1 \quad (4.4)$$

for $1 \leq j \leq K$:

$$\dot{\tilde{y}}^{(i)}(\tau_j) = f(\tilde{y}^{(i)}(\tau_j)). \quad (4.5)$$

The conditions (4.5) are called *collocation* conditions, and the intermediate points τ_j are called *collocation* points. If the coefficients $a^{(i)}$ satisfy the initial value in (4.3) and collocation conditions in (4.5), then the polynomial $\tilde{y}^{(i)}(t)$ given in (4.2) yields is an approximated solution of the ODE over the time interval $[t_i, t_{i+1}]$. There are many different collocation methods based on the form of polynomials and collocation conditions being used. The most commonly used collocation methods for the optimal control problem are the piece-wise continuous *standard collocation* method and the *orthogonal collocation* method. Here, we briefly overview the mathematical formulation of these two collocation methods.

4.1.1 Standard Collocation

As opposed to solving the state sequentially as in a time-marching method, the collocation methods solve the differential equations simultaneously and implicitly due to the fact that the solution of states at each collocation point are determined at the same time. The standard collocation method usually uses implicit Runge-Kutta methods, such as the *Trapezoidal* method and the *Hermite-Simpson* method, because implicit Runge-Kutta methods have better stability properties than explicit methods. The *Trapezoidal* and *Hermite-Simpson* methods are also referred as the Lobatto IIIA method of order 2 and the Lobatto IIIA method of order 4 [133]. For a *Lobatto* method, both endpoints of the time interval are also the collocation points, i.e. $\tau_1 = t_i$ and $\tau_K = t_{i+1}$. For instance, the trapezoidal approach uses a quadratic interpolating polynomial, in which the three coefficients are determined by the solution matches at the beginning of the interval and the derivative matches at the beginning and the end of the interval. On the other hand, the Hermite-Simpson scheme uses cubic interpolating polynomial with four coefficients determined by the solution matches at the beginning of the interval and the derivative matches at the beginning, midpoint, and the end of the interval.

The process starts with dividing the integration duration $[t_0, t_f]$ into N intervals

$$t_0 = t_0 \leq t_1 \leq \dots \leq t_N = t_f, \quad (4.6)$$

where the points are referred to as nodes or grid points. Suppose that $y^{(i)} = y(t_i)$ be the values of states at time t_i that to be determined, we first approximate the solution over each mesh with a polynomial function with coefficients determined via the collocation constraints, and then define a *defect constraint* at each grid

$$\zeta_i := y^{(i+1)} - \tilde{y}^{(i)}(t_{i+1}), \quad (4.7)$$

so that the approximated solution is piece-wise continuous between two meshes. Hence the goal becomes determining $(y^{(0)}, y^{(1)}, \dots, y^{(N)})$ at all nodes in order to

solve $\mathbf{Z} = (\zeta_0, \zeta_1, \dots, \zeta_{N-1})$ being zero. For example, the Hermite-Simpson collocation scheme uses a cubic Hermite interpolation polynomial to approximate the solution and then integrates the differential equations using the Simpson's quadrature rule. In particular, the defect constraints are defined as

$$\zeta_i = y^{(i+1)} - \left(y^{(i)} + \frac{\Delta t_i}{6} (f(y^{(i)}) + 4f(\bar{y}_{i+1}) + f(y^{(i+1)})) \right) \quad (4.8)$$

where the approximated value of the states at the midpoint \bar{y}_{i+1} is determined by the cubic interpolation polynomial:

$$\bar{y}_{i+1} = \frac{1}{2}(y^{(i)} + y^{(i+1)}) + \frac{\Delta t_i}{8}(f(y^{(i)}) - f(y^{(i+1)})). \quad (4.9)$$

Solving the defect constraints being zero yields a piecewise-cubic approximation of the system solution. Let $y(t)$ be the exact solution and $\tilde{y}(t)$ be a piece-wise cubic polynomial approximation over the interval $t \in [t_0, t_f]$ that satisfies the defect constraints in (4.8), then the approximation error is estimated by the following lemma.

Lemma 4.1. *Suppose that $y(t) \in C^4[t_0, t_f]$ and has bounded derivatives, then*

$$\|y(t) - \tilde{y}(t)\| \leq \frac{1}{384} \max_{t_0 \in [0, t_f]} |y^{(4)}(t)| h^4 \leq Ch^4, \quad (4.10)$$

for some constant C that is independent of N , with $h := \max_i \Delta t_i$.

Proof. This is a standard result of the Hermite-Simpson collocation; for more detail see [83, 39]. □

This shows that the Hermite-Simpson collocation method has $O(h^4)$ accuracy within a fixed time interval $[t_0, t_f]$. More importantly, the approximated solution converges to the true solution in a rate of at least 4.

4.1.2 Orthogonal Collocation

Different from the standard collocation, in an orthogonal collocation method the collocation points are chosen at roots of a family of orthogonal polynomials, such as Chebyshev polynomials or Legendre polynomials [126]. The three most commonly used sets

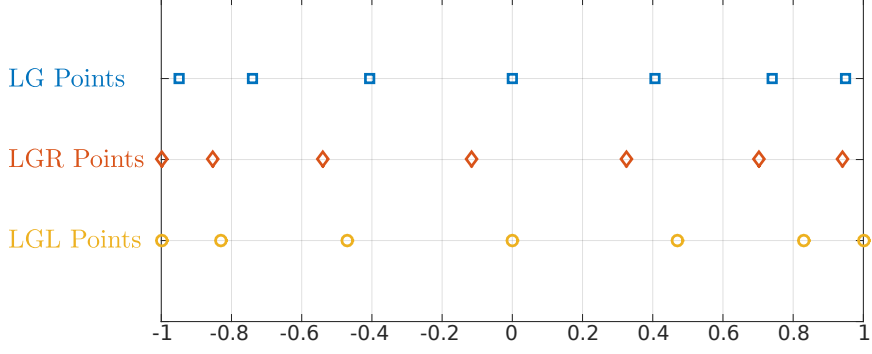


Figure 4.1: Distributions of LG, LGR, and LGL points ($N = 7$).

of collocation points are *Legendre-Gauss* (LG), *Legendre-Gauss-Radau* (LGR), and *Legendre-Gauss-Lobatto* (LGL) points. In particular, the LG points are the roots of K^{th} -order Legendre polynomials $P_K(\tau)$ that lies on the interval $\tau \in (-1, 1)$, the LGR points are the roots of $P_{K-1}(\tau) + P_K(\tau)$, and the LGL points are the roots of $(\tau^2 - 1)\dot{P}_{K-1}(\tau)$, respectively [47]. The LG points contain neither -1 nor 1 , the LGR points contain only one end point, and the LGL points contain both end points -1 and 1 , as shown in Figure 4.1.

In the orthogonal collocation, the states are typically approximated on the time interval $\tau \in [-1, 1]$ as

$$y(\tau) \approx \tilde{y}(\tau) = \sum_{k=0}^K c_k L_k(\tau) \quad (4.11)$$

where $L_k(\tau)$ are Lagrange basic polynomials defined as

$$L_k(\tau) := \prod_{\substack{0 \leq j \leq K \\ j \neq k}} \frac{\tau - \tau_j}{\tau_k - \tau_j}, \quad (4.12)$$

which has a *isolation property*

$$L_k(\tau_j) = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases}. \quad (4.13)$$

The isolation property yields the fact that

$$\tilde{y}(\tau_k) = c_k. \quad (4.14)$$

The *collocation* constraints are defined in a manner such that

$$\zeta_j := f(\tilde{y}(\tau_j)) - \dot{\tilde{y}}(\tau_j) = f(\tilde{y}(\tau_j)) - \sum_{k=0}^K \tilde{y}(\tau_j) \dot{L}_k(\tau_j) = 0, \quad (4.15)$$

for all $j \in [0, K]$, and where the derivatives of $L_k(\tau)$ at the collocation nodes are only depend on j and k . If there exist a series of approximated states $\tilde{y}(\tau_j)$ at orthogonal collocation points that satisfies the collocation constraints, the resulting polynomial in (4.11) is an accurate approximation of the system solution. For the accuracy of the orthogonal approximation, we refer the readers to [27, 54].

Remark 4.1. Because the collocation methods use implicit integration schemes to simultaneously simulate the dynamical equations, they are applicable to solve the differential algebraic equations, where the dynamics of the system is given implicitly in terms of states. Typically, the differential algebraic equations are more general than the ordinary differential equations, and often are represented by simpler expressions. Hence this adaptation of the direct collocation methods to a DAE system can be utilized to simplify the analytic expression of system dynamics. This is very useful when applying the collocation methods on high dimensional systems.

4.2 Constrained Dynamics

The direct collocation based optimization is based on the fundamental idea of collocation methods, converts the original trajectory optimization problem, e.g., the problem described in Section 3.1, into a nonlinear programming problem by replacing the forward integration of the dynamics by a set of defect constraints. The first step in the construction of such a NLP is to determine how to represent the dynamic equations of the system. Recall that in the direct shooting optimization, we use the restricted zero dynamics equations. While it indeed helps to reduce the dimension of the system, the zero dynamics equations bears a scalability issue; the closed form expression of zero dynamics is difficult to obtain analytically for the reason that the

full order dynamics needs to be inverted. This is why the direct shooting methods perform extremely poor for high-dimensional humanoid robots.

The states of the full order system can be re-constructed from the partial zero dynamics states using the inverse mapping of $\Phi_v(x)$ (see Appendix A for more detail). Though this approach yields low dimensional representation of the full order system, the closed form solution of the partial zero dynamics equations is not trivial to determine analytically. In addition, the partial zero dynamics derivation requires to invert the inertia matrix $D_v(q)$ and the decoupling matrix $\mathcal{A}_v(q, \dot{q})$, resulting in very complicated expressions that are usually extremely expensive to compute symbolically. Not to mention that the inverse map $\Phi_v^{-1}(\xi)$ is often difficult to solve for high dimensional humanoid. Further, the initial condition of the zero dynamics must satisfy the admissible conditions of the domain manifold of the full order system, hence additional conditions must be taken into consideration when determining the initial condition of the zero dynamics states.

On the other hand, the direct collocation method provides an alternative option. Different from the shooting methods which explicitly solves the system ODE via some time-marching forward integration techniques, the direct collocation methods allows to define system dynamics in more general implicit differential algebraic equations (DAE) forms, e.g., $F(y(t), \dot{y}(t)) = 0$. In this case, the $f(y(t))$ terms in the defect constraints (4.8) are replaced by the derivatives of the states $\dot{y}(t)$, which in turn must satisfy the system dynamics equations $F(y(t), \dot{y}(t)) = 0$.

Recall that the zero dynamics is based on the full order dynamics (2.17) subject to the holonomic constraints and virtual constraints being zero. In other words, the restricted dynamics can be described as differential algebraic equations:

$$\begin{cases} D(q)\ddot{q} + H(q, \dot{q}) - B_v u - J_v^T(q)\lambda_v = 0 \\ h_v(q) = 0 \\ y_{2,v}(q) = 0 \end{cases} \quad (4.16)$$

This problem has a differentiation index of 3 due to the fact that the acceleration terms appear in the second derivatives of the constraints [17, 25]. As we discussed in the Chapter 2, we could differentiate the constraints twice to reduce the index of (4.16), yields an index-1 implicit DAE problem:

$$F_v(q, \dot{q}, \ddot{q}, u, \lambda_v, \alpha_v) := \begin{cases} D(q)\ddot{q} + H(q, \dot{q}) - B_v u - J_v^T(q)\lambda_v = 0 \\ J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} = 0 \\ \dot{\eta}_v(q, \dot{q}, \ddot{q}, \alpha_v) - F_v^{cl}\eta(q, \dot{q}, \alpha_v) = 0 \end{cases} \quad (4.17a)$$

subject to the initial value conditions

$$h_v(q(t_0)) = \bar{h}_v, \quad J_v(q(t_0))\dot{q}(t_0) = 0, \quad (4.17b)$$

$$y_{2,v}(q(t_0), \alpha_v) = 0, \quad \dot{y}_{2,v}(q(t_0), \dot{q}(t_0), \alpha_v) = 0, \quad (4.17c)$$

where $\eta_v = (y_{1,v}, y_{2,v}, \dot{y}_{2,v})$ and \bar{h}_v is a vector of constants determined by the contact conditions. Note that we introduce the stable linear output dynamics that described in (2.42) in the virtual constraints differentiation. This system can be considered as an implicit form that is equivalent to the zero dynamics equation by construction. It turns out that fully utilizing this fact in the direct collocation optimization is the key to the unification the hybrid zero dynamics and direct collocation methods.

Lemma 4.2. *Suppose that $\phi_v(t) \subset \mathcal{X}$ is a solution of the initial value problem in (4.17), then $\phi_v(t) \subset \mathcal{PZ}_v$. That is, $\phi_v(t)$ is also a solution on the partial hybrid zero dynamics.*

Proof. The result follows immediately from the construction of the constrained dynamics in (4.17). By assumption, $\phi_v(t_0) = [q(t_0), \dot{q}(t_0)]^T$ satisfy the initial conditions in (4.17b). Further $\ddot{h}_v(q(t), \dot{q}(t), \ddot{q}(t)) = 0$ from the second equation in (4.17a) guarantees that $h_v(q) \equiv \bar{h}_v$ along any solution of (4.17a). Hence, $\phi_v(t) \subset \mathcal{X}_v$ where \mathcal{X}_v is a canonical projection of \mathcal{D}_v onto the state space \mathcal{X} . Similarly, the third equation in (4.17a) stabilizes the virtual constraints η_v exponentially to the origin. Considering

that $y_{2,v}(t_0) = 0$ and $\dot{y}_{2,v}(0) = 0$, therefore, we have

$$\begin{aligned} y_{2,v}(q(t), \alpha_v) &= 0, \\ \dot{y}_{2,v}(q(t), \dot{q}(t), \alpha_v) &= 0 \end{aligned}$$

for all $t_0 \leq t \leq t_f$ until the solution $\phi_v(t)$ reaches a guard, i.e., $\phi_v(t_f) \in \mathcal{X}_v \cap S_{\text{sor}^{-1}(v)}$. By the definition of the partial hybrid zero dynamics in (2.53), we conclude that $\phi_v(t) \subset \mathcal{PZ}_v$. \square

Remark 4.2. Due to the fact we do not need to invert the inertia matrix $D(q)$ anymore to explicitly solve the zero dynamics equations, the implicit DAE system given in (4.17a) yields much simpler expressions of the constrained system dynamics from the computational point of view.

4.3 Local Direct Collocation Optimization

The local direct collocation optimization utilizes the fundamental idea of the standard collocation methods of solving the differential equations. By replacing the forward integration of the dynamical systems with the series of defect constraints, it converts the optimal control problem into a nonlinear programming problem (NLP). Hence this method is also referred to as the *direct transcription* formulation [19]. In this section, we particularly use the Hermite-Simpson collocation method to construct the direct collocation formulation of the hybrid zero dynamics based gait trajectory optimization. For ease of communication, we refer it to the *direct HZD optimization* in the remainder of the thesis.

4.3.1 Collocation Constraints

In the Hermite-Simpson collocation scheme, the states at the midpoint are typically computed by the boundary states of the two collocation nodes as in (4.9). In [19], Betts et al. introduced a modified version of Hermite-Simpson scheme, in which the states at the midpoints are also included in the optimization variables which

satisfy the midpoint states constraints as described in (4.9). This modified scheme is termed as *Hermite-Simpson separated* (HSS) scheme. Based on the HSS scheme, we propose an improved formulation via introducing *defect variables*. Defect variables are supplementary decision variables that could be computed in closed-form originally. For instance, the states at the midpoints are defect variables in the HSS formulation. To formulate the collocation constraints for implicit DAEs, we also introduce the first order derivatives of state variables as the defect decision variables of the NLP. Consequently, the system dynamics in the form of implicit DAEs is then enforced for state variables and their derivatives as a set of equality constraints.

Assuming $T_{I,v} > 0$ is the time at which the system reaches the guard associated with a given domain, \mathcal{D}_v , the time discretization of the duration of continuous dynamics is given as,

$$0 = t_0 < t_1 < t_2 < \dots < t_{N_v} = T_{I,v}, \quad (4.18)$$

with $N_v = 2(N_v^c - 1)$, where the even points are called *cardinal nodes* and the odd points are called *interior nodes* (see Figure 4.2). The total number of cardinal nodes specified per domain, N_v^c , has to be greater or equal to 2. The distribution of cardinal nodes within a domain could be arbitrary, however, an interior point has to be placed at the center of two adjacent cardinal nodes. Our formulation allows the cardinal nodes to be placed at the uniformly distributed points or the *Chebyshev-Gauss-Lobatto* (CGL) points. While the former provides simplicity in implementation, the latter yields better accuracy due to the fact that more nodes are placed close to two terminals.

Given the discretization, the HSS scheme uses Hermite interpolation polynomials to approximate the solution within two neighboring cardinal nodes using the estimated states and their derivatives. As illustrated in Figure 4.2, there are two defect constraints defined at each collocation mesh: 1) the difference between the estimated states from the optimizer and the interpolated states at the interior node from the

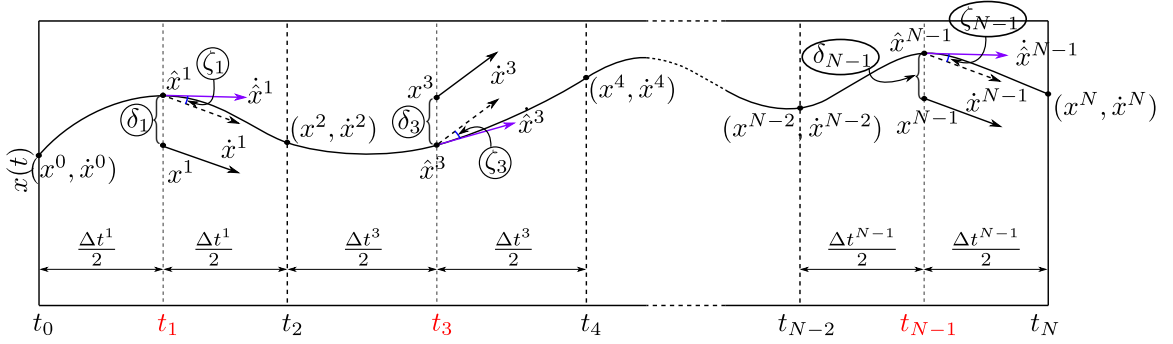


Figure 4.2: Illustration of defect constraints and node distribution.

approximated polynomial (δ_i in Figure 4.2), and 2) the difference between the derivatives of states from the optimizer and the first-order derivatives of the approximated polynomial at the interior point (ζ_i in Figure 4.2). These constraints can be stated as¹,

$$\delta^{(i)} := x^{(i)} - \tilde{x}^{(i)} = 0, \quad (4.19)$$

$$\zeta^{(i)} := \dot{x}^{(i)} - \dot{\tilde{x}}^{(i)} = 0, \quad (4.20)$$

where

$$\dot{\tilde{x}}^{(i)} = \frac{3}{2\Delta t^{(i)}}(x^{(i+1)} - x^{(i-1)}) - \frac{1}{4}(\dot{x}^{(i-1)} + \dot{x}^{(i+1)}), \quad (4.21)$$

$$\tilde{x}^{(i)} = \frac{1}{2}(x^{(i+1)} + x^{(i-1)}) + \frac{\Delta t^{(i)}}{8}(\dot{x}^{(i-1)} - \dot{x}^{(i+1)}), \quad (4.22)$$

and $x^{(i)} = (q^{(i)}, \dot{q}^{(i)})$ and $\dot{x}^{(i)} = (\dot{q}^{(i)}, \ddot{q}^{(i)})$, respectively. In particular, $\Delta t^{(i)} = t_{i+1} - t_{i-1}$, for $i \in \{1, 3, 5, \dots, N_v - 1\}$ and for all $v \in V$. For example,

$$\Delta t^{(i)} = \frac{T_{I,v}}{N_v^c - 1}, \quad (4.23)$$

if cardinal nodes are uniformly distributed, and

$$\Delta t^{(i)} = \frac{T_{I,v}}{2} \left(\cos\left(\frac{(i-1)\pi}{N_v}\right) - \cos\left(\frac{(i+1)\pi}{N_v}\right) \right), \quad (4.24)$$

¹An animated version of the illustration of direct collocation formulations can be found in <https://youtu.be/aL-B2eIoCK4>.

if cardinal nodes are placed at CGL points. These two cases are just two practical examples of the node distribution schemes. In Section 6.1, we will evaluate the discretization errors in these two different node distribution cases. It is important to note that the decision variables $x^{(i)}$ and $\dot{x}^{(i)}$ at each node must satisfy the system dynamics given in (4.17a). The detail of the dynamics constraints is presented in the next section together with other HZD specific constraints.

4.3.2 Problem Formulation

With the formulation of collocation constraints, now we can formally state the main result of the thesis—the direct HZD optimization problem. The ultimate goal of the optimization is to find a set of virtual constraints parameters α that result in a hybrid invariant periodic orbit \mathcal{O} for the multi-domain hybrid system in (2.43), yielding a periodic walking gait of the robot.

We start with defining the optimization variables of the problem. Consider the time discretization in (4.18) for each domain \mathcal{D}_v with $N_v + 1$ nodes, let

$$\begin{aligned}\mathbf{q}_v &= (q^{(0)}, q^{(1)}, \dots, q^{(N_v)}), \\ \dot{\mathbf{q}}_v &= (\dot{q}^{(0)}, \dot{q}^{(1)}, \dots, \dot{q}^{(N_v)}), \\ \ddot{\mathbf{q}}_v &= (\ddot{q}^{(0)}, \ddot{q}^{(1)}, \dots, \ddot{q}^{(N_v)}), \\ \mathbf{u}_v &= (u^{(0)}, u^{(1)}, \dots, u^{(N_v)}), \\ \boldsymbol{\lambda}_v &= (\lambda_v^{(0)}, \lambda_v^{(1)}, \dots, \lambda_v^{(N_v)}),\end{aligned}$$

be collections of optimization variables defined on all nodes. The defect constraints in (4.19) and (4.20) are defined on all interior nodes. Thus we define that

$$\boldsymbol{\zeta}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v) := \begin{bmatrix} \zeta^{(1)} \\ \zeta^{(3)} \\ \vdots \\ \zeta^{(N_v-1)} \end{bmatrix} = \mathbf{0}, \quad \boldsymbol{\delta}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v) := \begin{bmatrix} \delta^{(1)} \\ \delta^{(3)} \\ \vdots \\ \delta^{(N_v-1)} \end{bmatrix} = \mathbf{0},$$

for each $v \in V$. In addition, the state variables must satisfy the dynamic equations in (4.17a) at all nodes. Similarly, we have

$$\mathbf{F}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \mathbf{u}_v, \boldsymbol{\lambda}_v, \alpha_v) := \begin{bmatrix} F_v(q^{(0)}, \dot{q}^{(0)}, \ddot{q}^{(0)}, u^{(0)}, \lambda_v^{(0)}, \alpha_v) \\ F_v(q^{(1)}, \dot{q}^{(1)}, \ddot{q}^{(1)}, u^{(1)}, \lambda_v^{(1)}, \alpha_v) \\ \vdots \\ F_v(q^{(N_v)}, \dot{q}^{(N_v)}, \ddot{q}^{(N_v)}, u^{(N_v)}, \lambda_v^{(N_v)}, \alpha_v) \end{bmatrix} = \mathbf{0}.$$

In the context of the hybrid system model, any solution should satisfy the domain of admissibility conditions. Thus from (2.24), we define that

$$\mathbf{A}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \boldsymbol{\lambda}_v) := \begin{bmatrix} A_v(q^{(0)}, \dot{q}^{(0)}, \lambda_v^{(0)}) \\ A_v(q^{(1)}, \dot{q}^{(1)}, \lambda_v^{(1)}) \\ \vdots \\ A_v(q^{(N_v)}, \dot{q}^{(N_v)}, \lambda_v^{(N_v)}) \end{bmatrix} \geq \mathbf{0}.$$

Note that we state any constraints that involves the contact wrenches explicitly in terms of $\lambda^{(i)}$, considering that they are already parts of the optimization variables and there is no need to compute them from the states and control inputs. In addition, it must be guaranteed that the system reaches the corresponding guard surface of a domain $v \in V$ at $T_{I,v}$. This is equivalent to imposing the guard condition in (2.26) at the last node of each domain, i.e.,

$$\begin{aligned} H_e(q^{(N_v)}, \dot{q}^{(N_v)}, \lambda_v^{(N_v)}) &= 0, \\ \dot{H}_e(q^{(N_v)}, \dot{q}^{(N_v)}, \lambda_v^{(N_v)}) &< 0, \end{aligned}$$

for all $v \in V$ with e being the subsequent edge of vertex v , i.e., $e = \text{sor}^{-1}(v)$. We also replace u_v with λ_v because the guard condition is determined by the unilateral constraints which are defined as functions of either robot configuration or contact wrenches.

While the continuous dynamics is determined by collocation, the discrete dynamics given in (2.30) can be directly imposed as an equality constraint that connect

the solutions of two neighboring domains. Because this constraint involves variables defined on two different domains, we denote that $\square^{(0_{v+})}$ be a variable defined on the first node of the next domain of \mathcal{D}_v . Let $\mathcal{Z} = \{T_{I,v}, \mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \mathbf{u}_v, \boldsymbol{\lambda}_v, \alpha_v, \bar{h}_v\}_{v \in V}$ be the set of NLP variables, the direct HZD optimization problem can be stated as [66]:

$$\mathcal{Z}^* = \underset{\mathcal{Z}}{\operatorname{argmin}} \sum_{v \in v_1}^{v_{np}} \mathcal{J}_v(T_{I,v}, \mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \mathbf{u}_v, \boldsymbol{\lambda}_v, \alpha_v, \bar{h}_v) \quad (4.25a)$$

$$\text{s.t. } \boldsymbol{\zeta}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v) = 0, \quad (4.25b)$$

$$\boldsymbol{\delta}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v) = 0, \quad (4.25c)$$

$$\mathbf{F}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \mathbf{u}_v, \boldsymbol{\lambda}_v, \alpha_v) = 0, \quad (4.25d)$$

$$\mathbf{A}_v(\mathbf{q}_v, \dot{\mathbf{q}}_v, \boldsymbol{\lambda}_v) \geq 0 \quad (4.25e)$$

$$H_e(q^{(N_v)}, \dot{q}^{(N_v)}, \lambda_v^{(N_v)}) = 0, \quad (4.25f)$$

$$\dot{H}_e(q^{(N_v)}, \dot{q}^{(N_v)}, \lambda_v^{(N_v)}) < 0, \quad (4.25g)$$

$$\Delta_e(q^{(N_v)}, \dot{q}^{(N_v)}) - (q^{(0_{v+})}, \dot{q}^{(N_v)}) = 0, \quad (4.25h)$$

$$h_v(q^{(0)}) - \bar{h}_v = 0, \quad (4.25i)$$

$$y_{2,v}(q^{(0)}, \alpha_v) = 0, \quad (4.25j)$$

$$\dot{y}_{2,v}(q^{(0)}, \dot{q}^{(0)}, \alpha_v) = 0, \quad (4.25k)$$

$$u^{\min} \leq u^{(i)} \leq u^{\max}, \quad (4.25l)$$

$$q^{\min} \leq q^{(i)} \leq q^{\max}, \quad (4.25m)$$

$$\dot{q}^{\min} \leq \dot{q}^{(i)} \leq \dot{q}^{\max}, \quad (4.25n)$$

for all $v \in V$, where \mathcal{J}_v is the cost function. In the direct collocation optimization, any integral in the cost function is computed using quadrature rules, which we will introduce in Section 4.3.3.

Remark 4.3. Note that the control input u and the contact wrenches λ_v are defined as decision variables at each node explicitly. We will explain the reasoning of this

choice in Section 4.5. More importantly, it is straightforward to verify from (4.17a) that the constraint wrenches, $\lambda_v^{(i)}$, are determined (implicitly) via the *acceleration level* holonomic constraints and the control inputs, $u^{(i)}$, are determined (also implicitly) from the linear output dynamics. Hence by definition, the control inputs determined from the optimization are equal to the feedback controllers defined in (2.39). That is, the feedback control law is encoded in the trajectory optimization problem through the virtual constraints, which is different from other classic trajectory optimization formulations, in which the control inputs are often assumed to be open-loop, discrete or piece-wise continuous values determined by the optimizer. This is because we are more interested in determining a set of parameters $\alpha = \{\alpha_v\}_{v \in V}$ to represent the optimal gait behavior for bipedal robot, rather than open-loop control inputs that results in optimal trajectories. More importantly, for a set of optimal parameters from the nonlinear optimization, we can always construct a family of feedback control laws via rapidly exponentially stabilizing control Lyapunov functions (RES-CLF) to stabilize the output dynamics.

Remark 4.4. The equality constraint of the reset map can be simplified with the introduction of defect variable δF_v . More specifically, the joint configuration continuity constraints can be expressed as

$$q^{(0_{v+})} - \mathcal{R}(q^{(N_v)}) = 0, \quad (4.26)$$

for all $v \in V$. The impact velocity equation, however, involves inversions of complicated matrix expression as seen in (2.29). Therefore, we introduce the impact wrenches δF_v in the NLP decision variables, so that the impact velocity constraints can be imposed in its original, yet simpler, form:

$$J_v(q^{(0)})\dot{q}^{(0)} = 0, \quad (4.27)$$

$$D(q^{(0_{v+})})(\dot{q}^{(0_{v+})} - \mathcal{R}(\dot{q}^{(N_v)})) - J_{v+}^T(q^{(0_{v+})})\delta F_v = 0, \quad (4.28)$$

for all $v \in V$. Note that (4.27) is also one of the initial conditions (velocities of holonomic contact constraints) in (4.17). In a word, this initial condition is already enforced via the reset map.

Any feasible solution of the NLP in (4.25) yields a periodic orbit of the hybrid system \mathcal{H}^α , and more importantly, the orbit is hybrid invariant, which we state as a theorem below.

Assumption 4.1. In the local direct collocation optimization, the trajectories of state variables are approximated by piece-wise cubic polynomials. From Lemma 4.1, we know that the approximated trajectories converges to the true solution when $N_v \rightarrow \infty$. While we could not pick infinite number of nodes in practice, we assume that there exists sufficiently large number of nodes such that the error between the approximated and true solutions becomes negligible. More specifically, suppose that $\phi_v(t)$ and $\tilde{\phi}_v(t)$ are the true and approximated solution of the hybrid system during a given domain \mathcal{D}_v , respectively. Then for a sufficiently small tolerance $\varepsilon > 0$, there always exists an positive integer $\bar{N}_v > 0$ such that for all $N_v \geq \bar{N}_v$, the following inequality holds true:

$$\|\phi_v(t) - \tilde{\phi}_v(t)\| \leq \varepsilon. \quad (4.29)$$

Thus we assume that the solution of the NLP in (4.25) is the actual solution of the system. Note that the $\tilde{\phi}_v(t)$ with $t \in [0, T_{I,v}]$ is a piece-wise cubic polynomial determined by the time distribution $(t_0, t_1, \dots, t_{N_v})$ and the discrete state variables at each time point, $\{\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v\}$, for a given domain $v \in V$. Let $T_I = \sum_{v=v_1}^{v_{n_p}} T_{I,v}$, then the (approximated) solution, $\tilde{\phi}(t)$, of the multi-domain hybrid system is given by

$$\tilde{\phi}(t) = \begin{cases} \tilde{\phi}_{v_1}(t), & 0 \leq t < T_{I,v_1} \\ \tilde{\phi}_{v_2}(t - T_{I,v_1}), & T_{I,v_1} \leq t < T_{I,v_1} + T_{I,v_2} \\ \vdots & \vdots \\ \tilde{\phi}_{v_{n_p}}(t - T_{I,v_{n_p-1}}), & T_{I,v_1} + \dots + T_{I,v_{n_p-1}} \leq t < T_{I,v_1} + \dots + T_{I,v_{n_p}} \end{cases} \quad (4.30)$$

with $t \in [0, T_I)$.

Theorem 4.1. *Suppose that \mathcal{Z}^* is a feasible solution of the constrained nonlinear programming problem in (4.25), then the solution $\tilde{\phi}^*(t)$ given in (4.30) is periodic. Moreover, $\tilde{\phi}^*(t)$ is hybrid invariant under the virtual constraints based feedback control law in (2.39) with the parameters α^* , i.e., $\tilde{\phi}^*(t) \subset \mathcal{PZ}_{\alpha^*}$.*

Proof. Let $\tilde{\phi}_v^*(t)$ be a piece-wise continuous polynomial determined by the solution $\{T_{I,v}^*, \mathbf{q}_v^*, \dot{\mathbf{q}}_v^*, \ddot{\mathbf{q}}_v^*\}$ from the optimization (4.25) for each domain \mathcal{D}_v . Then from (4.25b)–(4.25d), $\tilde{\phi}_v^*(t)$ is a approximated solution of the continuous constrained dynamics (4.17a), and by Assumption 4.1, $\tilde{\phi}_v^*(t) \subset \mathcal{X}_v$ is also considered as the exact solution of (4.17a). Further from (4.25f) and (4.25g), we can conclude that

$$(q^{(N_v)}, \dot{q}^{(N_v)}) \in S_e \cap \mathcal{X}_v, \quad (4.31)$$

and from (4.25h), the solutions of two adjacent continuous domains are connected via the continuous reset map Δ_e , i.e.,

$$\tilde{\phi}_{v+}^*(0) = \Delta_e(q^{(N_v)}, \dot{q}^{(N_v)}). \quad (4.32)$$

So that $\tilde{\phi}^*(t)$ in (4.30) is a valid solution of the hybrid system model. Both F_v and Δ_e are C^1 continuous by definition, therefore, there exists an unique solution for some given initial condition $x(0)$. Moreover, with the constraint of reset maps, we have

$$\tilde{\phi}^*(T_I) = \Delta_e(q^{(N_{v_{np}})}, \dot{q}^{(N_{v_{np}})}) = (q^{(0_{v_1})}, \dot{q}^{(0_{v_1})}) = \tilde{\phi}^*(0). \quad (4.33)$$

Hence, $\tilde{\phi}^*(t)$ is periodic due to the uniqueness of the solution. In particular T_I is the period of the periodic solution.

The hybrid invariant of the periodic solution can be verified by constraints (4.25i)–(4.25k). In other words, $\tilde{\phi}^*(t) \subset \mathcal{PZ}_v$ by Lemma 4.2, and $(q^{(N_v)}, \dot{q}^{(N_v)}) \in S_e \cap \mathcal{PZ}_v$. Moreover from (4.25h), we know that

$$\Delta_e(q^{(N_v)}, \dot{q}^{(N_v)}) = \tilde{\phi}_{v+}^*(0).$$

By Lemma 4.2, $\tilde{\phi}_{v+}(0) \in \mathcal{PZ}_{v+}$. therefore

$$\Delta_e(q^{(N_v)}, \dot{q}^{(N_v)}) \in \mathcal{PZ}_{v+}, \quad (4.34)$$

for all $v \in V$. This shows that the solution is impact invariant over all discrete dynamics. We also know that the solution is forward invariant under the feedback controller \mathbf{u}_v^* . As a result $\tilde{\phi}^*(t) \subset \mathcal{PZ}_{\alpha^*}$, where the partial hybrid zero dynamics manifold, $\mathcal{PZ}_{\alpha^*} := \mathcal{PZ}_{v_1} \cup \mathcal{PZ}_{v_2} \cdots \cup \mathcal{PZ}_{v_{n_p}}$, depends on the parameters α^* . \square

Note that here we only prove that the periodic solution is hybrid invariant and lies on the restricted partial hybrid zero dynamics manifold. In other words, only one point on the surface $S_e \cap \mathcal{PZ}_v$ is impact invariant. This result cannot extend to the full switching surface in general cases, especially when the partial zero dynamics has dimension of more than two. However, in some special cases we can conclude that the optimal parameters α^* of the optimization in (4.25) guarantee that the hybrid system has partial hybrid zero dynamics (PHZD). In fact, this is a quite typical case that could encounter in practice. If the robot is fully actuated or there is only one degree of underactuation on all continuous domains, the corresponding partial zero dynamics has dimension of at most two. This can be verified by the construction of zero dynamics equations in Appendix A.

Corollary 4.1. *If all continuous domains \mathcal{D}_v with $v \in V$ in the hybrid system \mathcal{H}^α have partial zero dynamics \mathcal{PZ}_v of dimension two, the parameters α^* solving the constrained optimization (4.25) yields partial hybrid zero dynamics (PHZD).*

Proof. From Theorem 4.1, a feasible solution of (4.25) results in a periodic orbit that is hybrid invariant. In [7, 135], it shows that if the dimension of the partial zero dynamics (or zero dynamics for underactuated domain) has dimension of two, then $S_e \cap \mathcal{PZ}_v$ is a singleton. We have already shown that the singleton is hybrid invariant in Theorem 4.1, therefore, we have

$$S_e \cap \mathcal{PZ}_v \in \mathcal{PZ}_{v+}, \quad (4.35)$$

for all $v \in V$. This result completes the proof. \square

4.3.3 Cost Function

Normally the cost function of a trajectory optimization problem consists of the terminal cost and the running cost. While it is trivial to compute the terminal cost, the computation of the running cost requires further discussions. One possible approach is introducing an additional state variables and the differential equation

$$\dot{x}_{n+1}(t) = \mathbf{L}_v(x(t), u(t)) \quad (4.36)$$

with an initial condition $x_{n+1}(t_v^0) = 0$. Then the original running cost can be replaced with the Mayer form representation:

$$\int_{t_v^0}^{t_v^f} \mathbf{L}_v(x(t), u(t)) dt = x_{n+1}(t_v^f). \quad (4.37)$$

However it might be computationally expensive as it increase the dimension of the states so as the size of the problem [19].

