

Dynamic locomotion with a Hexapod Robot

by

Uluç Saranlı

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2002

Doctoral Committee:

Professor Daniel E. Koditschek, Chair

Professor Anthony M. Bloch

Professor Martin Buehler

Professor Jessy W. Grizzle

Professor John F. Laird

ABSTRACT

Dynamic locomotion with a Hexapod Robot

by

Uluç Saranlı

Chair: Daniel E. Koditschek

Legged vehicles offer superior mobility over natural terrain compared to traditional mobile platforms. Furthermore, their structural flexibility admits greater versatility in functionality. This thesis concerns the development of dynamically capable controllers for a hexapedal robot in order to achieve fast, agile and efficient locomotion.

The first contribution towards this end is the design and construction of a hexapedal robot, RHex. The experimental results we present establish RHex as the first power autonomous robot to achieve speeds exceeding one body length per second over terrain approaching the complexity and diversity of the natural landscape. The combination of simple open-loop control algorithms and RHex's morphology exploits mechanical feedback to yield surprising energetic performance and robustness. Furthermore, the versatility of the design becomes evident in additional behaviors that we present, including turning and dynamical back-flips.

The second contribution of this thesis is the development of high bandwidth state feedback controller for hexapedal locomotion. This approach lies in extreme opposition to our open-loop controllers and is inspired by the dynamical nature of running in animals. In particular, research in biomechanics demonstrates the descriptive utility

of simple spring mass models across a large range of sizes and morphologies. Consequently, we adopt the well studied Spring-Loaded Inverted Pendulum(SLIP) model as a literal control target for a hexapedal alternating tripod gait. Now the design effort focuses on speed and agility, with the possibility that such high bandwidth sensor feedback may be rather costly to implement regarding both platform resources as well as runtime efficiency.

Specifically, we introduce the idea of *template based control*, wherein we attempt to actively tune the natural dynamics of the robot to mimic those of SLIP. We use existing gait level SLIP controllers to achieve stable locomotion in simulation, with a simple and intuitive regulatory interface. Despite realistic actuation constraints, we identify stable limit cycles with large basins of attraction, significantly increasing the promise for an experimental implementation on RHex. Even though such an implementation is outside the scope of the thesis, only a few practical issues remain to be resolved before it becomes a reality.

© Uluc Saranlı 2002
All Rights Reserved

To my parents

ACKNOWLEDGEMENTS

I thank my advisor Dan Koditschek for helping me discover and grow the creativity and enthusiasm that I did not know I had. Throughout six years, it has been his vision, knowledge, creativity and energy which carried me forward to this day.

This work would not have been possible if it were not for the collaboration of many individuals that I have had the privilege to work with. I thank Bill Schwind for all his insight and help throughout the first years of my studies. I thank John Guckenheimer for his guidance and expertise in the small but invaluable time I have spent in Cornell, leading to the creation of SimSect. I thank Richard Altendorfer for helping me complete the final missing components of my research and for his colorful company on countless lunch hours. I also thank Phil Holmes and Bob Full for our few but unique conversations, where I have always been amazed and inspired by their expertise and endless sources of new ideas.

The creation of RHex has definitely been a turning point in my research. I owe many thanks to Martin Buehler, whose ingenuity and ideas made RHex possible. I am thankful for the privilege and pleasure to have worked with him and learn from the best, how to combine engineering and science. RHex would also not have been possible without the help and expertise of his students in the Ambulatory Robotics Laboratory. I thank Liana Mitrea for her help in the initial construction as well as Martin de Lasa, Geoff Hawker, Shervin Talebi and Sami Obaid for teaching me how to build real robots. I also thank all the past and present members of the RHex project for bringing the project its current success with their ideas, expertise and

creativity. It has been a great pleasure to work with Ned Moore, Don Campbell, Dave McMordie, Felix Grimminger, Matt Smith, Chris Prahacs and many others. I also thank Al Rizzi, Jay Gowdy, Ben Brown and all other CMU members of our project for the privilege to work with such experts in our field.

Perhaps the most rewarding aspect of my studies was the opportunity to work with the truly amazing group of people in Dan's lab. I thank Eric Klavins, one of the most creative and independent minds I know, for teaching me how to do proper software design in our quest for RHexLib's creation; Noah Cowan for his incredible technical prowess and patience in improving my juggling and foosball skills; Haldun Komsuoglu for his undying enthusiasm and infinite stream of ideas; Rick Groff and his ever growing links page for providing the answer to the life, universe and everything and Greg Sharp for many interesting conversations and his help on the RHex vision tracking system.

Outside the laboratory, many people directly or indirectly contributed to my completion of this dissertation. I thank my brother for his guidance; he has always been the first to take the next step and give me the courage to proceed. I thank the University of Michigan Ballroom Dance Team and all of its members for creating the wonderful social setting in which I have discovered my passion for dancing and was able to preserve my sanity. I thank TUBITAK, the Scientific and Technical Research Council of Turkey for their financial support over the first year of my studies. Most importantly, however, I thank my dance partner and beloved girlfriend Marie Eguchi, for her endless love, understanding and support throughout all the years that we have been together.

This thesis is dedicated to my parents, Tanay and Türel Saranlı, whose undying love supported me through long years of separation, and gave me the energy and motivation to continue.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF APPENDICES	xi
NOTATION	xii
CHAPTERS	
1 Introduction	1
1.1 Legged Vehicles: Motivations and Challenges	1
1.2 “Functional Biomimesis” for Legged Vehicle Design	4
1.3 The Spring-Loaded Inverted Pendulum	5
1.4 Contributions	7
2 RHex: A Compliant Hexapod Robot	11
2.1 Motivation	11
2.2 Design Concept and Morphology	12
2.3 Open Loop Control of Locomotion	14
2.3.1 Running	15
2.3.2 Turning	15
2.4 Experimental Platform	16
2.4.1 Hardware Description	16
2.4.2 Visual Measurement Apparatus	18
2.4.3 Experimental Performance	18
2.5 Conclusion	25
3 Backflip Controllers for RHex	28
3.1 Motivation	28
3.2 Flipping RHex	30

3.2.1	Simple Open Loop Control	30
3.2.2	Assumptions and Control Design Constraints	32
3.3	The Planar Flipping Model	33
3.3.1	A Generic Planar Model	33
3.3.2	Contact States and Constraints	35
3.3.3	The 1DOF Planar Model	36
3.3.4	Continuous Dynamics	37
3.3.5	Hybrid Leg Contacts	40
3.3.6	Leg-Ground Collisions	42
3.3.7	Body Contact Forces During Collision	44
3.4	An Improved Controller	46
3.4.1	Constraints on the Control Inputs	46
3.4.2	Maximal Thrust Control	47
3.4.3	Hybrid Energy Pumping	48
3.5	Simulations	48
3.5.1	Apex Return Maps	49
3.5.2	Maximal Thrust vs PD Control	51
3.6	Conclusion	51
4	The Basic Planar Spring-Loaded Inverted Pendulum	54
4.1	Motivation	54
4.2	The Basic SLIP	56
4.2.1	System Model and Assumptions	56
4.2.2	Parametrization of Equilibrium Gaits	57
4.3	SLIP Controllers	59
4.3.1	The Control Objective	59
4.3.2	The Approximate Deadbeat Controller	59
4.3.3	The Modified Raibert Controller and Integral Compensation	60
4.3.4	Simulation Studies	61
4.4	Template Control Example: Planar 4DOF Leg	63
4.4.1	The AKH Leg Model	63
4.4.2	Embedding the SLIP Template	64
4.4.3	Simulation Studies	66
4.5	Conclusion	67
5	Template Control of Planar Hexapedal Running	69
5.1	Motivation	69
5.2	The Bipedal SLIP Template	71
5.2.1	System Model and Dynamics	71
5.3	A Bipedal SLIP Controller	74
5.3.1	The Structure of the Controller	75
5.3.2	BSLIP Control Inputs	76
5.3.3	An Approximate Apex Return Map	77
5.3.4	The Bipedal Deadbeat Gait Controller	81

5.3.5	Coordination of Swing and Stance Legs	84
5.3.6	Stability with Limited Precompression	87
5.3.7	Basins of Attraction	91
5.4	Slimpod: A Planar Hexapod Model	94
5.4.1	System Model and Assumptions	94
5.4.2	The Hybrid State Space and Related Notation	95
5.4.3	Polar Leg Coordinates and Leg Forces	96
5.4.4	The Continuous Dynamics	97
5.4.5	Mode Transitions	98
5.5	Template Control of Slimpod Locomotion	99
5.5.1	Structure of the Embedding Controller	99
5.5.2	Imposing the Alternating Tripod Gait	99
5.5.3	Virtual Toe Coordinates and Foot Placement	101
5.5.4	Coordination and Control of the Recirculating Tripod	102
5.5.5	Control of the Idle Tripod	104
5.5.6	Dynamics in the Virtual Toe Coordinates	105
5.5.7	Control of the Active Tripod	106
5.6	Conclusion	109
6	Running with a Spatial Hexapod	111
6.1	Motivation	111
6.2	The Compliant Hexapod Model	112
6.2.1	System Model and Assumptions	112
6.2.2	The Continuous Dynamics	114
6.3	Spatial Extensions to the BSLIP Template	115
6.4	Template Control of Hexapod Locomotion	117
6.4.1	Virtual Leg Coordinates and Euler Angle Definitions	117
6.4.2	Dynamics in Virtual Leg Coordinates	118
6.4.3	Structure of the New Spatial Controller	119
6.4.4	Control of the Active Tripod	120
6.5	Simulation Studies	123
6.5.1	Numerical Environment and Model Parameters	123
6.5.2	Presence and Character of Stable Orbits	126
6.5.3	Stability and Basins of Attraction	128
6.6	Conclusion	131
7	Conclusion	135
	APPENDICES	139
	BIBLIOGRAPHY	179