A more computationally efficient approach would be using the quadrature approximation rules to compute function integrals. In numerical integration, a quadrature rule is an approximation of a function integral on a finite time interval, usually formulated as a *weighted sum* of function values at discrete points within the interval [126]. For example, the *Simpson's* quadrature rule between two cardinal nodes $t \in [t_{i-1}, t_{i+1}]$ with t_i is an interior node is given by

$$\int_{t_{i-1}}^{t_{i+1}} \mathbf{L}_v(x(t), u(t)) dt \approx \frac{\Delta t^{(i)}}{6} (\mathbf{L}_v(x^{(i-1)}, u^{(i-1)}) + 4\mathbf{L}_v(x^{(i)}, u^{(i)}) + \mathbf{L}_v(x^{(i+1)}, u^{(i+1)})). \quad (4.38)$$

For a sufficiently smooth cost function $\mathbf{L}_v(x, u)$, the local estimation error in the Simpson's quadrature approximation is given by

$$\varepsilon_i = \frac{1}{90} \left(\frac{\Delta t^{(i)}}{2} \right)^5 \mathbf{L}_v^{(4)}(x(\xi), u(\xi)) \quad (4.39)$$

for some $\xi \in [t_{i-1}, t_{i+1}]$ [126]. The integral of the cost function over the entire domain is computed via adding the quadrature approximation of each collocation mesh:

$$\begin{aligned}
& \int_{t_v^0}^{t_v^f} \mathbf{L}_v(x(t), u(t)) dt \\
&= \sum_{i=1}^{N_v^c-1} \int_{t_{2i-2}}^{t_{2i}} \mathbf{L}_v(x(t), u(t)) dt \\
&= \sum_{i=1}^{N_v^c-1} \frac{\Delta t^{(2i-1)}}{6} (\mathbf{L}_v(x^{(2i-2)}, u^{(2i-2)}) + 4\mathbf{L}_v(x^{(2i-1)}, u^{(2i-1)}) + \mathbf{L}_v(x^{(2i)}, u^{(2i)})) \\
&= \sum_{i=0}^{N_v} w_i \mathbf{L}_v(x^{(i)}, u^{(i)}), \tag{4.40}
\end{aligned}$$

where w_i is a weight given by

$$w_i = \begin{cases} \frac{1}{6} \Delta t^{(i+1)}, & i = 0, \\ \frac{2}{3} \Delta t^{(i)}, & \text{if } i \text{ is an interior node,} \\ \frac{1}{3} (\Delta t^{(i-1)} + \Delta t^{(i+1)}), & \text{if } i \text{ is a cardinal node but } i \neq 0 \text{ and } i \neq N_v, \\ \frac{1}{6} \Delta t^{(i-1)}, & i = N_v, \end{cases} .$$

Then the global estimation error in Simpson's quadrature approximation is

$$\varepsilon \leq \frac{T_{I,v}}{180} h^4 \mathbf{L}_v^{(4)}(x(\xi), u(\xi)) \tag{4.41}$$

for some $\xi \in [t_v^0, t_v^f]$, where $h = \max_{i \in [1, N_v^c]} \Delta v^{(2i-1)}$. That is, the Simpson's rule has an order of 4 and the error proportional to h^4 .

With the quadrature approximation, the cost function can be computed as

$$\mathcal{J} = \sum_{v \in V} \left(\mathbf{E}_v(x^{(0)}, u^{(0)}, x^{(N_v)}, u^{(N_v)}) + \sum_{i=0}^{N_v} w_i \mathbf{L}_v(x^{(i)}, u^{(i)}) \right). \tag{4.42}$$

Since the Simpson quadrature approximation rule is the same method that we use in the Hermite-Simpson collocation, this computation is equivalent to the Mayer form defined in (4.37).

4.4 Pseudospectral Optimization

As a subset of direct collocation methods, the pseudospectral optimization approach employs the idea of orthogonal collocation techniques. Instead of using piecewise

continuous polynomials, this approach uses a global orthogonal polynomial to approximate the solution of the system, and consequently, collocation constraints are enforced at particularly chosen points. Therefore, the pseudospectral optimization is also referred to as *global orthogonal collocation* method [108]. In this thesis, we particularly use LGL points as collocation points, however, the same formulation can be extended to other cases.

4.4.1 Orthogonal Collocation Constraints

As discussed in Section 4.1.2, in the Legendre pseudospectral method, the basic idea is to approximate the solution of a given continuous domain, \mathcal{D}_v , by Lagrange interpolating polynomials which interpolate the solutions at Legendre-Gauss-Lobatto (LGL) nodes (see Figure 4.1). Suppose that

$$-1 = \tau_0 < \tau_1 < \tau_2 < \dots < \tau_{N_v} = 1, \quad (4.43)$$

are the LGL points where $N_v + 1$ is the number of points chosen for the domain $v \in V$. These LGL points are located at the zeros of $(\tau^2 - 1)\dot{P}_{N_v}(\tau)$ distributed over the interval $\tau \in [-1, 1]$, where $\dot{P}_{N_v}(\tau)$ is the derivative of the N_v -th order Legendre polynomial $P_{N_v}(\tau)$ [53]. Let $x^{(i)} = (q^{(i)}, \dot{q}^{(i)})$ be approximated states at node τ_i , then the solution $x(\tau)$ on $\tau \in [-1, 1]$ is approximated by

$$x(\tau) \approx \hat{x}(\tau) = \sum_{i=0}^{N_v} x_i L_i(\tau) \quad (4.44)$$

where

$$L_i(\tau) = \frac{1}{N_v(N_v + 1)P_{N_v}(\tau_i)} \frac{(\tau^2 - 1)\dot{P}_{N_v}(\tau)}{\tau - \tau_i}, \quad (4.45)$$

for $i \in [0, N_v]$ is a family of Lagrange interpolating polynomials of order N_v . It can be easily checked that the isolation property as in (4.13) holds for $L_i(\tau)$. Similarly, the derivate of $x(\tau)$ is approximated by differentiating the approximation $\hat{x}(\tau)$ in (4.44),

$$\dot{\hat{x}}(\tau) = \sum_{i=0}^{N_v} x_i \dot{L}_i(\tau). \quad (4.46)$$

Interestingly, the derivative of $L_i(\tau_k)$ is a constant determined only by i and k and independent of the state variables [111]. This feature leads to the LGL differentiation matrix $\mathbf{D}^{\text{LGL}} \in \mathbb{R}^{(N_v+1) \times (N_v+1)}$. Let $\mathbf{D}_{ki}^{\text{LGL}} = \dot{L}_i(\tau_k)$ be the (k, i) entry of the differentiation matrix, we have

$$\mathbf{D}_{ki}^{\text{LGL}} = \begin{cases} \frac{L_{N_v}(\tau_i)}{L_{N_v}(\tau_k)(\tau_i - \tau_k)}, & \text{if } i \neq k, \\ -\frac{N_v(N_v+1)}{4}, & \text{if } i = k = 0, \\ \frac{N_v(N_v+1)}{4}, & \text{if } i = k = N_v, \\ 0, & \text{otherwise.} \end{cases} \quad (4.47)$$

The orthogonal collocation condition is defined such that at all LGL points, the approximated derivatives $\hat{x}(\tau_i)$ in (4.46) match the derivatives $\dot{x}^{(i)}$ that determined by the optimizer. Before formulating the collocation constraints, note that the Legendre interpolating polynomials are defined on the interval $[-1, 1]$, which must be transformed to the actual time interval $[0, T_{I,v}]$ via the affine transformation [46]:

$$t = \frac{T_{I,v}}{2}(\tau + 1).$$

Hence, the collocation constraint at the k -th LGL point is given as

$$\sum_{i=0}^{N_v} x^{(i)} \dot{L}_i(\tau_k) - \frac{T_{I,v}}{2} \dot{x}^{(k)} = \sum_{i=0}^{N_v} x^{(i)} \mathbf{D}_{ki}^{\text{LGL}} - \frac{T_{I,v}}{2} \dot{x}^{(k)} = 0, \quad (4.48)$$

for all $k \in \{0, 1, \dots, N_v\}$ (see Figure 4.3). We can combine this collocation constraints in a single equation by letting

$$\mathbf{X} = \begin{bmatrix} x^{(0)} \\ x^{(1)} \\ \vdots \\ x^{(N_v)} \end{bmatrix}, \quad \dot{\mathbf{X}} = \begin{bmatrix} \dot{x}^{(0)} \\ \dot{x}^{(1)} \\ \vdots \\ \dot{x}^{(N_v)} \end{bmatrix}, \quad (4.49)$$

be the stacked vectors of states and their derivatives in the NLP variables, then the collocation constraints can be stated as

$$(\mathbf{D}^{\text{LGL}} \otimes \mathbf{I}_{n_v}) \mathbf{X} - \frac{T_{I,v}}{2} \dot{\mathbf{X}} = 0, \quad (4.50)$$

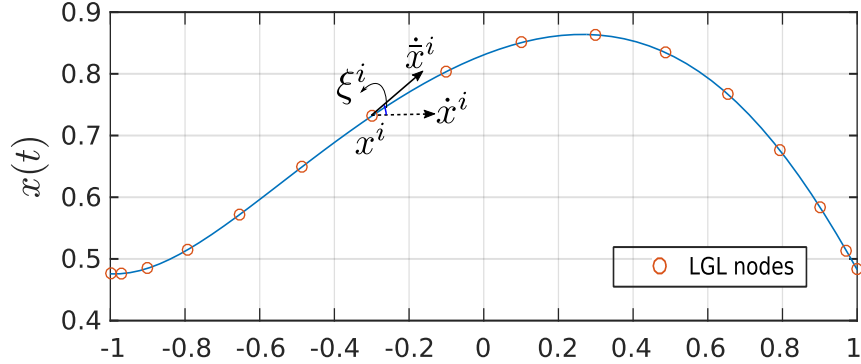


Figure 4.3: Demonstration of the global orthogonal collocation at Legendre-Gauss-Lobatto (LGL) points.

where \otimes represents the Kronecker products and \mathbf{I}_{n_v} is an identity matrix of dimension $n_v = \dim \mathcal{X}_v$.

In the pseudospectral optimization, the function integral is calculated using the Gauss quadrature approximation. The running cost function of the NLP is computed as

$$\int_{t_v^0}^{t_v^f} \mathbf{L}_v(x(t), u(t)) dt = \sum_{k=0}^N w_k \mathbf{L}_v(x^{(i)}, u^{(i)}) \quad (4.51)$$

where w_k is the LGL weight given by

$$w_k = \frac{2}{N_v(N_v + 1)} \frac{1}{[P_{N_v}(\tau_k)]^2}. \quad (4.52)$$

The pseudospectral optimization can be considered as an extension of the local direct collocation optimization. The key difference is the way of imposing collocation conditions. Different from the local direct collocation, in which the collocation constraints depend on three neighboring nodes, in pseudospectral optimization the collocation constraints depend on states of all LGL nodes. By using global orthogonal polynomials which interpolate at Legendre-Gauss-Lobatto (LGL) points, the pseudospectral method provides an approximation of the solution that has exponential convergence (as a rate of the number of LGL nodes increases) to the smooth solution [111].

With the definition of orthogonal collocation constraints and the Gauss quadrature rule, the HZD based pseudospectral optimization can be formulated as Section 4.3.2. The problem can be stated similar to (4.25), except that we replace the collocation constraints (4.25b) and (4.25c) in the original optimization problem with the global collocation constraints given in (4.50).

4.5 Structured NLP Construction

It is known that the direct collocation formulations significantly increase the number of constraints and optimization variables, leading to a *large* nonlinear programming problem. Yet, the Jacobian matrix of constraints is very sparse; the density of the matrix is far less than 1% in most cases. This feature allows the problem to be solved efficiently using appropriate large sparse NLP solvers such as IPOPT [132], SNOPT [51], etc. The fact that the optimization variables and constraints are defined in terms of collocation nodes inspire us to formulate the problem in a structured manner. This structured formulation is based on the defect variables and smart indexing of NLP variables and constraints. We have already introduce many defect variables in our previous formulation, for example, \dot{x} , u and λ , etc. The idea of introducing defect variables in the NLP is that instead of computing these variables explicitly using relatively complicated functions, we impose implicit but equivalent equality constraints, which are often computationally easier. The purposes are twofold:(1) it helps simplify the constraint expression, so that symbolically computing the analytic first-order Jacobian of constraints becomes feasible, (2) it distributes the “decision” weight of some particular variables. In large-scale sparse NLPs, providing analytic Jacobian of constraints would significantly reduce the computation time and improve the robustness the optimization convergence. The second reason particularly applies to some constant variables.

Different from the definition of optimization variables as in Section 4.3.2, we define

optimization variables at each node following the idea of defect variables as follows:

$$\mathbf{Y}_v^{(i)} := (T_{I,v}^{(i)}, q^{(i)}, \dot{q}^{(i)}, \ddot{q}^{(i)}, u^{(i)}, \lambda_v^{(i)}, \alpha_v^{(i)}, \bar{h}_v^{(i)}) \quad (4.53)$$

be a vector of optimization variables defined at each node $i \in \{0, 1, 2, \dots, N_v\}$ based on the time discretization of the domain \mathcal{D}_v . By defining variables in this way, we could group both optimization variables and constraints of one node as a unit, which simplifies the indexing of variables and constraints.

Remark 4.5. In particular, we include the constant vectors \bar{h}_v as decision variables for the (desired) holonomic constraints values so that they can be determined by the optimizer. These constants often include the gait properties such as the step length and width. By including \bar{h}_v as optimization variables, we have direct control over these properties in the optimization.

Remark 4.6. In addition, it can be noted that we define the time $T_{I,v}$ and parameters α_v , and \bar{h}_v at each node in spite of the fact that they are constants on a given domain. The reason of defining them in this way is that it distributes these parameters at isolated nodes, so that the “weights” of these parameters are spread over the entire domain. If these constant parameters are defined locally on each node, the following constraints must be enforced to guarantee that they are indeed constant. The consistency of parameters is formulated as:

$$T_{I,v}^{(i)} - T_{I,v}^{(i+1)} = 0, \quad (4.54)$$

$$\alpha_v^{(i)} - \alpha_v^{(i+1)} = 0, \quad (4.55)$$

$$\bar{h}_v^{(i)} - \bar{h}_v^{(i+1)} = 0, \quad (4.56)$$

for all $i \in \{0, 1, 2, \dots, N_v - 1\}$ and $v \in V$. Some parameters are in fact the same on different domains, therefore, they should be enforced to be consistent throughout these domains. For example, the desired holonomic constraints \bar{h}_c should be consistent

if a particular holonomic constraint $c \in \mathcal{C}_v$ is defined on both neighboring domains. Hence, we impose that

$$\bar{h}_c^{(N_v)} - \bar{h}_c^{(0_{v+})} = 0, \quad \forall c \in (\mathcal{C}_v \cap \mathcal{C}_{v+}). \quad (4.57)$$

for all vertices except the last one, i.e., $v \in V \setminus \{v_{n_p}\}$.

We start with stacking the decision variables of the NLP that were defined on all domains into a single vector of optimization variables, i.e.,

$$\mathbf{Y} = (\mathbf{Y}_{v_1}^{(0)}, \mathbf{Y}_{v_1}^{(1)}, \dots, \mathbf{Y}_{v_1}^{(N_v)}, \mathbf{Y}_{v_2}^{(0)}, \mathbf{Y}_{v_2}^{(1)}, \dots, \mathbf{Y}_{v_2}^{(N_v)}, \dots, \mathbf{Y}_{v_{n_p}}^{(0)}, \mathbf{Y}_{v_{n_p}}^{(1)}, \dots, \mathbf{Y}_{v_{n_p}}^{(N_v)}),$$

where $\mathbf{Y}_v^{(i)}$ given in (4.53). Due to the introduction of defect variables, the size of the resulting NLP problem is very large. To facilitate the construction of the sparse Jacobian matrix of the constraints or the gradient of the cost function, we arrange these variables in terms of collocation nodes (as defined in (4.53)) and then index them based on the positions in the stacked vector \mathbf{Y} . Similarly, we organize all constraints in (4.25) into a stacked vector of NLP constraints, $\mathbf{C}(\mathbf{Y})$. Then we state the structured NLP problem as,

$$\begin{aligned} \mathbf{Y}^* &= \underset{\mathbf{Y}}{\operatorname{argmin}} \mathcal{J}(\mathbf{Y}) \\ \text{s.t. } & \mathbf{C}^{\min} \leq \mathbf{C}(\mathbf{Y}) \leq \mathbf{C}^{\max}, \\ & \mathbf{Y}^{\min} \leq \mathbf{Y} \leq \mathbf{Y}^{\max}, \end{aligned} \quad (4.58)$$

where the cost function $\mathcal{J}(\mathbf{Y})$ is computed via (4.42). In addition, \mathbf{Y}^{\min} and \mathbf{Y}^{\max} are the vectors containing the minimum and maximum values of optimization variables, respectively, and \mathbf{C}^{\min} and \mathbf{C}^{\max} are the vectors containing the minimum and maximum values for the constraints, respectively. In the case of equality constraints, the corresponding minimum and maximum values are set to zero. It is also important to notice that physical constraints, such as torque bounds, joint velocity and angle limits, etc., can be imposed directly as the boundary value of corresponding optimization variables in \mathbf{Y}^{\min} or \mathbf{Y}^{\max} .

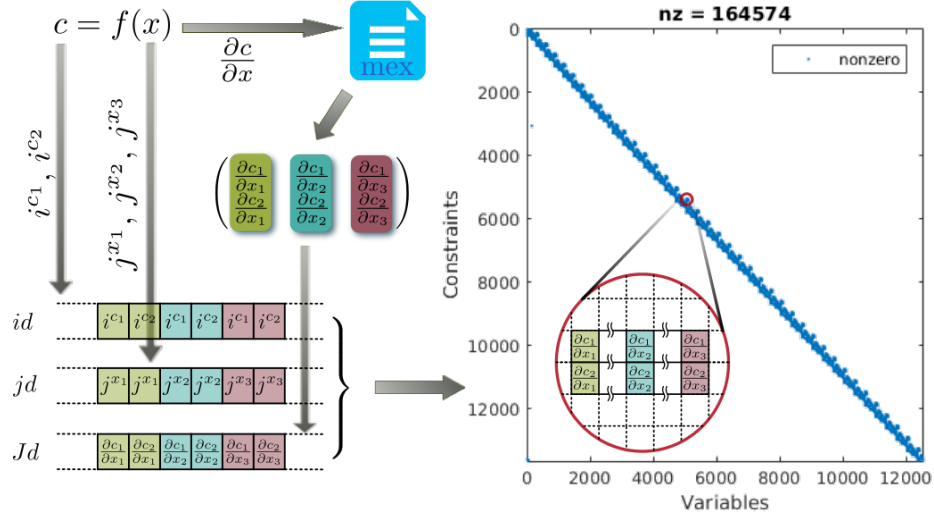


Figure 4.4: The illustration of sparse Jacobian matrix construction. For each constraint c , assuming it has two rows and depends on variables x_1, x_2, x_3 , the partial derivatives are computed using symbolically generated function, and three vectors id , jd , and Jd are formulated based on indices of constraints and variables and corresponding partial derivatives to construct the sparse Jacobian matrix.

Usually, the Jacobian matrix of constraints is obtained through the finite differencing or the automatic differentiation (also referred as algorithmic differentiation) [55]. Despite being straightforward to compute, the finite difference approach is very slow to evaluate numerically and often has very low accuracy. Automatic differentiation provides better accuracy, however, it often suffers from limitations caused by restrictions of the tools available. In our formulation, however, the introduction of defect variables significantly reduce the complexities of the analytic expression of the constraints and the cost function, so that symbolically generating the constraints' Jacobian becomes feasible. Furthermore, the symbolic Jacobian of a constraint equation could be generated *a priori* and stored as a function to be called by the optimizer during optimization process. For the same constraints that are imposed on multiple nodes, only one symbolic Jacobian function is required to be generated. This greatly reduces the overhead time of generating the full Jacobian matrix directly.

To expedite the optimization process, we exploit the sparsity pattern of the Jacobian matrix further. First, let i^c be the indices of an arbitrary constraint c , and

j^x be the indices of dependent variables x^c of the constraint c . Then the Jacobian of this constraint is given by a $n^c \times n^x$ matrix with $n^c = \text{Dim}(i^c)$, $n^x = \text{Dim}(j^x)$. Based on the indices of variables and constraints, the large sparse Jacobian matrix can be constructed based on the compressed column storage (CCS) format [50]. The following vectors are determined for the constraint c :

$$\begin{aligned} id^c &:= [i^c, && i^c, && \dots, && i^c], \\ jd^c &:= [j^x[1]_{1 \times n^c}, && j^x[2]_{1 \times n^c}, && \dots, && j^x[n^x]_{1 \times n^c}], \\ Jd^c &:= \left[\frac{\partial c}{\partial x^c_1}, && \frac{\partial c}{\partial x^c_2}, && \dots, && \frac{\partial c}{\partial x^c_{j^x}} \right] \end{aligned}$$

with $j^x[k]_{1 \times n^c}$ a $1 \times n^c$ vector of $j^x[k]$ at each element, and $\frac{\partial c}{\partial x^c_k}$ the k th column in the Jacobian matrix of the constraint c obtained by evaluating the associated symbolic Jacobian function of the current constraint. Among them, id^c and jd^c store the information of row and column indices of the constraints Jacobian, and Jd^c stores the values of the partial derivatives, respectively. Combining id^c , jd^c , and Jd^c of all constraints together, the whole Jacobian matrix of constraints is created using the MATLAB function, `sparse`. See Figure 4.4 for an illustration of this process.

4.6 Control Lyapunov Function Synthesis

With the construction of the direct HZD optimization problem as discussed in previous section, any feasible solution of (4.58) yields an optimal walking gait of the robot. While we explicitly constrain the output dynamics based on the input-output linearization controllers, there are wider ranges of optimal controllers that can be utilized to stabilize the resulting gaits. They are the rapidly exponentially stabilizing control Lyapunov functions (RES-CLF) as we introduced in Section 2.3.4. In this section, we explicitly construct the RES-CLF controllers in terms of a quadratic programming (QP) problem for the hybrid zero dynamics optimal gaits.

Theorem 4.2. *If the parameters α^* that solves the direct HZD optimization problem,*

then there exists a family of rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs) based feedback controllers that can be constructed to exponentially stabilize the transverse dynamics.

Proof. Recall that under the feedback control law u^{α_v} given in (2.39), we have a linear output dynamics of the form:

$$\begin{bmatrix} \dot{y}_{1,v} \\ \ddot{y}_{2,v} \end{bmatrix} = \mu_v, \quad (4.59)$$

for some auxiliary inputs μ_v . Therefore, given that $\eta_v = (y_{1,v}, y_{2,v}, \dot{y}_{2,v})$, the above output dynamics can be equivalently written as a linear control system:

$$\dot{\eta}_v = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix}}_{F_v} \eta_v + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}}_{G_v} \mu_v. \quad (4.60)$$

Then in the context of this control system, we consider the continuous time algebraic Riccati equations (CARE):

$$F_v^T P_v + P_v F_v - P_v G_v G_v^T P_v + Q_v = 0, \quad (4.61)$$

for $Q_v = Q_v^T > 0$ with solution $P_v = P_v^T > 0$. One can use P_v to construct a RES-CLF that can be used to exponentially stabilize the output dynamics at a user defined rate of $\frac{1}{\varepsilon}$ (see [9, 10]). In particular, define

$$V_v^\varepsilon(\eta_v) = \eta_v^T \underbrace{I^\varepsilon P_v I^\varepsilon}_{P_v^\varepsilon} \eta_v, \quad \text{with } I^\varepsilon = \text{diag}(\varepsilon I, I), \quad (4.62)$$

wherein it follows that:

$$\dot{V}_v^\varepsilon(\eta_v) = L_{F_v} V_v^\varepsilon(\eta_v) + L_{G_v} V_v^\varepsilon(\eta_v) \mu_v,$$

with

$$L_{F_v} V_v^\varepsilon(\eta_v) = \eta_v^T (F_v^T P_v^\varepsilon + P_v^\varepsilon F_v) \eta_v, \quad (4.63)$$

$$L_{G_v} V_v^\varepsilon(\eta_v) = 2\eta_v^T P_v^\varepsilon G_v.$$

With the goal of exponentially stabilizing the η_v to zero, we wish to find μ_v such that,

$$L_{F_v} V_v^\varepsilon(\eta_v) + L_{G_v} V_v^\varepsilon(\eta_v) \mu_v \leq -\frac{\gamma}{\varepsilon} V_v^\varepsilon(\eta_v),$$

for some $\gamma > 0$. In particular, it allows for specific feedback controllers, e.g., the min-norm controller, which can be stated as the closed form solution to a quadratic program (QP) stated in (2.62).

Recalling that $\mathcal{A}_v u = -L_{f_v} + \mu_v$, it follows that:

$$\mu_v^T \mu_v = u^T \mathcal{A}_v^T \mathcal{A}_v u + 2L_{f_v}^T \mathcal{A}_v u + L_{f_v}^T L_{f_v},$$

which allows for reformulating the QP problem in terms of u instead of μ_v , so that additional constraints on torques or reaction forces can be directly implemented in the formulation. To achieve an optimal control law, we can relax the CLF constraints and penalize this relaxation. In particular, we consider the following modified CLF-based QP in terms of u and a relaxation factor δ_v :

$$\underset{(u, \delta_v)}{\operatorname{argmin}} \quad p_v \delta_v^2 + u^T \mathcal{A}_v^T \mathcal{A}_v u + 2L_{f_v}^T \mathcal{A}_v u \quad (4.64)$$

$$\text{s.t} \quad \tilde{A}_v^{\text{CLF}}(q, \dot{q}) u \leq \tilde{b}_v^{\text{CLF}}(q, \dot{q}) + \delta_v \quad (\text{CLF})$$

where,

$$\tilde{A}_v^{\text{CLF}}(q, \dot{q}) := L_{G_v} V_v^\varepsilon(q, \dot{q}) \mathcal{A}_v(q, \dot{q}), \quad (4.65)$$

$$\tilde{b}_v^{\text{CLF}}(q, \dot{q}) := -\frac{\gamma}{\varepsilon} V_v^\varepsilon(q, \dot{q}) - L_{F_v} V_v^\varepsilon(q, \dot{q}) - L_{G_v} V_v^\varepsilon(q, \dot{q}) L_{f_v}, \quad (4.66)$$

and $p_v > 0$ is a large positive constant that penalizes violations of the CLF constraint. Note that we use the fact that η_v is a function of the system states (q, \dot{q}) , so the constraints can be expressed in the term of system states. \square

The end result of solving this QP is the optimal control law that guarantees exponential convergence of the control objective $y_v \rightarrow 0$ if $\delta_v \equiv 0$. In the case of

sufficiently small δ_v , we still achieve exponential convergence of the outputs, which motivates the minimization of δ_v in the cost of QP.

4.7 *Summary*

We presented a generalized optimization framework for synthesizing formally stable gaits on robots as complex as humanoids. This framework blends the theoretical foundation of hybrid zero dynamics (HZD) with direct collocation trajectory optimization techniques.

While at first glance this direct collocation approach adds thousands of design variables and constraints that are (seemingly) superfluous, it bears counterintuitive advantages. For one, the result is a very sparse NLP, for which the vast majority of constraint Jacobian entries are zero. In essence, each design variable only needs to worry about its effect on its neighboring points, not the full problem. Due to local linearity, this makes the effective problem much more linear at each iteration of the optimization, which is helpful for NLP solvers that iterate on linear approximations of the problem (e.g. sequential quadratic programming (SQP) and interior point (IP)). Second, these additional constraints can be expressed in closed form, not requiring an computationally iterative integration process, even when the original dynamical equations are unsolvable.

Further, this method optimizes the interactions of the full multi-domain multi-body dynamics of humanoid system models, without conforming motions to simpler more-tractable dynamics. Admittedly, optimizing the full order dynamics increases the number of optimization variables significantly. However, the simplification of expressions and the sparsity of the constraints Jacobian by the implicit constrained dynamics form and defect variables at the expense of the problem size can be fully utilized to make the optimization more robust and fast, as a result it can be scaled to higher degrees of freedom systems. With these simplified expressions, we are able

to rigorously compute analytic expressions of the gradient of the cost function and the Jacobian of the constraints, even the Hessian of the optimization problem, using any proper symbolic mathematics software. The end result is a fast and reliably converging nonlinear programming problem for generating dynamic walking gaits.

More importantly, for any feasible periodic gait obtained from the direct HZD optimization, there exists a class of feedback controllers that can be applied to stabilize the transverse dynamics while respecting the torque limits of the physical robots. This is a distinctive property of the proposed direct HZD gait optimization from other locomotion gait generation problems that appear in bipedal locomotion literatures [37, 69, 82].

CHAPTER V

APPLICATIONS TO HUMANOID ROBOTS

In this chapter, we present applications of the direct collocation formulations of hybrid zero dynamics gait optimization framework on three different bipedal robots. These examples demonstrates the scalability, robust and fast convergence of the direct HZD optimization even though a robot has many underactuated degrees of freedom (DURUS) or very high DOFs (HUBO). By exploiting of the full-order dynamics of the robot in the optimization, it results in not only energy-efficient but also very natural, human-like locomotion. Finally, we will show that the advanced pseudospectral optimization method is even capable of generating new gaits on-line to accommodate the environmental changes.

With the systematic framework of the hybrid zero dynamics and direct collocation optimization, the process of implementing dynamic walking on a robot can be automated, see Figure 5.1 for the illustration of the process. Given a robot of the interest, we first model the desired behavior as a multi-domain hybrid system model, and then design a set of virtual constraints as our control objectives. This direct collocation HZD optimization framework parses the multi-body model of the robot and a set of parameterized virtual constraints into a large and sparse nonlinear program (NLP) with upwards of 10,000 design variables and constraints. Large-scale algorithms can typically solve this NLP, thereby optimizing a dynamic gait for the robot that exploits the full multi-body dynamics of the machine, despite underactuation. The solution of such a NLP yields a periodic behavior for the system. By implementing the resulting joint trajectories on the actual hardware, we realize dynamic locomotion of the robot.

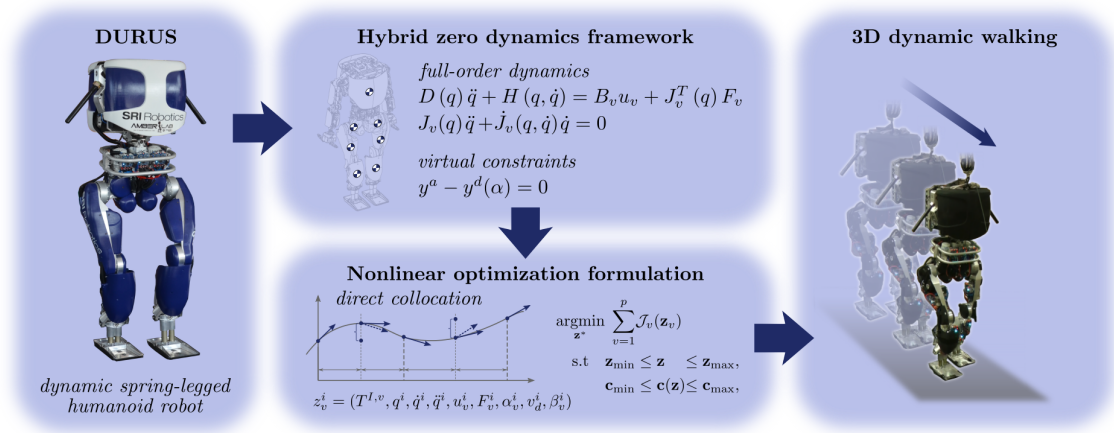


Figure 5.1: Illustration of the process used to generate dynamic walking gaits.

5.1 DURUS – Efficient Humanoid Walking

In this section, we explicitly generate energy efficient walking gaits on a spring-legged humanoid robot–DURUS. The goal is to demonstrate that the proposed gait generation framework is capable of generating energy efficient walking gaits by embracing the high-dimensional full-order dynamics of the particular mechanical design that consists of underactuated linear springs.

5.1.1 Robot Model

Developed by SRI International, DURUS was designed with the overarching goal of achieving never before seen efficiency in locomotion, thereby allowing for longer autonomous battery-powered operation. This goal is in response to the current state of the art in humanoid robots. The underlying mechanical and electrical components incorporated into the design of DURUS provide an essential foundation from which the control design can build upon.

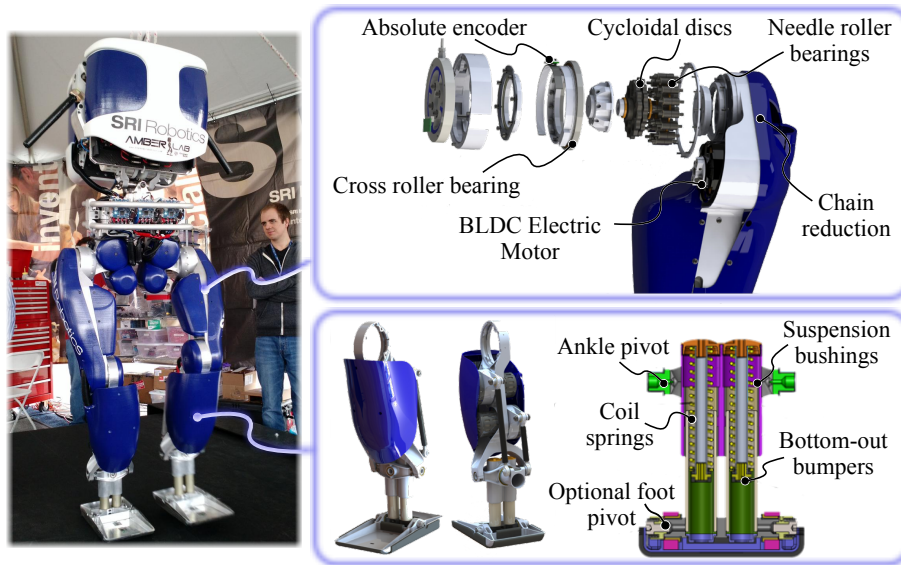


Figure 5.2: The mechanical design of the humanoid DURUS. The design of linear springs at the end of legs is to absorb the energy at impact.

5.1.1.1 Mechanical Design

The primary mechanical components which provided gains in overall efficiency were the actuator and transmission elements (see Figure 5.2). Each actuator-gearbox combination consists of an electric motor connected via a chain reduction to a custom-designed cycloid transmission, which can achieve up to 97% efficiency. Each actuator-gearbox unit was lightweight, weighing only 2.7 kg and able to output 250 Nm of torque with maximum joint accelerations exceeding 130 rad/s^2 . To ultimately realize dynamic and efficient locomotion on the humanoid robot DURUS, precision in control implementation is required at every level of the hardware. Therefore, an essential component in the process of realizing locomotion is a motor controller which can accurately track the trajectories. Custom motor controllers are employed on DURUS, allowing for 10 kHz control of torque, current, and position. For the duration of the walking, these motor controllers tracked joint positions with an overall rms error of 0.005 rad and a peak error of 0.026 rad . Additionally, DURUS is self-powered with a 1.1 kWh battery pack weighing 9.5 kg .

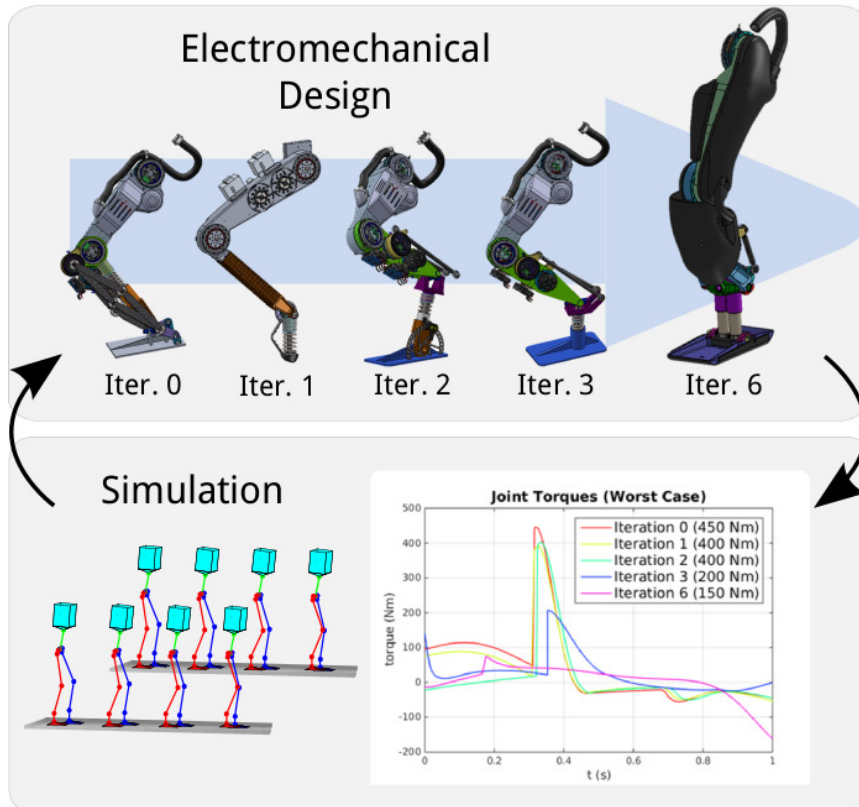


Figure 5.3: Iterative design process which facilitated a leg morphology satisfying hardware constraints in simulation.

A key difference between DURUS and many humanoid robots is the use of passive springs in the ankles with significant compliance. To leverage the greatest returns, the springs are much more compliant than typically seen on powered humanoid robots leveraging springs for efficiency [105, 141]. The morphology of DURUS, and specifically the role of passive-compliant elements, directly impacts how well a control scheme can achieve efficient and stable dynamic gaits. The leg morphology of DURUS is the result of an iterative collaboration between the designer and control engineers. Specifically, designs for the leg geometry and passive-compliant ankles were passed to rigorous simulation for evaluation. In particular, the nonlinear control and gait approach in [64] was utilized to realize walking in simulation and the design was evaluated with regard to performance parameters such as the joint torques and walking stability. These findings were then compiled and passed back to the design engineer

for improvement. The result of this iterative process was a leg design which walked in simulation with worst case torques of 150 Nm, as opposed to initial leg designs which demonstrated peak torques of 450 Nm; this procedure, along with several of the leg designs and their associated simulation torques, is illustrated in Figure 5.3. We believe that this “control in the loop” mechanical design methodology was a key factor in the ability of the control scheme presented in Section 2.3 to maintain smooth, stable walking while exploiting the energy saving capabilities of the passive-compliant ankle structures. The higher degree of compliance in DURUS allows for the design of gaits which can be designed with significant energy savings at impact. The drawback to this compliance is the injection of additional passive degrees of freedom in the robot which are difficult to control. This difficulty motivates the use of hybrid zero dynamics control framework combined with direct HZD optimization techniques which can generate and control stable walking gaits on an underactuated robots.

5.1.1.2 Kinematic Model

DURUS consists of fifteen actuated joints and two passive springs (see Figure 5.4). The passive springs, which are rigidly perpendicular to the feet, are designed to reduce the energy loss during foot impact while walking. Here we use the floating base coordinates, assuming that the origin of the six-dimensional base coordinates, R_b , is rigidly attached at the center of the pelvis link. In addition, the robot consists of 15 actuated joints and 2 passive springs shown in Figure 5.4, where R_0 is the inertial frame, R_b is the robot base frame located at the center of the pelvis with p_b , ϕ_b is the position and orientation of R_b . ψ_w , ϕ_w , and θ_w are the waist yaw, roll, and pitch angles, ψ_{lh} , ϕ_{lh} , θ_{lh} , θ_{lk} , θ_{la} , ϕ_{la} , and r_{ls} are the left hip yaw, hip roll, hip pitch, knee pitch, ankle pitch, ankle roll angles, and spring deflection, respectively, and ψ_{rh} , ϕ_{rh} , θ_{rh} , θ_{rk} , θ_{ra} , ϕ_{ra} , and r_{rs} are the right hip yaw, hip roll, hip pitch, knee pitch, ankle pitch, ankle roll angles, and spring deflection, respectively. The red arrow of each joint represents the

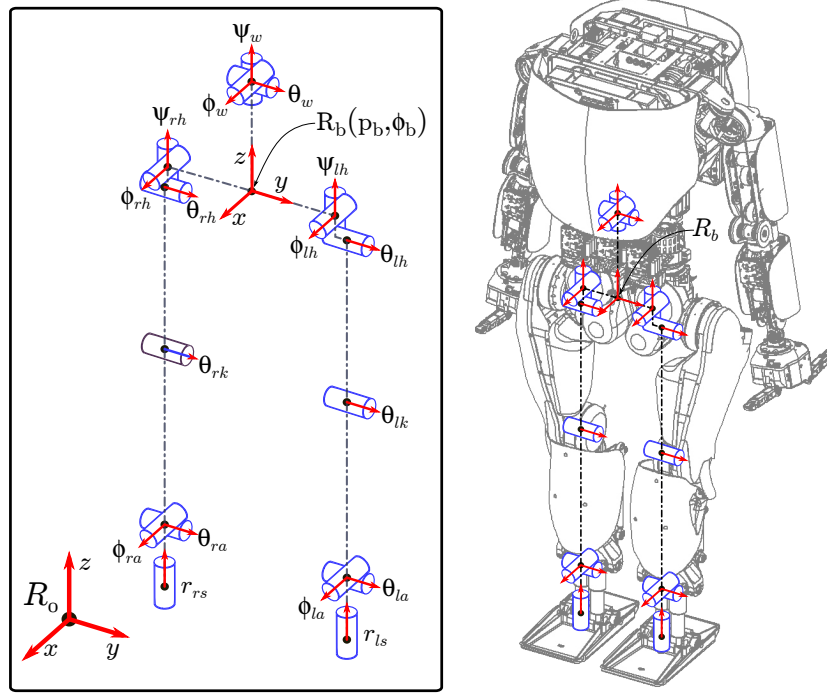


Figure 5.4: The coordinates of DURUS robot.

positive rotation (or translation) axis of the corresponding joint using the right hand rule. The kinematic tree structure of body coordinates consists of three branches: the waist joints, $q_w = [\psi_w, \phi_w, \theta_w]^T$, the left leg joints, $q_l = [\psi_{lh}, \phi_{lh}, \theta_{lh}, \theta_{lk}, \theta_{la}, \phi_{la}, r_{ls}]^T$, and the right leg joints, $q_r = [\psi_{rh}, \phi_{rh}, \theta_{rh}, \theta_{rk}, \theta_{ra}, \phi_{ra}, r_{rs}]^T$, respectively. Specifically, we model the passive spring as prismatic joints, and then apply the spring forces as “feedback controllers” on these joints by ensuring that the “feedback controllers” indeed equal to the spring forces. Including the floating base coordinates, the robot has 23 degrees of freedom, therefore, the generalized coordinates of the robot are given by

$$q = (p_b, \phi_b, q_w, q_l, q_r) \in \mathcal{Q} \subset \mathbb{R}^{23}. \quad (5.1)$$

The continuous dynamics of the robot is determined by (2.17). In addition, we incorporate the reflected motor inertia of actuators as a decoupled addition to the inertia matrix as in [36]. Let $D_m \in \mathbb{R}^{23 \times 23}$ be a diagonal matrix with each element

in the diagonal line corresponding to the reflected inertia of a rotational actuator through the gearbox (zero if not actuated), then for a given domain \mathcal{D}_v , we modify (2.17) such that:

$$D^R(q)\ddot{q} + H^R(q, \dot{q}) = B_v u_v + J_v^T(q)\lambda_v, \quad (5.2)$$

where $D^R(q) = D(q) + D_m$, and $H^R(q, \dot{q}) = H(q, \dot{q}) + \kappa(q, \dot{q})$ with $\kappa(q, \dot{q}) : T\mathcal{Q} \rightarrow \mathbb{R}^{23}$ is a vector of spring forces. Let k_s be the stiffness and b_s be the damping coefficient of the spring, the spring force vector $\kappa(q, \dot{q}) : T\mathcal{Q} \rightarrow \mathbb{R}^{23}$ is a vector that has only two non-zero entries, given by

$$\kappa(q, \dot{q}) = [\mathbf{0}_{1 \times 15}, k_s r_{ls} + b_s \dot{r}_{ls}, \mathbf{0}_{1 \times 6}, k_s r_{rs} + b_s \dot{r}_{rs}]^T, \quad (5.3)$$

where $\mathbf{0}_{n \times m}$ represents $n \times m$ matrix with all entries being zero.

Remark 5.1 (Symmetric Gait Model). In the study of symmetric walking gaits, a “common” trick is to define the coordinates in terms of stance/non-stance legs instead of using left/right legs. Followed by this definition is a relabeling of coordinates at foot impact due to the change of stance leg, which can be done by a linear map, $\mathcal{R} : \mathcal{Q} \rightarrow \mathcal{Q}$, in which left and right leg angles are switched accordingly and the signs of all roll and yaw angles as well as the base position in y -axis direction are “flipped” [59]. For the coordinates system defined in (5.1), the relabeling matrix can be given as

$$\mathcal{R} := \begin{bmatrix} \mathcal{R}_b & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_w & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{R}_l \\ \mathbf{0} & \mathbf{0} & \mathcal{R}_l & \mathbf{0} \end{bmatrix}, \quad (5.4)$$

where \mathcal{R}_b , \mathcal{R}_w , and \mathcal{R}_l are the sign “flipping” matrix for base, waist and leg coordinates, given by

$$\mathcal{R}_b = \text{diag}(1, -1, 1, -1, 1, -1), \quad (5.5)$$

$$\mathcal{R}_w = \text{diag}(-1, -1, 1), \quad (5.6)$$

$$\mathcal{R}_l = \text{diag}(-1, -1, 1, 1, 1, -1, 1), \quad (5.7)$$

where diag represents a diagonal matrix. For the sake of simplicity, we assume the right leg is the stance leg in the remainder of the section. The advantage of defining in this manner is that it reduces the number of domains by half. In this thesis, all gaits considered have left-right symmetry, therefore, we assume the right leg is the stance leg and the left leg is the non-stance leg in the following discussion.

Given the configuration of DURUS, we now construct hybrid system models of different walking patterns based on contact conditions and domain orderings. To define the different foot contact configurations, we define the following foot contact points: the stance foot, p_{sf} , the stance toe, p_{st} , the stance heel, p_{sh} , the non-stance foot, p_{nsf} , the non-stance toe, p_{nst} , and the non-stance heel, p_{nsh} , as depicted in Figure 5.5. In the following discussions, we denote that $p_{\square}(q) = (p_{\square}^x, p_{\square}^y, p_{\square}^z) : \mathcal{Q} \rightarrow \mathbb{R}^3$ be the three dimensional Cartesian position of a point in the world frame, and $\phi_{\square}(q) = (\phi_{\square}^x, \phi_{\square}^y, \phi_{\square}^z) : \mathcal{Q} \rightarrow SO(3)$ be the three dimensional orientations of the link—to which the point attached—with respect to the world inertial frame. Correspondingly, we denote $\lambda_{\square} := (\lambda_{\square}^{fx}, \lambda_{\square}^{fy}, \lambda_{\square}^{fz}, \lambda_{\square}^{mx}, \lambda_{\square}^{my}, \lambda_{\square}^{mz})$ be the six-dimensional wrenches associated with the contact constraints at the point \square , among which $(\lambda_{\square}^{fx}, \lambda_{\square}^{fy}, \lambda_{\square}^{fz})$ be the ground reaction forces acting on the point along x , y , and z directions, and $(\lambda_{\square}^{mx}, \lambda_{\square}^{my}, \lambda_{\square}^{mz})$ be the ground reaction moments acting on the foot around x , y , and z axes, respectively.

5.1.2 3D Flat-Footed Walking

The term “flat-footed” indicates that the feet remain flat with respect to the ground plane when on the ground. It is used to distinguish this type of walking from the multi-contact case—which will be discussed in the next subsection—where feet can be angled in any number of ways. To characterize the unsupported walking of the humanoid robot from the restricted planar 2D walking gaits, we specifically name

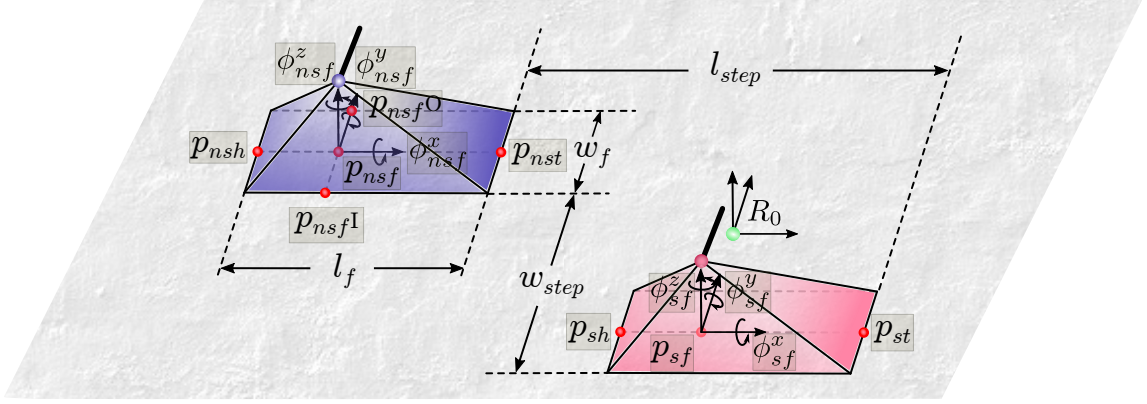


Figure 5.5: Illustration of the position of different contact points.

the walking as 3D walking. In this case, the two-domain hybrid system model of the flat-footed walking gaits is constructed based on the discussion in Chapter 2. The direct HZD optimization framework is then formulated to generate energy efficient walking gaits that fully exploits the special mechanical design of the robot.

5.1.2.1 Multi-Domain Hybrid System Model

The design of passive springs at the end of each leg permits a non-trivial double support phase, therefore, the hybrid system model of 3D flat-footed walking, consists of two domains: a *double-support* domain, \mathcal{D}_{ds} , when both feet are on the ground, and a *single-support* domain, \mathcal{D}_{ss} , when only one foot is on the ground (see Figure 5.6).

Contact constraints. Let p_{sf} and p_{nsf} be a point on the stance foot and non-stance foot respectively (see Figure 5.5), the holonomic constraints of each domain are defined as follows:

$$h_{ds}(q) := (p_{sf}, \phi_{sf}, p_{nsf}, \phi_{nsf}) \in \mathbb{R}^{12}, \quad (5.8)$$

$$h_{ss}(q) := (p_{sf}, \phi_{sf}) \in \mathbb{R}^6. \quad (5.9)$$

Further, all orientation constraints, ϕ_{sf} and ϕ_{nsf} , should be zero because the feet should remain flat on the ground. The desired values of the position depend on the relative position of the point within the inertia frame. Assume that the origin of the

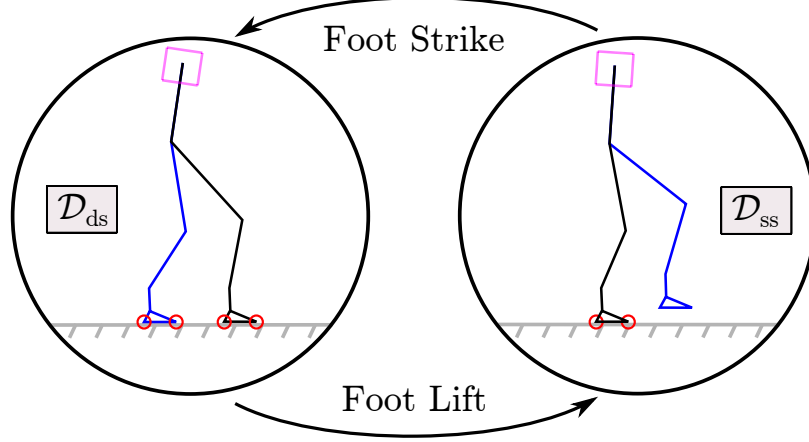


Figure 5.6: Domain graph of the two-domain 3D flat-footed walking model.

inertia frame, R_0 , is located at a position as shown in Figure 5.5, and l_{step} be the step length and w_{step} be the step width, respectively, then we have

$$\bar{p}_{sf} = [0, -\frac{w_{step}}{2}, 0]^T, \quad \bar{p}_{nsf} = [-l_{step}, \frac{w_{step}}{2}, 0]^T. \quad (5.10)$$

Domains. The admissibility conditions for each domain are determined so that constraint wrenches associated with foot contacts should ensure that any foot on the ground should remain flat. These conditions include: (1) normal forces of the ground acting against the feet should be positive; (2) feet should not slide on the ground; and (3) the robot should not roll over the edge of the feet. For instance, these constraints regarding the stance foot contact are given as:

$$\lambda_{sf}^{fz} \geq 0, \quad (5.11)$$

$$|\lambda_{sf}^{fx}| < \frac{\mu}{\sqrt{2}} \lambda_{sf}^{fz}, \quad (5.12)$$

$$|\lambda_{sf}^{fy}| < \frac{\mu}{\sqrt{2}} \lambda_{sf}^{fz}, \quad (5.13)$$

$$-\frac{w_f}{2} \lambda_{sf}^{fz} < \lambda_{sf}^{mx} < \frac{w_f}{2} \lambda_{sf}^{fz}, \quad (5.14)$$

$$-l_h \lambda_{sf}^{fz} < \lambda_{sf}^{my} < l_t \lambda_{sf}^{fz}, \quad (5.15)$$

where $l_h = |p_{sf} - p_{sh}|$ and $l_t = |p_{sf} - p_{st}|$ are the distances from the stance foot point to stance heel and stance toe, respectively, and μ is the assumed friction coefficient

of the ground contact. Considering that both feet remains flat on the ground during the double-support domain, these constraints should apply to both feet. The domain of admissibility condition for the double-support domain is given as:

$$A_{\text{ds}}(q, \dot{q}, u) = \begin{bmatrix} \nu_{sf}(q)\lambda_{sf}(q, \dot{q}, u) \\ \nu_{nsf}(q)\lambda_{nsf}(q, \dot{q}, u) \end{bmatrix} \geq 0, \quad (5.16)$$

where $\nu_{sf}(q)\lambda_{sf}(q, \dot{q}, u)$ is obtained by grouping constraints (5.11)–(5.15) together. The same is done for the non-stance foot constraint wrenches. For the single-support domain, only the stance foot is on the ground. In addition, the non-stance foot should be always above the ground during the single-support domain, which could be formulated as an unilateral constraint of the domain. Hence we have

$$A_{\text{ss}}(q, \dot{q}, u) = \begin{bmatrix} \nu_{sf}(q)\lambda_{sf}(q, \dot{q}, u) \\ p_{nsf}^z \end{bmatrix} \geq 0. \quad (5.17)$$

Consequently, we can define \mathcal{D}_{ds} and \mathcal{D}_{ss} as in (2.24).

Guards. Accordingly, a transition from double-support to single-support domain takes place when the normal force on non-stance foot reaches zero, and a transition from single-support to double-support domain occurs when the non-stance foot strikes the ground, i.e.,

$$H_{\text{ds}}(q, \dot{q}, u) := \lambda_{nsf}^{fz}(q, \dot{q}, u), \quad (5.18)$$

$$H_{\text{ss}}(q, \dot{q}, u) := p_{nsf}^z(q), \quad (5.19)$$

and thereby the guards $S_{\text{ds} \rightarrow \text{ss}}$ and $S_{\text{ss} \rightarrow \text{ds}}$ are determined by (2.26).

Reset maps. No impact or coordinate change occurs when transitioning from a double-support to single-support domain, i.e., $\Delta_{\text{ds} \rightarrow \text{ss}} = I$ where I is an identity matrix. On the other hand, the reset map, $\Delta_{\text{ss} \rightarrow \text{ds}}$, needs to incorporate the impact of the non-stance foot strikes and the change of coordinates caused by the switching of stance leg, as described in (2.30). In particular, the coordinate relabeling map \mathcal{R} for DURUS is defined in (5.4).

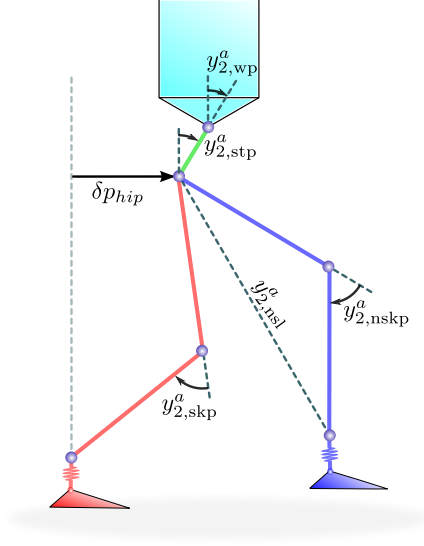


Figure 5.7: 3D flat-footed walking outputs defined in the sagittal plane.

With the definition of individual elements, the two-domain hybrid control system model $\mathcal{H}\mathcal{C}_{\text{ff}}$ for the 3D flat-footed walking behavior can be constructed as in (2.1).

5.1.2.2 Virtual Constraints

The choice of virtual constraints is inspired by Ames et al. previous work regarding the human-inspired bipedal locomotion [12]. Instead of using the canonical walking function as the desired output function form, we use 4th-order Bézier polynomial, as given in (2.34). Due to the fact that the robot has actuated ankle pitch joints that are active during both double-support and single-support domains, we pick the linearized hip position given by

$$y_{1,v}(q) := \delta p_{\text{hip}}(q) = L_a \theta_{ra} + (L_a + L_c) \theta_{rk} + (L_a + L_c + L_t)(\theta_{rh} + \phi_b^y) \quad (5.20)$$

as the velocity-modulating output $y_{1,v}^a(q)$ for both domains, where L_a , L_c , and L_t are the height of ankle¹, the length of calf and thigh of the robot, respectively. For simplicity, we mostly consider linear outputs as our position-modulating outputs.

¹This is measured when the spring has zero deflection.

Table 5.1: Position-modulating outputs library for DURUS 3D walking.

name	definition
stance knee pitch (skp):	$y_{2,\text{skp}}^a = \theta_{rk},$
stance ankle pitch (sap):	$y_{2,\text{sap}}^a = \theta_{ra},$
stance torso pitch (stp):	$y_{2,\text{stp}}^a = -\theta_{ra} - \theta_{rk} - \theta_{rh},$
stance ankle roll (sar):	$y_{2,\text{sar}}^a = \phi_{ra},$
stance torso roll (str):	$y_{2,\text{str}}^a = -\phi_{ra} - \phi_{rh},$
stance hip yaw (shy):	$y_{2,\text{shy}}^a = \psi_{rh}.$
waist roll (wr):	$y_{2,\text{wr}}^a = \phi_w,$
waist pitch (wp):	$y_{2,\text{wp}}^a = \theta_w,$
waist yaw (wy):	$y_{2,\text{wy}}^a = \psi_w,$
non-stance ankle pitch (nsap):	$y_{2,\text{nsap}}^a = \theta_{la}.$
non-stance knee pitch (nskp):	$y_{2,\text{nskp}}^a = \theta_{lk},$
linearized non-stance slope (nsl):	$y_{2,\text{nsl}}^a = -\theta_{ra} - \theta_{rk} - \theta_{rh} + \frac{L_c}{L_c+L_t}\theta_{lk} + \theta_{lh},$
non-stance leg roll (nslr):	$y_{2,\text{nslr}}^a = \phi_{rh} - \phi_{lh},$
non-stance foot roll (nsfr):	$y_{2,\text{nsfr}}^a = p_{\text{nsf}^I}^z(q) - p_{\text{nsf}^O}^z(q),$
non-stance foot pitch (nsfp):	$y_{2,\text{nsfp}}^a = p_{\text{nst}}^z(q) - p_{\text{nsh}}^z(q),$
non-stance foot yaw (nsfy):	$y_{2,\text{nsfy}}^a = p_{\text{nst}}^y(q) - p_{\text{nsh}}^y(q).$

Specifically, we define a library that consists of all available outputs for use, as listed in Table 5.1. We assume the right leg is the stance leg here. For domains in which the left leg is the stance leg, we swap the left and right leg angles accordingly in the definition. Figure 5.7 illustrates outputs that are defined in the sagittal plane. The last three outputs of the single-support domain are nonlinear outputs equivalently representing the orientations of the non-stance foot. The locations of points nst , nsh , nsf^I , and nsf^O are shown in Figure 5.5. These outputs were chosen over Euler angles in order to avoid expressions which contain inverse trigonometric functions. To guarantee that the non-stance foot remains flat, the desired outputs associated with these three outputs should be zero.

Due to the holonomic constraints imposed on the non-stance foot, non-stance leg joints should not be controlled via virtual constraints, otherwise, the system will be over-constrained. Notwithstanding, we include the non-stance knee pitch angle output for the double support domain because the passive spring introduces one additional degree of freedom to the non-stance leg. For the single-support domain, the outputs

selection takes the fact that the non-stance foot is no longer constrained in contact with the ground into consideration. As a result, we have the following output indexing sets for each domain:

$$\mathcal{O}_{ds} = \{\text{skp, stp, sar, str, shy, wr, wp, wy, nskp}\}, \quad (5.21)$$

$$\mathcal{O}_{ss} = \{\text{skp, stp, sar, str, shy, wr, wp, wy, nskp, nsl, nslr, nsfr, nsfp, nsfy}\}. \quad (5.22)$$

With the selection of virtual constraints for each domain, we can apply the feedback controllers as in (2.39) to make the system closed loop. Let $\alpha := \{\bar{v}_v, \alpha_{ds}, \alpha_{ss}\}$ be the parameters of desired velocity and position modulating outputs, the next step is find a set of parameters α so that the closed-loop hybrid system has a hybrid invariant periodic orbit.

5.1.2.3 Gait Optimization

The formulation of the optimization problem follows the direct HZD optimization framework presented in Chapter 4.

Cost Function. To achieve efficient walking, we set the objective function to minimize the mechanical cost of transport of the walking gait, which is given as the total mechanical work done by the actuators divided by the weight of the robot and the distance traveled during one step [32]. Hence, the running cost is defined as

$$\mathbf{L}_v(\bar{h}_v, \dot{q}^{(i)}, u^{(i)}) := \frac{\|P_v(u^{(i)}, \dot{q}^{(i)})\|}{mgl_{step}(\bar{h}_v)}, \quad (5.23)$$

where mg is the robot weight, $l_{step}(\bar{h}_v)$ is the distance traveled during a gait which could be determined from the desired holonomic constraints, and $P_v(u^{(i)}, \dot{q}^{(i)})$ is the total power consumed assuming no power-regeneration (see [77]) computed at each node. No terminal cost is defined. The cost function of the gait generating NLP is then can constructed using the Simpson's quadrature rule as described in Section 4.3.3.

Constraints. The necessary NLP constraints can be formulated based on the hybrid

system model defined in the previous subsection with proper upper and lower boundaries imposed on both optimization variables and constraints according to physical limitations of the robot hardware. For more detailed description of these necessary constraints, please refer to Section 4.3. In addition to the necessary constraints of the HZD direct collocation optimization in (4.25), we iteratively add additional physical constraints based on the observations of actual hardware implementation as gaits are tested on the physical hardware. Using this approach, the following constraints are added to the gait optimization and are configured specifically to provide favorable conditions for experimental walking in order to achieve sustainable 3D flat-footed walking gaits

Restricting torso movement. The robot tends to fall more easily when the upper body wobbles. This can be prevented by constraining the torso movement in the gait design. Let $\phi_{tor}(q) : \mathcal{Q} \rightarrow \mathbb{R}^3$ be the three dimensional orientations of the upper torso link, we restrict them within a small range $[\phi_{tor}^{\min}, \phi_{tor}^{\max}]$, i.e.,

$$\phi_{tor}^{\min} \leq \phi_{tor}(q) \leq \phi_{tor}^{\max}. \quad (5.24)$$

Keeping non-stance foot flat. the non-stance foot is constrained to be flat in the air so that it strikes the ground flat, i.e., three outputs associated with the orientation of the non-stance foot, $\{y_{2,nsfr}^d, y_{2,nsfp}^d, y_{2,nsfy}^d\}$, should be zero. This can be enforced by setting parameters of the desired functions to zero.

Constraining impact velocities. It became apparent through testing that if the swing foot impacts the ground too hard, it can destabilize the robot's balance. Therefore, we constrain the impact velocities of the heel to be within a reasonable range. Let $v_x^{\max}, v_y^{\max}, v_z^{\max} > 0$ be the maximum allowable impact velocities in x , y , and z direction respectively, then the swing foot velocities $\dot{h}_{nsf}(q^-, \dot{q}^-)$ should satisfy

$$|\dot{h}_{nsf}^x(q^-, \dot{q}^-)| \leq v_x^{\max}, \quad (5.25)$$

$$|\dot{h}_{nsf}^y(q^-, \dot{q}^-)| \leq v_y^{\max}, \quad (5.26)$$

$$|\dot{h}_{nsf}^z(q^-, \dot{q}^-)| \leq v_z^{\max}, \quad (5.27)$$

where $(q^-, \dot{q}^-) \in \mathcal{D}_{ss} \cap \mathcal{S}_{ss \rightarrow ds}$.

Avoiding swing leg collision. Due to the existence of compliance in the mechanical system, the swing leg can strike the stance leg if they are not separated enough. The separation of legs can be expressed as the difference between stance and swing hip roll angles. Therefore during the single-support domain, the non-stance leg roll output is constrained as

$$\phi^{\min} \leq y_{2,nsf}^a(q) \leq \phi^{\max}, \quad (5.28)$$

where $\phi^{\max} > \phi^{\min} \geq 0$ are the maximum and minimum allowable separation angles.

We incorporate these practical constraints into the direct HZD optimization problem in (4.58). For this application, we use the piece-wise continuous local direct collocation method particularly. In addition, the number of cardinal nodes are picked as 10 and 20 for the double-support and single-support domain, respectively. Solving the direct HZD optimization problem results in periodic energy-efficient flat-footed walking gaits for DURUS. More importantly, these gaits does not comply to any simplified models, thereby exhibiting more dynamic motions.

The gait optimization NLP is solved using IPOPT within MATLAB on a laptop computer with an Intel Core i7-6820HQ processor (2.7 GHz x 8) and 8 GB of RAM. In addition, we use the linear solver ma57 for IPOPT and set the feasibility tolerance to be 1×10^{-6} . The convergence of the NLP depends on the initial guess seeded to the solver and the constraints imposed. For the 3D flat-footed walking of DURUS, the NLP converges typically from 5-20 minutes from randomly generated initial guesses. It can be reduced to less than 3 minutes if seeded with better initial guesses. More thorough discussions of the performance of the direct HZD optimization are presented in Chapter 6.

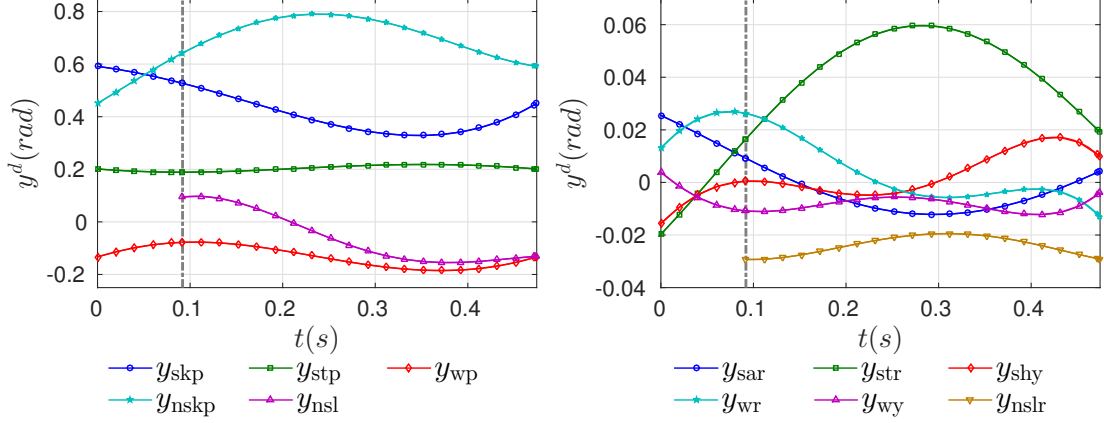


Figure 5.8: Desired outputs of the 3D flat-footed walking gait. The vertical dotted lines represent the moment of domain transition.

5.1.2.4 Simulation and Experimental Results

In the subsection, we present the simulation and experimental results of one of many stable periodic gaits obtained from the optimization. The step length of this particular gait is 0.1304 m and the step width is 0.2370 m . The total elapsed time of one complete gait step is 0.4767 seconds. The desired linearized hip velocity is 0.2974 m/s for this gait, and the resulting desired trajectories of position-modulating outputs are shown in Figure 5.8. The desired last three outputs, $(y_{2,\text{nsfr}}^d, y_{2,\text{nsfp}}^d, y_{2,\text{nsfy}}^d)$, are all zero, therefore, we omit them from the plot. A video of this 3D flat-footed walking gait in simulation and experiment can be found in [1].

Simulation results. To demonstrate the convergence of actual outputs to given desired trajectories under the feedback controllers, we simulate the system starting from a disturbed rest position. This initial condition is determined by slightly disturbing the fixed point of the original periodic orbit on the Poincaré section, and set all joint velocities to be zero. As shown in Figure 5.9, the outputs y_2 and derivatives \dot{y}_2 converged to zero at the end of every step under the input-output linearization control given in (2.39). More importantly, the output errors due to the swing foot impact decreased in every step, and reduced to less than 1×10^{-3} after just two steps. The

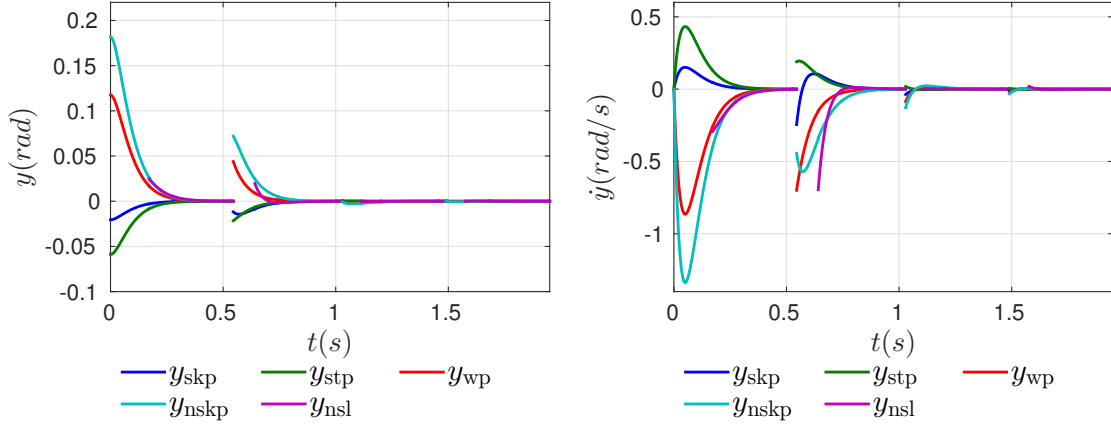
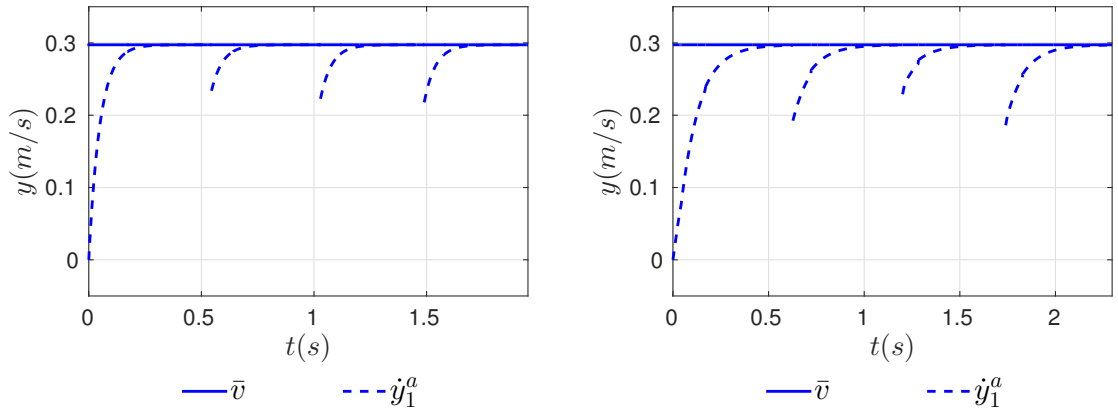


Figure 5.9: Convergence of outputs y_2 and \dot{y}_2 in 4 steps with input-output linearization controllers when simulating the flat-footed walking gait from rest. We only show errors of sagittal plane outputs because these outputs have larger errors than others.



(a) Input-output linearization controllers

(b) QP based CLF controllers

Figure 5.10: Tracking of the velocity-modulating output using both IO linearization controllers and QP based CLF controllers in flat-footed walking simulation.

convergence of the relative degree one output y_1 is also shown in Figure 5.10a, from which it can be seen that the initial velocity of the robot is zero. The torque inputs when using the input-output linearization controllers, however, exceeds the maximum 250 Nm torque limit of the robot during the first step, as shown in Figure 5.11a. Hence, we simulated the system from the same initial condition using QP based CLF controllers given in (4.64) with proper torque limits imposed in the QP. In particular, all position-modulating relative degree two outputs are relaxed. Figure 5.10b shows the tracking of the linearized hip velocity using QP based CLF controllers, and the

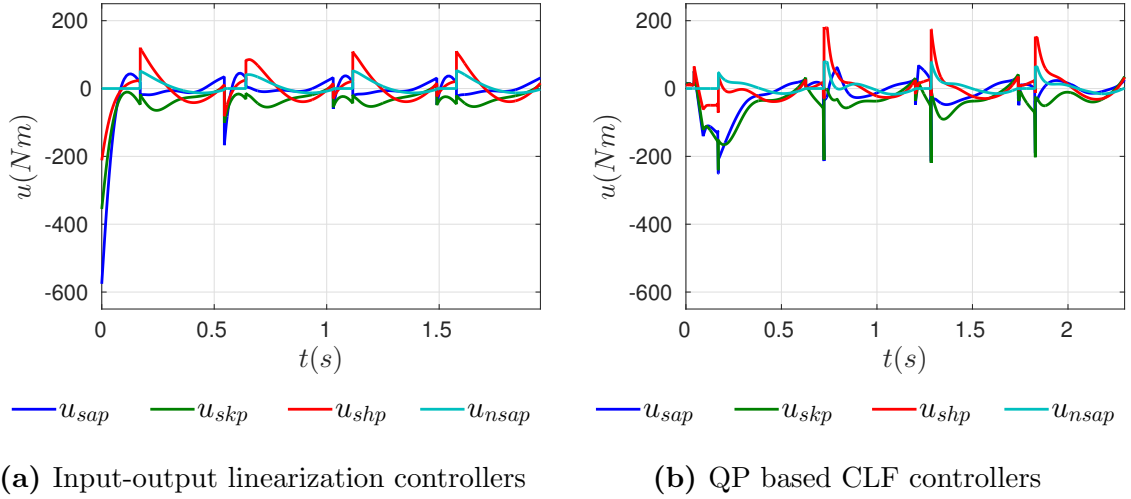


Figure 5.11: Joint torque profiles when using two different feedback control laws in DURUS flat-footed walking simulation. Here we show torques of stance ankle pitch, u_{sap} , stance knee pitch, u_{skp} , stance hip pitch, u_{shp} , and non-stance ankle pitch, u_{nsap} , considering that these joints often require larger torques than others.