LIST OF TABLES

Table

2.1	Failure statistics on RHex's running experiments	20
2.2	Controller parameters for turning at different speeds	21
3.1	Success rates of the first generation flipping controller for 10 experiments each on different surfaces.	32
3.2	RHex's Kinematic and dynamic parameters.	36
4.1	Structural simulation parameters for human-like and kangaroo-like four degree of freedom legs	67
5.1	Bipedal SLIP threshold functions for each possible mode transition. . .	74
5.2	Structural and controller parameters for all BSLIP simulations	88
6.1	Leg attachment points for the spatial hexapod	124
6.2	Dynamical parameters of the spatial hexapod model	124
6.3	Leg model parameters for the spatial hexapod	125

LIST OF FIGURES

Figure		
2.1	The RHex experimental platform	11
2.2	The compliant hexapod design	13
2.3	Open-loop motion profiles for the left and right tripods	16
2.4	RHex’s turning yaw rate as a function of forward velocity	20
2.5	Sagittal plane data during simple obstacle crossing	22
2.6	Sagittal plane data during composite obstacle crossing	23
2.7	Scale drawing of RHex and the random obstacle course	24
2.8	The two fastest (top) and the two slowest (bottom) robot trajectories in the horizontal plane as RHex moves over the rough surface	25
2.9	Comparison of RHex’s average forward velocity and energetics for all running experiments	26
2.10	Sample profiles of row 6 (columns 8,9 and 10) of the rough surface with RHex	27
3.1	Sequence of states for the flipping controller	31
3.2	Simple rigid planar model of RHex	34
3.3	Hybrid regions in the planar flipping model in (a) the $b_z - \alpha$ plane (b) polar coordinates with angle α and radius b_z using RHex’s morphology	36
3.4	Free body diagrams for the body and one of the legs	38
3.5	Impulsive contact forces during leg-ground collision	45
3.6	An example simulation with the flipping model using RHex-like pa- rameters	49
3.7	Return maps under the maximal thrust feedback controller, numer- ically computed using the planar model with representative surface parameter settings	50
3.8	Outcomes for flipping attempts with lower coefficient of restitution, $k_c = 0.75$	52
3.9	Outcomes for flipping attempts with higher coefficient of restitution, $k_c = 0.9$	53
4.1	The basic Spring Loaded Inverted Pendulum(SLIP) leg model	55

4.2	Sample runs of the deadbeat and modified Raibert controllers applied to the basic SLIP leg for step and sinusoid references	61
4.3	Simulation results for step and sinusoid references	63
4.4	The Ankle-Knee-Hip(AKH) leg model	64
4.5	Tracking performance of the deadbeat and modified Raibert controllers under a step reference input	67
4.6	Tracking performance of the deadbeat and modified Raibert controllers under a sinusoid reference input	68
5.1	The Bipedal Spring-Loaded Inverted Pendulum(BSLIP) Model	70
5.2	Mode transitions in the Bipedal SLIP model	74
5.3	The finite state machine governing leg alternation in BSLIP	75
5.4	Evolution of the stride phase values for left and right legs of the bipedal SLIP, together with an illustration of the limit cycle on the torus	86
5.5	Stable reachable goal regions for the Bipedal SLIP model for different precompression limits	89
5.6	A sample period BSLIP run illustrating period two behavior	91
5.7	Goal points within the reachable apex goal set of the Bipedal SLIP	92
5.8	Basins of attraction of the Bipedal SLIP with the deadbeat controller for six different apex goals	93
5.9	Slimpod: The planar compliant hexapod model	94
5.10	Touchdown kinematics of the recirculating tripod of the Slimpod model	103
6.1	The compliant hexapod model	112
6.2	Horizontal plane extensions to the Bipedal SLIP template	116
6.3	The virtual foot coordinates for the spatial hexapod model	118
6.4	Numbering convention for the spatial hexapod legs	123
6.5	Illustration of the attracting limit cycle for an example run with the spatial hexapod for $\dot{b}_y^* = 1.6m/s$	127
6.6	Yaw, pitch and roll trajectories over time for an example run with the spatial hexapod for $\dot{b}_y^* = 1.6m/s$	128
6.7	Projection of the basin of attraction of the hexapedal controller for 6 different speed goals onto BSLIP apex coordinates	132
6.8	Projection of the basin of attraction of the hexapedal controller for 6 different speed goals onto pitch and roll coordinates	133
6.9	Projection of the basin of attraction of the hexapedal controller for 6 different speed goals onto horizontal plane velocity coordinates	134
A.1	Height distribution over the rough surface	144
F.1	All of the three different possible arrangements for the allowable torque space, the affine embedding subspace \mathcal{T}_s and the desired solution $\boldsymbol{\tau}^*$	166
F.2	All of the three different possible arrangements for the allowable torque space, the affine embedding subspace \mathcal{T}_s and the desired solution $\boldsymbol{\tau}^*$	167
G.1	The life cycle of a Module in RHexLib	171

LIST OF APPENDICES

APPENDIX

A	Details of the RHex Experiments, Experimental Setup and Failure Modes	140
B	Derivation of the Deadbeat Control Laws for the Basic SLIP	145
C	Preliminary Analysis for the Template Based Control of the Ankle-Knee-Hip Model	149
D	Existence and Uniqueness of the Maximal Contact State for the Flipping Model	155
E	Derivation and Analysis of the Jacobians for the Planar and Spatial Hexapod Models	158
F	Projections onto the Limited Torque Space	164
G	RHexLib: A Real-Time Control Software Library	169

NOTATION

Basic, bipedal and spatial SLIP and AKH models

Spaces and sets

$\mathcal{P}_a, \mathcal{P}_t$ apex and touchdown gait parametrization spaces

$\mathcal{P}_b, \mathcal{P}_l$ bottom and liftoff gait parametrization spaces

$\mathcal{X}_a, \mathcal{X}_l$ apex and liftoff state spaces

\mathcal{U} basic SLIP control input space

\mathcal{U}_b BSLIP control input space

\mathcal{M}_b set of BSLIP hybrid modes

\mathcal{Q}_Ω BSLIP state space within mode $\Omega \in \mathcal{M}_b$

\mathcal{Q} hybrid BSLIP state space

\mathcal{L}_i state spaces for individual BSLIP legs

States and various coordinates

\mathbf{b} cartesian position of the body, $\mathbf{b} := [b_y, b_z]^T$

ξ, ψ basic SLIP leg length and angle

ξ_i, ψ_i BSLIP leg lengths and angles

ξ_r, ψ_r states of recirculating BSLIP legs

ξ_s, ψ_s states of BSLIP legs in stance

f_i BSLIP toe positions

Model parameters

m	body mass
$U(\xi)$	radial leg spring potential law
$U_A(\xi)$	potential law for the air spring
$U_g(\xi, \psi)$	total potential energy with gravity
$\widehat{U}_g(\xi)$	linearized gravity approximation to the total potential
k, ξ_0	leg spring constant and rest length
d_s	BSLIP viscous leg damping coefficient

Control system definitions

ψ_t	touchdown angle command
ψ_n	neutral touchdown angle
k_c, k_d	compression and decompression spring constants for basic SLIP
ξ_t, ξ_l	touchdown and liftoff leg lengths for BSLIP
X_a^*	apex state goal
f_a	apex return map; $f_a : \mathcal{X}_a \times \mathcal{U}_b \mapsto \mathcal{X}_a$
\widehat{f}_a	approximate apex return map; $\widehat{f}_a : \mathcal{X}_a \times \mathcal{U}_b \mapsto \mathcal{X}_a$
g	reference governor function; $g : \mathcal{X}_a \times \mathcal{U}_b \mapsto \mathcal{U}_b$
v_i, w_i	radial and angular actuations for recirculating legs of BSLIP
Φ_i	stride phase map for BSLIP legs; $\Phi_i : \mathcal{Q} \mapsto S^1$
Δ_ψ	constraint margin on the touchdown angle command
Δ_z	constraint margin on the target apex height
K_ψ, K_ξ	feedback gains for the control of BSLIP leg recirculation
k_y	modified Raibert controller forward speed gain

The planar flipping model

Spaces and sets

- \mathcal{C} configuration space of the constrained template
- \mathcal{Q} state space of the constrained template
- $\mathcal{H}_b, \mathcal{H}_t$ hybrid contact state spaces for the body and the toes
- $\mathcal{T}(\mathbf{q}, p)$ allowable torque space under physical and practical constraints