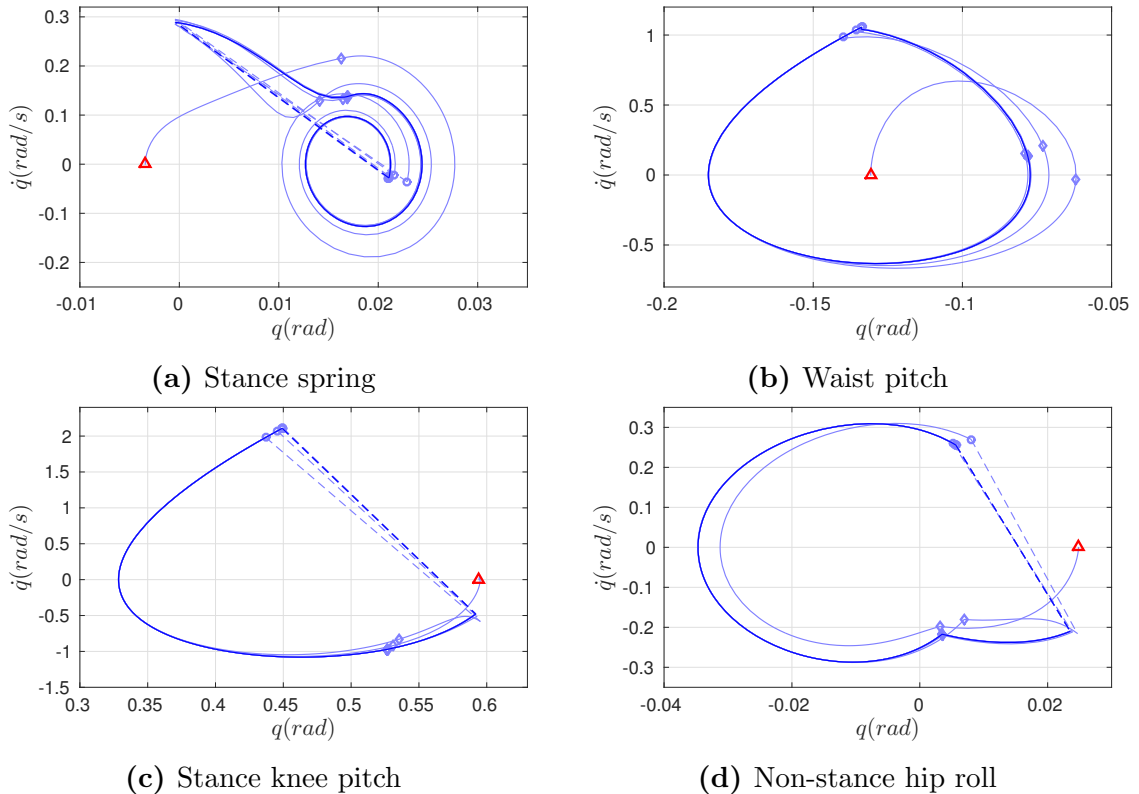


Figure 5.12: Phase portraits of representative joints in the 3D flat-footed walking simulation starting from the rest. The red \triangle shows the initial point of the simulation. The dashed lines represent the discrete jump in system states at the end of each step. The domain transitions are represented by \diamond and blue \circ , respectively.

feedback control torques in simulation are shown in Figure 5.11b. The torque plot demonstrates that the RES-CLF controllers respect the torque limit while still be able to track the outputs exponentially, as shown in Figure 5.10b.

Further, the feedback controllers drive the system to periodic limit cycles even starting from a point that is not on the orbit. This is demonstrated in Figure 5.12, where we show phase portrait plots of four representative joints. As shown in these figures, both uncontrolled states (stance spring) and controlled states (waist pitch, stance knee pitch, and non-stance hip roll) converge to periodic limit cycles under the feedback controllers. In particular, the waist pitch and stance knee pitch joints exhibit nice and smooth transitions from the double-support to single support domain because they are determined by the same set of virtual constraints. On the other hand, the non-stance hip roll joint is determined by the holonomic constraints during the double-support domain but is determined by virtual constraints during the single support domain. As a result, there is discrete changes in the acceleration of these non-stance leg joints, even though the angles and velocities are continuous.

It is important to note that there is no NLP constraint that guarantees the stability of the periodic orbit in the current optimization framework. In fact, incorporating the stability constraint of periodic walking gaits to the direct collocation optimization remains computationally challenging. Therefore, the stability of a walking gait is evaluated *a posteriori* through checking whether the maximum magnitude of eigenvalues of the Jacobian of the Poincaré return map of the closed loop hybrid system. In particular, we select the guard $S_{ss \rightarrow ds}$ as the Poincaré section. The evaluation of the Poincaré return map of this periodic orbit revealed that the maximum magnitude of the eigenvalues of the Jacobian of the Poincaré return map is 0.24, indicating that the gait is stable.

Experimental setup. The experimental setup of the DURUS 3D walking is shown in Figure 5.13. During the experiments, the robot walked on a large treadmill platform.

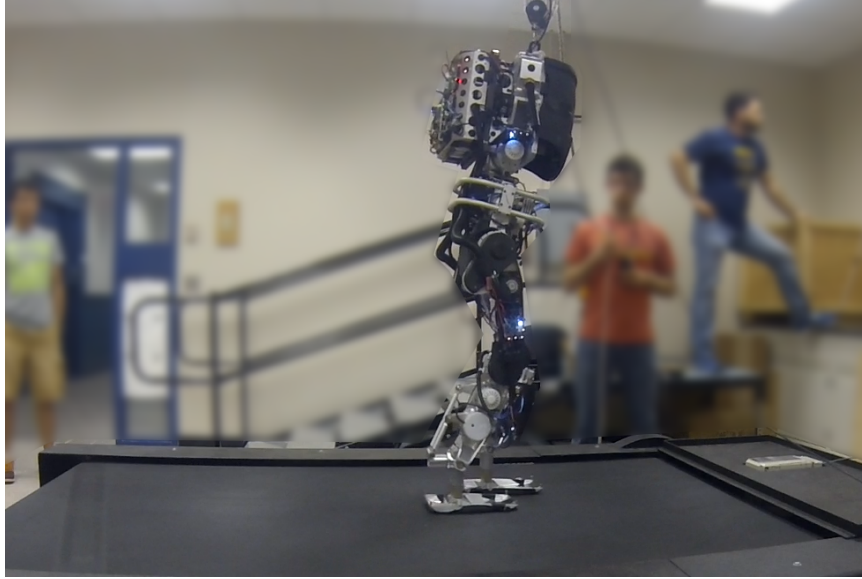


Figure 5.13: The setup of unsupported 3D DURUS walking experiments on treadmill. The treadmill speed is programmed to match the actual walking velocity.

The treadmill speed is programmed to match the actual walking velocity. There is a rope that loosely hangs the torso of the robot from the roof due to the safety concern in the case the robot falls. During the walking experiments, there is enough slack on the rope to guarantee that it will not intervene the robot behaviors.

Sustained walking in experiments. The 3D flat-footed gait walked stably over hours in multiple occasions. The stable walking of DURUS was showcased at the Robot Endurance Test at DARPA Robotic Challenge finals, during which DURUS exhibited sustained walking over large distances with a consistently low cost of transport. Regarding the detail of the experimental implementation of DURUS 3D walking, we refer the readers to [109]. Figure 5.14 shows the periodic phase portraits of each of the actuated joint angles in one of the experiments. Particularly, we show the periodic limit cycles of in term of left/right leg angles instead of stance and non-stance. The comparison of corresponding periodic orbit in simulation is also plotted in the figure, which shows that very close match between experimental and simulation trajectories despite noise signals from the hardware. These periodic limit cycles in

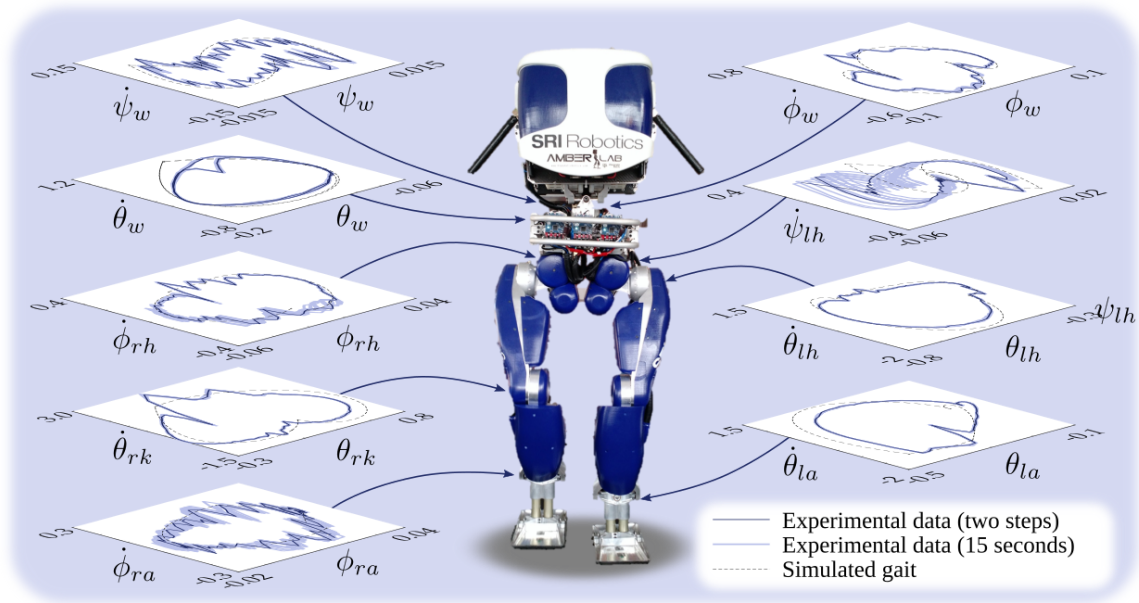


Figure 5.14: Periodic limit cycles of actuated joints in experiment and overlaid on the simulated gait (units: rad and rad/s; symmetric joints omitted for clarity).

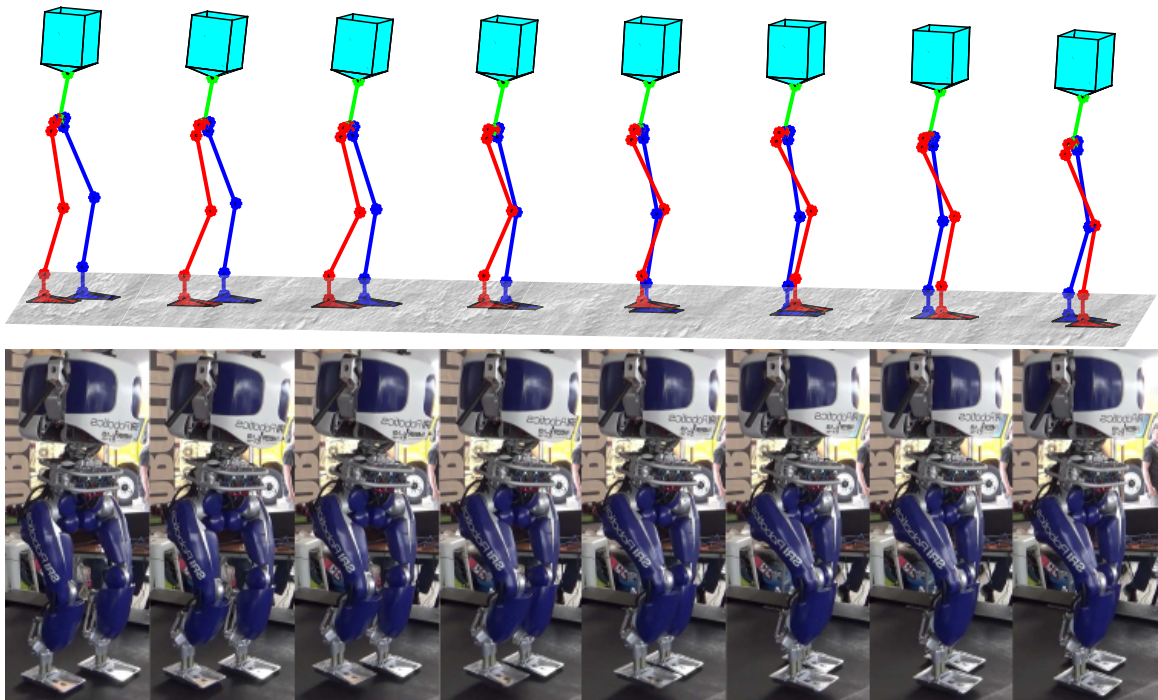


Figure 5.15: Tiled still images from the simulation and experiment of DURUS flat-footed walking in 3D at $0.3m/s$.

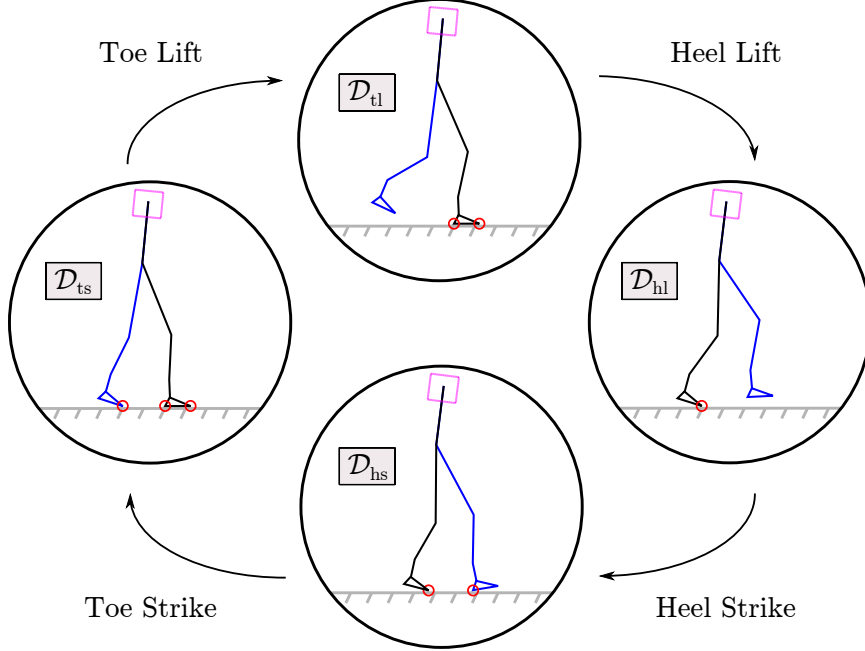


Figure 5.16: Domains graph of 3D multi-contact walking. The red circles represent foot contact points.

turn represent a cycle behavior of the robot, i.e., periodic walking gaits. We show tiled images of during one gait step of the stable 3D walking in experiment and simulation in Figure 5.15. This successful stable walking in 3D is a strong indication of the practicality of the presented optimization approach on humanoid robots, even in the presence of compliant elements.

5.1.3 3D Multi-Contact Walking

Different from the flat-footed walking, 3D multi-contact walking allows rotation of the foot about the toe or the heel edges. We no longer constrain feet to remain flat, resulting in heel- or toe-only contacts with the ground. Based on Ames et al. study on the human locomotion data analysis in [12, 117], we develop a four-domain hybrid system model for the 3D multi-contact walking gaits using the same procedure as its counterpart flat-footed walking.

5.1.3.1 Multi-Domain Hybrid System Model

This style of locomotion has four domains: a *toe-strike* domain, \mathcal{D}_{ts} , a *toe-lift* domain, \mathcal{D}_{tl} , a *heel-lift* domain, \mathcal{D}_{hl} , and a *heel-strike* domain, \mathcal{D}_{hs} , as illustrated in Figure 5.16. We assume the walking gait cycle starts from the toe-strike domain, and ends at the heel-strike domain.

Contact constraints. The four-domain hybrid system model of 3D multi-contact walking, can be determined from the domain graph and contact conditions of all domains shown in Figure 5.16. We particularly exclude the cases in which the robot tilt over the side edge of the foot. In other words, we assume that the heel- or toe-only contact only allows the rotation along the y -axis, rotations along other two axes are constrained by the contact. Consequently, the holonomic constraints for each domain are defined as:

$$h_{ts}(q) := (p_{st}, \phi_{st}, p_{nst}, \phi_{nst}^x, \phi_{nst}^z) \in \mathbb{R}^{11}, \quad (5.29)$$

$$h_{tl}(q) := (p_{st}, \phi_{st}) \in \mathbb{R}^6, \quad (5.30)$$

$$h_{hl}(q) := (p_{st}, \phi_{st}^x, \phi_{st}^z) \in \mathbb{R}^5, \quad (5.31)$$

$$h_{hs}(q) := (p_{st}, \phi_{st}^x, \phi_{st}^z, p_{nsh}, \phi_{nsh}^x, \phi_{nsh}^z) \in \mathbb{R}^{10}, \quad (5.32)$$

where p_{st} , p_{nst} , and p_{nsh} are the position of the stance toe, the non-stance toe, and the non-stance heel, respectively, as depicted in Figure 5.5. Given the origin of the inertia frame, R_0 , depicted in Figure 5.5, the desired position for these contact points are given in term of gait metrics—step length and step width—as

$$\bar{p}_{st} = \begin{bmatrix} l_t \\ -\frac{w_{step}}{2} \\ 0 \end{bmatrix}, \quad \bar{p}_{nst} = \begin{bmatrix} l_t - l_{step} \\ \frac{w_{step}}{2} \\ 0 \end{bmatrix}, \quad \bar{p}_{nsh} = \begin{bmatrix} -l_h - l_{step} \\ \frac{w_{step}}{2} \\ 0 \end{bmatrix} \quad (5.33)$$

Domains. Similar to flat-footed walking model, domains are determined by the limiting conditions on the ground reaction wrenches and additional unilateral constraints

of the robot configuration. We have listed these constraints for the flat-foot contact in (5.11)–(5.15). For heel- or toe-only contacts, the constraint in (5.15) is no longer required. In addition, the unilateral constraints are determined so that toe or heel position of the foot should be above the ground, i.e.,

$$v_{ts}(q) := (p_{nsh}^z), \quad v_{tl}(q) := (p_{nsh}^z, p_{nst}^z), \quad (5.34)$$

$$v_{hl}(q) := (p_{sh}^z, p_{nsh}^z, p_{nst}^z), \quad v_{hs}(q) := (p_{sh}^z, p_{nst}^z). \quad (5.35)$$

Therefore, the admissibility condition $A_v(q, \dot{q}, u)$ for each $v \in \{ts, tl, hl, hs\}$ is determined from the reaction force constraints and unilateral constraints accordingly.

Guards and Reset Maps. It can be noted that, from the graph shown in Figure 5.16, the name of a given domain indicates the previous discrete event that results in the transition to that domain. In other words, the guard conditions are specified as,

$$H_{ts \rightarrow tl}(q, \dot{q}, u) := \lambda_{nst}^{fz}(q, \dot{q}, u), \quad (5.36)$$

$$H_{tl \rightarrow hl}(q, \dot{q}, u) := \lambda_{sh}^{fz}(q, \dot{q}, u), \quad (5.37)$$

$$H_{hl \rightarrow hs}(q, \dot{q}, u) := p_{nsh}^z(q), \quad (5.38)$$

$$H_{hs \rightarrow ts}(q, \dot{q}, u) := p_{nst}^z(q). \quad (5.39)$$

There is no impact or coordinate change involved with the toe-lift and the heel-lift event, therefore, the associated reset map is an identity map for these two transitions. The non-stance heel impacts the ground when the heel-strike event occurs. In addition to the non-stance toe impact, there exists a change of coordinates of switching of the stance and the non-stance foot at the toe-strike event.

With the definition of individual elements, the four-domain hybrid control system model $\mathcal{H}\mathcal{C}_{mc}$ for the 3D multi-contact walking behavior can be modeled as in (2.1).

5.1.3.2 Virtual Constraints

For domains \mathcal{D}_{ts} and \mathcal{D}_{tl} , we define the same velocity-modulating output as stated in (5.20). Considering the fact that the non-flat stance foot will make it difficult

(perhaps impossible) to *directly* control the forward speed of the system, there is no velocity-modulating output defined on \mathcal{D}_{hl} and \mathcal{D}_{hs} . In addition, we determine the position-modulating outputs of each domain in terms of the following output indexing sets:

$$\mathcal{O}_{ts} = \{\text{skp, stp, sar, str, shy, wr, wp, wy, nskp}\}, \quad (5.40)$$

$$\mathcal{O}_{tl} = \{\text{skp, stp, sar, str, shy, wr, wp, wy, nskp, nsl, nsap, nslr, nsfr, nsfy}\}, \quad (5.41)$$

$$\mathcal{O}_{hl} = \{\text{skp, stp, sar, str, shy, wr, wp, wy, nskp, sap, nsl, nsap, nslr, nsfr, nsfy}\}, \quad (5.42)$$

$$\mathcal{O}_{hs} = \{\text{skp, stp, sar, str, shy, wr, wp, wy, nskp, sap, nsap}\}, \quad (5.43)$$

where each position-modulating output is chosen from the output library given in Table 5.1. The rule how to select these outputs, we refer to the definition of position-modulating outputs in Definition 2.5.

5.1.3.3 Gait Optimization

Similar to the flat-footed walking gait optimization, we incorporate a few additional user preference constraints into the direct HZD optimization problem. In addition to the torso movement restriction and swing leg collision avoidance constraints, we also consider the following gait constraints:

Impact Velocity. It became apparent through testing that if the swing foot impacts the ground too hard, it can destabilize the robot's balance. Therefore, we constrain the impact velocities of the heel to be within a reasonable range. Let $v_x^{\max}, v_y^{\max}, v_z^{\max} > 0$ be the maximum allowable impact velocities in x , y , and z direction respectively, then the swing heel velocities $\dot{h}_{nsh}(q^-, \dot{q}^-)$ should satisfy

$$|\dot{h}_{nsh}^x(q^-, \dot{q}^-)| \leq v_x^{\max}, \quad |\dot{h}_{nsh}^y(q^-, \dot{q}^-)| \leq v_y^{\max}, \quad |\dot{h}_{nsh}^z(q^-, \dot{q}^-)| \leq v_z^{\max}, \quad (5.44)$$

where $(q^-, \dot{q}^-) \in \mathcal{D}_{hl} \cap S_{hl}$.

Ground Reaction Wrench Constraints. The ground reaction wrenches resulting from the contact conditions cannot be infinitely large. The limitations of ground reaction

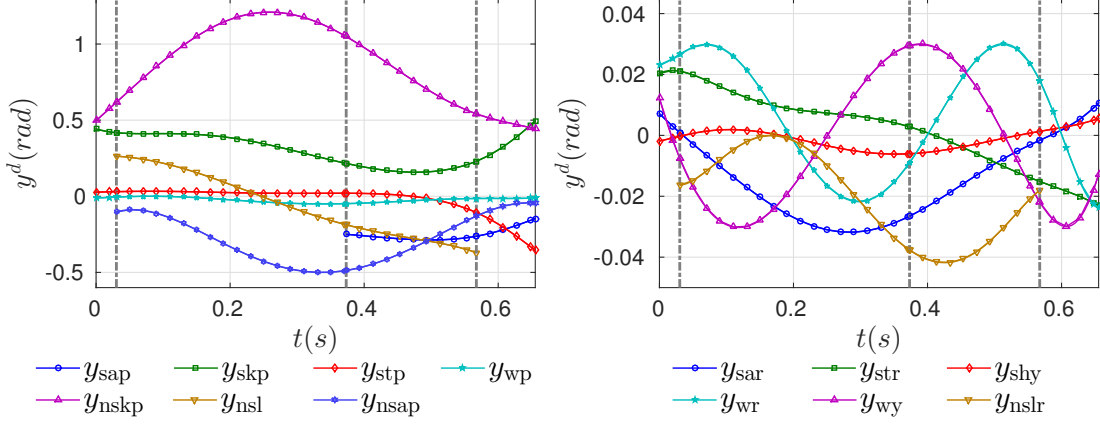


Figure 5.17: Desired outputs of the 3D multi-contact walking gait. The vertical dotted lines represent the moment of domain transition.

wrenches are often described as the Zero Moment Point (ZMP) constraints, which are discussed thoroughly in [58]. In particular, we only enforce the ZMP constraints only during the single support domain \mathcal{D}_{tl} when the stance foot is flat on the ground. In addition, we also constrain the yaw reaction moment of the stance foot, λ_{sf}^{mz} , being reasonably small:

$$\|\lambda_{sf}^{mz}\| \leq \lambda^{\max} \quad (5.45)$$

where λ^{\max} is the maximum acceptable yaw reaction moment.

The 3D multi-contact walking gait optimization problem is formulated based on the hybrid system model and virtual constraints of the walking gait according to the standard procedure as the 3D flat-footed walking. Specifically, the number of cardinal nodes are picked as 10, 15, 20, and 12 for the toe-strike, toe-lift, heel-strike, and toe-strike domain, respectively. With the same goal of minimizing the energy cost of the gait, We use the same cost function as the flat-footed walking case given in (5.23). Because the 3D multi-contact walking gait consists of four domains, the wall time of the optimization is often twice as much as the flat-footed gait optimization.

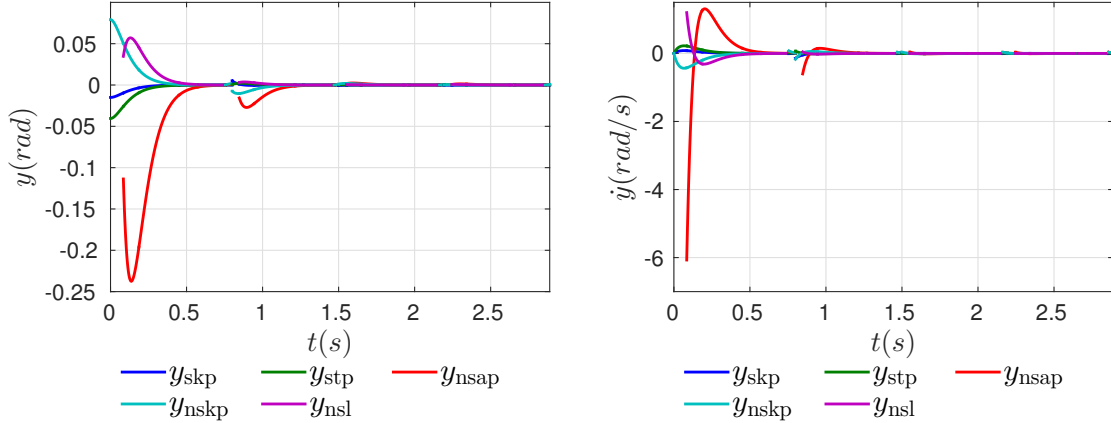


Figure 5.18: Convergence of outputs y_2 and \dot{y}_2 in 4 steps with input-output linearization controllers when simulating the multi-contact walking gait from rest. We only show errors of sagittal plane outputs because these outputs have larger errors than others.

5.1.3.4 Simulation and Experimental Results

The end result of the gait optimization is a stable multi-contact dynamic walking gait that can be implemented on DURUS. Different from the 3D flat-footed walking gait, the feet are no longer required to must be flat when on the ground or in the air for the 3D multi-contact walking.

Human-like walking. The most apparent benefit of the multi-contact walking is the longer step length than its counterpart flat-footed gaits, exhibiting more aesthetically human-like behaviors, likely because the feet can stretch further out in front by landing on their heels. As a result, the step length of this multi-contact walking gait is 0.4 m , which is almost three times larger than the flat-footed walking gait. The

Table 5.2: A comparison of domain durations between human locomotion data versus the multi-contact walking gait on DURUS through optimization.

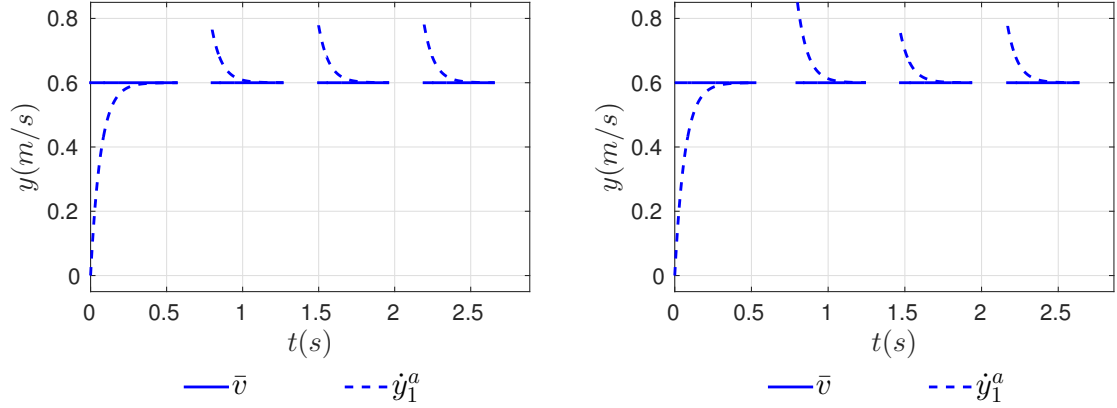
Domain	Human	DURUS
\mathcal{D}_{ts}	6%	4.6%
\mathcal{D}_{tl}	59%	52.4%
\mathcal{D}_{hl}	18%	29.6%
\mathcal{D}_{hs}	17%	13.4%

robot walks twice as fast as the flat-footed walking with the multi-contact behavior with a desired velocity of 0.6 m/s . In addition, the elapsed time of each gait step is 0.6559 seconds. In Table 5.2, we show that the duration percentages of each discrete domain of the optimal gait match very closely to that of the human walking data [12], despite forcing no explicit human-like domain interval constraints. That is, we recover human-like behavior without explicit human reference as a consequence of the natural dynamics of the robot.

Simulation results. Similarly, we run the DURUS 3D multi-contact walking gait simulation starting from a disturbed rest position. As shown in Figure 5.18, the outputs y_2 and \dot{y}_2 decreased to zero under the input-output linearization feedback controllers, and the maximum error after impact reduced to less than 2×10^{-3} after two steps. The torques applied in this simulation is shown in Figure 5.20a, which obviously exceeds the maximum torque limits. On the other hand, the QP based RES-CLF controllers reduces the maximum torques within the limits, as shown in Figure 5.20b. More importantly, both feedback controllers demonstrate exponential convergence of virtual constraints, as depicted in Figure 5.19 and Figure 5.19b respectively. These plots of actual and desired (linearized) hip velocity also indicate that the velocity of the robot is zero at the starting moment.

The convergence to periodic limit cycles is illustrated by phase portrait plots of several robot joints, including un-actuated and actuated joints, in Figure 5.21. Despite these limit cycles are totally different than that of the flat-footed walking gait simulation, they all have a common feature: exponential convergence to periodic limit cycles under the feedback controllers. The maximum magnitude of eigenvalues of the Jacobian of the Poincaré return map of this gait is 0.21, which demonstrates the stability of the walking gait.

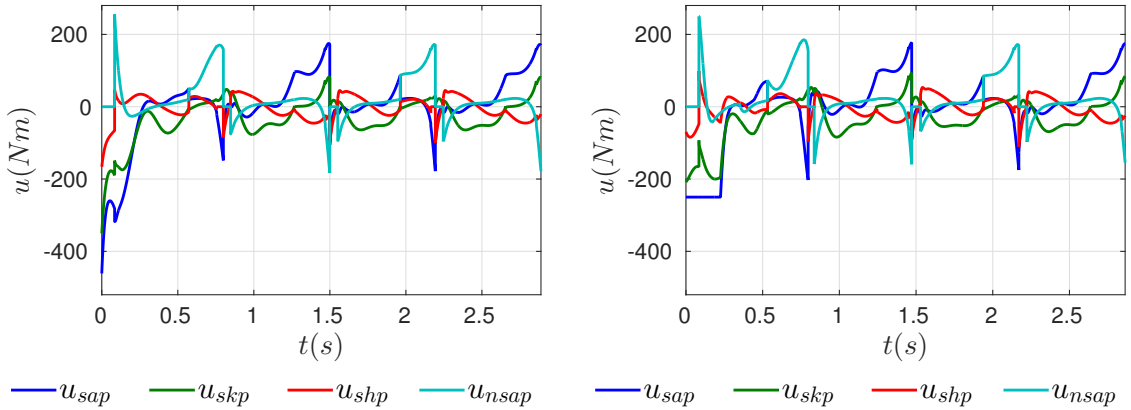
Sustained walking in experiments. In the context of bipedal walking, a stable limit cycle implies stable walking. The limit cycles achieved experimentally on



(a) Input-output linearization controllers

(b) QP based CLF controllers

Figure 5.19: Tracking of the velocity-modulating output using both IO linearization controllers and QP based CLF controllers in multi-contact walking simulation.



(a) Input-output linearization controllers

(b) QP based CLF controllers

Figure 5.20: Joint torque profiles when using two different feedback control laws in DURUS multi-contact walking simulation. Here we show torques of stance ankle pitch, u_{sap} , stance knee pitch, u_{skp} , stance hip pitch, u_{shp} , and non-stance ankle pitch, u_{nsap} , considering that these joints often require larger torques than others.

DURUS and shown in Figure 5.22 exhibit a closed behavior, indicating that the multi-contact walking behaviors are stable in both the saggital and coronal planes. It is clear from the hip roll limit cycle that this is the joint of primary deviation from theory through the implementation of feedback regulators. Figure 5.23 shows the synchronized tiled images of the multi-contact walking gait in simulation and experiment. From these images, we can see the heel and toe contact during the gaits, resulting in more dynamic and human-like behaviors. A video illustrating the experimental

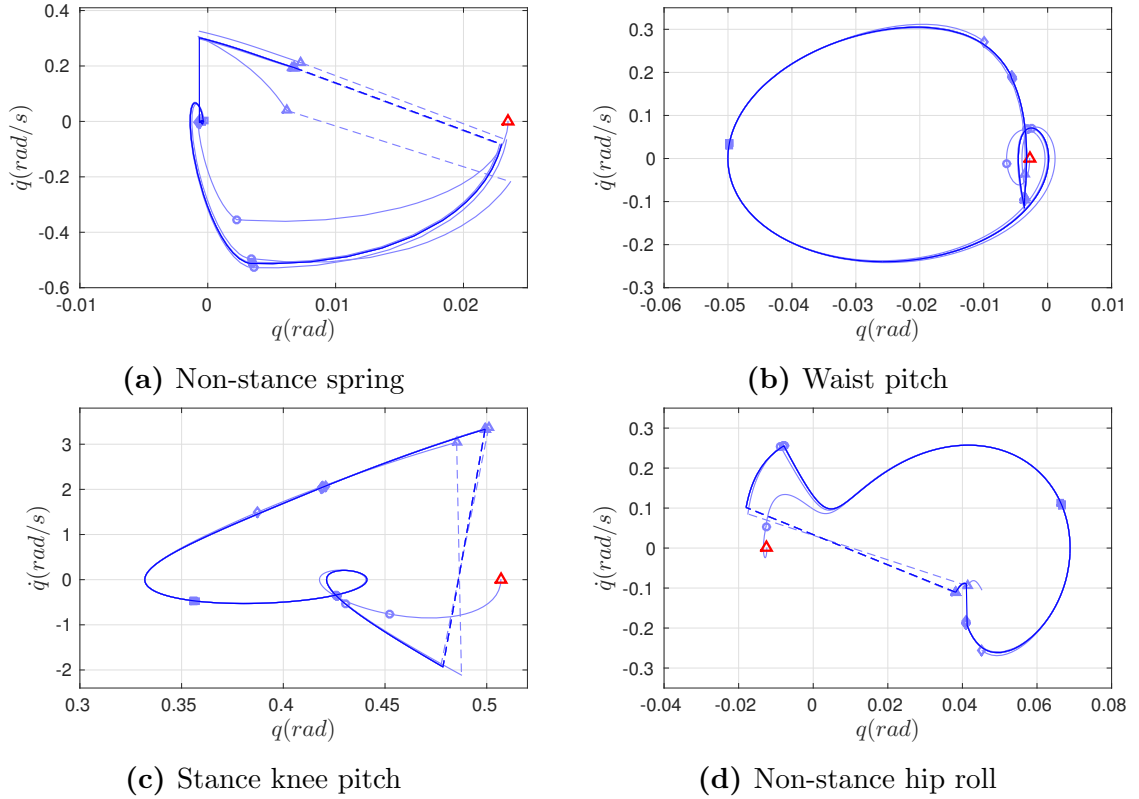


Figure 5.21: Phase portraits of representative joints in the 3D multi-contact walking simulation starting from the rest. The red \triangle shows the initial point of the simulation. The dashed lines represent the discrete jump in system states at the end of each step. The domain transitions are illustrated with \circ , \square , \diamond , and \triangle , respectively.

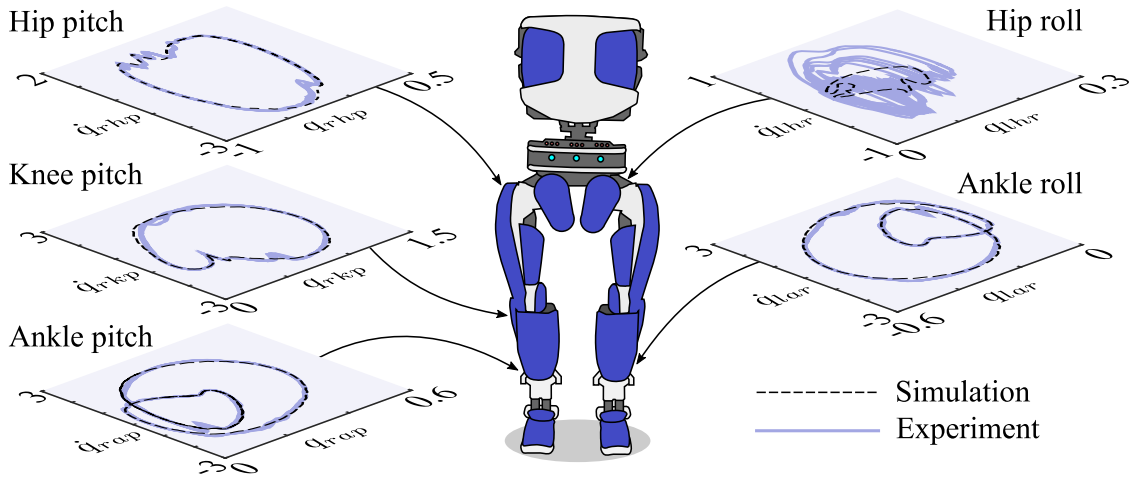


Figure 5.22: Experimental results of multi-contact walking with DURUS in 3D, showing phase plots of some representative joints in both in simulation and experiment (units: rad and rad/s).

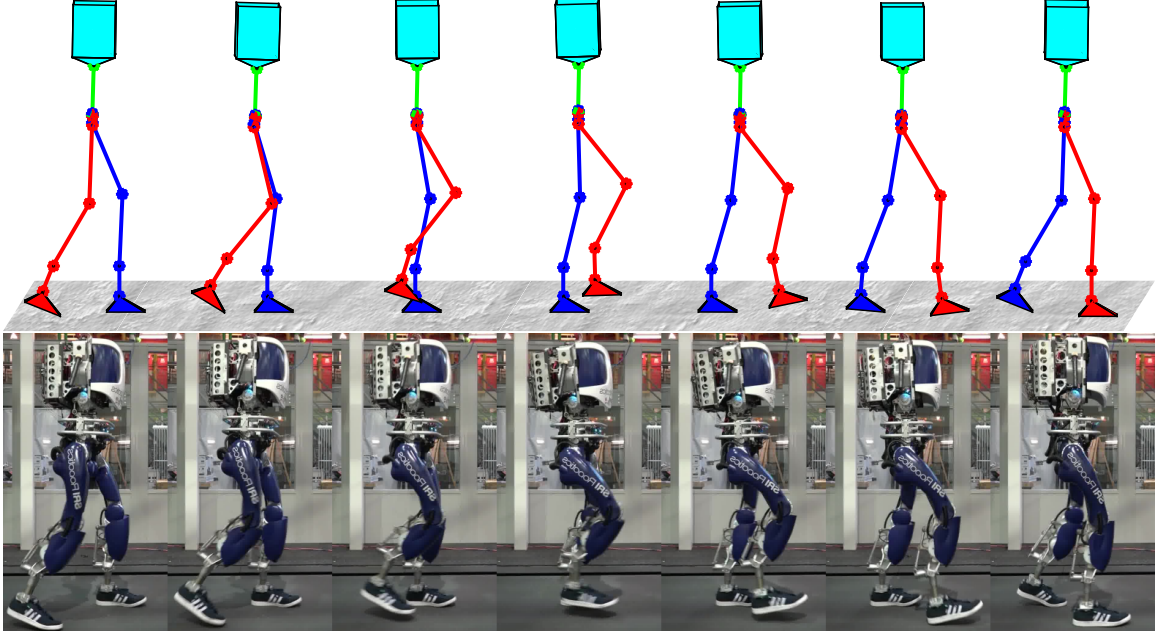


Figure 5.23: Tiled still images from the simulation and experiment of DURUS multi-contact walking in 3D at 0.6 m/s .

results of the 3D multi-contact walking gait can be found in [2].

5.1.4 3D Walking Efficiency

The efficiency of the walking gait is demonstrated by the specific cost of electrical transport based on all power consumed, including the electrical power of central control computer, motor drivers and controllers, motors and sensors. The specific cost of electrical transport c_{et} was calculated as in [36], where the total energy consumed over the weight and distance traveled is represented for step i as:

$$c_{et,i} = \frac{1}{mgd_i} \int_{t_i^+}^{t_i^-} \left(P_{el} + \sum_{j=1}^{15} I_j(t)V_j(t) \right) dt, \quad (5.46)$$

where $P_{el} = 86.4\text{W}$ is logic power consumed by the onboard computer and motor controllers, d_i is the distance traveled through the i^{th} step, and $I_j(t)$ and $V_j(t)$ are the currents and voltage recorded for the j^{th} motor. We compute the electrical COT numbers of each step for both flat-footed and multi-contact walking experiments of DURUS robot. In particular, Figure 5.24 shows measurements of the electrical cost

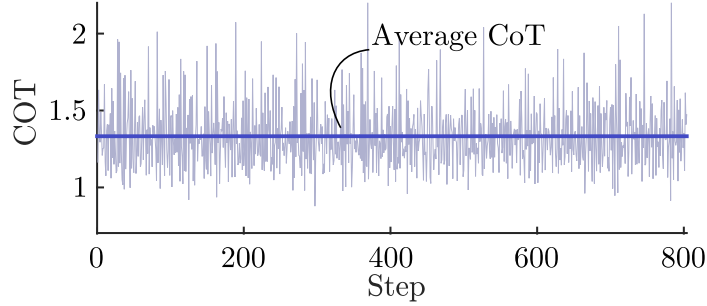


Figure 5.24: Electrical cost of transport (COT) for a 3D flat-footed walking experiment over 800 steps with a mean cost of transport of 1.33.

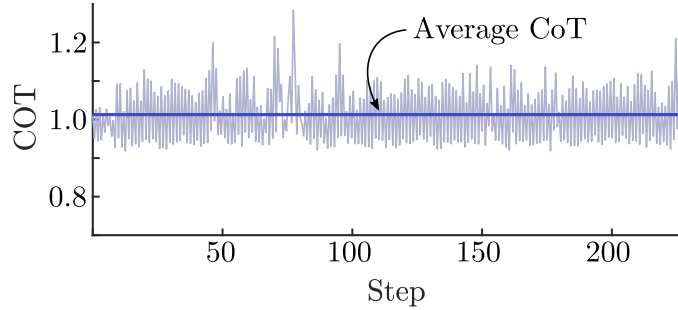


Figure 5.25: Electrical cost of transport (COT) for a 3D multi-contact walking experiment over 200 steps with a mean cost of transport of 1.02.

of transport of 3D flat-footed walking gait over 800 sustainable steps, and Figure 5.25 shows measurements of 3D multi-contact walking gait over 200 steps. The average transport cost of DURUS when walking with flat-foot was measured to be 1.33, which represents a significant improvement in humanoid locomotion economy. Even better, the mean COT for DURUS during steady-state multi-contact locomotion is $\bar{c}_{et} = 1.02$, which is 23% more efficient than experimental results obtained on DURUS for flat-footed walking, which was previously the lowest recorded CoT on a humanoid robot.

The reported electrical cost of transport for several robots is summarized in Table 5.3, from which we observe the robots utilizing passive elements, small motors, or anthropomorphic designs to leverage energy savings demonstrating the lowest energy expenses (Cornell Ranger and Biped). Notably, other non-humanoid bipeds¹ have been built specifically to demonstrate more efficient locomotion [20], yielding

¹Specifically, the Cornell Ranger is pseudo-planarized to avoid frontal-plane tipping.

Table 5.3: Comparison of gait efficiency on various robot platforms

Name	\bar{c}_{et}	m (kg)
Human	0.2	.
ATLAS [20]	5	102
ASIMO [33]	3.23	52
AMBER 1 [140]	1.88	3.3
ATRIAS [110]	1.13	62
DURUS-2D [36]	0.63	31.5
Cornell Biped [33]	0.2	13
Cornell Ranger [20]	0.19	9.9
DURUS (flat-footed)	1.33	79.5
DURUS (multi-contact)	1.02	79.5

transport costs under 0.2. Additionally, robots employing HZD to achieve locomotion exhibit efficient locomotion (AMBER 1 and DURUS-2D), although these are restricted to walking in a 2D plane. The closest efficiency numbers come from ATRIAS—possibly since it inspired the compliment elements in the design of DURUS—yet this robot not humanoid in nature. Therefore, in the category of full-scale bipedal humanoid robots (e.g., ATLAS and ASIMO) the electrical cost of transport on DURUS is the lowest ever reported. By these results, we demonstrate that gait economy can be advanced in more-traditional 3D humanoid forms, at least in part, as a result of our scalable and energy-optimized gait generation.

5.2 DRC-HUBO – Walks with Arm Swing

Humanoid robots are designed with dozens of actuated joints to suit a variety of tasks, but walking controllers rarely make the best use of all of this freedom. Achieving dynamic walking on humanoids is hard; their dynamics are inherently nonlinear and their numerous actuators render their computational search spaces very high-dimensional. As such, it’s often simplest to just ignore their arms when generating their locomotion patterns. This can be a missed opportunity. Beyond the obvious manipulation tasks, arms can be helpful for improving the balance [101] and economy [34] of locomotion by swinging them as part of a dynamic gait. As such, we use

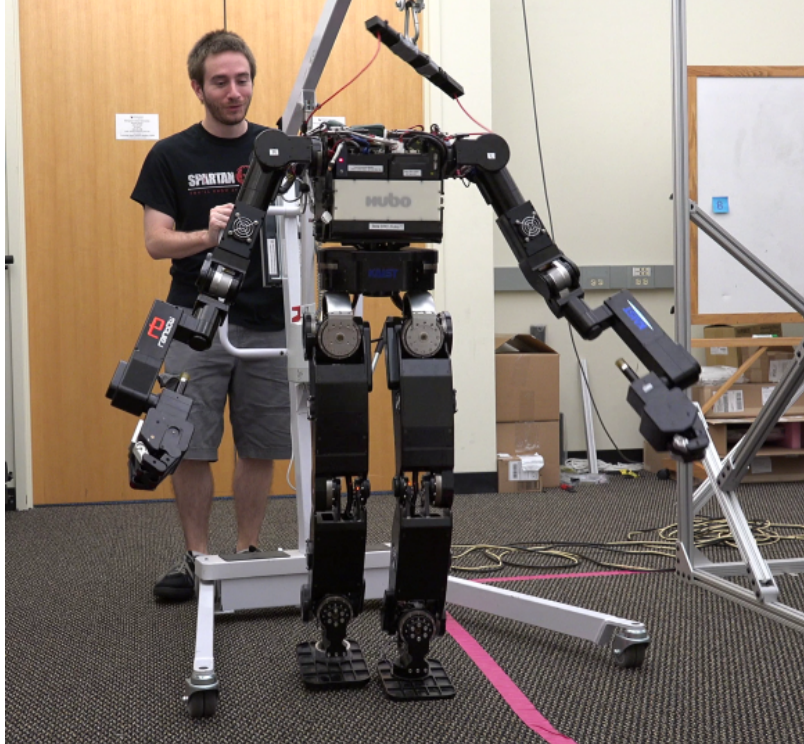


Figure 5.26: DRC-HUBO, a humanoid robotic platform with 27 actuated degrees of freedom from its feet to its wrists.

the direct HZD optimization to fully exploit the internal dynamics of the robot. In this section, we present dynamic and stable walking with dynamic arm swing of the DRC-HUBO, a fully-actuated humanoid with 27 actuators (see Figure 5.26). Importantly, the humanoid swings its arms as a consequence of optimizing the dynamic gait for energy-efficient locomotion subject to no-net-moment constraints, not by *a priori* specification.

We present simulations of two different stable full-body-optimized walking gaits (including arm motions) on a DRC-HUBO model in the DART simulation environment¹. This includes an optimized startup motion to accelerate from rest to the steady periodic gait. We also present a preliminary implementation of one of these walking gaits on the DRC-HUBO robot using open-loop position control.

¹The DART simulation environment is available at <https://github.com/dartsim/dart>.

5.2.1 Hybrid System Model

The DRC-HUBO model has 27 actuated degrees of freedom from its wrists to its feet [100]¹, which are depicted in Figure 5.27. The robot has 6 actuated joints on each leg, 7 actuated joints on each arm and 1 actuated torso yaw joint. Assume that the floating base coordinate, R_b , is attached to the pelvis link of the robot, the generalized coordinates of the robot is given as

$$q = (p_b, \phi_b, q_{lleg}, q_{rleg}, \psi_{tor}, q_{larm}, q_{rarm}) \in \mathcal{Q} \subset \mathbb{R}^{33} \quad (5.47)$$

Given the coordinate system, the equations of motion of the robot is then determined via the classic Euler-Lagrangian equations given in (2.17).

We constrain our continuous domains to only allow flat-footed contact with the ground as to minimize the necessary torques at the foot actuators. We thus model the domains of DRC-HUBO as a sequence of alternating single-support domains, \mathcal{D}_{ss} , as depicted in Figure 5.28. In the later discussion, we assume that the right leg is the stance leg. During the single-support domain, only the stance foot is in contact with the ground. The holonomic constraints of the domain then consists of the Cartesian positions of a point on the stance foot link and the orientations of the stance foot link being zero:

$$h_{ss} = (p_{sf}^x, p_{sf}^y, p_{sf}^z, \phi_{sf}^x, \phi_{sf}^y, \phi_{sf}^z) \in \mathbb{R}^6. \quad (5.48)$$

Besides, the non-stance foot should be above the ground all the time, i.e., $p_{nsf}^z(q) \geq 0$. Combining them yields the domain of admissibility condition $A_{ss}(q, \dot{q}, u) \geq 0$, as discussed in (2.25), where $u \in U \subset \mathbb{R}^{27}$ is the vector of actuator inputs. Therefore,

$$\mathcal{D}_{ss} = \{(q, \dot{q}, u) \in T\mathcal{Q} \times U \mid A_{ss}(q, \dot{q}, u) \geq 0\}. \quad (5.49)$$

The switching surface is determined by the condition in which the non-stance foot

¹Fingers are excluded from dynamic optimization in this study.

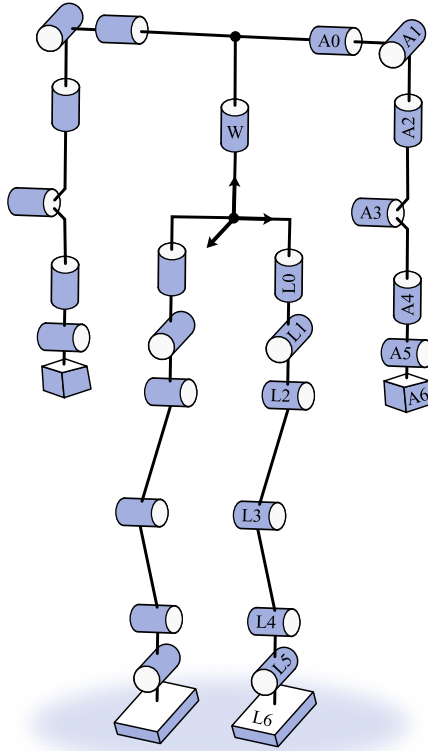


Figure 5.27: Kinematic tree diagram of the DRC-HUBO platform. The floating base frame R_b is located at the pelvis link. The body coordinates of the robot consists of the left leg joints, $q_{leg} = (\psi_{lh}, \phi_{lh}, \theta_{lh}, \theta_{lk}, \theta_{la}, \phi_{la})$, the right leg joints, $q_{rleg} = (\psi_{rh}, \phi_{rh}, \theta_{rh}, \theta_{rk}, \theta_{ra}, \phi_{ra})$, the torso yaw joint, ψ_{tor} , the left arm joints, $q_{larm} = (\theta_{ls}, \phi_{ls}, \psi_{ls}, \theta_{le}, \psi_{lw}, \theta_{lw}, \phi_{lw})$, and the right arm joints, $q_{rarm} = (\theta_{rs}, \phi_{rs}, \psi_{rs}, \theta_{re}, \psi_{rw}, \theta_{rw}, \phi_{rw})$. Here, we use θ to denote pitch joints, ϕ to denote roll joints, and ψ to denote yaw joints respectively.

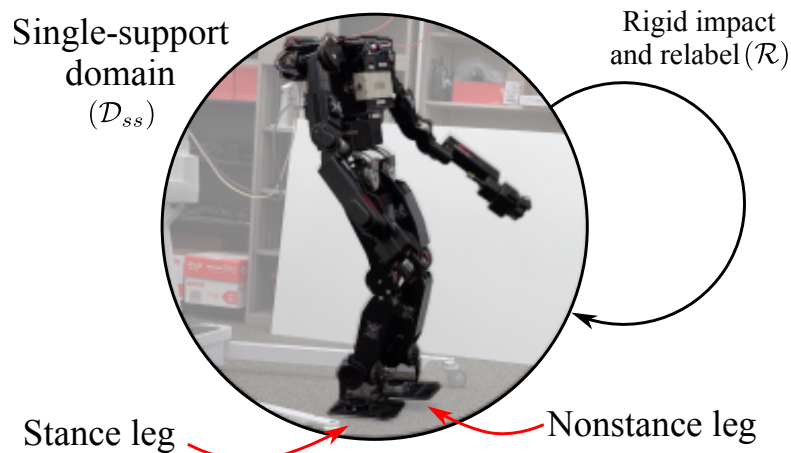


Figure 5.28: Flow of domains for single-support walking with DRC-HUBO.

hits the ground. That is,

$$S_{ss \rightarrow ss} = \{(q, \dot{q}, u) \in T\mathcal{Q} \times U | p_{nsf}^z(q) = 0, \dot{p}_{nsf}^z(q, \dot{q}) \leq 0\}. \quad (5.50)$$

Correspondingly, the reset map $\Delta_{ss \rightarrow ss}$ is determined by the plastic impact equation given in (2.30). Similar to the DURUS robot that we described in Remark 5.1, a relabeling of coordinates takes place at the discrete domain change. With the definition of each element, the single domain hybrid system model for DRC-HUBO flat-footed walking is given as

$$\mathcal{H}\mathcal{C}^H = \{\mathcal{D}_{ss}, S_{ss \rightarrow ss}, \Delta_{ss \rightarrow ss}, FG\}. \quad (5.51)$$

5.2.2 Virtual Constraints

The feedback controller is determined through virtual constraints, as given in (2.39). Due to the fact that DRC-HUBO is fully-actuated, we have one velocity-modulating output, given as

$$y_{1,ss}(q) := \delta p_{\text{hip}}(q) = L_c(-\theta_{ra}) + L_t(-\theta_{ra} - \theta_{rk}), \quad (5.52)$$

where L_c and L_t are lengths of the calf and thigh. In addition, the following position-modulating outputs are defined:

$$y_{2,ss}(q) := (y_{2,\text{nsl}}^a, y_{2,\text{stp}}^a, y_{2,\text{str}}^a, y_{2,\text{nslr}}^a, y_{2,\text{nsfr}}^a, y_{2,\text{nsfp}}^a, y_{2,\text{nsfy}}^a, \psi_{rh}, \theta_{rk}, \phi_{ra}, \theta_{lk}, \psi_{tor}, \theta_{rs}, \phi_{rs}, \psi_{rs}, \theta_{re}, \psi_{rw}, \theta_{rw}, \phi_{rw}, \theta_{ls}, \phi_{ls}, \psi_{ls}, \theta_{le}, \psi_{lw}, \theta_{lw}, \phi_{lw}) \in \mathbb{R}^{26}, \quad (5.53)$$

where

- non-stance slope:

$$y_{2,\text{nsl}}^a = -\theta_{ra} - \theta_{rk} - \theta_{rh} + \frac{L_c}{L_c + L_t} \theta_{lk} + \theta_{lh},$$

- stance torso pitch: $y_{2,\text{stp}}^a = -\theta_{ra} - \theta_{rk} - \theta_{rh}$,
- stance torso roll: $y_{2,\text{str}}^a = -\phi_{ra} - \phi_{rh}$,
- non-stance leg roll: $y_{2,\text{nslr}}^a = \phi_{rh} - \phi_{lh}$,
- non-stance foot roll: $y_{2,\text{nsfr}}^a = p_{\text{nsfl}}^z(q) - p_{\text{nsfo}}^z(q)$,
- non-stance foot pitch: $y_{2,\text{nsfp}}^a = p_{\text{nst}}^z(q) - p_{\text{nsh}}^z(q)$,
- non-stance foot yaw: $y_{2,\text{nsfy}}^a = p_{\text{nst}}^y(q) - p_{\text{nsh}}^y(q)$.

and other outputs are directly defined as joint angles. Specifically, the non-stance foot roll, pitch and yaw outputs are required to be zero, so that the non-stance foot is flat. In particular, the desired position-modulating outputs are given in terms of 4th order Bézier polynomials.

5.2.3 Optimization and Task Definitions

With the definition of the hybrid system model and the virtual constraints, We use direct HZD optimization framework to generate dynamic locomotion that reasons about the multibody dynamics of all degrees of freedom of DRC-HUBO, including its swinging arms, without restricting motions to a planning template. In addition to the base-level constraints defined in Section 4.3, we also enforced a number of task-specific constraints in the optimization. For one, we ensure there is no net moment about the foot (note: we do not explicitly compute a zero moment point or enforce LIPM, see [58]). We impose impact velocity constraints of the nonstance foot, which limits potential damage to DRC-HUBO’s drive mechanisms, as well as limits on joint rotation, velocity, and acceleration. Further, the feet must be separated by a minimum horizontal distance, and the nonstance foot height must exceed a clearance function above the ground.

We hypothesized that optimizing the robotic gait for energy economy would encourage DRC-HUBO to swing its arms. As such, we chose an objective to minimize

the mechanical cost of transport over the course of the stride cycle:

$$\mathbf{L}(\bar{h}, \dot{q}^{(i)}, u^{(i)}) := \frac{\|P(u^i, \dot{q}^i)\|}{mg l_{step}(\bar{h})}, \quad (5.54)$$

where mg is the robot weight, $l_{step}(\bar{h})$ is the distance traveled during a gait which could be determined from the desired holonomic constraints, and $P(u^i, \dot{q}^i)$ is the total power consumed assuming no power-regeneration (see [77]) computed at each node. The cost function of the gait generating NLP is then can constructed as described in Section 4.3.3. The optimization problem was formulated with 20 cardinal nodes.

In this study, we use our direct collocation optimization to generate outputs for two types of motions: startup and periodic locomotion. We first optimize an efficient periodic gait subject to all listed constraints and objectives. Secondly, we optimize a startup motion, which requires all initial generalized coordinate velocities (\dot{q}_0) be set to zero, with final positions and velocities (q_f, \dot{q}_f) equal to our periodic gait.

We also tested this optimization method by generating two different periodic gaits. The first was allowed more liberal constraints on arm-joint velocities and effective foot size as well as slower stepping frequencies¹. The result of this optimization (**Gait A**) was a very natural-looking, counter-rotating arm swing, but was less likely to be successful on hardware. The second result (**Gait B**) was restricted by tighter constraints on arm-joint velocities, smaller effective foot size, and higher step frequencies. In addition to simulation, **Gait B** is presented with preliminary hardware results. Each of these gaits were solved with IPOPT (using the linear solver `ma57`) in approximately 7-10 minutes on a laptop computer (Intel Core i7-3820QM processor, 2.7 GHz, with 12 GB of RAM).

¹In open-loop position control, dynamic gaits tend to be more robust with higher stepping frequencies than typical humanoid gaits.

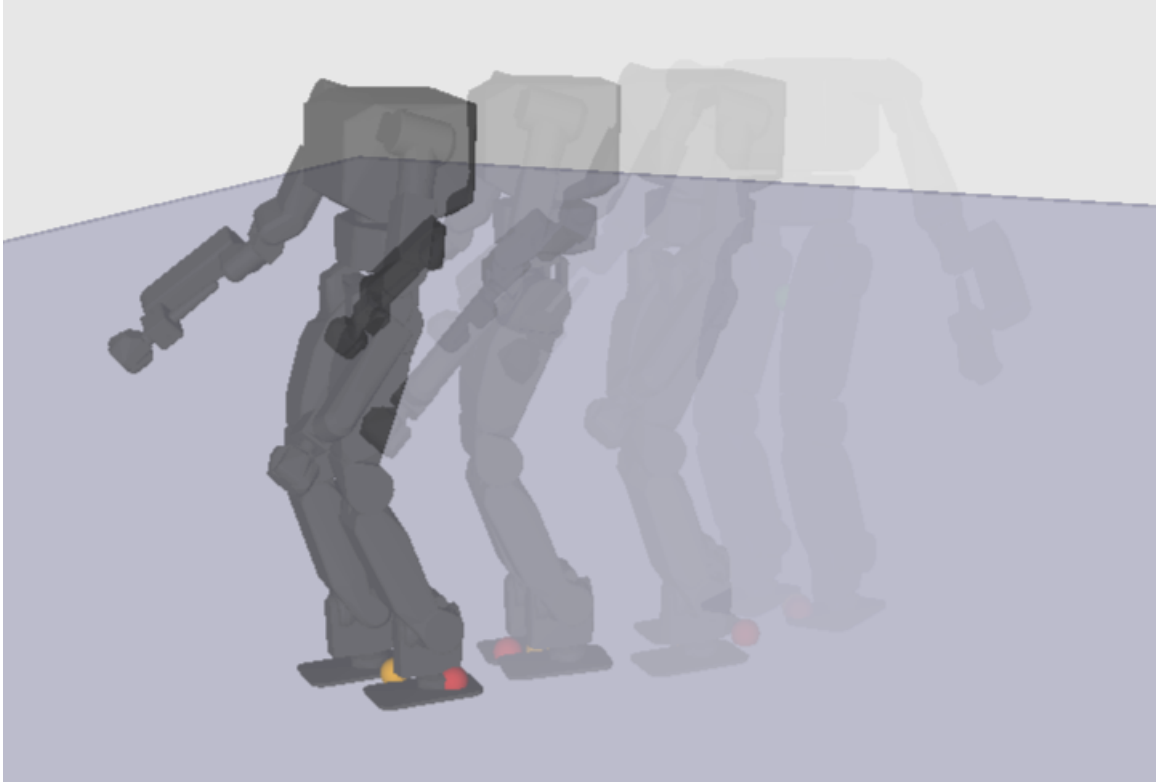


Figure 5.29: A stroboscopic image of full-body-optimized stable walking in simulation (DART environment). The robot starts at rest, executes an optimized startup motion, and runs a pre-scripted open-loop trajectory representing optimized PHZD outputs for ten steps. Outputs are computed by inverse kinematics methods, which solve at an average rate of 5kHz.

5.2.4 Simulation Results

We tested both **Gait A** and **Gait B** in the DART simulation environment. DART is a wholly different dynamics engine than the equations of motion seen by the optimizer to generate stable and dynamic gaits (for instance, DART uses LCP-based contact solving, while the optimizer sees an explicit formulation of the holonomic constraints). This makes for a good independent check that the optimizer is in fact generating stable walking controllers. After executing the optimized startup sequence, the robot was commanded to walk ten steps, playing back the trajectory generated by the gait optimization problem.

Both **Gait A** and **Gait B** resulted in stable, dynamic walking in the DART

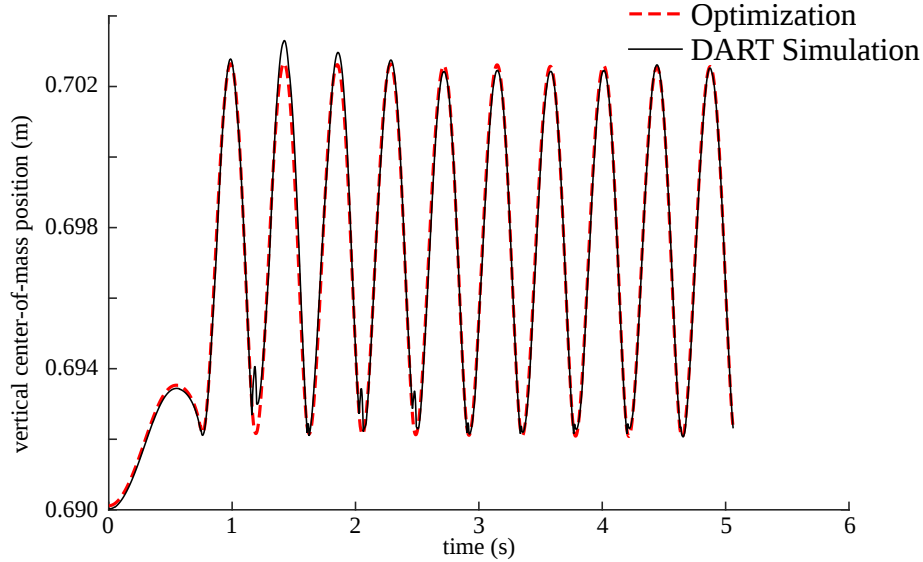


Figure 5.30: Center-of-mass position over time for the simulated gait, compared to the raw kinematic trajectory from the optimization. This pendulum-like bouncing while walking demonstrates that the resulting gait is not restricted to a constant CoM height, as per LIPM-based techniques.

simulator. Figure 5.29 shows a stroboscopic image of **Gait B** walking in Cartesian space, while Figure 5.31 shows a tiled comparison of the optimization dynamics and the DART simulation for both gaits. The exhibited gaits showed a bouncing center-of-mass behavior, not restricted to constant CoM heights as demanded by many LIPM-based methods, as plotted in Figure 5.30 for **Gait B**. Further, the arms in **Gait A** exhibit a natural-looking counter-rotating swing.

5.2.5 Preliminary Hardware Experiments

As a test of the framework’s ability to produce stable control, we use only DRC-HUBO’s encoders for position control feedback, and no inertial measurement. We fed the open-loop position trajectory from **Gait B** to DRC-HUBO. We noticed that the robot had a significant propensity to lean forward and fall during the beginning static position, which we determined was likely due to modeling errors. As such, we used the gait optimization to adjust the gait intuitively, asking it to move the feet forward by 4cm to tune its balance. With this adjustment made, DRC-HUBO would start

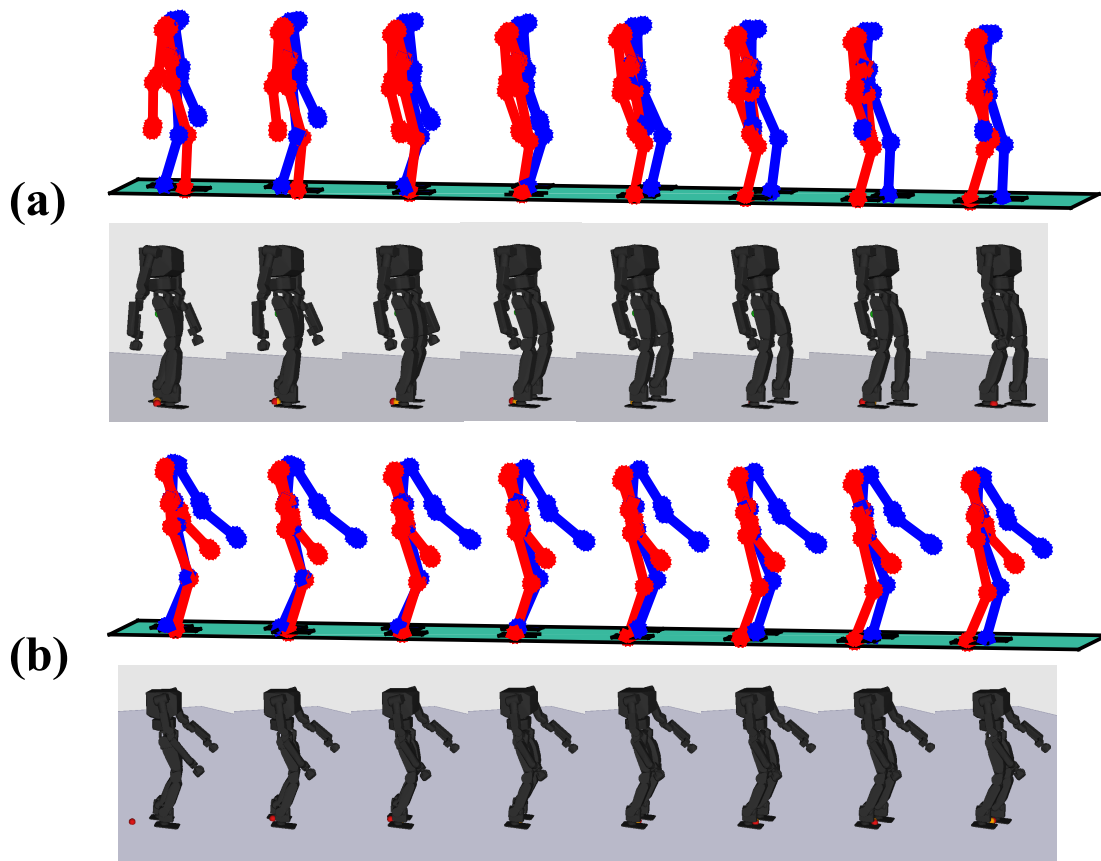


Figure 5.31: A walking tiles figure comparing the optimized gait as simulated by the equations of motion used by the optimizer (top) to the open-loop walking controller simulated in DART (bottom) using optimized gaits after the startup procedure has finished: **(a) Gait A** and **(b) Gait B**. Both gaits were stable using open-loop position control and **Gait A** resulted in a solution with natural-looking counter-rotating arm swing.

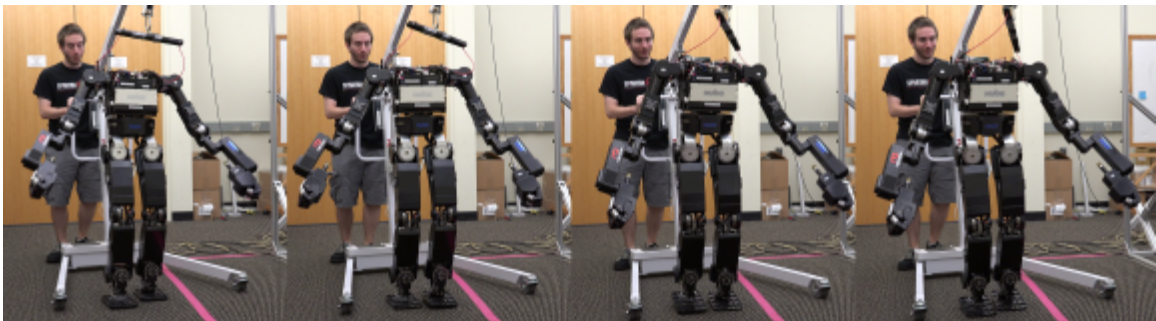


Figure 5.32: Tiled images of a preliminary dynamic walking experiment with DRC-HUBO with an optimized open-loop controller. After a dynamic startup sequence, DRC-HUBO walked for nine steps before modeling errors caused a forward fall.

from rest, execute the startup procedure, and walk nine dynamic steps, open-loop, before falling forward. Figure 5.32 shows tiled images of the experimental gait.

This preliminary experiment aims to show that these generated dynamic motions are reasonable to implement on a real humanoid robot, which are subject to physical limitations like motor saturation limits. Clearly, while stable in simulation, the result is not very robust to the modeling errors present. **Gait A**, which has more exaggerated arm swing, was also implemented on DRC-HUBO but had difficulty performing the startup maneuver without falling, despite its stability in the independent simulator. We suspect this failure is also likely due to inaccurate modeling of the link inertias, which are more drastically accelerated in **Gait A**. Future work will incorporate feedback control in order to compensate for modeling errors.

5.3 PROXI – Online Optimal Gait Generation

This section presents an application of the pseudospectral HZD gait optimization, and—for the first time—experimentally realizes online HZD gait generation for a planar underactuated robot. We show that by taking advantage of the direct transcription formulation and the exponential convergence of the pseudospectral method, the reliability of the HZD gait optimization and the convergence speed are significantly increased. As a result, generating of HZD gaits online becomes feasible with an average computation time less than 0.5 seconds, as will be demonstrated experimentally on a bipedal robot.

5.3.1 Online Motion Planning

The goal of bipedal robots is to demonstrate dynamic and agile locomotion that allows for navigation of terrain not approachable by wheeled robots. Yet the ability to accommodate changes in terrain present in uncontrolled environments, however, is a challenging problem. The difficulty arises from the fact that planning dynamic motions that are consistent to the full body dynamics of the complex robot model is

often computationally expensive.

Existing methods of online motion planning typically use reduced-order models, such as the linear inverted pendulum model (LIPM) [80], to mitigate the complexity of full body dynamics. By balancing the robot about the Zero Moment Point (ZMP) [131], these approaches plan trajectories for the simplified model and then generate the whole body motion by conforming the robot to these analytically tractable dynamics [79, 100]. These simplifications, however, place stringent requirements on the design of the robot (e.g. all joints must be actuated with no significant compliance) and restrictions on the overall locomotion capabilities of the machine (e.g. the robot must always walk with a constant center-of-mass height). Embracing planning and formal control that exploits the full body dynamics of the robot is a path toward unlocking the fully dynamic capabilities of the machine. An increasing number of methods have been developed to generate optimal gaits using full-body dynamics optimization [103, 92]. Some researchers also explore a middle path, in which whole body motion is planned with the robot’s centroidal dynamics subject to full body kinematic constraints [37]. While these methods can realize more dynamic behaviors, the optimization requires excessive amount of time to run and may not be able to converge reliably, and therefore, are only suitable for off-line *a priori* motion planning.

In the hybrid zero dynamics (HZD) framework, existing methods for generating hybrid invariant gaits are often time-consuming, and thus must be performed off-line. While advanced control methods [10, 61] have been developed to robustify the gait under perturbations and switch between gaits (often encoded as *motion primitives*) [104], these methods are restricted to a limited number of gaits generated in advance.

Exploiting the advantages of direct transcription formulation of the pseudospectral method, we show that this approach makes the gait optimization amendable to being solved in a fast and reliable fashion utilizing existing NLP solvers. More importantly, the proposed approach opens the possibility of generating optimal gaits online

while considering the whole body dynamics of the robot. We experimentally evaluate the performance of this method on a planar underactuated robot which walks on a treadmill with varying speeds. As a result, the optimizer successfully generates gaits with different walking speeds online to enable the robot to adjust to the changing speed of the treadmill—this provides the first example of online HZD gait generation.