States and various coordinates

- \mathbf{b} cartesian position of the body, $\mathbf{b} := [b_y, b_z]^T$
- α body pitch
- \mathbf{q} state of constrained template system; $\mathbf{q} := [\alpha, \dot{\alpha}]^T$
- ϕ_i leg angles at the hips with respect to the body
- γ_i leg angle with respect to the ground
- y_i positions of the toes
- s_n, s_t, s_i contact states of the nose, the tail and the legs

Model parameters

- d, d_i, l dimensions of the body and the legs
- m, I body mass and inertia
- m_t toe mass
- μ coulomb friction coefficient for the toes
- k_d viscous friction coefficient for the

Control system definitions

- τ_i torque commands at the hips
- γ_c toe angle at the time of leg ground collision
- $k_r(\gamma_c)$ leg coefficient of restitution

The planar Slimpod and spatial hexapod models

Coordinate frames, spaces and sets

\mathcal{W} inertial world reference frame

\mathcal{V} inertial virtual foot reference frame

\mathcal{B} body reference frame

\mathcal{M}_h set of modes for the Slimpod or the hexapod

$\mathcal{I}_1, \mathcal{I}_2$ leg index sets for the left and right tripods

$\mathcal{I}_a, \mathcal{I}_i, \mathcal{I}_r$ leg index sets for the active, idle and recirculation tripods

States and various coordinates

\mathbf{b} cartesian position of the body, $\mathbf{b} := [b_x, b_y, b_z]^T$

γ, α, β body yaw, body pitch and body roll

${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$ rotation matrix from \mathcal{B} to \mathcal{W}

\mathbf{f}_i positions of the feet in \mathcal{V}

ϕ_i hip angles around the body x axis

θ_i sideways leg sprawl around the body y axis

ρ_i lengths of the legs

s_i binary contact states of the toes. $s_i \in \{0, 1\}$

\mathbf{l}_i cartesian coordinates of leg vectors, expressed in \mathcal{B}

$\bar{\mathbf{l}}_i$ polar coordinates of leg vectors, expressed in \mathcal{B}

Model parameters

- m, m_t mass of the body and the toes
- I body inertia for the Slimpod
- \mathbf{I}_0 spatial hexapod body inertia expressed in \mathcal{B}
- \mathbf{a}_i leg attachment points at the hips, expressed in \mathcal{B}
- $V_i(\rho_i)$ radial spring potential for the Slimpod legs
- $V_{\rho_i}(\rho_i)$ radial spring potential for the spatial hexapod legs
- $V_{\theta_i}(\theta_i)$ lateral torsional spring potential for the spatial hexapod legs
- ρ_0 rest length of radial leg springs
- θ_0 rest angle of lateral torsional leg springs
- d_i radial damping constant for the Slimpod legs
- d_{ρ_i}, d_{θ_i} radial and lateral damping constants for the spatial hexapod legs.

Control system definitions

- τ_i individual torque commands at the hips
- $\boldsymbol{\tau}$ hip torque command vector
- F_{r_i} individual radial leg spring forces
- \mathbf{F}_r radial leg spring force vector
- τ_{θ_i} individual lateral leg spring torques
- $\boldsymbol{\tau}_\theta$ lateral leg spring torque vector

CHAPTER 1

Introduction

This thesis concerns the design of dynamically capable hexapedal running controllers as well as the development of RHex, a hexapedal robot that is capable of locomoting over terrain approaching the complexity of the natural landscape and breaks the speed record for power autonomous legged locomotion. In contrast to existing research on static and quasi-static stabilization of legged locomotion, our goal is to achieve speed and efficiency by adopting a dynamical mode of operation. Towards this end, we present a decompositional approach, where the problem is split into the task specification through simple dynamical systems, “templates” and their embedding in the original system to achieve stable control of locomotion.

1.1 Legged Vehicles: Motivations and Challenges

It might seem that there are already many existing mobile platforms, such as wheeled or tracked vehicles, which are capable of locomotion at very high speeds, and we need not necessarily consider legged locomotion as an alternative. Instead, one could argue that there are many, equally important and challenging problems to be solved at various other levels of abstraction, such as robot navigation, localization and mapping. Clearly, all of these problems are of great importance in robotic locomotion,

and have received well deserved attention in the robotics literature.

On the other hand, one must also note that the performance of traditional mobile robots largely results from the structured nature of their operating environments. In fact, approximately half of the world's land mass is inaccessible to such vehicles [78]. Robotic mobility over highly broken and unstable terrain requires legged machines. Even though for many applications, traditional wheeled platforms provide sufficient robustness and energetic performance, in the long run, systems capable of operating in the widest variety of terrain conditions, will be legged robots.

Nonetheless, legged platforms present many difficulties from an engineering point of view. Unlike traditional mobile robots, the control of these platforms requires a thorough understanding of their dynamics. Most of their behavioral and energetic performance relies on the inherent properties of their mechanical structure, for which we currently have very few well developed analytical tools. The coordination of the large number of joints and the redundancy in the actuated degrees of freedom compared to the small number of task degrees of freedom, present novel challenges in the design of controllers for such systems.

The simpler solutions to these problems have been adopted by most legged machines such as Brooks' Genghis [12], and Case Western Robot II [22], using statically stable gaits by keeping their center of mass within the support area of the legs [39, 52, 73, 74]. The effectiveness of this paradigm, however, has been hostage to the necessity of operating at very low speeds, where the kinematics dominate the behavior. The effective actuation bandwidth is thus limited by sacrificing speed for force. Moreover, the stiff coupling of the actuators to the environment precludes the possibility of dynamic gaits where impacts and collisions with objects in the environment are inevitable.

Another limitation of such platforms arises from their mechanical complexity. The design of legged robots always exercises the familiar trade-off between freedom of movement and decreased reliability as the number of actuators increases. Examples on

one extreme include the Case Western Robot III [5, 55] whose commitment to detailed zoomorphic mimicry results in great kinematic complexity (24 actuated degrees of freedom) and the attendant limitations of contemporary commercial actuators incur significant limitations on performance and robustness. At this level of complexity, it is virtually impossible to obtain the bandwidth and reliability required by faster speeds of operation with today's technology.

Not surprisingly, the speed and agility of animals in various modes of locomotion, results from their efficient and dynamically dexterous use of their bodies. Even though for tasks where precision is required, static stability may become desirable, it cannot match the speed and efficiency that a dynamical mode of operation can achieve [1, 48]. Furthermore, in many cases systems need to interact with an inherently dynamic environment, often to overcome limitations of hardware. As a result, most existing legged systems are structurally unsuitable for tasks requiring dynamical dexterity, such as running or leaping. In this context, Raibert's runners were the first robots to demonstrate the feasibility of working with the natural dynamics of the underlying mechanical system, achieving dynamically dexterous behaviors [60].

On the other hand, the design of such dynamically capable robots is only part of the challenge. Tight coupling between behavioral primitives and an inherently dynamical environment challenges the traditional view of mobile robotics, wherein it has been possible to adopt fairly rigid abstractions in isolation from the physical environment. The familiar enforcement of static stability and the use of kinematic planning are not at all suitable in describing or controlling dexterous maneuvers. Increasingly, there is a need for the dynamics of the motion to reflect back through the interface between the behavioral primitives and higher level planning and navigation systems. In this context, proper design and instantiation of appropriate task representations is an outstanding challenge.

1.2 “Functional Biomimesis” for Legged Vehicle Design

Our need for a machine naturally capable of incorporating both dynamical dexterity, as well as the precise and slow maneuvers appropriate for other settings, is clear. A natural solution is suggested by biological locomotors: Hexapods obviously afford simple low torque solutions to static (parked) and quasi-static (walking) operation. More surprisingly, they also admit dynamical behaviors such as running [75], primarily attributed to bipeds and quadrupeds. As a consequence, part of this thesis concerns the design and building of a hexapod robot, RHex, to satisfy the constraints of physically realizable dynamical dexterity.

RHex’s design represents what might be considered the minimal configuration for a legged vehicle that leaves room for a reasonably wide range of behaviors. With only one actuated degree of freedom per leg, it relies on the passive dynamics of its compliant legs to overcome the very limited power density of contemporary actuation and energy storage technology. In a way that might be termed “functional biomimesis”, this design exercise is informed by specific principles governing animal locomotion [28], inspired by the stunning success of Raibert’s runners and later embodiments of similar principles in the Scout class of quadrupeds [14, 15].

Even though RHex is equipped with the necessities of autonomous dynamic behaviors, the control of its locomotion still remains a formidable challenge. Achieving these behaviors using the seemingly small number of actuated degrees of freedom, through interactions with the passive mechanics requires an understanding of the underlying dynamics. Additional complications spawn from numerous modeling issues such as actuator limitations, unknown dynamics of elastic materials as well as the imperfect sensory data.

Equally challenging is the specification and characterization of the underlying

task, in a way that informs its realization. One of the central problems in the design of controllers is that of characterizing a task in ways that are useful, intuitive and can capture important aspects of the underlying mechanism. There is often a multitude of ways in which the interface between a controlled system and its supervising authority can be chosen. Traditional solutions include trajectory tracking style interfaces, where the kinematic behavior of the system is explicitly specified. The nature of our problem, however, does not suitably lend itself to such solutions. Rigid abstractions that are not informed by the dynamical properties of the underlying mechanism are unlikely to withstand the demands of rich environments. One of the major contributions of this thesis is the explicit use of simple “template” dynamical systems as a means of specifying and characterizing dynamical locomotion tasks.

1.3 The Spring-Loaded Inverted Pendulum

Research in the biomechanics of running and walking in animals [3, 17, 18], demonstrated that simple inverted pendulum and spring-mass models are very successful in describing the motions of animals of very different sizes and morphologies [10, 11, 24, 50]. A more formal version of these models, the Spring-Loaded Inverted Pendulum(SLIP) model has become a very useful descriptive tool in understanding animal running. Thanks to its relatively simple mathematical structure, analytical studies have succeeded in characterizing basic aspects of running, including stability and parameterizations of stable gaits [44, 49, 69, 70].

Similarly, finding such low degree of freedom characteristic behaviors across a very wide range of animals, spawned the idea that the SLIP model of locomotion may not only be a descriptive model of running, but could also be a literal control target. Animals could in fact be recruiting their muscles and sensors, in a way that forces their center of mass to mimic a low degree of freedom template dynamical system. Surprisingly, numerical studies on human running data seems to support

this hypothesis [23, 24].

The idea of using low degree of freedom dynamical systems as control targets also addresses the problem of suitably encoding the goal task. Especially in cases where all the essential aspects of a behavior can be captured with a low degree of freedom model such as SLIP, the advantages become much clearer. The ability of such a model to express the target behavior with the least amount of complexity is a very important step in overcoming the curse of dimensionality.

The dynamical template control paradigm is largely inspired by these observations. Properly chosen, the simple, low degree of freedom control target can achieve efficiency and dexterity by exploiting the natural dynamics of the underlying system. Analytical studies of simple templates can reveal the essential features of particular tasks, yielding formally characterized controllers and their stability properties. Finally, the resulting natural decomposition of the problem may admit the use of the redundancy in the system in a multitude of ways, without interfering with the essential features of the task.

The first explicit instantiation of these ideas was Raibert’s work on controlling multi-legged robots as well as more complex animated locomotors [61]. Similarly, simple dynamics of a swinging pendulum were recently employed as “target dynamics” for various dynamic brachiating tasks [54]. Their work demonstrated that the representation of essential task coordinates through a simple dynamical system admits the design of several task level controllers independent from the details of joint level torque controllers.

Similarly, recent work in the asymptotically stable control of walking through the use of tuned zero dynamics [79, 80] also employs variants of the same principle, even though their domain admits a much more careful and formal treatment. Several output functions on the state of the robot are defined to encode a two dimensional manifold in the state space, together with a finite time controller to ensure convergence of trajectories onto the manifold. The resulting dynamics on the manifold concisely

capture the progression of the walking motion, while the definition of the output maps encode the detailed motions of individual joints. The end result portrays a similar decomposition of the task coordinates from the shape coordinates.

Among other similar examples is the virtual model control scheme, which implements a motion control language where virtual mechanical components are composed to generate imagined forces [58, 59]. These forces are then projected back to real joint torques and actuations to achieve control of the underlying system. Virtual model control has been successfully used in several applications, including the Case Western Robot III [55] and the MIT Spring Turkey [58]. The dynamical template control approach is a more formal version of the same principle, where the virtual dynamical systems and their embedding in the overall system, are much more carefully defined and analyzed.

1.4 Contributions

The first major contribution of this thesis is the design and construction of an autonomous hexapedal robot, RHex, together with simple open-loop controllers for basic locomotion tasks. RHex’s design incorporates specific principles from biomechanics of insect locomotion such as passive compliance and lateral sprawl. Combined with seemingly simple control strategies, it yields surprising behavioral performance. We present carefully designed experiments that establish RHex as the first robot in the literature to achieve speeds exceeding one body length per second over terrain approaching the complexity and diversity of the natural landscape. Our systematic experiments include basic walking on various surfaces and turning as well as locomotion over obstacles of various sizes and shapes. These studies also represent one of the most careful performance characterization efforts in the mobile robotics literature.

It should be noted that neither the set of behaviors we introduced, nor their characterization are by any means exhaustive. There are many other capabilities

feasible with RHex’s morphology, such as stair climbing, leaping and pronking, as demonstrated in recent work by members of the RHex project [46, 51]. Detailed mathematical explanations for the performance of these solutions, particularly in relation to various design choices in the morphology, still remain to be discovered.

The second contribution of this thesis is the development of dynamical template based control methods within the context of hexapedal locomotion. We define dynamical templates as simple (one or two degrees of freedom) systems that accurately and concisely characterize essential features of a task, while admitting careful analysis and understanding of their behavior and stability properties. Even though this idea is not new in itself, being discovered relatively early by researchers in biomechanics [9, 28] and employed later within robotics control problems [16, 54, 61], our work represents its first application to hexapedal locomotion. We not only adopt the idea of using dynamical templates as literal control targets, but also develop novel methods to explicitly address singularities and actuator limitations inherent in the system.

As a first illustrative example of dynamical templates, we present algorithms for RHex to perform self-righting through dynamical flipping. Primarily motivated by our ultimate goal of building robots that are capable of autonomous survival in unstructured environments, an open-loop controller capable of flipping on a variety of indoor surfaces is developed. Subsequently, a model based controller is introduced to overcome the limitations of this first generation controller. Furthermore, through the use of specific constraints on the design of the controller, we are able to reduce the initial 3 DOF model to a much simpler one degree of freedom system, whose numerical characterization yields performance and stability properties of the resulting behavior.

Following the flipping analysis, we describe two locomotion models of increasing complexity where we achieve effective locomotion control through the embedding of appropriate templates. Both for a planar 4 DOF leg morphology, the Ankle-Knee-Hip(AKH) model, and a planar compliant hexapod (Slimpod), model based

torque control strategies are devised to embed variants of the basic Spring-Loaded Inverted Pendulum(SLIP) model within their more complicated, higher degree of freedom structures.

The simple planar 2 DOF SLIP template is sufficient to control the task level behavior of the planar AKH model. Simulation studies demonstrate that even a strongly model dependent controller at the gait level — in particular, a deadbeat strategy that relies on approximate inversion of SLIP return maps — is able to successfully regulate the locomotion of the AKH without any further modifications. Furthermore, this work also emphasizes the simplicity and convenience at the task level that results from the explicit decomposition of the control problem. Simple commands such as forward velocity, hopping height and duty factor are much better suited for interfacing with locomotion systems than individual joint trajectories or torque commands for a highly complex system.

For the control of hexapedal locomotion, the basic SLIP template turns out to be inadequate. Consequently, this thesis also develops bipedal extensions to the basic model and introduces the Bipedal SLIP template. BSLIP is particularly important in capturing the recirculation of flight legs that is characteristic of RHex’s alternating tripod gait. A new deadbeat controller is presented, together with simulation studies to establish the satisfactory gait level stability properties of the resulting closed loop system.

As the last step towards a spatial hexapedal controller, we introduce a planar hexapod model, Slimpod, and its control through the embedding of BSLIP. This model captures nearly all the characteristic aspects of RHex’s morphology. In particular, we investigate within Slimpod, the enforcement of the alternating tripod gait, the continuous control of leg recirculation, the singularities arising from the underactuation of the legs as well as the torque limitations of the actuators. The extension of these constructs to the spatial controller turns out to be fairly straightforward with minor changes to control the additional degrees of freedom in the body orientation

and a rudimentary horizontal plane motion model.

The thesis concludes with simulation studies characterizing the stability of the closed loop system resulting from the action of the embedding controller on the spatial compliant hexapod dynamics. It is important to note that this controller lies in extreme opposition to the pure open-loop strategies as it eliminates all explicit time dependency from the closed loop dynamics and yields a completely autonomous dynamical system. Even so, we were able to numerically identify asymptotically stable limit cycles of this system with large basins of attraction. The numerical evidence hence suggests that an experimental implementation of our template based locomotion controllers is indeed feasible. Even though the scope of this thesis does not cover such an implementation, only a few practical issues remain to be resolved before it becomes a reality.

CHAPTER 2

RHex: A Compliant Hexapod Robot

2.1 Motivation

RHex is a hexapedal robot that can run over terrain approaching the complexity and diversity of the natural landscape. This chapter presents a characterization of RHex's performance with a simple, open-loop alternating tripod gait controller as a baseline for the template based control strategies of later chapters.



Figure 2.1: The RHex experimental platform (<http://ai.eecs.umich.edu/RHex>).

The original design idea for RHex spawned from close interactions between researchers from biomechanics, mathematics and dynamics [56]. Its morphology com-

bines principles from biomechanics such as compliance in the saggital plane and sprawled posture, with novel mechanical design principles to achieve robustness and efficiency. Similar ideas were embodied in earlier legged platforms such as the Scout class of quadrupeds [14, 15].

Preliminary simulation studies on a simplified, planar version of the morphology suggested impressive locomotion and obstacle crossing capabilities. Throughout 1999, I extended these simulation studies to a spatial implementation of the hexapod, together with realistic actuator and energy storage models. These extensions also yielded very successful results and demonstrated that a physical implementation of the original design was indeed feasible.