5.3.2 Hybrid System Model

We will consider a 5-link underactuated planar robot—PROXI—in this case. As shown in Figure 5.33a, (p_x, p_z, q_{sf}) represents the Cartesian position of stance foot and the angle of stance calf with respect to fixed inertia frame respectively, and $(q_{sk}, q_{sh}, q_{nsh}, q_{nsk})$ are the four actuated joints: stance knee, stance hip, non-stance hip and non-stance knee, respectively. The floating base coordinates of the planar robot then can be given as,

$$q = (p_x, p_z, q_{sf}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk}) \in \mathcal{Q} \subset \mathbb{R}^7. \quad (5.55)$$

Because there is no compliant joint in the system, the walking of PROXI only has one domain. During the continuous event (swinging of the non-stance leg), the normal ground reaction forces constrain the foot to the ground, therefore, the holonomic constraints for this single domain \mathcal{D} is

$$h(q) = (p_x, p_z) \equiv \text{constant}. \quad (5.56)$$

The ground reaction forces associated with the holonomic constraints must satisfy the friction cone constraints and the normal force must be positive. In addition, the non-stance foot must be above the ground throughout a step, i.e., the vertical non-stance foot position, p_{nsf}^z , must be greater than or equal to zero. These unilateral constraints form the domain admissibility condition given as

$$A(q, \dot{q}, u) = \begin{bmatrix} \mathcal{R}F(q, \dot{q}, u) \\ p_{nsf}^z(q) \end{bmatrix} \geq 0, \quad (5.57)$$

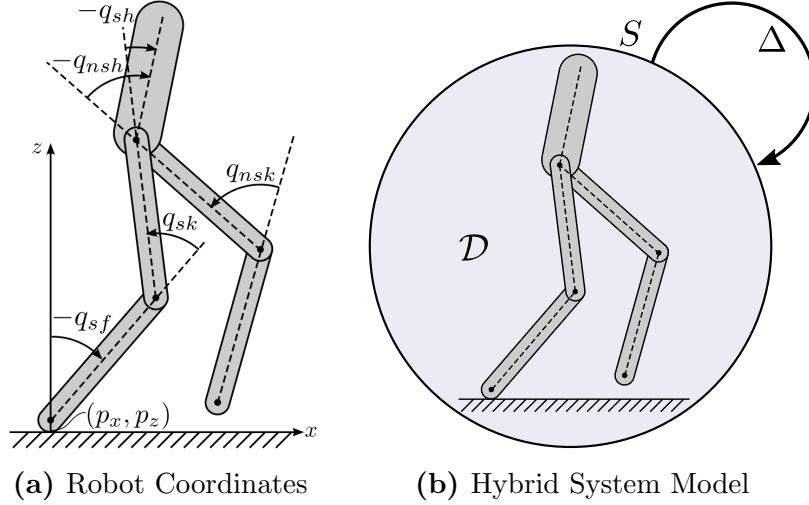


Figure 5.33: Robot model and hybrid system structure of the 5-link underactuated planar robot–RPOXI.

where $u \in U \subset \mathbb{R}^4$ is the vector of control inputs and \mathcal{R} is a constant matrix capturing the friction cone and the positive normal force condition. Hence, the domain of admissibility is defined as:

$$\mathcal{D} = \{(q, \dot{q}, u) \in T\mathcal{Q} \times U \mid A(q, \dot{q}, u) \geq 0\}. \quad (5.58)$$

For the walking of this planar robot, this switching surface is defined when the height of the non-stance foot crosses zero. Correspondingly, the guard is defined as

$$S = \{(q, \dot{q}, u) \in \mathcal{D} \mid p_{nsf}^z(q) = 0, \dot{p}_{nsf}^z(q, \dot{q}) < 0\}. \quad (5.59)$$

5.3.3 Virtual Constraints

The definition of virtual constraints follows from the discussion in Chapter 2. For simplicity of implementation, we pick actuated joint angles as our actual outputs,

$$y_2^a(q) = (q_{sk}, q_{sh}, q_{nsh}, q_{nsk}), \quad (5.60)$$

and desired outputs, y_2^d , are defined in terms of 4th order Bézier polynomials. There is no velocity-modulating output because the robot has underactuated point feet.

5.3.4 Online Gait Optimization

In this section, we employ the pseudospectral gait generation formulation in Section 4.4 to design optimal gaits online. In particular, we optimize new HZD gaits subject to a specific desired forward velocity that changes when the robot is still walking, and apply the newly optimized gait parameters α^* to change the walking velocity of the robot in real time.

Objective-Oriented Constraints. In addition to all necessary HZD constraints introduced in Section 4.3, we formulate following objective-oriented (OC) constraints for our particular speed-regulation purpose. For periodic walking gait, the distance travelled during a step equals the step length of the gait. Let \bar{v}^d be the desired forward speed of the robot, the optimized gait should satisfy the following condition:

$$\left\| \frac{l_{step}(q_N)}{T} - \bar{v}^d \right\| \leq \delta \quad (5.61)$$

for a small constant $\delta > 0$, where $l_{step}(q_N) = p_{nsf}^x(q_N)$ is the step length, and T is the duration of one step. With a goal to apply optimized gaits to the robot hardware on the fly, additional constraints must be enforced so that the resulting gaits are reasonable and feasible from the viewpoint of the actual hardware. In particular, we consider the following constraints.

- The upper body should be upright as much as possible, i.e., given $q_{tor}(q) = -q_{sf} - q_{sk} - q_{sh}$, we impose that

$$q_{tor}^{\min} \leq q_{tor}(q) \leq q_{tor}^{\max}. \quad (5.62)$$

- There should be enough swing foot clearance to prevent scuffing, which we constrain in such a way that the non-stance foot is always above a curve:

$$p_{nsf}^z(q) - h_{nsf}^d \geq 0, \quad (5.63)$$

where $h_{nsf}^d(q)$ is a pre-determined desired foot clearance curve.

- The joint velocities and actuator torques must be within the specifications of the motors:

$$-\dot{q}^{\max} \leq \dot{q} \leq \dot{q}^{\max}, \quad (5.64)$$

$$-u^{\max} \leq u \leq u^{\max}. \quad (5.65)$$

Cost Function. With a goal to achieve energy efficient locomotion at a specific speed, the cost function of the NLP is defined as the mechanical cost of transport of the gait, with the running cost function given as:

$$\mathbf{L}(u, \dot{q}) := \frac{1}{mgl_{step}} P(u, \dot{q}), \quad (5.66)$$

with $P(u, \dot{q})$ is the total power consumed on the actuated joints:

$$P(u, \dot{q}) = \|u \circ (B^T \dot{q})\|^2, \quad (5.67)$$

where B is the torque distribution matrix. The cost function of the pseudospectral optimization is formulated via (4.51).

5.3.5 Experimental Results

In this section, we present the experimental results of the online speed regulation optimization.

Experimental Setup. As shown by Figure 5.34, the robot is mounted on the treadmill utilizing a cage on which a boom is mounted. The robot is constrained to the lateral plane to restrict the robot to up-down and forward-backward movements only—that is, the boom *only* supports the robot in the lateral plane. The boom structure is freely allowed to slide on the cage in order to ensure that the motion of the robot is not restricted in its general direction of walking. The two main processing units: one for actuation, and the other for generating new desired trajectories, are both mounted on the stationary gantry and connected to the robot via ETHERCAT.

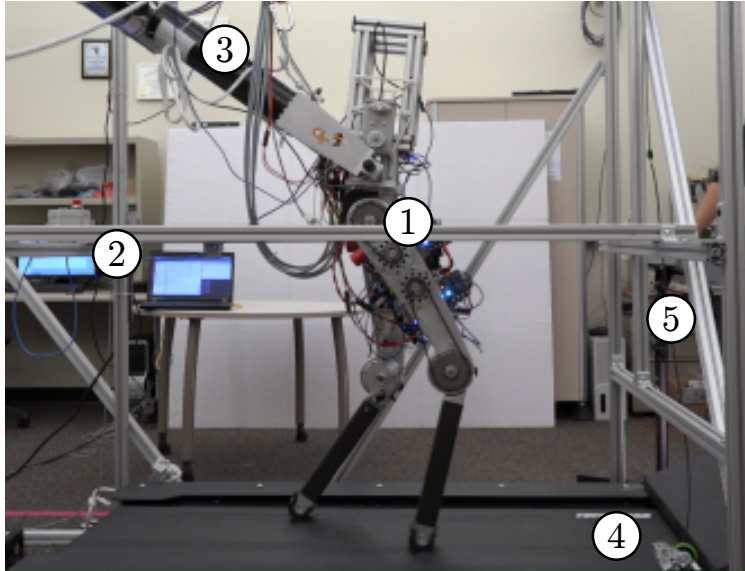


Figure 5.34: Experimental setup of the speed-regulated walking of a planar underactuated robot on a treadmill. The labels represents: (1) PROXI robot, (2) the control workstation, (3) the linear boom used to constrain the robot to the sagittal plane, (4) the treadmill whose actual velocity is measured by an encoder attached, (5) the control panel used to change the velocity of the treadmill.

The speed of the treadmill can be changed from its control panel, and the actual speed of the treadmill is measured by an encoder, the readings are updated and sent to the off board processing unit at the rate of 1 kHz. The online gait optimizer immediately generates new virtual constraints and sends them to the first processing unit, waiting to be updated at the beginning of each robotic step. The desired joint angles and velocities are generated by the off board processing unit in real time using the optimal gait parameters α^* from the gait optimization, and sent to the local controller at the rate of 1 kHz.

Nominal Gait Generation. First, we use the pseudospectral gait optimization to generate a nominal gait off-line. In this case, we run the optimization from a set of random initial guesses (of the decision variables). The desired walking speed of the nominal gait is chosen to be 0.65 m/s . The offline optimization, which runs on a laptop computer with an Intel Core i7-3820QM processor (2.7 GHz) and 12 GB of

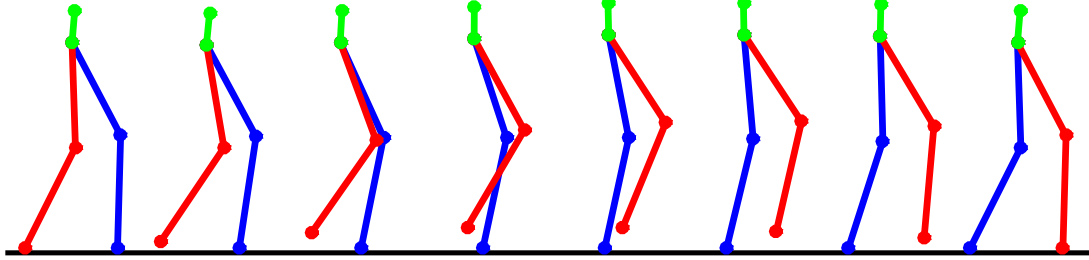


Figure 5.35: Snapshots of the nominal gait in simulation.

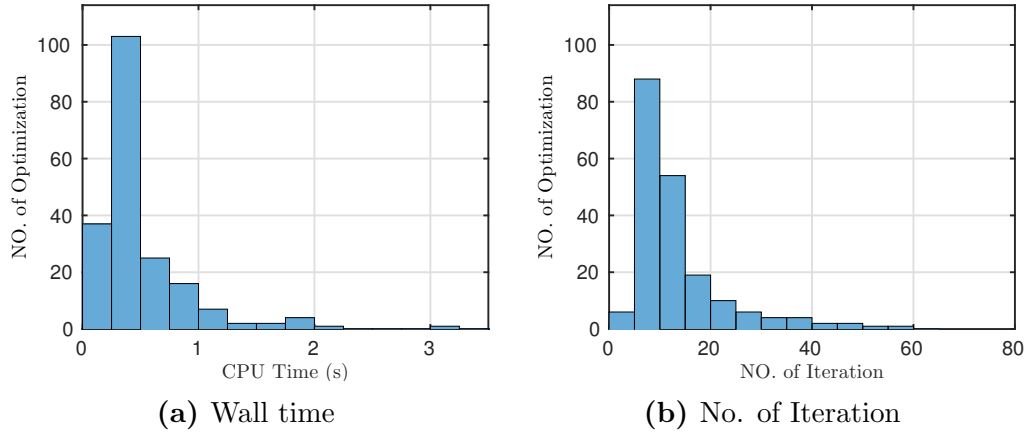


Figure 5.36: The histogram plots of total 198 gait optimizations.

RAM, converges to an optimal solution successfully with an average CPU time of 9.5 seconds. Figure 5.35 shows a series of snapshots of the nominal gait during one step in simulation.

Performance of Online Optimization. The measured treadmill speed serves as the desired speed \bar{v}^d in the pseudospectral online gait optimization and is published to a ROS message at the beginning of every step. Once the online gait optimizer, which runs on the same laptop computer, receives a new message from the ROS message, the optimizer runs a new optimization subject to the new desired walking speed if the difference between the updated speed and previously optimized speed is greater than a certain threshold. In particular, we pick both the threshold and the δ in (5.61) to be 0.01. Once the optimization converges, new gait parameters are sent to the off board processing unit immediately and applied to the gait controller at the beginning of the next robotic step.

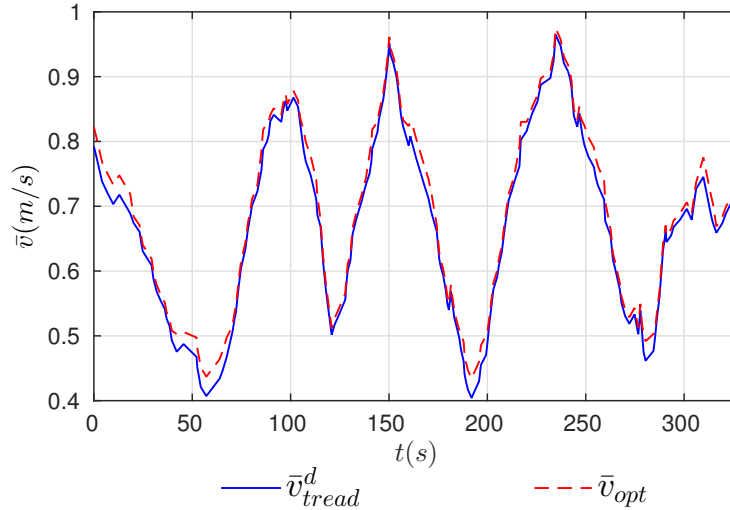


Figure 5.37: The tracking of varied treadmill speed, \bar{v}_{tread}^d . The dashed line represents the speed, \bar{v}_{opt} , of the optimized gait.

In order to achieve faster convergence, we enabled the `warm start` feature of the optimizer in IPOPT where we used the results from the previous optimization as the initial guess for the next optimization. Further we provided the exact Hessian of the problem using the analytical second order derivatives of all constraints and cost function instead of the Quasi-Newton approximation of the Hessian. By doing so, the optimizations converged faster and more reliably. Figure 5.36 shows the histogram figures of the CPU time spent and total number of iterations of each optimization. The average CPU time spent of the total 198 gait optimizations during the experiment of 328 seconds is 0.4964 second, which is less than the average time of one step. There are only two occasions when the optimizer ran for more than 2 seconds. Furthermore, the online optimizer converged successfully to an optimal solution in all occurrences within the maximum allowed iterations of 100. In fact, the average number of iterations is just about 13 with the maximum number being 67.

For the experimental results reported in this thesis, we started from the nominal gait and then changed the treadmill speed from its control panel within a range from 0.43 m/s to 0.97 m/s . The treadmill speed was slowed down and speeded up

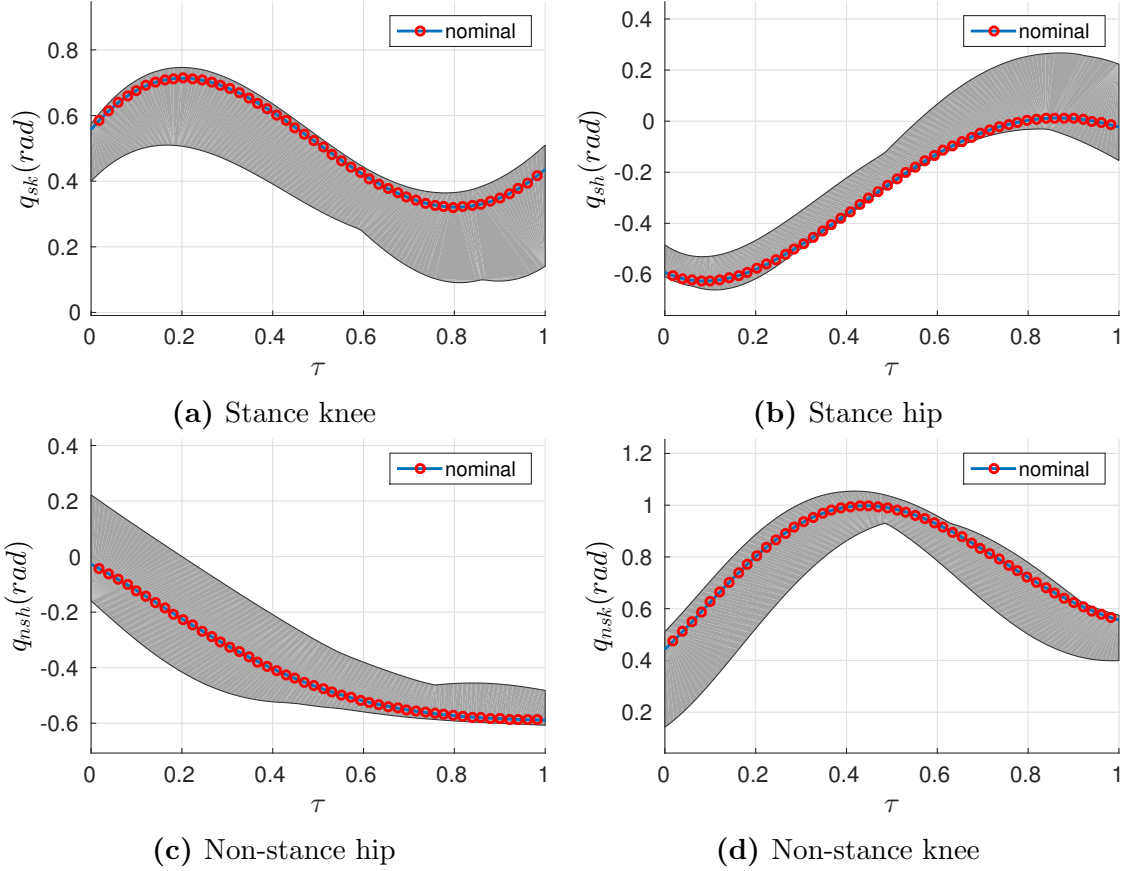


Figure 5.38: The gray areas represent desired outputs of optimal gaits at different walking speeds generated by the online optimizer. The blue lines with red circles show the desired outputs of the nominal gait.

Table 5.4: Performance comparison of online gait optimization with different methods of computing the Hessian (Total number of optimizations is: 198.)

Method	CPU Time (sec)		NO. of Iteration	
	Average	Std.	Average	Std.
Exact Hessian	0.4954	0.3997	13.1414	10.13
Quasi-Newton	0.8927	0.8749	44.5657	36.60

multiple times, as the blue line shown in Figure 5.37. The dashed red line showed that the online optimizer generates new gaits that closely tracked the desired walking speed. The video demonstration of this experiment can be found in [3].

The performances of the online gait optimization when providing the exact and Quasi-Newton approximation of the Hessian are also studied. The comparison results

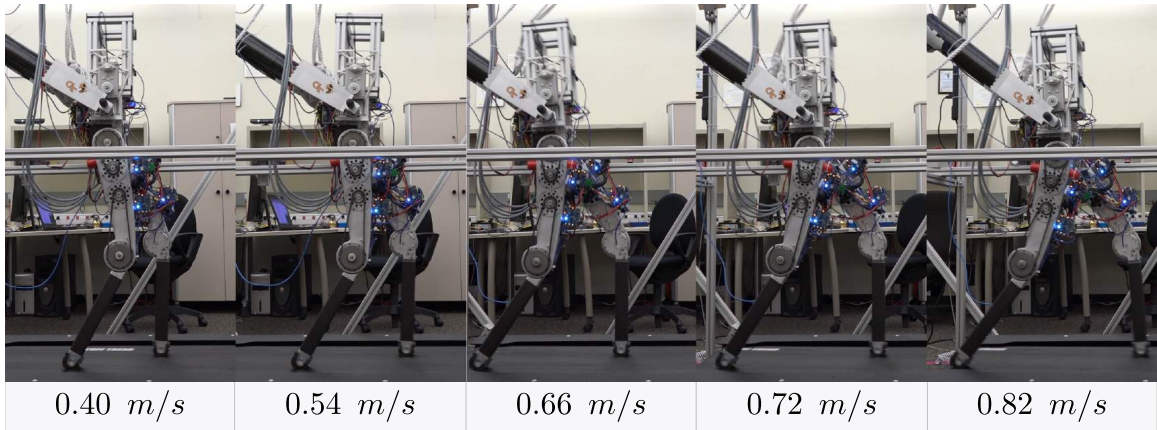


Figure 5.39: Snapshots of optimal gaits generated from the online optimizer at different walking speeds.

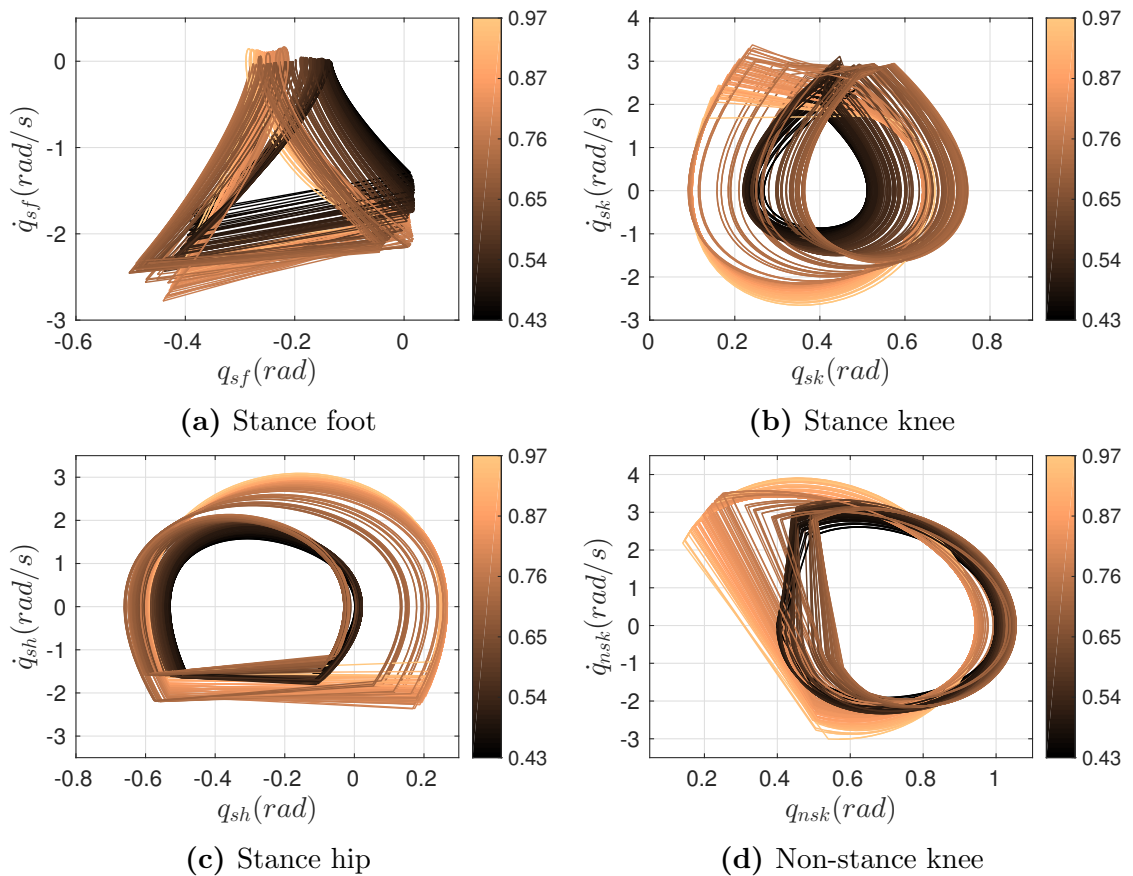


Figure 5.40: Phase portraits of optimal gaits at different walking speeds generated by the online optimizer. The color bar indicates the speed of the gaits, where darker lines represent slower speeds and lighter lines represent faster speeds.

are shown in Table 5.4, in which the optimizer spent more time and required more iterations to converge when using Quasi-Newton approximation. Further, in 50 of total 198 occurrences, the optimization stalled at an almost feasible solution, which is indicated as the “`Restoration Failed`” in IPOPT outputs¹. These failures do not occur when we were using the exact Hessian for the problem.

Gait Performance. With the proper objective-orientated constraints presented in Section 5.3.4, all gaits generated from the online optimizer are physically realizable on the robot hardware. Figure 5.39 shows the snapshots of the gaits during the speed regulated walking experiment. Figure 5.38 shows the range of desired outputs of all gaits generated. These gray areas are then compared to the desired outputs of the nominal gait, which shows very good similarities. Figure 5.40 shows the phase portraits of robot joints for all gaits generated, where darker lines represent the slower gaits and lighter lines represent the faster gaits. It can be noted that each gait produces periodic orbits for the robot joints. Also, the size of the orbits is expanded when the speed of the robot is increased, which shows that the optimizer generated gaits with faster joint velocities and wider joints movements as the speed increased.

These results demonstrate that the pseudospectral method empowers a fast and reliable gait optimization method which is, for the first time, capable of generating HZD gaits online. We experimentally validated the optimization method on a planar 5-link underactuated robot, PROXI, to generate gaits online subject to varying desired walking speeds. The online full body dynamics optimizer successfully optimized energy-efficient walking gaits in average of 0.5 seconds, while satisfying all dynamical and kinematics constraints.

¹<http://www.coin-or.org/Ipopt/documentation/node35.html>

5.4 *Summary*

These applications that we introduced in this chapter show that the hybrid zero dynamics based direct collocation optimization approaches provides a generalized framework for designing dynamic walking gaits for legged robots. The particular formulation of constraints using defect variables enables the problem can be scaled to very high dimensional system, as shown in both DURUS and DRC-HUBO application. Particularly in the case of DURUS, the optimization still manages to converge even in the presence of many degrees of underactuation and multiple domains of locomotion. More importantly, the fact that the optimal solution is based on distributed information due to the very sparse formulation yields reliable convergence and much more robustness to initial guess. Especially, the advanced pseudospectral optimization even makes it feasible to solve the whole body dynamics gait optimization online so that new gaits can be updated to accommodate the changes in external environment, as we seen in the last application.

CHAPTER VI

OPTIMIZATION PERFORMANCES

In the previous chapter, we applied the direct collocation optimization on several real world examples to successfully generate optimal walking gaits of different configurations. Through these example we have shown that the proposed methods can be scaled to very high dimensional systems, even in the presence of unactuated components. Yet, it is an inevitable step to measure overall computational performances of the proposed methods. In this chapter, we will try to answer the following questions regarding the direct HZD gait optimization problems presented in Chapter 4:

- How accurate a solution of the problem is to the exact solution of the system?
- How reliable these algorithms perform in case of bad initial guesses?
- How large the problems are and how fast they converge to an optimal solution?

If not specified otherwise, all NLP evaluations in the following discussion are solved using IPOPT within MATLAB¹ on a laptop computer with an Intel Core i7-6820HQ processor (2.7 GHz x 8) and 8 GB of RAM. Albeit the computer is able to perform parallel computation, we only use a single core for the computation for the reason that the results will not be affected by the particular algorithms used by parallel computing, and therefore, more consistent. In addition, we use the linear solver `ma57` for IPOPT and set the feasibility tolerance to be 1×10^{-6} .

¹The MATLAB interface for IPOPT: <https://projects.coin-or.org/Ipopt/wiki/MatlabInterface>.

6.1 Solution Accuracy

The solution obtained via the direct collocation optimization is an approximation of the actual solution of the system by definition. However, it is often desired that this approximation is an accurate estimation of the system solution within a certain tolerance. There are two main measures to represent the accuracy of a discrete solution. Suppose that $x(t)$ be the exact solution of the system and $x^{(i)}$ be the computed discrete state variables at time t_i from the optimization, the *global error* at a point t_{i+1} is the difference between $x(t_{i+1})$ and $x^{(i+1)}$, and the *local error* is the difference between the compute solution $x^{(i+1)}$ and the exact solution $x(t_{i+1})$ that passes through $x^{(i)}$. Typically, if the global error of a solution is $\mathcal{O}(h^p)$, then the local error is $\mathcal{O}(h^{p+1})$. Thus, we particularly estimate the local error of the solution from the direct collocation optimization to assess the solution accuracy in this chapter.

6.1.1 Error Estimation

We start with representing the approximated solution using the discrete state variables in the case of the local direct collocation method. Suppose $i \in \{1, 3, \dots, N_v - 1\}$ be the index of a interior node, then the approximated solution $\tilde{x}(t)$ within two cardinal nodes, i.e., $t_{i-1} \leq t \leq t_{i+1}$, is given as

$$\tilde{x}(s) = C_0 + C_1 s + C_2 s^2 + C_3 s^3 \quad (6.1)$$

where $s = (t - t_{i-1})/\Delta t_i \in [0, 1]$ is the scaled time. The coefficients are determined by the boundary state Variables:

$$\tilde{x}(0) = x^{(i-1)}, \quad \dot{\tilde{x}}(0) = \dot{x}^{(i-1)}, \quad (6.2)$$

$$\tilde{x}(1) = x^{(i+1)}, \quad \dot{\tilde{x}}(1) = \dot{x}^{(i+1)}. \quad (6.3)$$

To calculate the local error, we consider a single integration step between two neighboring cardinal nodes $t_{i-1} \leq t \leq t_{i+1}$. Specifically, we consider the affine

control system of the full order dynamics in (2.20), and the closed-loop dynamics $f_v^{cl}(x, \alpha_v)$ obtained via the feedback controllers computed from the virtual constraints parameters. This is another advantage of considering the feedback controllers in the NLP, so that the computed control is correct without additional assumptions. Then the exact solution at point t_{i+1} that passes through $x^{(i-1)}$ is:

$$x(t_{i+1}) = x^{(i-1)} + \int_{t_{i-1}}^{t_{i+1}} f_v^{cl}(x(t), \alpha_v) dt. \quad (6.4)$$

Note that the computation involves the exact solution of $x(t)$, which is unknown. Here, we use the approach that is discussed in [19] to estimate the local error of interval i . That is, we consider the approximation of the solution $\tilde{x}(t)$ in (6.1) to approximate the integration. Therefore, we have

$$\hat{x}(t_{i+1}) := x^{(i-1)} + \int_{t_{i-1}}^{t_{i+1}} f_v^{cl}(\tilde{x}(t), \alpha_v) dt, \quad (6.5)$$

and define the *absolute local error* on a particular time interval i as,

$$\begin{aligned} e_i &= |x^{(i+1)} - \hat{x}(t_{i+1})| \\ &= |x^{(i-1)} + \int_{t_{i-1}}^{t_{i+1}} \dot{\tilde{x}}(t) dt - (x^{(i-1)} + \int_{t_{i-1}}^{t_{i+1}} f_v^{cl}(\tilde{x}(t), \alpha_v) dt)| \\ &= \int_{t_{i-1}}^{t_{i+1}} |\dot{\tilde{x}}(t) - f_v^{cl}(\tilde{x}(t), \alpha_v)| dt. \end{aligned} \quad (6.6)$$

To calculate this error, we numerically integrate the error dynamics $|\dot{\tilde{x}}(t) - f_v^{cl}(\tilde{x}(t), \alpha_v)|$ over the time interval between two cardinal nodes, using MATLAB's ODE solver `ode45`¹, and compare the states values resulted from the NLP at all collocation points. From the definition of the absolute error, the maximum *relative local error* at the mesh i is defined as the maximum relative error over all components $k \in [1, n_v]$:

$$\varepsilon_i \approx \max_k \frac{e_{i,k}}{\omega_k + 1}, \quad (6.7)$$

¹In particular, we set the relative error tolerance `RelTol` and the absolute error tolerance `AbsTol` to be 1×10^{-12} .

where the scaling weight

$$\omega_k = \max_{i \in [1, \dots, N_v - 1]} \{|x_k^{(i)}|, |\dot{x}_k^{(i)}|\} \quad (6.8)$$

defines as the maximum value of the k^{th} state variable or its first-order derivative over all nodes within a domain.

6.1.2 Numerical Results

For simplicity of calculation, we assess the solution accuracy using two different direct collocation methods—local direct collocation and pseudospectral method—for the planar PROXI robot (see Section 5.3.1 for the model description). The estimated errors of solution are computed for direct collocation NLPs with different number of collocation nodes. Specifically, the initial guess for the NLP is determined in the same way for each trial, given as follows: the discrete states, torques, etc., are obtained via interpolating by simulation results of a walking gait that does not satisfy the current physical constraints, and the initial guess of virtual constraints parameters are given as zero. In addition, the same physical constraints are enforced for all tests.

Local Direct Collocation. The results of the local direct collocation using two different node distribution schemes are shown in Table 6.1 and Table 6.2. The number of nodes is the total number of collocation nodes, $N_v + 1$, including both cardinal and interior nodes.

It can be seen from Table 6.1 that the discretization error reduces when there are more collocation nodes. It can also be noticed that the discretization error does not reduce further significantly when the total number of nodes exceeds a certain value, e.g. 49, which corresponds to the number of cardinal nodes is 25. A similar phenomenon can be seen in the final values of the cost function for each trial: its changes become negligibly small with more number of collocation nodes. On the other hand, the total run time of the NLP escalates noticeably when more collocation nodes are used in the problem. We also notice that the number of iteration tends to increase

Table 6.1: Solution accuracy of local direct collocation optimization when using the CGL node distribution.

$N_v + 1$	$\max(\varepsilon_i)$	Runtime(s)	Iteration	Cost
9	2.6000×10^{-3}	0.86626	46	0.051557
19	1.1508×10^{-4}	2.0375	62	0.052609
29	2.0688×10^{-5}	3.0186	62	0.052747
39	7.5922×10^{-6}	4.5471	73	0.052723
49	5.9023×10^{-6}	5.5129	75	0.052737
59	4.8004×10^{-6}	13.502	124	0.052753
69	3.9683×10^{-6}	9.6407	96	0.052741
79	3.3533×10^{-6}	12.8	90	0.052747
89	2.8189×10^{-6}	20.116	151	0.052751
99	2.3507×10^{-6}	13.81	98	0.052747

Table 6.2: Solution accuracy of local direct collocation optimization when using the uniform node distribution.

$N_v + 1$	$\max(\varepsilon_i)$	Runtime(s)	Iteration	Cost
9	3.5966×10^{-3}	1.2876	74	0.052753
19	1.7699×10^{-4}	2.9308	72	0.053664
29	3.2316×10^{-5}	2.9287	63	0.053216
39	9.8434×10^{-6}	10.602	116	0.052975
49	4.0203×10^{-6}	7.3709	108	0.052844
59	3.0967×10^{-6}	9.9454	119	0.052775
69	2.596×10^{-6}	10.808	111	0.052742
79	2.2136×10^{-6}	11.617	104	0.052726
89	1.8851×10^{-6}	13.562	102	0.052723
99	1.5809×10^{-6}	17.383	121	0.052727

albeit very slightly. This is because the total number of optimization variables and constraints grows proportional to the number of collocation nodes, therefore, the optimizer consumes more time to evaluate these constraints functions. This observation shows that using more collocation nodes does not yields better approximated solutions, instead it increases the total computational burden significantly.

A same observation can be made from Table 6.2 in the case the collocation nodes are uniformly distributed. When comparing Table 6.2 with Table 6.1, the maximum estimated approximation error of the uniformly distributed collocation is larger than

that of the CGL distribution when the total number of nodes is small, and similar when the total number of nodes is large. Hence, the CGL distribution provides better approximation when using small number of collocation nodes, however, has little effect on the approximation accuracy when there is enough number of nodes used. This shows that placing more points at two ends will benefit the accuracy of the solution when there is less collocation nodes.

The total run-time and total number of NLP iterations to converge in two distribution schemes does not differ a lot in general. There is a few examples in which the run time or the iteration number differs in two cases, however, we believe that is mostly due to the random initial guess seeded to the NLPs. Despite using different numbers of collocation nodes, we observe that the final values of the cost function are almost the same in both distribution schemes. This shows that the local direct collocation optimization has a very good robustness property, which we will discuss more in the later section.

Pseudospectral Optimization. In the case of the pseudospectral method, we use the similar formula in (6.7) to estimate the local discretization error of the NLP solution. Different from the local direct collocation method, the approximated solution is given by the Lagrange polynomials as shown in (4.44), and the time interval is between each two neighboring collocation nodes instead of two cardinal nodes. The numerical results of the pseudospectral method are shown in Table 6.3.

As expected, solutions of the pseudospectral optimization exhibits much better accuracy even with very small number of collocation nodes. The estimation error can be reduced around 1×10^{-8} with enough nodes, which is very difficult to obtain with the local direct collocation methods. Due to the better accuracy, we normally have smaller size of problem with the pseudospectral methods, and thus the wall time of each iteration becomes shorter. However, this approach tends to less robust to bad initial guesses because it uses a single high order polynomial to approximate the

Table 6.3: Solution accuracy of the pseudospectral optimization.

$N_v + 1$	$\max(\varepsilon_i)$	Runtime(s)	Iteration	Cost
6	1.9672×10^{-4}	1.4679	126	0.1483
9	1.5330×10^{-5}	2.8681	139	0.0640
12	6.9968×10^{-6}	5.9900	231	0.0576
15	5.6788×10^{-6}	8.7357	253	0.0525
18	4.6710×10^{-7}	5.4900	145	0.0527
21	4.0865×10^{-8}	6.2709	146	0.0527
24	1.7321×10^{-8}	8.9091	175	0.0529
27	1.4928×10^{-8}	8.3726	149	0.0528

solution, and thus relies on the global information rather than distributed information. This can be seen from the total number of iterations, which are typically greater than its counterpart local collocation method. Moreover, when using small number of collocation nodes, e.g., 6 or 9, the final value of the cost function is greater than the local methods. The total run time of the pseudospectral methods seems to be in a similar order of the local methods despite it often takes more iterations to converge.

6.1.3 Summary

It is a mandatory requirement of a NLP formulation that the discrete solution obtained from the direct methods should be sufficiently close to the exact solution of the system. Numerical results in this section show that both collocation techniques yield sufficiently small discretization errors with enough collocation points are imposed. Especially the pseudospectral methods exhibits an excellent accuracy which makes it possible to reduce the size of the NLP problem.

In addition, the error estimation approach can be utilized to refine the collocation meshes of the NLP in the local collocation schemes [19, 98]. Admittedly, this process often requires extensive and iterative evaluations of discretization errors in order to obtain slightly more accurate solutions. Hence, we do not perform any mesh refinement algorithm in current formulation by assuming that the resulting solutions

are accurate enough for our purpose. This leads to choosing the number of collocation nodes becomes very important. All factors should be considered to balance the trade-offs between the solution accuracy and computational efforts. For example, one possible good choice in the above tests would be 25 of cardinal nodes, because it yields an acceptable approximation accuracy for the solution, and at the same time, it requires reasonably little computational effort.