Following the simulation studies, I built the first version of RHex during a period of six weeks, starting in August 1999¹. At the end of this period, RHex was able to perform basic locomotion tasks such as walking forward and backward and turning, with computational and power autonomy.

The experiments presented in this chapter were performed during December 1999 in The University of Michigan. This material was also published in an archival paper [64].

2.2 Design Concept and Morphology

In all robotics applications, mechanical complexity is one of the major sources of failure and considerably increases the cost. Our design emphasizes mechanical simplicity and thereby promotes robustness. Autonomy, a critical component of our aspiration toward real-world tasks in unstructured environments outside the laboratory, imposes very strict design constraints on the hardware and software components. It is often impossible to achieve with simple modifications to a system otherwise de-

¹The construction took place in the Ambulatory Robotics Laboratory, McGill University, with help from Liana Mitrea as well as other graduate students working with Prof. Martin Buehler.

signed for non-autonomous operation. These constraints also justify our preference for overall simplicity, in particular the minimum amount of actuation and sensing.

Our design, depicted in Figure 2.2, consists of a rigid body with six compliant legs, each possessing only one independently actuated revolute degree of freedom. The attachment points of the legs as well as the joint orientations are all fixed relative to the body and the leg compliance is mainly in the unactuated spherical degrees of freedom.

This configuration admits an alternating tripod gait for forward and backward locomotion, and possibly other more elaborate behaviors such as leaping, stair climbing etc. Moreover, the symmetry of the design allows identical upside-down operation and imposes no restrictions on forward directionality. We explore some of this behavioral repertoire experimentally in Section 2.4.

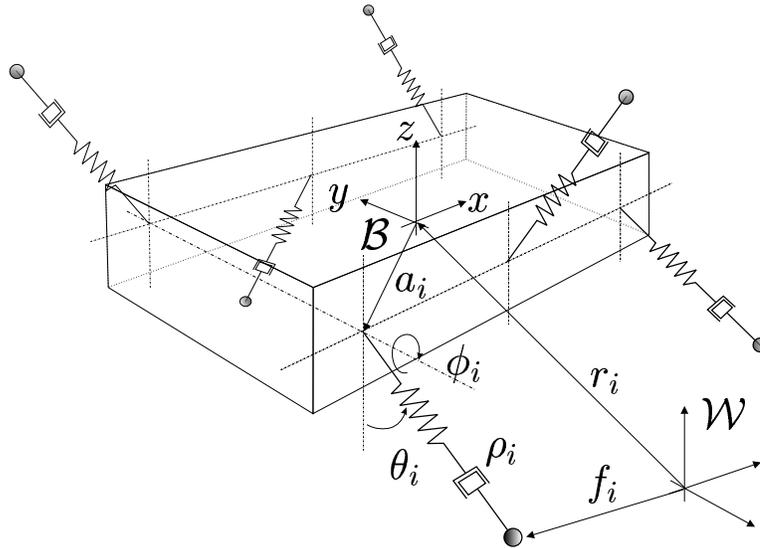


Figure 2.2: The compliant hexapod design.

2.3 Open Loop Control of Locomotion

The first prototype of RHex did not have any external sensors by which its body coordinates could be estimated. As a consequence, the experiments presented in this chapter use joint space closed loop (“proprioceptive”) but task space open loop control strategies. The algorithms that we describe are tailored to demonstrate the intrinsic stability properties of the compliant hexapod morphology and emphasize its ability to operate without a sensor-rich environment.

Not surprisingly, the robustness of such open-loop algorithms which exercise mechanical feedback will be much higher against sensor noise and failure than those with tighter feedback loops. On the other hand, one would ultimately need task level feedback to deal with uncertainties in the environment and model inaccuracies in a predictable and reliable manner. As a consequence, exploring and characterizing the range of controllers between pure open-loop algorithms and very tight feedback loops is very important in effectively combining these extremes.

In this section, we present a four-parameter family of controllers, that yields stable running and turning of the hexapod on flat terrain, without explicit enforcement of quasi-static stability. In Section 2.4, we demonstrate the capabilities of this family of controllers on our experimental platform, over a wide range of terrain conditions, from flat terrain to a rough, broken surface.

All controllers generate periodic desired trajectories for each hip joint, which are then enforced by six local PD controllers (one for each individual hip actuator). As such, they represent examples near one extreme of possible control strategies, which range from purely open-loop controllers to control laws which are solely functions of the leg and rigid body state. The experiments presented in this chapter attempt to characterize the properties associated with the sensorless feedforward extreme, which, when RHex has been endowed with sensors, we hope to complement with feedback to explore the aforementioned range.

An alternating tripod pattern governs both the running and turning controllers, where the legs forming the left and right tripods are synchronized with each other and are 180° out of phase with the opposite tripod, as shown in Figure 2.3.

2.3.1 Running

The running controller’s target trajectories for each tripod are periodic functions of time, parametrized by four variables: t_c , t_s , ϕ_s and ϕ_o . The period of both profiles is t_c . In conjunction with t_s , it determines the duty factor of each tripod. In a single cycle, both tripods go through their slow and fast swing phases, covering ϕ_s and $2\pi - \phi_s$ of the complete rotation, respectively. The duration of double support t_d (where all six legs are in contact with the ground) is determined by the duty factors of both tripods. Finally, the ϕ_o parameter offsets the motion profile with respect to the vertical (see Figure 2.3). Note that both profiles are monotonically increasing in time; but they can be negated to obtain backward running. Throughout this chapter, $\mathbf{u} := [t_c, t_s, \phi_s, \phi_o]$ is used as the definition of the parameter vector for a particular instantiation of this gait.

Control of running behavior is achieved by modifying these parameters for a particular desired behavior during locomotion.

2.3.2 Turning

We have developed two different controllers for two qualitatively different turning modes: turning in place and turning during running.

The controller for turning in place employs the same leg profiles as for running except that contralateral sets of legs rotate in opposite directions. This results in the hexapod turning in place in the direction determined by the rotational polarity of the left and right sets of legs. Note that the tripods are still synchronized internally, maintaining three supporting legs on the ground. Similar to the control of the forward

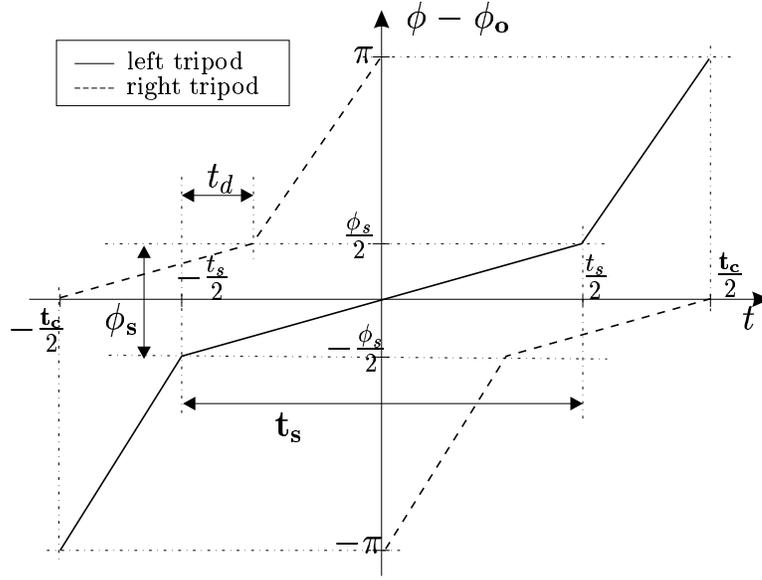


Figure 2.3: Open-loop motion profiles for the left and right tripods.

locomotion speed, the rate of turning depends on the choice of the particular motion parameters, mainly t_c and ϕ_s .

In contrast, we achieve turning during forward locomotion by introducing differential perturbations to the forward running controller parameters for contralateral legs. In this scheme, t_c is still constrained to be identical for all legs, which admits differentials in the remaining profile parameters, ϕ_o and t_s , while ϕ_s remains unchanged. Two new gain parameters, Δt_s and $\Delta \phi_o$ are introduced. Consequently, turning in $+x$ (right) direction is achieved by using $\mathbf{u}_l = [t_c, t_s + \Delta t_s, \phi_s, \phi_o + \Delta \phi_o]$ and $\mathbf{u}_r = [t_c, t_s - \Delta t_s, \phi_s, \phi_o - \Delta \phi_o]$ for the legs on the left and right sides, respectively.

2.4 Experimental Platform

2.4.1 Hardware Description

As part of my dissertation research, I have built an experimental platform (see Figure 2.1) as an instantiation of the design concepts of Section 2.2. RHex is an

autonomous hexapod robot with compliant legs. All the computational and motor control hardware is on board, together with two Panasonic 12V 2.2Ah sealed lead-acid batteries for power autonomous operation. A PC104 stack with a 100 MHz Intel 486 microprocessor, together with several I/O boards performs all the necessary computation and implements the controllers of Section 2.3. A remote control unit provides the user input for giving higher level commands such as the running speed, and turning direction, presently via a joystick.

Each leg is directly actuated by a Maxon RE118751 20W brushed DC motor combined with a Maxon 114473 two-stage 33:1 planetary gear [42], delivering an intermittent stall torque of 6 Nm. Even though each motor is PWM voltage controlled, additional back-EMF compensation in software permits approximate motor torque control. The motor angle, and thus the leg angles, are controlled via 1 kHz PD control loops.

The main body measures 53x20x15 cm. The legs are made from 1 cm diameter Delrin rods and are "C" shaped to provide compliance primarily in the radial direction and permit easy clamping to the gear shaft. The leg length is 17.5 cm, measured as the vertical distance from ground to the gear shaft when standing up. The encoder/motor/gear stacks protrude from the main body and the maximum widths of the front and back legs amount to 39.4 cm, measured at half the leg length. To provide clearance for the rotating front and back legs, the motors for the middle legs are further offset and result in a maximum width of 52 cm. The total mass of the robot is 7 kg with each leg contributing only approximately 10 g.

As part of RHex's development, I have also developed the control software that implements the low level motor control as well as the control strategies described in this chapter. RHexLib is a software library that runs both under Linux and QNX operating systems, the former for development within a simulation environment and the latter for real-time control of the robot. Its detailed design and capabilities are described in Appendix G.

2.4.2 Visual Measurement Apparatus

Absent any inertial sensing on RHex, we devised a simple visual tracking system to record the robot’s position and orientation in the sagittal (obstacle crossing experiments) and the horizontal (turning and rough surface experiment) planes². Four Light Emitting Diodes (LEDs) were attached to the robot’s body and a set of stationary calibration LEDs were placed close to the extremes of the camera’s field of view. The experiments were then conducted in complete darkness, which provided for very high contrast recordings of the LED markers. Thanks to this greatly simplified visual data, standard computer vision algorithms were then employed to extract the planar robot position and orientation up to a 1% accuracy in the average velocity computations.

2.4.3 Experimental Performance

In the sequel, we will document the robot’s speed over various terrains, its maneuverability, obstacle crossing capability, and payload. Furthermore, energy efficiency and runtime are critical performance criteria for any untethered robot. Thus the energetic performance of the robot is carefully documented, but it must be noted that to date no efforts have been made to optimize it. All experiments - except the random obstacles experiment - were run untethered and we document the average power consumption, based on recordings of the battery voltage and current.

To measure energy efficiency we use the “Specific Resistance” [30], $\varepsilon = P/(mgv)$, based on the robot’s weight, mg , and its average power consumption, P , at a particular speed, v . Specific resistance was originally used to compare the energy efficiency of animals of vastly different sizes, where the average power measured the rate of metabolic energy expenditure, based on oxygen consumption. The same measure

²Special thanks to Greg Sharp who has built and tested the tracking algorithms for these experiments.

has been used to compare the energy efficiency of a range of different robots [1, 36]. Unfortunately, attention to energy efficiency and its reporting is a fairly recent phenomenon in robotics, and not consistent. For example, the power, if documented at all, is given as the mechanical power delivered by the actuators, the peak mechanical power of the main power source, or the total electrical power consumption. Therefore, quantitative energetic comparisons of past robots are not always precise. For electrically actuated mobile robots like RHex it makes most sense to report the total electrical power consumption (which includes the power for sensing and computing), since it will determine, together with the battery capacity, the all important runtime.

Throughout the experiments, the control parameters were set to fixed values, and these were only modified by the operator via the joystick commands in an attempt to steer the robot along a straight line. The speed command input was used solely for starting and stopping the robot.

Experimental findings are summarized in Table 2.1 and Figures 2.4-2.9. A detailed account of the setup, measurement protocols and failure modes is presented in the Appendix.

Running

This first set of experiments documents RHex's maximum velocity, power and specific resistance with the two-stroke open loop controller of Section 2.3 while traversing carpet, Linoleum, grass and coarse gravel. The robot moved well over these indoor and outdoor surfaces, with only minor velocity variations between 0.45 m/s and 0.55 m/s as shown in Figure 2.9. The velocity on Linoleum was lowest due to intermittent slipping, which also causes a larger standard deviation of the runs compared to carpet. The surface irregularities of the outdoor grass and gravel surfaces provided improved traction, and therefore average velocities slightly above 0.5 m/s, but also resulted in larger variations between the runs. The specific resistance (power consumption) was

	carpet	linoleum	grass	gravel	rough	single obst.	comp. const.	rand. obst.
total # of runs	10	11	16	25	32	14	14	26
successful runs	10	10	10	10	16	10	10	10
electronics & hardware problems	-	-	1	5	6	-	-	2
deviation from course	-	1	-	5	7	-	-	5
operator mistake ^a	-	-	5	5	-	-	-	2
stuck on obstacle	-	-	-	-	3	1	2	7
measurement errors	-	-	-	-	-	3	2	-

Table 2.1: Experimental statistics.

^aThese failure modes include steering in unwanted directions, failure to trigger the timing switch and the power cord wrapping around the legs.

lowest on carpet with 2.21 (80 W) and highest on gravel with 3.74 (140 W).

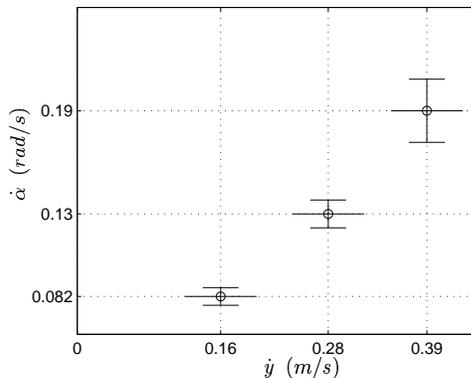


Figure 2.4: RHex’s turning yaw rate as a function of forward velocity.

We experimented with control parameter settings to reach the maximum robot velocity on office carpet and linoleum, and selected $\mathbf{u} = [0.45, 0.2, 51, 0]$. The maximum speed in the current prototype is limited by the onset of vertical body oscillations which reduce traction. The grass and gravel surfaces were not tested with these settings prior to the reported experiments. Figure 2.9 shows the average velocity, power

$\dot{x}(m/s)$	t_c	t_s	ϕ_s	ϕ_o	$\Delta\phi_o$	Δt_s
0	1.0	0.6	35	0.0	0.0	0.0
0.16	1.2	0.7	25	0.0	6.0	-0.02
0.28	0.8	0.45	35	0.0	7.5	0.0
0.39	0.53	0.33	40	0.0	6.5	0.0

Table 2.2: Controller parameters for turning at different speeds

consumption and the specific resistance over ten runs, with standard deviations for all the experiments. Table 2.1 summarizes the failure modes and statistics for all the experiments described in this Section.

Turning

As our simulation study had predicted, steering is possible, even though the leg actuation is limited to motion in the sagittal plane only, via differential motion between left and right legs. We selected control parameters which resulted in turns in place and robot speeds up to about 0.4 m/s (see Table 2.2).

The maximum forward velocity is reduced during turning, because the differential leg motion precipitate the onset of the speed limiting vertical body oscillations. The maximum yaw angular velocities increase almost linearly with forward velocity up to 0.19 rad/s at 0.39 m/s, as illustrated in Figure 2.4. Interestingly, the resulting turn radius is almost constant with approximately 2 m. Turning in place provides the highest yaw angular velocity of 0.7 rad/s. At present, we do not understand completely the relationship between the controller parameters and effective yaw rates, a subject of ongoing research.

Obstacle Crossing

The obstacle crossing capabilities of the simple open loop walking controller were evaluated with two different obstacles - a 15 cm high Styrofoam block and a composite

obstacle with a maximum height of 22 cm, as shown in Figure 2.5 and Figure 2.6, respectively. The robot was able to surmount both obstacles, neither sensing them, nor with any modification to the control parameters of the walking experiments. The data in the top portions of the two graphs shows the forward velocity averages before, during and after the obstacle, averaged over ten runs. Surprisingly, the average velocity decreases only slightly as the robot climbs over the obstacle, and increases again afterward. Since the robot’s trajectory over the obstacle depends greatly on how the legs engage it, the standard deviation of the average velocities increases over the obstacle. The average speeds varied most (largest standard deviation) after the composite obstacle, since it depended much on how the robot landed. As a further illustration of the robot’s motion, the forward velocity from a particular run, and the robot’s body in the sagittal plane at 0.5 s intervals during the same run are shown as well.

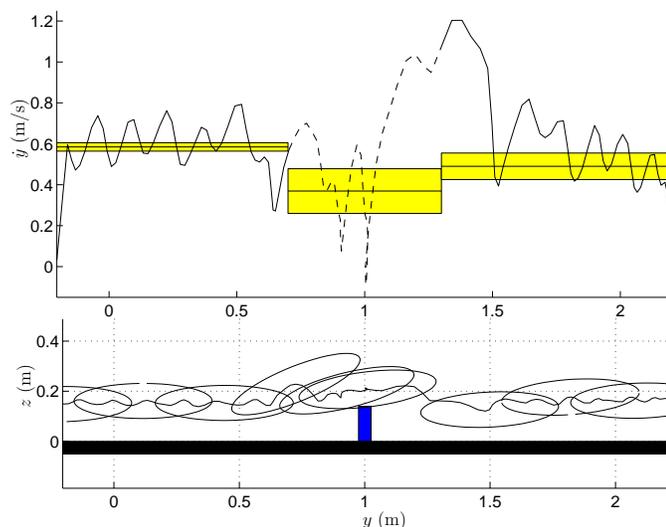


Figure 2.5: Sagittal plane data during simple obstacle crossing. The data in the top portion of the graph shows forward velocity averages and standard deviations before, during and after the obstacle, averaged over ten runs. The solid line is one particular run. The bottom half shows a projection of the robot’s body onto the sagittal plane in 0.5 s intervals (horizontal and vertical axes at the same scale).