6.2 Robustness Analysis of NLP Convergence

It is very important for a direct NLP formulation to converge reliably regardless of good or bad initial guesses. Though there is no formal guarantee that a direct HZD optimization problem converges to a local minima when starting from arbitrary initial guesses, it is required that the optimization problem should be able to find a feasible solution in most instances.

6.2.1 Robustness Measurement

In the previous section, we briefly discuss the robustness of the direct collocation methods while seeding with relatively good initial guesses. In those experiments, we obtain the initial guess from the simulated solution of a walking gait that does not satisfy current physical constraints. To evaluate the global robustness of the optimization, here we seed the NLPs with arbitrary random initial guesses at each trial. Typically a NLP that uses direct collocation methods has a very large size of decision variables, which causes a certain difficulties to generate initial guesses for the problem. A straightforward approach could be just simply assigning arbitrary random values to all variables, however, some assigned values might not lie in the feasible regions of particular decision variables, and as a consequence, could be modified by the NLP solvers to make sure they are all within their boundary limitations. For our tests, we use a systematic procedure to generate partially-random partially-deterministic initial guesses for the problem. First, the time duration of domains, as

well as step features such as step length or width, are chosen as some reasonable fixed values ¹ because they can be approximated easily by our own experiences. The state variables $(q^{(i)}, \dot{q}^{(i)})$ and control inputs $u^{(i)}$ are chosen randomly within the boundaries. Suppose that $w_j^{(i)}$ is one of these variables with a lower and upper boundary of w_j^{\min} and w_j^{\max} , then at each node, we set the random initial guess of this variable to be

$$w_j^{(i)} = (w_j^{\max} - w_j^{\min}) \times r + w_j^{\min} \quad (6.9)$$

where $r \sim U([0, 1])$ is an uniformly distributed random number in the interval $[0, 1]$, which we use MATLAB's `rand` function to generate for every variables at every single node. The virtual constraints parameters, $\alpha_v^{(i)}$, are determined in a similar manner within a specified range of $[-1, 1]$. Other optimization variables, such as joint accelerations $\ddot{q}^{(i)}$ and ground reaction wrenches $\lambda_v^{(i)}$, are then set to be zero because their upper and lower boundaries are often difficult to determine *a priori*. In practice, we often set reasonably large boundary values for these variables to make sure that the optimization will not be jeopardized by some unreasonably small boundary values.

Seeding with different sets of initial guess that are randomly generated at every trial, we run the gait optimization 100 times subject to same physical constraints. For each trial, we first check if the problem converges to a feasible solution before it reaches the maximum allowable number of NLP iterations. If the problem converges, we then analyze the following metrics: the total number of iterations, the final cost value, and the correlations between other solutions. The correlations between different solutions are computed via the correlation coefficients of two variables, which is a measure of their linear dependence. If the coefficient is 1, then they are perfectly positive correlated; there is no correlation between these two variables if the coefficient is 0, and perfectly negative correlations if this number is -1 . Considering that the solution

¹These values are not obtained from any previous walking gaits, instead approximately chosen from human walking patterns.

of whole NLP variables are often very large and their correlation coefficients might not be able to correctly represent the similarity between two resulting gaits, we instead compute the coefficients between the virtual constraints parameters of two solutions as a mean to examine their similarity. Numerical results of several robustness tests are presented in the next section.

6.2.2 Numerical Results

From the computational point of view, we first use the simplest model that we discussed—PROXI—to conduct our numerical evaluations of robustness. In particular, we set the maximum number of iterations of the NLP solver to be 2000, and the convergence tolerance to be 1×10^{-6} (see Eq. (6) in [132]). The number of collocation nodes for the local direct collocation and pseudospectral methods are set to be 29 and 15, respectively, and same physical constraints are enforced for both cases.

Local Collocation Optimization. We start with the local collocation method. The result shows that in 79 out of 100 trials, the optimization problem converges to an optimal solution. More importantly, the final costs of all converged solutions are the same, despite they are obtained by seeding with different initial guesses. Further the correlation coefficients between these optimal solutions are 1, which show that the problem actually converges to an exact same optimal solution even though the problem starting from different initial points at each trial. It is difficult to formally verify whether it reaches a global optimal solution, however, this observation still demonstrates the excellent reliability and robustness of the direct hybrid zero dynamics optimization formulations. It shows that the local collocation scheme is able to find a “very good” local minima in most scenarios regardless of the initial guess. The failed trials indicate that the location of the starting point does affect the final convergence to some degree. The outputs from IPOPT solver further reveal that the constraint violations in these cases are less than the tolerance, however, the dual

infeasibility condition is not satisfied within the given tolerance. That is, these are feasible but not optimal solutions.

Unlike the final convergence properties, the total number of iterations and corresponding CPU run time are affected by different initial guesses seeded to the optimizer. The total number of iterations varies from 147 to 1837 and the total run time range from 6.13 seconds to 74.82 seconds, respectively, for those cases in which the problem converges to the same optimal solution. Moreover, the average number of iterations is 322 with standard deviation of 218, and the average total CPU time is 16.62 seconds with standard deviation of 10.19 seconds. It is understandable that the computational performance has such a large deviation considering the fact the initial guesses are randomly selected without any prior knowledge of a feasible solution.

Pseudospectral Optimization. We also evaluate the pseudospectral method that uses global orthogonal collocation schemes. In previous section, we observed that the pseudospectral method appears to be less robust than the local collocation method. However, our more extensive robustness test indicates otherwise. Under the same circumstances, the pseudospectral optimization problem converges to an optimal solution in 87 out of the total 100 trials. Similar to the local methods, the constraint violations in these “failed” cases are quite small but the dual infeasibility conditions are not satisfied.

We observe that there are three different values appears in the final cost of these optimal solutions, indicating that the problem might converges to different solutions when seeding with different initial guesses. Notwithstanding, the numbers of occurrences of these three optimal solutions are 82, 4, and 1 respectively, indicating that the problem converges to a same optimal solution in most instances.

Further, the global method uses almost twice as much iterations in order to converge in most cases, yielding an average of 778 iterations. The average CPU run time of 17.15 seconds, which show that the average evaluation time of the pseudospectral

optimization is about the same as of the local collocation, despite requiring more iterations. The reason behind this result is that the problem size of the global method is only about the half of the local collocation problem.

Testing on High Dimensional Systems. In previous two tests, we use a planar robot for simplicity. It is necessary to evaluate the robustness of the direct HZD optimization for higher dimensional systems with more complex nonlinear dynamics. For this goal, we particularly test the robustness of the local direct collocation optimization for the two-domain 3D flat-footed walking of DURUS robot. In particular, we set the numbers of collocations nodes to be 15 and 25 for the single-support and double-support domains, respectively. The maximum allowable number of iterations in this case is set to 3000.

Similar to the local collocation optimization of planar robots, the problem still converges to an optimal solution in 79 out of 100 trials despite the complexity of the full humanoid model dynamics. Different from the planar case in which the problem converges to an universal solution, the gait optimization of DURUS resulted in nine different optimal solutions. Unfortunately, the “global” convergence property that exhibited in the planar case does not extend to more complicated 3D humanoids. An interesting fact is that some solutions are more frequent to obtain, where some only occur once. For instance, the numbers of occurrence of the trials that corresponds to the cost value 0.1523, 0.1773, and 0.1861 are 17, 24, and 24, respectively. On the other hand, there are four instances in which the optimal solution only occurs during the 100 trials.

In particular, the total number of iterations varies from 464 to 2845 and the total run time range from 174.06 seconds to 1318.2 seconds, respectively, for those cases in which the problem converges to the same optimal solution. Moreover, the average number of iterations is 1080 with standard deviation of 533, and the average total CPU time is 511.5 seconds with standard deviation of 286.2 seconds.

6.2.3 Summary

In this section, we tested the robustness to two direct collocation methods under much harsher conditions. Fortunately, our methods still successfully converged to in most tests regardless, exhibits a excellent robustness performance.

In principle, it is very difficult to have a formal guarantee of global convergences from arbitrary starting points for a NLP problem, especially when the problem subject to complex nonlinear constraints. This is the case for the direct HZD optimization, for the reason that this method optimizes the interactions of the full body nonlinear dynamics of the robots hybrid system model, without restricting motions to simpler more-tractable dynamics (e.g., inverted pendulum models). Even the planar robot—which we consider as the "best case scenario"—has a state space dimension of 14, not to mention the full-size underactuated humanoid, DURUS. Even more, the way we generate our random initial guesses are quite intensive, and we believe that the results in this section are enough to demonstrate the reliability of the proposed direct HZD optimization methods.

Admittedly, there are some discrepancies in robustness among different methods used and different models tested. To sum up, the local collocation method appears to be more robust than the global method, as shown in both the final cost and the number of iterations it takes. Nonetheless, the pseudospectral method has its own advantage in the computational time due to the smaller problem size. We have also observed an degradation of robustness in the local collocation method with more complex systems. Considering that the systems involves two discrete domains and also consists of compliant elements in the mechanical design, 80% of success rate is still an impressive result in this regard. After all, we do not expect our optimization methods are robust to arbitrary random initial guesses. In practice, it would not be too difficult to find some initial guesses that are much better than those randomly generated ones we used in our tests.

6.3 Computational Efficiency

Arguably the most important performance measure of a nonlinear optimization problem would be its computational effectiveness. While there is no formal mean to measure the computational efficiency of a NLP problem, the total CPU time used to converge is a very commonly used metric. The total number of iterations and the number of function calls are also considered in the evaluation of the effectiveness.

In this section, we first compare the computational performance against other gait optimization methods, mostly against the direct shooting methods. To have reasonable comparisons as much as possible, we perform all optimization evaluations on the same laptop under the same hardware and software environments. Indeed, there are many possible reasons that might affect the results. As a consequence, it would be very difficult to have completely commensurable comparisons between two different methods. Notwithstanding, the following results demonstrates the better computational efficiency of the direct HZD optimization methods over other methods.

6.3.1 Comparison versus the Single Shooting Optimization

We start with comparing the computational performance between the direct collocation methods and the traditional single shooting optimization. In particular, the single shooting optimization is formulated as described in Section 3.2.2 for a 5 link planar biped with a single continuous domain hybrid system model. Then we use the local direct collocation methods to formulated the HZD gait optimization for the same robot model subject to same physical constraints.

Specifically, both optimization problems are tested with MATLAB's `fmincon` NLP solver instead of IPOPT in these tests. We use the gradient based interior-point algorithm, which is the same algorithm that is used in IPOPT solver [132]. Typically, `fmincon` is more suitable for general purpose nonlinear constrained optimization,

Table 6.4: Comparison results between the (local) direct collocation optimization vs the single shooting optimization.

Method	CPU time (s)	Iteration	Function calls
Single shooting (<code>fmincon</code>)	162.59	21	612
Direct collocation (<code>fmincon</code>)	5.17	23	55
Direct collocation (<code>IPOPT</code>)	1.60	47	59

whereas `IPOPT` is more efficient in solving large-scale sparse problems. We also included the results of the direct collocation optimization using `IPOPT` as a reference.

Considering that the single shooting method is not robust enough to handle random initial guesses, we take the parameters from previous results as the initial guess. The initial guess of the direct collocation optimization is determined in a same manner. Moreover, we set 10 collocation intervals for the direct collocation optimization, and at the same time, physical constraints are checked in 10 uniformly distributed points along the trajectory in the single shooting optimization. The comparison results are shown in Table 6.4.

The results show that the direct collocation method has approximately 30 to 100 times faster than the single shooting method depends on which solver has been used. In addition, the total numbers of function calls are significantly less due to the fact that the analytic first order derivatives of the cost and constraints are provided to the solvers in the direct collocation optimization. We also noticed that the number of iterations is in fact very close in two cases. It appears to be that the capabilities of explicitly solving the pre-impact states and the closed form solution of the two-dimensional zero dynamics benefit the overall convergence of the single shooting optimization. Unfortunately, the same approach cannot be extend to general humanoid robots. Besides, using numerical finite difference methods to compute the gradient information requires extensive amount of computing resources. As a result, the overall CPU time used in the single shooting optimization is much more than the collocation method. Therefore, even the partial zero dynamics can be computed

in closed form as it appears in many fully actuated humanoids [104, 38], the single shooting optimization requires incredible amounts of computing time. Interestingly, even though the number of iterations is in fact greater when using IPOPT, yet still it uses the least amount of wall time. This indicates the fact that IPOPT performs better for large-scale sparse problems.

6.3.2 Comparison versus Multiple Shooting Optimization

As stated in Section 3.2, the single shooting method lacks necessary reliability to be applied to more complex system, particularly the degree of underactuation exceeds one. In [64], we demonstrated that the direct multiple shooting method can increase the reliability of the problem. For simple applications, such as the one we used in previous section, the convergence speed of the multiple shooting optimization is on par with that of the single shooting algorithm. In this section, we evaluate the computational performances of the multiple shooting optimization for some slightly more complex system, and compare the computational performance with the more advanced direct HZD optimization method.

In particular, we use the DURUS-2D model that we introduced in Section 3.3.2. The walking behavior consists of two continuous domains due to the passive springs in the mechanical design. The robot has three degrees of underactuation during the single support phase, which corresponds to dimensions of zero dynamics of six. The multiple shooting optimization of DURUS-2D is formulated as in Section 3.3.2 and Ref. [64]. A direct HZD gait optimization problem of DURUS-2D is also formulated using the (local) direct collocation method subjects to same set of physical constraints. For each optimization, we set the number of shooting grids and the collocation nodes as 15 and 25 for the single-support and double-support domain, respectively.

Similarly, we use `fmincon` for both optimization, and provide the reference results of IPOPT at the same time. The results in Table 6.5 show that the direct collocation

Table 6.5: Comparison results between the (local) direct collocation optimization vs the multiple shooting optimization.

Method	CPU time (s)	Iteration	Function calls
Multiple shooting	5027.45	1155	1624687
Direct collocation (<code>fmincon</code>)	84.30	44	123
Direct collocation (<code>IPOPT</code>)	41.47	207	845

method completely outruns the multiple shooting methods speed-wise: the CPU time elapsed in the multiple shooting optimization is approximately 60 to 100 times greater than its rival—the direct collocation method. In addition, the number of iterations and functions calls are notably more than that of the collocation method. Apparently, the forward integration of zero dynamics using explicit Runge-Kutta methods is much less reliable than the implicit Runge-Kutta methods used in the collocation methods.

6.3.3 Comparison versus DIRCON

Due to the nonlinear complexity of the humanoid dynamics, many trajectory optimization of bipedal locomotion gait tends to use simplified reduced dimensional model, e.g. LIPM [42], instead of the whole body dynamics. In such applications, the system dynamics is linear and relatively simple. An notable exception is DIRCON, developed by Posa et al. at, which optimizes an open loop trajectory for the constrained whole body dynamics [103, 102] and then an constrained LQR-based feedback controller is formulated to stabilize the optimal trajectory. Similar to our proposed optimization, DIRCON also uses the Hermite-Simpson local collocation method. In [102], it is reported that this framework can generate an optimal trajectory for a 3D Atlas robot in ten minutes to two hours on a desktop computer.

In this section, we use our hybrid zero dynamics based direct collocation gait optimization formulation to generate dynamic walking gaits on the same 3D humanoid model of Atlas. A preliminary simulation result of a generated optimal walking gait of Atlas is illustrated in this section, as well as the typical wall time of the gait optimization problem elapsed is given, providing a simple comparison versus DIRCON.

There are many other factors, for example, hardware/software environments and different constraints used or different number of collocation nodes defined, would affect the results of the optimization. Hence this comparison should not be considered as the strict evidence of one formulation is superior than the other. We consider the comparison as an extension of our algorithm for some popular humanoids, e.g., Atlas.

In this example, based on the Atlas robot model described in [85] we consider a 3D humanoid model with six degrees of freedom in each leg. Including the floating base coordinates, this model has 18 total degrees of freedom. Similar to the designed motion in [102], we consider a walking with 3 different contact conditions, resulting in a multi-domain hybrid system model with a directed cycle $\Gamma = \{V, E\}$, given as

$$V = \{ts, tl, hs\} \tag{6.10}$$

$$E = \{ts \rightarrow tl, tl \rightarrow hs, hs \rightarrow ts\} \tag{6.11}$$

where each vertex is defined similar to the DURUS multi-contact walking case (see Figure 5.16). The detailed construction of the hybrid system model and corresponding virtual constraints follows the discussion in Chapter 2, and the direct HZD gait optimization problem is formulated accordingly as in Section 4.3. In particular the number of cardinal nodes are picked as 8, 15, and 8 for each domain, respectively. We also set the objective function to minimize the mechanical cost of transport of the walking gait, which is given as the total mechanical work done by the actuators divided by the weight of the robot and the distance traveled during one step.

The gait optimization problem converged to a feasible solution after 988 seconds and 744 iterations when uses randomly generated initial guess. These numbers reduced to 269 seconds and total of 246 iterations when seeded with relatively “good” initial guess taken from the previous results. We also show the simulation result of one particular optimal walking gait in Figure 6.1. The step length of this gait is 0.48 m, the time duration of a complete step is 0.94 seconds and the mechanical cost

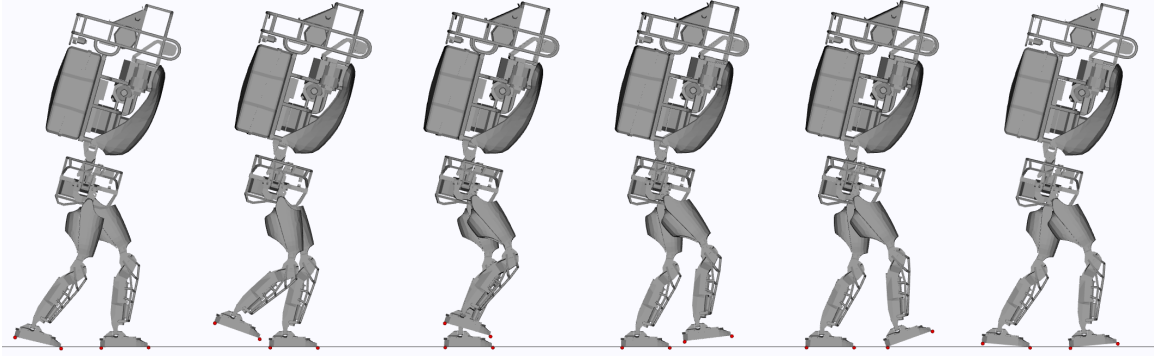


Figure 6.1: Tiled still images from the simulation of Atlas multi-contact walking in 3D at $0.51m/s$.

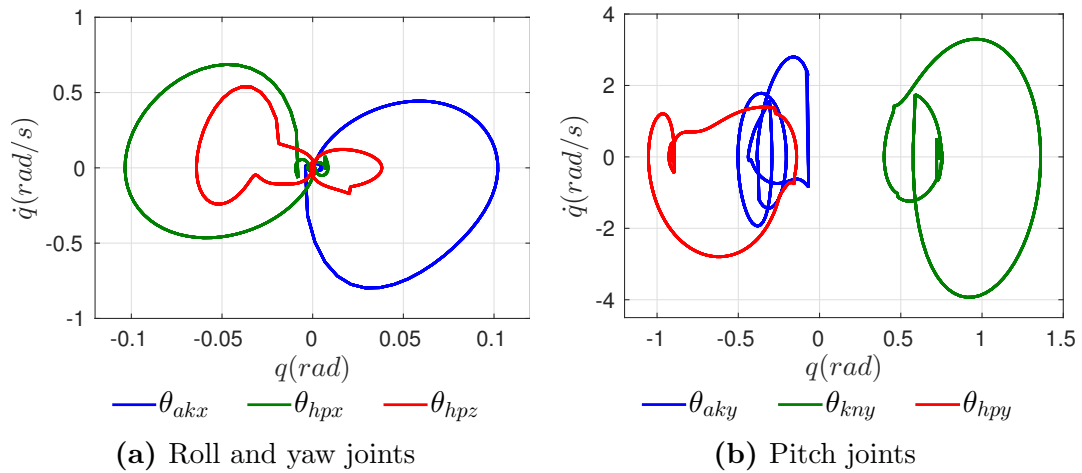


Figure 6.2: Periodic limit cycles of right leg joints of the Atlas 3D multi-contact walking gait in the simulation under the feedback control defined in (2.39). We omit the symmetric joints for simplicity.

of transport (COT) is 0.201. Because in the simulation we use the same feedback controllers as used in the optimization via the linear output dynamics constraints, there is no increase of the COT numbers in simulation. When using QP-based CLF feedback controllers in the optimization, we noticed a slight reduction in the COT number, yielding 0.19. Compared to the result in [102], the gait discussed in this work has slower walking speed but marginally smaller cost of transport number.

In addition, the periodic limit cycles obtained from the simulation over 20 steps are shown in Figure 6.2. Considering that the gait is symmetric, we only show the phase portraits plots of right leg joints. Figure 6.2 demonstrates that the resulting

gaits is periodic and stable.

6.4 Performance Summary of Direct Collocation Methods

Besides the initial guess, the run time and convergence of the optimization problem are also affected by the reachable set of the optimization variables as determined by the boundary values and constraints. Hence in this section, we evaluate the performances of the optimization with respect to different initial guesses and different boundary conditions, by which the reachable set of the problem is determined. Table 6.6 shows some statistical results of the (local direct collocation) optimization of 3D flat-footed walking with DURUS using different initial guesses and constraints¹. There are two different seeds: a random generated initial guess and an initial guess obtained from a walking gait that does not satisfy all constraints. We also considered three different conditions of boundary values: a “relaxed” condition indicates the reachable set of optimization variables set to be large, for example, larger joint velocity, higher actuator torques, and relaxed angle limits, etc.; a “regular” condition suggests that the boundaries of variables are set to reasonable values, mostly set to slightly tighter limits than what the hardware is capable of; a “restricted” condition restricts the reachable set of the optimization variables within a very small region so as to achieve walking gaits with certain fixed behaviors, such as maintaining a straight torso.

Our results showed that adjusting the initial guess and relaxing the constraints can improve the convergence time of the optimization. However, we observe that even from a completely random initial guess subject to strict hardware constraints the NLP problem still converges successfully. This suggests that creative seeding is a helpful but perhaps unnecessary measure for solving such high-dimensional problems.

It is often difficult to quantitatively evaluate the scalability of the method. Hence,

¹An example of 3D flat-footed walking gait optimization is shown <https://youtu.be/jBVUfDZua4s>.

Table 6.6: Runtime and convergence test of the optimization.

Initial Guess	Bound.	Iteration	Feasibility	Runtime (s)
random	relaxed	856	2.24×10^{-9}	675.49
	regular	1209	7.89×10^{-13}	911.48
	restricted	2811	3.08×10^{-9}	2027.08
good	relaxed	193	4.91×10^{-09}	150.28
	regular	357	3.68×10^{-11}	375.9
	restricted	693	5.21×10^{-13}	620.03

we recap typical performances of all robot model that we have discussed in the study. The results are summarized in Table 6.7¹. In particular, six walking behaviors of five different bipedal robots are included: PROXI, DURUS-2D, DURUS 3D flat-footed walking, DURUS 3D multi-contact walking, full-size DRC-HUBO, and Atlas robot. For each walking behavior of a robot model, the degrees of freedom of the robot model, total numbers of domain, and degrees of underactuation in each discrete domain are given. As a reference, we define 15 total cardinal nodes for each domain, which in turn determines the size of the optimization problem, e.g., the total number of NLP variables and constraints. As expected, the total number of NLP variables and constraints increase as the system dimension and the number of domains increase. Yet the sparsity of the constraints' Jacobian is very sparse.

Typically, the computational performance of a NLP problem is evaluated metrics like total computer time or number of iterations. While these two metrics are often affected by many external conditions, they might not be a good scale to measure how complex a problem is computational-wise. It is often necessary to check how long do the NLP functions require to evaluate in per iteration. Hence, in Table 6.7, we also listed how much CPU time the optimizer requires to evaluate the cost function, constraints, gradients of the cost function, and Jacobian of constraints in per iteration. The evaluation time of these functions, particularly the constraints and Jacobians, is

¹This result is limited to the local direct collocation optimization.

noticeably affected by the complexity of the model.

The typical total CPU time and iterations to converge are evaluated by initial guesses obtained from some previously obtained gait. To add the difficulty to the evaluation, we provide random values as initial guesses for the parameters of virtual constraints. Despite these metrics would vary in practice, we could consider them as a rough reference how the performance of the problem scales up when the size of the problem and the complexity of the system increases.

Table 6.7: Performance summary of the (local) direct collocation HZD optimization on different robot platforms.

	PROXI	DURUS-2D	DURUS Flat ¹	DURUS Multi ²	DRC-HUBO	Atlas
Degrees of freedom ³	7	11	23	23	33	18
Domains	ss	ds, ss	ds, ss	ts, tl, hl, hs	ss	ts, tl, hs
Degrees of underactuation	1	2, 3	1, 2	2, 2, 3, 2	0	0, 0, 0
Number of NLP variables ⁴	1510	3723	8974	21 309	7996	11 811
Number of NLP constraints	1577	4198	9893	22 721	9763	12 151
Sparsity of Jacobian ⁵	0.44%	0.78%	0.13%	0.05%	0.15%	0.09%
Cost function runtime ⁶ (sec)	0.0006	0.0010	0.0009	0.0018	0.0014	0.0013
Cost gradient runtime ⁷ (sec)	0.0006	0.0011	0.0014	0.0020	0.0015	0.0015
Constraints function runtime ⁸ (sec)	0.0104	0.0231	0.0517	0.0671	0.0372	0.1124
Constraints Jacobian runtime ⁹ (sec)	0.0114	0.0571	0.2385	0.3870	0.4271	0.3062
Typical total CPU time (sec)	3	21	146	647	663	282
Typical number of iterations	54	114	226	418	604	321

¹DURUS 3D flat-footed walking gait.

²DURUS 3D multi-contact walking gait.

³It includes floating base coordinates, which has 3 for planar robots and 6 for 3D robots.

⁴The number of NLP variables and constraints depends on the number of nodes defined. Here we list the values assuming that there are 15 cardinal nodes for each domain.

⁵The sparsity of the Jacobian matrix is the ratio of nonzero entries in the whole Jacobian matrix.

⁶This is an average time to run the cost function on a laptop computer.

⁷This is an average time to run the gradient function of the cost function on a laptop computer.

⁸This is an average time to run all constraints function on a laptop computer.

⁹This is an average time to run the Jacobian functions of all constraints on a laptop computer.

CHAPTER VII

CONCLUSION

The work presented in this study is built upon and contributes to the well-established hybrid zero dynamics control framework for bipedal locomotion. We develop a theoretical foundation for solving hybrid zero dynamics based dynamic humanoid walking gait optimization via computationally efficient direct collocation methods. This systematic framework removes the limitations of hybrid zero dynamics application on high dimensional systems, and establishes hybrid zero dynamics based controllers in the analytically tricky domain of humanoid robots.

7.1 Summary of Contributions

The key contributions of the thesis are summarized as follows:

Scalable direct collocation HZD optimization framework. We present a systematic methodology that allows for the fast and reliable generation of multi-domain robotic walking gaits in the context of HZD framework. Specifically, we formulate an optimization framework which builds upon the novel unification of hybrid zero dynamics and direct collocation methods. The direct collocation methods convert the continuous trajectory optimization problem into a large-scale nonlinear programming (NLP) problem by enforcing defect constraints at discretized collocation points. By eschewing the need for time-marching integration schemes (as per to single- and multiple-shooting methods) in favor these local defect constraints, we open the possibility of expressing all optimization constraints in closed-form. Fully algebraic constraint expressions allow for symbolic Jacobians with fast evaluation times, high accuracy and robustness, which is critical for the scalability of the gait optimization.

By necessity, this work takes care in addressing the scalability of constraint expression sizes, which can easily explode to intractable proportions. To understand this complexity, consider symbolically transcribing the NLP constraints for one of the presented humanoids—DURUS. Take just one component of humanoid-scale dynamics; the symbolic inertial matrix alone ($D(q)$) consists of 13 million subexpressions¹, which would then be inverted analytically ($D(q)^{-1}$), and then symbolically differentiated for the Jacobian ($\delta D(q)^{-1}\ddot{q}/\delta q$). So while ostensibly straightforward, such a trivial approach becomes wildly infeasible very quickly. Instead, we present our formulae for systematically introducing *defect variables* into the constraints which avoid symbolically verbose operations, keeping the resulting expressions simple. Further, by carefully indexing the optimization variables and constraints, we further simplify the Jacobian matrix to have a banded structure, enabling efficient evaluation and use by standard large-scale NLP solvers. All such components of this process are crucial in engineering fast and reliable optimization which synthesizes HZD gaits for humanoid robots. Encouragingly, despite the high-dimensionality of the 10,000-variable problem, the optimization even converges with random initial guesses. Building upon the theoretical foundation of HZD, this method optimizes the interactions of the full body dynamics of the robots hybrid system model, without restricting motions to simpler more-tractable dynamics (e.g., inverted pendulum models), resulted in more agile and efficient motions of humanoids.

Experimental realizations of energy efficient 3D humanoid walking with hybrid zero dynamics controllers. The computationally efficient direct HZD optimization enables the hybrid zero dynamics framework can be applied to 3D walking

¹These subexpressions of $D(q)$ (as counted by Mathematica’s `LeafCount`) require over 430 MB of memory to store (via Mathematica’s `ByteCount`). To date, we have still not been able to successfully perform even the symbolic inverse of $D(q)$, which is one of the first steps in explicitly generating symbolic constraints.



Figure 7.1: The competition space inhabited by both DURUS and STEPPR at the DARPA Robot Endurance Test.

of high DOFs humanoids, even in the presence of underactuation. By taking advantages of the natural dynamics—including absorbing impact via passive and compliant elements as well as counter-balancing upper-body and arm motions—we generate optimal energy efficient 3D walking gaits of several humanoids and experimentally demonstrate the efficient locomotion on the hardware platforms.

Specifically, we produce two different walking motions of DURUS: a conserved flat-footed walking and a fast multi-contact walking gait with heel-toe contacts, and first time experimentally realize sustained 3D walking motions on an underactuated humanoid using hybrid zero dynamics controllers. The exceptionally efficient and stable flat-footed locomotion was able to be showcased at the DARPA Robot Endurance Test, completing against the Sandia National Labs robot STEPPR [91] to demonstrate humanoid efficiency available on full-scale disaster relief humanoid robots. DURUS walked for just under five hours in a single day, traveling 3.9 km. Most notably, we report the best energy economy reported for a full-scale humanoid to date [109], and approximately twice as much as efficient than STEPPR robot, which is particularly designed to efficiently walk with the ZMP based controllers.

The same procedure is applied to DRC-HUBO, which has over 27 actuators. As a consequence of optimizing energy consumption, the resulting gaits swing their arms,

not by a user defining swinging motions a priori or superimposing them on gaits *post hoc*, but as an emergent behavior from optimizing the dynamic gait [72]. We also present preliminary dynamic walking experiments with DRC-HUBO in hardware, thereby building a case that hybrid zero dynamics is becoming a viable approach for controlling the full complexity of very high dimensional humanoid locomotion.

Online whole body dynamics gait optimization. We present an optimal gait synthesis method that exploits the whole body dynamics of robots—for the first time—experimentally realizes online HZD gait generation for a planar underactuated robot. Due to the nonlinearity and complexity of the full body dynamics of the robot, gait optimization using the full body dynamics requires excessive amount of time to run and may not be able to converge reliably, and therefore, is limited to off-line *a priori* motion planning only. In this thesis, we improve the reliability of the HZD gait optimization and significantly increase the convergence speed by taking advantage of the pseudospectral optimization. As a result, generating of HZD gaits online becomes feasible with an average computation time less than 0.5 seconds. We experimentally evaluate the performance of this method on a planar underactuated robot which walks on a treadmill with varying speeds. The optimizer successfully generates gaits with different walking speeds online to enable the robot to adjust to the changing speed of the treadmill [67].

7.2 *Perspectives on Future Work*

Exploring the great potentials of the hybrid zero dynamics framework, our method is not limited to dynamic bipedal walking, moreover, is capable of extending to a wide variety of dynamic behaviors. In this section, we exhibit a few extensions of the work and preliminary results of the applications.

Dynamic Running. Bipedal running is an important benchmark for humanoid control for a number of mathematical and practical reasons. Unlike walking, running is

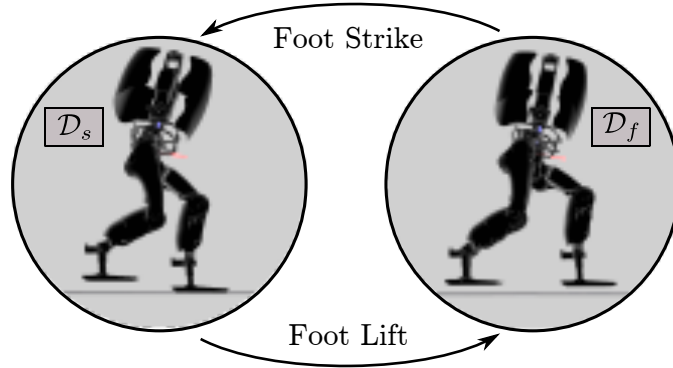


Figure 7.2: Two-domain hybrid system model of dynamic running on DURUS.

an inherently underactuated control problem. Whenever the robot leaves the ground, it fundamentally loses its ability to actuate all of its degrees of freedom, and is at the mercy of its ballistic trajectory. Due to its capability to seamlessly handle underactuation, the hybrid zero dynamics framework has long-held promise of realizing dynamic running of bipedal robots [121, 122, 147].

In [88], our group present a preliminary result of extending the direct collocation HZD optimization method to generate stable dynamic running of 3D DURUS humanoid. Similar to the flat-footed walking of DURUS, the dynamic running behavior is modeled as a two-domain hybrid system, as shown in Figure 7.2. The system has degrees of underactuation of 2 and 5 for the stance and flight domain respectively, resulting in very high dimensional of zero dynamics manifold. By formulating the gait optimization problem in a similar manner with a few necessary modifications, we are able to generate a family of dynamic walking gaits over a range of speeds (1.5–3.0

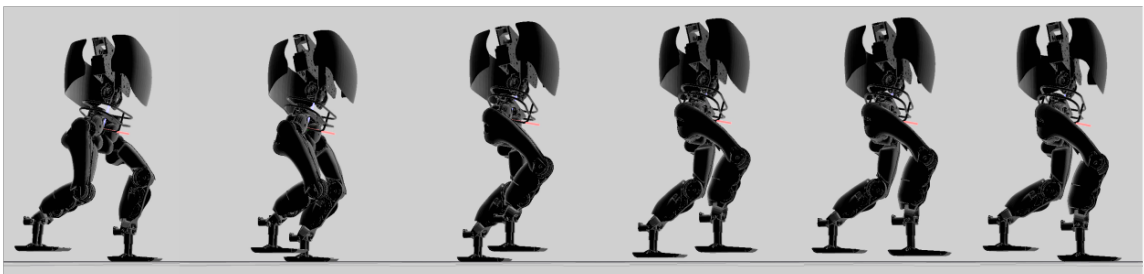


Figure 7.3: Tiled still images the DURUS running at 2.0 m/s .

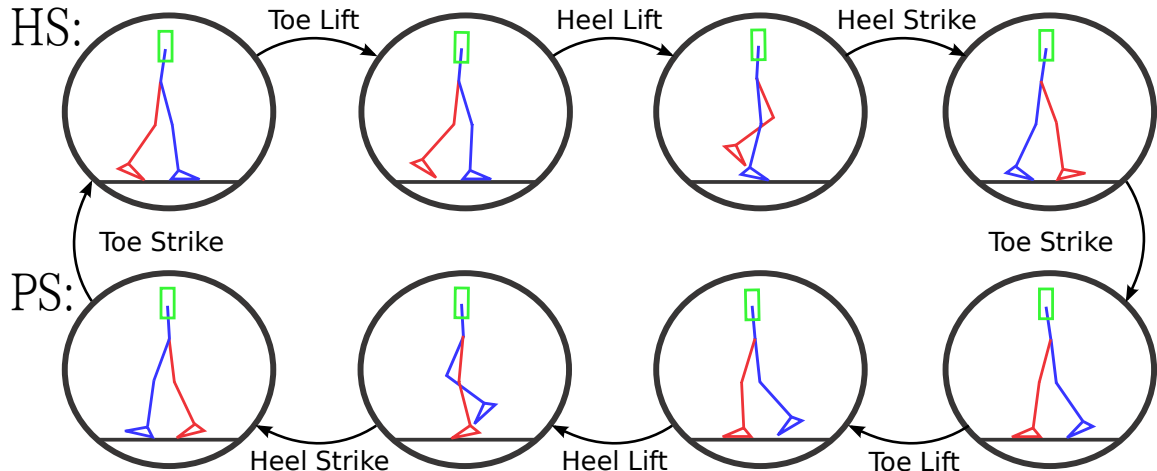


Figure 7.4: Two-step domain graph of the asymmetric amputee- prosthesis gait.

m/s) with an average convergence time of 323 seconds and 609 iterations.

The ability of generating multiple different gaits that respects all physical limitations of the robot within an acceptable time frame places a great expectation on experimentally realizing dynamic walking on actual robots in the next step.

Prosthetic Leg. Prosthetic devices are the most commonly seen daily-life applications of robotic techniques. Amputees with energetically passive prosthetic devices are found to be less stable, constrained with locomotion capabilities and require more force and energy during locomotion than healthy human. Powered-lower-limb prostheses capable of providing net power in conjunction with various prostheses controllers have been developed in recent decades with the potential to regain fully mobility in various terrains of amputees.