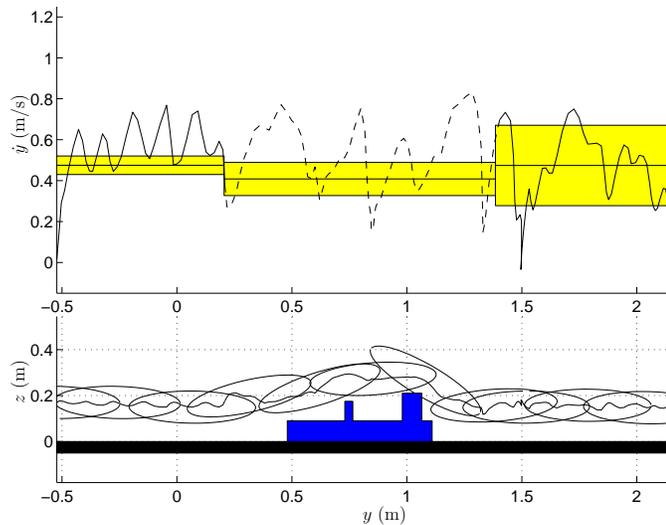


Figure 2.6: Sagittal plane data during composite obstacle crossing. The data in the top portion of the graph shows forward velocity averages and standard deviations before, during and after the obstacle, averaged over ten runs. The solid line is one particular run. The bottom half shows a projection of the robot’s body onto the sagittal plane in 0.5 s intervals (horizontal and vertical axes at the same scale).

Obstacle Course

To demonstrate RHex’s rough terrain capabilities, we constructed the obstacle course depicted in Figure 2.7. It consisted of ten randomly spaced obstacles of 12.2 cm height (that is, 60% of the leg length and exceeding ground clearance by 1.8 cm). This was by far the most challenging of the experiments, requiring the largest number of runs before ten successful completions. Most failures can be attributed to the open loop nature of the walking controller, which had to climb blindly over ten randomly spaced obstacles, sometimes as little as half a body width apart, but all higher than the ground clearance. This had to be done successfully over a distance of 8.13 m, avoiding all the failure modes detailed in the Appendix. Yet, for the ten successful runs, RHex was able to maintain an average velocity of 0.36 m/s over the length of the obstacle course (Figure 2.9). The best run finished in only 17.78 s, or an average velocity of 0.46 m/s, with a specific resistance of 8.17. The punishing

nature of this course is reflected in the power consumption of more than three times of walking on carpet, over five times the specific resistance, and a high rate of component breakdown: During the obstacle course experiments, RHex broke three legs, burned several circuit traces, and fractured its frame.

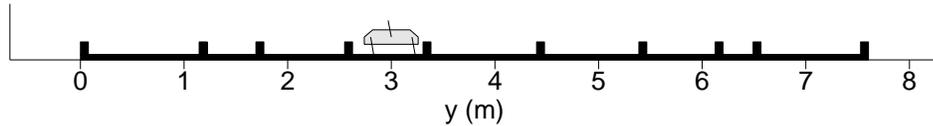


Figure 2.7: Scale drawing of RHex and the random obstacle course.

Rough Surface

This last rough terrain experiment is an attempt to evaluate RHex’s performance in a similar environment to that negotiated by the *Blaberus cockroach* in [27]. To our surprise, RHex was able to traverse this surface with random height variations of up to 20.32 cm (116% leg length, Figure 2.10) with relative ease at an average velocity of 0.42 m/s (Figure 2.9). RHex’s planar trajectories during the two fastest and the two slowest successful runs are shown in Figure 2.8.

Payload and runtime

To demonstrate RHex’s payload capacity, we mounted an additional mass of 7.94 kg (one 10 lb and one 7.5 lb weight-lifting barbell) to the bottom of RHex’s body and adjusted the control parameters for lower speed and a small sweep angle ($c_6 = 0.7, 0.2, 14, 0$). The robot was able to transport this additional mass, more than its own total mass, at about half its maximum speed (0.25 m/s) with a specific resistance of about 3. These averages were obtained from ten runs. This payload is close to the limit of the current design, and may not be practical, since the motors are not powerful enough to raise the robot when lying on the floor.

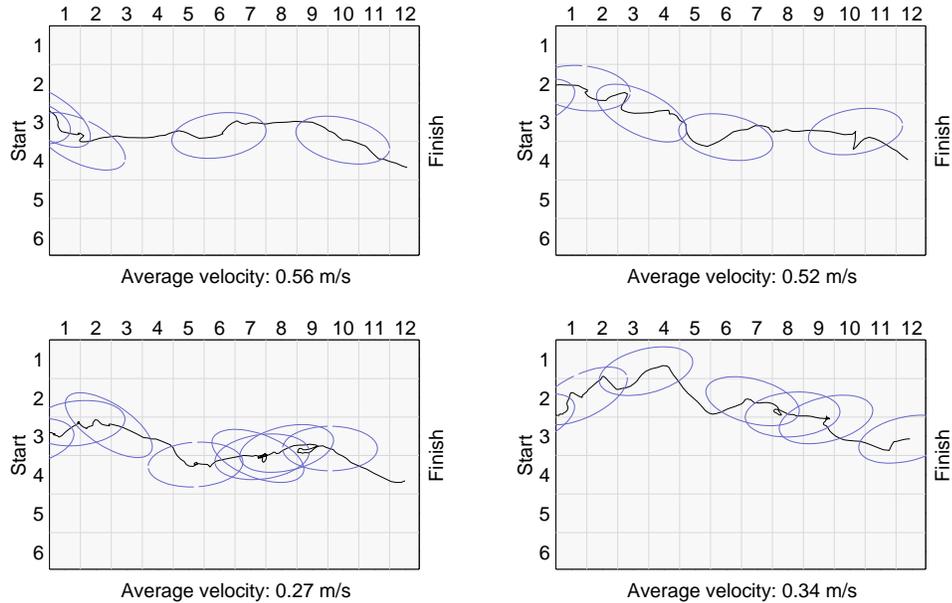


Figure 2.8: The two fastest (top) and the two slowest (bottom) robot trajectories in the horizontal plane as RHex moves over the rough surface. The plots also show projections of the body onto the horizontal plane in 1 s intervals.

Finally, the endurance of RHex was tested in standby mode, with the motors enabled, and maintaining a standing position, and while walking at maximum speed on carpet. The average standby runtime was 48 minutes, and 18 minutes for continuous walking (both values averaged over 5 successive experiments).

2.5 Conclusion

Nimble, robust locomotion over general terrain remains the sole province of animals, notwithstanding our functional prototype, RHex, nor the generally increased recent interest in legged robots. RHex, endowed with only a rudimentary controller, uses what might be termed the engineering equivalent of “preflexes” [13, 27] to negotiate relatively badly broken terrain at relatively high speeds — performance beyond that heretofore reported for autonomous legged vehicles in the archival literature.

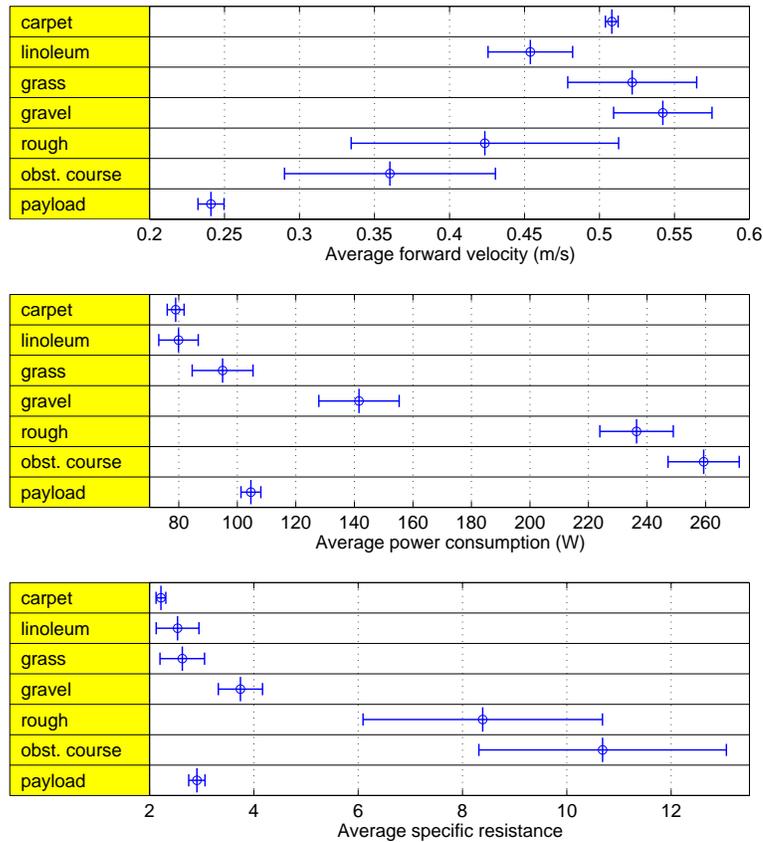


Figure 2.9: Comparison of RHex’s average forward velocity and energetics for all running experiments.