Among many other powered prosthetic lower leg application, AMPRO utilized the hybrid zero dynamics framework to realize stable periodic prosthetic walking that is human-like both kinetically and kinematically [143]. Despite the improvements achieved by the framework of virtual constraints, the current researches are still limited with several basic assumptions and open problems: a) forward human walking is simplified as a 2D model, i.e., a simple 2D planar model is usually chosen for generating the desired prosthetic trajectory; b) the amputee-prosthesis system is

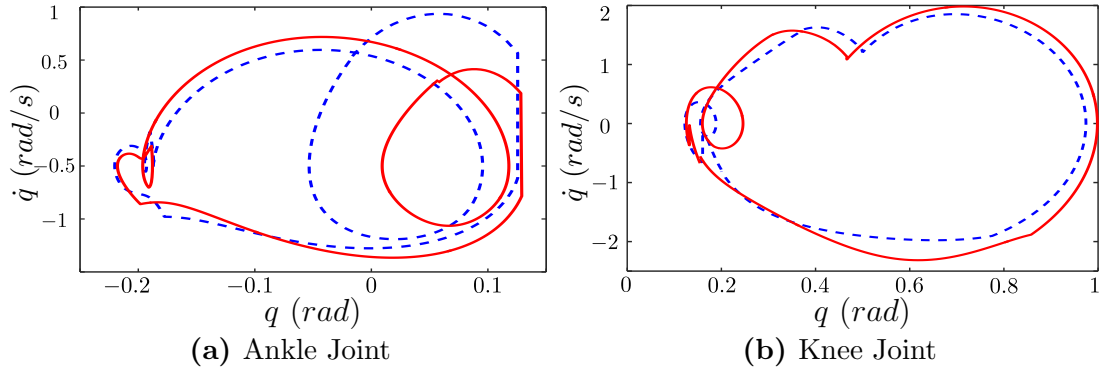


Figure 7.5: Phase portraits of the ankle and knee joints of both the amputee and prosthesis over 20 steps. The solid lines represent the prosthetic joints, and the dashed lines represent the amputee joints.

assumed to be symmetric, i.e., the amputee side and prosthetic side share the same requirements and model parameters; c) realistic requirements (human comfortability, energy consumption, hardware torque and velocity limitations) of a prosthetic gait have not yet been considered intuitively during the gait design procedure.

Based on the work presented in the study, a preliminary result is presented to successfully generate two-step asymmetric 3D human-like multi-contact walking gaits in simulation [145]. Based on the asymmetric amputee-prosthesis model, an eight domain hybrid system model is constructed, as shown in Figure 7.4. A two-step direct collocation virtual constraints optimization is formulated to generate stable walking gaits for the asymmetric system. More importantly, the flexibility of the direct collocation formulation, human-likeness constraints and comfortability constraints can be easily imposed in the nonlinear constrained optimization. The resulting gait exhibits two-step periodic locomotion as shown in Figure 7.5. Because of the general formulation of this work, this two-step modeling and optimization methods can potentially be applied to other asymmetric robot systems.

APPENDIX A

EQUATIONS OF MOTION OF ZERO DYNAMICS

By enforcing virtual constraints via feedback controllers, the full order dynamics of the robotic system is constrained on a reduced dimensional manifold, termed as *zero dynamics*. The low-dimensional representation of the restricted manifold, however, is not trivial to be obtained. In this chapter, we present the derivation of the equations of motion of zero dynamics. Due to the fact that the relative degree 1 output is excluded from the condition of partial zero dynamics manifolds, we determine the governing equation of motion of the reduced dimensional dynamics in two different cases based on whether the velocity-modulating output is defined or not on a given domain.

Before introducing the derivation, we start with formulating the affine control system of the continuous dynamics in (2.20). Instead of computing the constraint wrenches explicitly, we can consider them as parts of (pseudo) control inputs that satisfy the holonomic constraints. Recall that the equations of motion for a domain \mathcal{D}_v is determined by the classical Euler-Lagrange equation and holonomic constraints [58]:

$$D(q)\ddot{q} + H(q, \dot{q}) = B_v u_v + J_v^T(q)\lambda_v, \quad (\text{A.1})$$

$$J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} = 0, \quad (\text{A.2})$$

Then let

$$\bar{B}_v(q) = \begin{bmatrix} B_v & J_v^T(q) \end{bmatrix}, \quad \bar{u}_v = \begin{bmatrix} u_v & \lambda_v \end{bmatrix}^T, \quad (\text{A.3})$$

to yield equations of motion as the affine control system

$$\dot{x} = \bar{f}(x) + \bar{g}_v(x)\bar{u}_v \quad (\text{A.4})$$

from (A.1) with $x = (q, \dot{q})$ the system states, where

$$\bar{f}(x) = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)H(q, \dot{q}) \end{bmatrix}, \quad \bar{g}_v(x) = \begin{bmatrix} \mathbf{0} \\ D^{-1}(q)\bar{B}_v(q) \end{bmatrix}.$$

Case A.1. If the velocity-modulating output is not defined on a domain \mathcal{D}_v , then we have $y_v(q) = y_{2,v}(q)$. Let $z_v(q)$ be a real-valued function representing the local coordinates of the partial zero dynamics \mathcal{PZ}_v , i.e., the coordinates that are neither actuated nor constrained, we have

$$\Phi_v(q) := \begin{bmatrix} y_v(q) \\ h_v(q) \\ z_v(q) \end{bmatrix} : \mathcal{Q} \rightarrow \mathbb{R}^n \quad (\text{A.5})$$

is a diffeomorphism onto its image and there exists at least one point where both the virtual and holonomic constraints vanish, i.e., the *partial zero dynamics* manifold.

We consider $(\xi_{1,v}, \xi_{2,v})$ be the states of the zero dynamics, which are given by

$$\xi_{1,v} = z_v(q), \quad \xi_{2,v} = \gamma_v(q, \dot{q}), \quad (\text{A.6})$$

where

$$\gamma_v(q, \dot{q}) := \gamma_v^0(q)\dot{q}, \quad (\text{A.7})$$

where $\gamma_v^0(q) = \ell_v(q)D(q)$ with $\ell_v(q) \in \text{Null}(\bar{B}_v(q))$. It is obvious to verify that $L_{\bar{g}_v}\gamma_v(q, \dot{q}) = 0$. Hence, the equations of motion of the zero dynamics can be expressed explicitly:

$$\dot{\xi}_{1,v} = L_{\bar{f}}z_v(q, \dot{q}), \quad (\text{A.8})$$

$$\dot{\xi}_{2,v} = L_{\bar{f}}\gamma_v(q, \dot{q}), \quad (\text{A.9})$$

which is independent of the control torque input \bar{u}_v . When the system evolves on the

partial zero dynamics manifold, the full order states can be reconstructed as follows:

$$q = \Phi_v^{-1}(0, 0, \xi_{1,v}), \quad \dot{q} = \begin{bmatrix} \frac{\partial y_v(q)}{\partial q} \\ J_v(q) \\ \frac{\partial \xi_{2,v}(q)}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \xi_{2,v} \end{bmatrix}, \quad (\text{A.10})$$

where $\frac{\partial \xi_{2,v}(q)}{\partial q} = \gamma_v^0(q)$ by its definition in this case.

Case A.2. In the case if a velocity-modulating output is defined on a domain \mathcal{D}_v , we define a diffeomorphism of the robot generalized coordinates via:

$$\Phi_v(q) := \begin{bmatrix} y_{1,v}^a \\ y_{2,v}(q) \\ h_v(q) \\ z_v(q) \end{bmatrix} : \mathcal{Q} \rightarrow \mathbb{R}^n. \quad (\text{A.11})$$

When the system evolves on the partial zero dynamics manifold, both $y_{2,v}(q)$ and $h_v(q)$ vanish. Hence the states of the partial zero dynamics are given as:

$$\xi_{1,v} = \begin{bmatrix} y_{1,v}^a(q) \\ z_v(q) \end{bmatrix}, \quad \xi_{2,v} = \begin{bmatrix} \dot{y}_{1,v}^a(q, \dot{q}) \\ \gamma_v(q, \dot{q}) \end{bmatrix}, \quad (\text{A.12})$$

where $\gamma_v(q, \dot{q})$ is given by (A.7). Consequently, the governing equation of motion of the partial zero dynamics can be obtained straightforwardly:

$$\dot{\xi}_{1,v} = \begin{bmatrix} \dot{y}_{1,v}^a(q, \dot{q}) \\ L_{\bar{f}} z_v(q, \dot{q}) \end{bmatrix}, \quad \dot{\xi}_{2,v} = \begin{bmatrix} -\varepsilon(\dot{y}_{1,v}^a - \bar{v}_v) \\ L_{\bar{f}} \gamma_v(q, \dot{q}) \end{bmatrix}. \quad (\text{A.13})$$

Similarly, the full order states of the system can be re-constructed from the reduced dimensional manifold as in (A.10).

REFERENCES

- [1] “Durus 3d flat-footed walking gait:.” <https://youtu.be/zpWmKQzexSQ>.
- [2] “Durus walks like a human:.” <https://www.youtube.com/watch?v=1fC7b2LjVW4>.
- [3] “Online hybrid zero dynamics optimal gait generation using legendre pseudospectral optimization:.” <https://youtu.be/pvH3c2G2Pj4>.
- [4] ACKERMANN, M. and VAN DEN BOGERT, A. J., “Optimality principles for model-based prediction of human gait,” *Journal of biomechanics*, vol. 43, no. 6, pp. 1055–1060, 2010.
- [5] ALTENDORFER, R., KODITSCHKEK, D. E., and HOLMES, P., “Stability analysis of legged locomotion models by symmetry-factored return maps,” *International Journal of Robotics Research*, vol. 23, no. 10-11, pp. 979–999, 2004.
- [6] ALTENDORFER, R., MOORE, N., KOMSUOGLU, H., BUEHLER, M., BROWN JR, H. B., MCMORDIE, D., SARANLI, U., FULL, R., and KODITSCHKEK, D. E., “RHex: a biologically inspired hexapod runner,” *Autonomous Robots*, vol. 11, no. 3, pp. 207–213, 2001.
- [7] AMES, A. D., “Human-inspired control of bipedal walking robots,” *IEEE Transactions on Automatic Control*, vol. 59, pp. 1115–1130, May 2014.
- [8] AMES, A. D., COUSINEAU, E. A., and POWELL, M. J., “Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics,” in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pp. 135–144, ACM, 2012.
- [9] AMES, A. D., GALLOWAY, K., and GRIZZLE, J. W., “Control Lyapunov functions and hybrid zero dynamics,” in *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 6837–6842, Dec 2012.
- [10] AMES, A. D., GALLOWAY, K., SREENATH, K., and GRIZZLE, J. W., “Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control (TAC)*, vol. 59, pp. 876–891, April 2014.
- [11] AMES, A. D. and POWELL, M. J., “Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs,” in *Control of Cyber-Physical Systems*, pp. 219–240, Springer, 2013.

- [12] AMES, A. D., VASUDEVAN, R., and BAJCSY, R., “Human-data based cost of bipedal robotic walking,” in *Proceedings of the 14th international conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 153–162, ACM, 2011.
- [13] AMES, A. D., “First steps toward automatically generating bipedal robotic walking from human data,” in *Robot Motion and Control 2011*, pp. 89–116, Springer, 2011.
- [14] AMES, A. D. and HOLLEY, J., “Quadratic program based nonlinear embedded control of series elastic actuators,” in *Submitted to the 53rd IEEE Conference on Decision and Control*, 2014.
- [15] ANITESCU, M. and POTRA, F. A., “Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems,” *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.
- [16] ARTSTEIN, Z., “Stabilization with relaxed controls,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.
- [17] ASCHER, U. M., CHIN, H., PETZOLD, L. R., and REICH, S., “Stabilization of constrained mechanical systems with daes and invariant manifolds,” *Mechanics of Structures and Machines*, vol. 23, no. 2, pp. 135–157, 1995.
- [18] BETTS, J. T., “Survey of numerical methods for trajectory optimization,” *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [19] BETTS, J. T., *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19. Siam, 2010.
- [20] BHOUNSULE, P. A., CORTELL, J., GREWAL, A., HENDRIKSEN, B., KARSSSEN, J. D., PAUL, C., and RUINA, A., “Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge,” *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, 2014.
- [21] BIRN-JEFFERY, A., HUBICKI, C., BLUM, Y., RENJEWSKI, D., HURST, J., and DALEY, M., “Don’t break a leg: Running birds from quail to ostrich prioritise leg safety and economy on uneven terrain,” *Journal of Experimental Biology*, vol. 217, no. 21, pp. 3786–3796, 2014.
- [22] BLICKHAN, R., “The spring–mass model for running and hopping,” *Journal of Biomechanics*, vol. 22, no. 11, pp. 1217–1227, 1989.
- [23] BLICKHAN, R. and FULL, R., “Similarity in multilegged locomotion: bouncing like a monopode,” *Journal of Comparative Physiology A*, vol. 173, no. 5, pp. 509–517, 1993.
- [24] BOCK, H. G. and PLITT, K. J., *A multiple shooting algorithm for direct solution of optimal control problems*. Sonderforschungsbereich 72, Approximation u. Optimierung, Univ. Bonn, 1983.

- [25] BRENAN, K. E., CAMPBELL, S. L., and PETZOLD, L. R., *Numerical solution of initial-value problems in differential-algebraic equations*, vol. 14. Siam, 1996.
- [26] BUSS, B. G., RAMEZANI, A., AKBARI HAMED, K., GRIFFIN, B. A., GALLOWAY, K. S., and GRIZZLE, J. W., “Preliminary walking experiments with underactuated 3d bipedal robot marlo,” in *Intelligent Robots and Systems (IROS 2014)*, pp. 2529–2536, IEEE, 2014.
- [27] CANUTO, C., HUSSAINI, M. Y., QUARTERONI, A. M., THOMAS JR, A., and OTHERS, *Spectral methods in fluid dynamics*. Springer Science & Business Media, 2012.
- [28] CAPPELLINI, G., “Motor patterns in human walking and running,” *Journal of Neurophysiology*, vol. 95, pp. 3426–3437, feb 2006.
- [29] CARPENTIER, J., TONNEAU, S., NAVEAU, M., STASSE, O., and MANSARD, N., “A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion,” in *IEEE International Conference on Robotics and Automation (ICRA), May 2016, Stockholm, Sweden*, (Stockholm, Sweden), May 2016.
- [30] CHEVALLEREAU, C., ABBA, G., Aoustin, Y., PLESTAN, F., WESTERVELT, E. R., CANUDAS-DE-WIT, C., and GRIZZLE, J. W., “Rabbit: a testbed for advanced control theory,” *IEEE Control Systems*, vol. 23, pp. 57–79, Oct 2003.
- [31] CHEVALLEREAU, C., GRIZZLE, J., and SHIH, C., “Asymptotically stable walking of a five-link underactuated 3D bipedal robot,” *IEEE Transactions on Robotics*, vol. 25, pp. 37–50, February 2009.
- [32] COLLINS, S., RUINA, A., TEDRAKE, R., and WISSE, M., “Efficient bipedal robots based on passive-dynamic walkers,” *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [33] COLLINS, S. H. and RUINA, A., “A bipedal walking robot with efficient and human-like gait,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1983–1988, April 2005.
- [34] COLLINS, S. H., ADAMCZYK, P. G., and KUO, A. D., “Dynamic arm swinging in human walking,” *Proceedings. Biological sciences / The Royal Society*, vol. 276, no. 1673, pp. 3679–3688, 2009.
- [35] CONWAY, B. A. and PARIS, S. W., “Spacecraft trajectory optimization using direct transcription and nonlinear programming,” in *Spacecraft Trajectory Optimization* (CONWAY, B. A., ed.), pp. 37–78, Cambridge University Press, 2010. Cambridge Books Online.
- [36] COUSINEAU, E. and AMES, A. D., “Realizing underactuated bipedal walking with torque controllers via the ideal model resolved motion method,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5747–5753, May 2015.

- [37] DAI, H., VALENZUELA, A., and TEDRAKE, R., “Whole-body motion planning with centroidal dynamics and full kinematics,” in *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 295–302, IEEE, 2014.
- [38] DANTAM, N., HEREID, A., AMES, A., and STILMAN, M., “Correct software synthesis for stable speed-controlled robotic walking,” in *Proceedings of the 2013 Robotics: Science and Systems Conference IX (RSS)*, pp. 24–28, 2013.
- [39] DE BOOR, C., DE BOOR, C., MATHÉMATICIEN, E.-U., DE BOOR, C., and DE BOOR, C., *A practical guide to splines*, vol. 27. Springer-Verlag New York, 1978.
- [40] DEDONATO, M., DIMITROV, V., DU, R., GIOVACCHINI, R., KNOEDLER, K., LONG, X., POLIDO, F., GENNERT, M. A., PADIR, T., FENG, S., MORIGUCHI, H., WHITMAN, E., XINJILEFU, X., and ATKESON, C. G., “Human-in-the-loop control of a humanoid robot for disaster response: A report from the darpa robotics challenge trials,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015.
- [41] DIEHL, M., BOCK, H. G., DIEDAM, H., and WIEBER, P.-B., “Fast direct multiple shooting algorithms for optimal robot control,” in *Fast Motions in Biomechanics and Robotics*, pp. 65–93, Springer, 2006.
- [42] DIMITROV, D., SHERIKOV, A., and WIEBER, P. B., “A sparse model predictive control formulation for walking motion generation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2292–2299, Sept 2011.
- [43] ELNAGAR, G., KAZEMI, M. A., and RAZZAGHI, M., “The pseudospectral legendre method for discretizing optimal control problems,” *IEEE Transactions on Automatic Control*, vol. 40, pp. 1793–1796, Oct 1995.
- [44] FREEMAN, R. A. and KOKOTOVIC, P. V., *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer, 2008.
- [45] GALLOWAY, K., SREENATH, K., AMES, A. D., and GRIZZLE, J. W., “Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs,” *IEEE Access*, vol. 3, pp. 323–332, 2015.
- [46] GARG, D., *Advances in global pseudospectral methods for optimal control*. PhD thesis, University of Florida, 2011.
- [47] GARG, D., PATTERSON, M. A., HAGER, W. W., RAO, A. V., BENSON, D. A., and HUNTINGTON, G. T., “An overview of three pseudospectral methods for the numerical solution of optimal control problems,” *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 475–487, 2009.

- [48] GATH, P. F. and WELL, K. H., “Trajectory optimization using a combination of direct multiple shooting and collocation,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit. Montreal, Canada. AIAA 2001*, vol. 4047, 2001.
- [49] GEYER, H., SEYFARTH, A., and BLICKHAN, R., “Compliant leg behaviour explains basic dynamics of walking and running,” *Proceedings of the Royal Society B-Biological Sciences*, vol. 273, pp. 2861–2867, NOV 22 2006.
- [50] GILBERT, J. R., MOLER, C., and SCHREIBER, R., “Sparse matrices in matlab: design and implementation,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 333–356, 1992.
- [51] GILL, P. E., MURRAY, W., and SAUNDERS, M. A., “Snopt: An sqp algorithm for large-scale constrained optimization,” *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.
- [52] GLOCKER, C. and PFEIFFER, F., “Dynamical systems with unilateral contacts,” *Nonlinear Dynamics*, vol. 3, no. 4, pp. 245–259, 1992.
- [53] GONG, Q., KANG, W., and ROSS, I. M., “A pseudospectral method for the optimal control of constrained feedback linearizable systems,” *IEEE Transactions on Automatic Control*, vol. 51, pp. 1115–1129, July 2006.
- [54] GOTTLIEB, D. and ORSZAG, S. A., *Numerical analysis of spectral methods: theory and applications*, vol. 26, ch. 7, pp. 79–88. Siam, 1977.
- [55] GRIEWANK, A. and WALTHER, A., *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Siam, 2008.
- [56] GRIFFIN, B. and GRIZZLE, J., “Walking gait optimization for accommodation of unknown terrain height variations,” in *2015 American Control Conference (ACC)*, pp. 4810–4817, July 2015.
- [57] GRIZZLE, J. W., ABBA, G., and PLESTAN, F., “Asymptotically stable walking for biped robots: analysis via systems with impulse effects,” *IEEE Transactions on Automatic Control*, vol. 46, pp. 51–64, Jan 2001.
- [58] GRIZZLE, J. W., CHEVALLEREAU, C., SINNET, R. W., and AMES, A. D., “Models, feedback control, and open problems of 3D bipedal robotic walking,” *Automatica*, vol. 50, no. 8, pp. 1955 – 1988, 2014.
- [59] GRIZZLE, J. W., CHEVALLEREAU, C., AMES, A. D., and SINNET, R. W., “3d bipedal robotic walking: models, feedback control, and open problems,” in *IFAC Symposium on Nonlinear Control Systems*, vol. 2, p. 8, 2010.
- [60] GUCKENHEIMER, J. and JOHNSON, S., “Planar hybrid systems,” in *Hybrid Systems II*, vol. 999 of *Lecture Notes in Computer Science*, pp. 202–225, Springer Berlin Heidelberg, 1995.

- [61] HAMED, K. A. and GRIZZLE, J. W., “Robust event-based stabilization of periodic orbits for hybrid systems: Application to an underactuated 3d bipedal robot,” in *2013 American Control Conference*, pp. 6206–6212, June 2013.
- [62] HARGRAVES, C. R. and PARIS, S. W., “Direct trajectory optimization using nonlinear programming and collocation,” *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.
- [63] HEBERT, P., BAJRACHARYA, M., MA, J., HUDSON, N., AYDEMIR, A., REID, J., BERGH, C., BORDERS, J., FROST, M., HAGMAN, M., LEICHTY, J., BACKES, P., KENNEDY, B., KARPLUS, P., SATZINGER, B., BYL, K., SHANKAR, K., and BURDICK, J., “Mobile manipulation and mobility as manipulation-design and algorithms of robosimian,” *Journal of Field Robotics*, vol. 32, pp. 255–274, mar 2015.
- [64] HEREID, A., HUBICKI, C. M., COUSINEAU, E. A., HURST, J. W., and AMES, A. D., “Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 5734–5740, May 2015.
- [65] HEREID, A., POWELL, M. J., and AMES, A. D., “Embedding of slip dynamics on underactuated bipedal robots through multi-objective quadratic program based control,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 2950–2957, Dec 2014.
- [66] HEREID, A., COUSINEAU, E. A., HUBICKI, C. M., and AMES, A. D., “3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016.
- [67] HEREID, A., KOLATHAYA, S., and AMES, A. D., “Online hybrid zero dynamics optimal gait generation using legendre pseudospectral optimization,” in *Submitted to IEEE Conference on Decision and Control (CDC)*, IEEE, 2016.
- [68] HEREID, A., KOLATHAYA, S., JONES, M. S., VAN WHY, J., HURST, J. W., and AMES, A. D., “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, (New York, NY, USA), pp. 263–272, ACM, 2014.
- [69] HERZOG, A., ROTELLA, N., SCHAAL, S., and RIGHETTI, L., “Trajectory generation for multi-contact momentum control,” in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 874–880, IEEE, 2015.
- [70] HUANG, Q. H. Q., KAJITA, S., KOYACHI, N., KANEKO, K., YOKOI, K., ARAI, H., KOMORIYA, K., and TANIE, K., “A high stability, smooth walking pattern for a biped robot,” *Proceedings 1999 IEEE International Conference on*

Robotics and Automation (Cat. No.99CH36288C), vol. 1, no. May, pp. 65–71, 1999.

- [71] HUBICKI, C., GRIMES, J., JONES, M., RENJEWSKI, D., SPRÖWITZ, A., ABATE, A., and HURST, J., “ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot,” *The International Journal of Robotics Research*, p. 0278364916648388, 2016.
- [72] HUBICKI, C. M., HEREID, A., X., M. G., L., A. T., and AMES, A. D., “Work those arms: Toward dynamic and stable humanoid walking that optimizes full-body motion,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016.
- [73] HUNTINGTON, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems*. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2007.
- [74] HURMUZLU, Y., GÉNOT, F., and BROGLIATO, B., “Modeling, stability and control of biped robots—a general framework,” *Automatica*, vol. 40, no. 10, pp. 1647–1664, 2004.
- [75] HURMUZLU, Y. and MARGHITU, D. B., “Rigid body collisions of planar kinematic chains with multiple contact points,” *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.
- [76] HYON, S.-H. and MITA, T., “Development of a biologically inspired hopping robot-” kenken”,” in *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, vol. 4, pp. 3984–3991, IEEE, 2002.
- [77] JAFARI, R., FLYNN, L. L., HELLUM, A., and MUKHERJEE, R., “Energy-conserving gaits for point-foot planar bipeds: A five-dof case study,” in *ASME 2013 Dynamic Systems and Control Conference*, pp. V001T10A001–V001T10A001, American Society of Mechanical Engineers, 2013.
- [78] JOHNSON, M., SHREWSBURY, B., BERTRAND, S., WU, T., DURAN, D., FLOYD, M., ABELES, P., STEPHEN, D., MERTINS, N., LESMAN, A., CARFF, J., RIFENBURGH, W., KAVETI, P., STRAATMAN, W., SMITH, J., GRIFFIOEN, M., LAYTON, B., DE BOER, T., KOOLEN, T., NEUHAUS, P., and PRATT, J., “Team ihmc’s lessons learned from the darpa robotics challenge,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.
- [79] KAJITA, S., KANEHIRO, F., KANEKO, K., FUJIWARA, K., HARADA, K., YOKOI, K., and HIRUKAWA, H., “Biped walking pattern generation by using preview control of zero-moment point,” in *Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on*, vol. 2, pp. 1620–1626 vol.2, Sept 2003.

- [80] KAJITA, S., KANEHIRO, F., KANEKO, K., YOKOI, K., and HIRUKAWA, H., “The 3d linear inverted pendulum model: a simple modeling for a biped walking pattern generation,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 239–246, 2001.
- [81] KANEKO, K., KANEHIRO, F., MORISAWA, M., MIURA, K., NAKAOKA, S., and KAJITA, S., “Cybernetic human hrp-4c,” in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pp. 7–14, Ieee, dec 2009.
- [82] KOCH, K. H., MOMBAUR, K., and SOUERES, P., “Optimization-based walking generation for humanoid robot,” *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 498–504, 2012.
- [83] KRAFT, D., “On converting optimal control problems into nonlinear programming problems,” in *Computational Mathematical Programming*, pp. 261–280, Springer, 1985.
- [84] KUDRUSS, M., NAVEAU, M., STASSE, O., MANSARD, N., KIRCHES, C., SOUERES, P., and MOMBAUR, K., “Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations,” in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 684–689, Nov 2015.
- [85] KUINDERSMA, S., DEITS, R., FALLON, M., VALENZUELA, A., DAI, H., PERMENTER, F., KOOLEN, T., MARION, P., and TEDRAKE, R., “Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot,” *Auton. Robots*, vol. 40, pp. 429–455, Mar. 2016.
- [86] LANCZOS, C., *The Variational Principles of Mechanics*, vol. 4. Courier Corporation, 1970.
- [87] LENGAGNE, S., VAILLANT, J., YOSHIDA, E., and KHEDDAR, A., “Generation of whole-body optimal dynamic multi-contact motions,” *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.
- [88] MA, W.-L., HEREID, A., HUBICKI, C. M., and AMES, A. D., “Efficient hzd gait generation for three-dimensional underactuated humanoid running,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE/RSJ, 2016.
- [89] MARTIN, A. E., POST, D. C., and SCHMIEDELER, J. P., “Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet,” *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 988–1005, 2014.
- [90] MARTIN, W. C., WU, A., and GEYER, H., “Robust spring mass model running for a physical bipedal robot,” *International Conference on Robotics and Automation*, pp. 6307–6312, 2015.

- [91] MAZUMDAR, A., SPENCER, S., SALTON, J., HOBART, C., LOVE, J., DULLEA, K., KUEHL, M., BLADA, T., QUIGLEY, M., SMITH, J., BERTRAND, S., WU, T., PRATT, J., and BUERGER, S., “Using parallel stiffness to achieve improved locomotive efficiency with the sandia steppr robot,” *IEEE Conference on Robotics and Automation*, pp. 835–841, 2015.
- [92] MORDATCH, I., LOWREY, K., and TODOROV, E., “Ensemble-cio : Full-body dynamic motion planning that transfers to physical humanoids,” in *Robotics and Automation (ICRA), IEEE International Conference on*, 2015.
- [93] MOREAU, J. J., “Quadratic programming in mechanics: Dynamics of one-sided constraints,” *SIAM Journal on Control*, vol. 4, no. 1, pp. 153–158, 1966.
- [94] MORRIS, B. and GRIZZLE, J. W., “A restricted poincare map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 4199–4206, Dec 2005.
- [95] MORRIS, B. and GRIZZLE, J. W., “Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots,” *IEEE Transactions on Automatic Control*, vol. 54, pp. 1751–1764, Aug 2009.
- [96] MURRAY, R. M., LI, Z., SASTRY, S. S., and SASTRY, S. S., *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [97] NAGASAKA, K., INOUE, H., and INABA, M., “Dynamic walking pattern generation for a humanoid robot based on optimal gradient method,” in *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 6, pp. 908–913, 1999.
- [98] PARIS, S., RIEHL, J., and SJAUW, W., “Enhanced procedures for direct trajectory optimization using nonlinear programming and implicit integration,” in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pp. 21–24, 2006.
- [99] PARK, H., SREENATH, K., RAMEZANI, A., and GRIZZLE, J. W., “Switching control design for accommodating large step-down disturbances in bipedal robot walking,” in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, pp. 45–50, Ieee, may 2012.
- [100] PARK, I. W., KIM, J. Y., LEE, J., and OH, J. H., “Online free walking trajectory generation for biped humanoid robot khr-3(hubo),” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. May, pp. 1231–1236, 2006.
- [101] PARK, J., “Synthesis of natural arm swing motion in human bipedal walking,” *Journal of Biomechanics*, vol. 41, no. 7, pp. 1417–1426, 2008.

- [102] POSA, M., KUINDERSMA, S., and TEDRAKE, R., “Optimization and stabilization of trajectories for constrained dynamical systems,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1366–1373, May 2016.
- [103] POSA, M., CANTU, C., and TEDRAKE, R., “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [104] POWELL, M. J., HEREID, A., and AMES, A. D., “Speed regulation in 3d robotic walking through motion transitions between human-inspired partial hybrid zero dynamics,” in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4803–4810, May 2013.
- [105] PRATT, J., KOOLEN, T., DE BOER, T., REBULA, J., COTTON, S., CARFF, J., JOHNSON, M., and NEUHAUS, P., “Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid,” *The International Journal of Robotics Research*, vol. 31, pp. 1117–1133, Aug. 2012.
- [106] RAIBERT, M. H. and OTHERS, *Legged robots that balance*, vol. 3. MIT press Cambridge, MA, 1986.
- [107] RAMEZANI, A., HURST, J. W., HAMED, K. A., and GRIZZLE, J. W., “Performance analysis and feedback control of ATRIAS, A three-dimensional bipedal robot,” *Journal of Dynamic Systems Measurement and Control*, vol. 136, Mar 2014.
- [108] RAO, A. V., “A survey of numerical methods for optimal control,” *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.
- [109] REHER, J., COUSINEAU, E. A., HEREID, A., HUBICKI, C. M., and AMES, A. D., “Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [110] REZAZADEH, S., HUBICKI, C., JONES, M., PEEKEMA, A., VAN WHY, J., ABATE, A., and HURST, J., “Spring-mass walking with atrias in 3d: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot,” in *ASME 2015 Dynamic Systems and Control Conference*, pp. V001T04A003–V001T04A003, American Society of Mechanical Engineers, 2015.
- [111] ROSS, I. M. and FAHROO, F., “A unified computational framework for real-time optimal control,” in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2210–2215 Vol.3, Dec 2003.
- [112] RUMMEL, J., BLUM, Y., MAUS, H. M., RODE, C., and SEYFARTH, A., “Stable and robust walking with compliant legs,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 5250–5255, May 2010.

- [113] RUMMEL, J., BLUM, Y., and SEYFARTH, A., “Robust and efficient walking with spring-like legs,” *Bioinspiration & biomimetics*, vol. 5, no. 4, p. 046004, 2010.
- [114] SAKAGAMI, Y., WATANABE, R., AOYAMA, C., MATSUNAGA, S., HIGAKI, N., and FUJIMURA, K., “The intelligent asimo: system overview and integration,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2478–2483 vol.3, 2002.
- [115] SASTRY, S., *Nonlinear systems: analysis, stability, and control*, vol. 10. Springer New York, 1999.
- [116] SHIH, C., GRIZZLE, J., and CHEVALLEREAU, C., “Asymptotically stable walking of a simple underactuated 3D bipedal robot,” in *The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, (Taipei, Taiwan), pp. 2766–2771, Novembre 2007.
- [117] SINNET, R. W., POWELL, M. J., SHAH, R. P., and AMES, A. D., “A human-inspired hybrid control approach to bipedal robotic walking,” in *18th IFAC World Congress*, pp. 6904–11, 2011.
- [118] SONTAG, E. D., “A lyapunov-like characterization of asymptotic controllability,” *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, 1983.
- [119] SONTAG, E. D., “A universal construction of artstein’s theorem on nonlinear stabilization,” *Systems & control letters*, vol. 13, no. 2, pp. 117–123, 1989.
- [120] SREENATH, K., PARK, H., POULAKAKIS, I., and GRIZZLE, J. W., “A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL,” *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.
- [121] SREENATH, K., PARK, H.-W., POULAKAKIS, I., and GRIZZLE, J., “Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel,” *The International Journal of Robotics Research*, vol. 32, pp. 324–345, mar 2013.
- [122] SREENATH, K., *Feedback Control of a Bipedal Walker and Runner with Compliance*. PhD thesis, University of Michigan, 2011.
- [123] SRINIVASAN, M. and RUINA, A., “Computer optimization of a minimal biped model discovers walking and running,” *Nature*, vol. 439, no. 7072, pp. 72–75, 2006.
- [124] STEPHENS, B. J. and ATKESON, C. G., “Dynamic balance force control for compliant humanoid robots,” *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1248–1255, Oct. 2010.

- [125] STEWART, D. E. and TRINKLE, J. C., “An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulumb friction,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.
- [126] STOER, J. and BULIRSCH, R., *Introduction to numerical analysis*. Texts in applied mathematics, New York: Springer, 3 ed., 2002.
- [127] VASUDEVAN, R., AMES, A., and BAJCSY, R., “Persistent homology for automatic determination of human-data based cost of bipedal walking,” *Nonlinear Analysis: Hybrid Systems*, vol. 7, no. 1, pp. 101–115, 2013.
- [128] VON STRYK, O., *Numerical solution of optimal control problems by direct collocation*. Springer, 1993.
- [129] VON STRYK, O. and BULIRSCH, R., “Direct and indirect methods for trajectory optimization,” *Annals of operations research*, vol. 37, no. 1, pp. 357–373, 1992.
- [130] VUKOBRATOVIC, M. and JURICIC, D., “Contribution to the synthesis of biped gait,” *Biomedical Engineering, IEEE Transactions on*, no. 1, pp. 2–7, 1969.
- [131] VUKOBRATOVIĆ, M. and BOROVIĆ, B., “Zero-moment point: thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.
- [132] WÄCHTER, A. and BIEGLER, T. L., “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2005.
- [133] WANNER, G. and HAIRER, E., *Solving ordinary differential equations II: Stiff and Differential-Algebraic Problems*, vol. 14. Springer-Verlag Berlin Heidelberg, 2 ed., 1996.
- [134] WENSING, P. M. and ORIN, D. E., “High-speed humanoid running through control with a 3d-slip model,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5134–5140, Nov. 2013.
- [135] WESTERVELT, E. R., GRIZZLE, J. W., CHEVALLEREAU, C., CHOI, J. H., and MORRIS, B., *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.
- [136] WESTERVELT, E. R., GRIZZLE, J. W., and KODITSCHKE, D. E., “Hybrid zero dynamics of planar biped walkers,” *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.
- [137] WESTERVELT, E. R., MORRIS, B., and FARRELL, K. D., “Analysis results and tools for the control of planar bipedal gaits using hybrid zero dynamics,” *Autonomous Robots*, vol. 23, no. 2, pp. 131–145, 2007.

- [138] WILLIAMS, P. and TRIVAILO, P., “Optimal parameter estimation of dynamical systems using direct transcription methods,” *Inverse Problems in Science and Engineering*, vol. 13, no. 4, pp. 377–409, 2005.
- [139] XI, W., YESILEVSKIY, Y., and REMY, C. D., “Selecting gaits for economical locomotion of legged robots,” *The International Journal of Robotics Research*, 2015.
- [140] YADUKUMAR, S. N., PASUPULETI, M., and AMES, A. D., “From formal methods to algorithmic implementation of human inspired control on bipedal robots,” in *Algorithmic Foundations of Robotics X*, pp. 511–526, Springer, 2013.
- [141] YAMAGUCHI, J., INOUE, S., NISHINO, D., and TAKANISHI, A., “Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk,” in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pp. 96–101, IEEE, 1998.
- [142] YOSHIDA, E., ESTEVES, C., BELOUSOV, I., LAUMOND, J.-P., SAKAGUCHI, T., and YOKOI, K., “Planning 3-d collision-free dynamic robotic motion through iterative reshaping,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1186–1198, 2008.
- [143] ZHAO, H., HORN, J., REHER, J., PAREDES, V., and AMES, A. D., “A hybrid systems and optimization-based control approach to realizing multi-contact locomotion on transfemoral prostheses,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 1607–1612, Dec 2015.
- [144] ZHAO, H. H., MA, W. L., AMES, A. D., and ZEAGLER, M. B., “Human-inspired multi-contact locomotion with amber2,” in *Cyber-Physical Systems (ICCPs), 2014 ACM/IEEE International Conference on*, pp. 199–210, April 2014.
- [145] ZHAO, H., HEREID, A., AMBROSE, E., and AMES, A. D., “3d multi-contact gait design for prostheses: Hybrid system models, virtual constraints and two-step direct collocation,” in *Submitted to IEEE Conference on Decision and Control (CDC)*, IEEE, 2016.
- [146] ZHAO, H., HEREID, A., MA, W.-L., and AMES, A. D., “Multi-contact bipedal robotic locomotion,” *Robotica*, vol. FirstView, pp. 1–35, 4 2016.
- [147] ZHAO, H., YADUKUMAR, S. N., and AMES, A. D., “Bipedal robotic running with partial hybrid zero dynamics and human-inspired optimization,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1821–1827, IEEE, 2012.
- [148] ZUCKER, M., JOO, S., GREY, M. X., RASMUSSEN, C., HUANG, E., STILMAN, M., and BOBICK, A., “A general-purpose system for teleoperation of the drc-hubo humanoid robot,” *Journal of Field Robotics*, vol. 32, pp. 336–351, may 2015.