The control strategies presented in this chapter reside on one extreme of possible approaches, wherein the task level control is purely open-loop. Experimental results of this chapter demonstrate that even such a simple approach, combined with the specific morphology of RHex, is capable of yielding stable forward locomotion and turning as well as surprising performance over obstacles of varying size and shapes. In the following chapters, I will investigate the opposite end of the spectrum and demonstrate what can be accomplished with high bandwidth feedback.

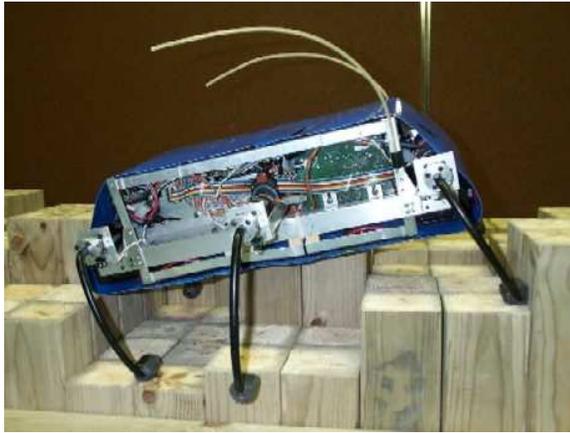


Figure 2.10: Sample profiles of row 6 (columns 8,9 and 10) of the rough surface with RHex.

CHAPTER 3

Backflip Controllers for RHex

3.1 Motivation

This chapter concerns the design and analysis of a dynamical self-righting controller for RHex. In this context, we first present a rigid planar model of RHex, motivated by an empirically tuned controller capable of inducing successful flips on a limited range of surfaces. The rest of the chapter develops a model based controller that promises to significantly extend the terrain over which the flipping succeeds.

The controller design also demonstrates how a simple dynamical template can be employed to gain a better understanding of the flipping task. In particular, we choose a one dimensional manifold in the configuration space and ensure its invariance through explicit constraints on the controller design. Under the assumption of initial conditions on the manifold, these constraints reduce the system to a one degree of freedom dynamical system, admitting the design of a thrust maximizing controller as well as a preliminary investigation of its stability and performance. Even though convergence onto the manifold is not explicitly enforced, we suspect that it implicitly results from the effect of gravity and unilateral ground contact constraints.

Interestingly, the resulting dynamical “template” turns out to be similar to dynamic juggling tasks [16, 62] as well as one degree of freedom vertical hopping [44, 45].

Motivated by earlier work on these domains, we employ one dimensional return maps and associated fixed points to characterize performance of the flipping task.

The use of flipping as a behavioral primitive in this context is also not a coincidence. Autonomy is clearly an essential component for any robotic platform designed to operate in the real world. In addition to strict power and computational constraints it imposes on the design, it also requires basic self-manipulation capabilities for the survivability of the robot in the absence of a human operator. Even during teleoperation, where the computational demands on the platform are less stringent, the ability to recover from unexpected situations through self-manipulation is essential. Application such as planetary rovers and space missions are among the best examples where these requirements are most critical [4].

In this context, recovery of correct body orientation is among the simplest of self-manipulation tasks. In cases where it is impossible for a human operator to intervene, the inability to recover from a simple fall can completely render a robot useless. Especially in outdoor environments with badly broken terrain and obstacles of various shapes and sizes, the debilitating effects of such accidents have been observed in the past [6].

RHex’s morphology is roughly symmetric with respect to the horizontal plane, and allows nearly identical upside-down or right-side up operation, a solution adopted by other mobile platforms [47]. However, various scenarios such as teleoperation and vision based navigation entail a nominal orientation as a result of the accompanying instrumentation and algorithms. Under these constraints, most existing robotic designs with self-righting capabilities incorporate special kinematic structures such as long extension arms or reconfigurable wheels [38, 76]. In consequence of weight and power limitations, RHex is not equipped with such structures and must rely on its existing morphology together with dynamic maneuvers to perform a flip-over.

The first part of the chapter presents an empirically designed controller capable of inducing a dynamical back flip in RHex. The applicability of this controller, however,

is limited to a small range of simple surfaces, such as linoleum, smooth concrete and carpet, outside which the robot cannot successfully flip over. As a consequence, the rest of the chapter presents a new model and a controller derived from it that promises to extend significantly the variability of terrain over which the desired behavior will be achieved.

Motivated by the successes and limitations of an empirically developed “energy pumping” scheme, we introduce a careful multi-point contact and collision model so as to derive the maximum benefit of our robot’s limited power budget. A comparative simulation study suggests that the new controller will extend significantly the terrain over which the self-righting maneuver succeeds.

The new torque control strategy is designed to maximize the energy injected into the system through a careful modeling of a simple planar hexapod model. In addition to the resulting performance increase, the controller design also greatly simplifies the characterization of the stability properties of the resulting flipping behavior, by reducing the closed loop system to a simple one degree of freedom discrete system.

3.2 Flipping RHex

3.2.1 Simple Open Loop Control

Our first generation flipping controller consists of a state machine (see Figure 3.1). Starting from a stationary position on the floor, the robot very quickly (in 0.2s) goes through two configurations (poses I and II in Figure 3.1), with front and middle legs successively leaving the ground. Depending on the frictional properties of the ground, these motions result in some initial kinetic energy of the body that may in some cases be sufficient to allow “escape” from the gravitational potential well of the initial configuration and fall into the other desired configuration. However, on most surfaces — gravel, grass and asphalt, but even some indoor settings such as

carpet — this is not sufficient to flip the body over. Instead, the robot reaches some maximum pitch lying within the basin of the original configuration, and the robot falls back toward its initial state. Under these circumstances, the controller brings the legs back to Pose I of Figure 3.1 and waits for the impact of the front legs with the ground, avoiding negative work — a waste of battery energy given the familiar power-torque properties of RHex’s conventional DC motors. The impact of the front legs with the ground in their kinematically singular configuration recovers some of the body’s kinetic energy, followed by additional thrust from the middle and back legs, during the period of decompression and flight of the front leg — i.e., during a phase interval when it is possible for the legs in contact to perform positive work on the robot’s mass center. Thrusting is achieved by running a high gain proportional derivative control (PD) law around a judiciously selected constant velocity leg sweep motion. The maximum pitch attained by the body increases with each bounce up until the point where collision losses are exceed the energy that can be be imparted by the PD controller during the leg sweep phase interval. As Table 3.1 suggests, this pumping strategy works very reliably on a number of common surfaces such as linoleum, smooth concrete, carpet and asphalt.

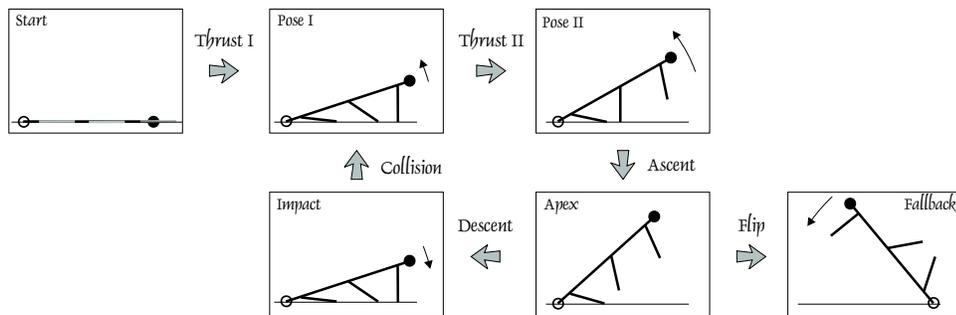


Figure 3.1: Sequence of states for the flipping controller.

However, on many surfaces — the outdoor environments most relevant to RHex’s presumed mission [4, 64] such as loose gravel, grass and soft ground — it does not

concrete	linoleum	carpet	asphalt	grass	gravel
90%	100%	90%	100%	0%	0%

Table 3.1: Success rates of the first generation flipping controller for 10 experiments each on different surfaces.

perform nearly as well. To permit a reasonable degree of autonomous operation, we would like to improve on the range of conditions flipping can function. This requires a more aggressive torque generation strategy for the middle and rear legs. However, empirically, we find that driving all available legs with the maximum torque allowed by the hip motors results in the body lifting off the ground into stance mode, still in the wrong configuration. We require a strategy that can be tuned carefully enough to produce larger torques aimed specifically at pitching the body over. This requires a detailed model of the manner in which the robot can elicit ground reaction forces in consequence of hip torques operating at different body states and assuming varying leg contact configurations.

3.2.2 Assumptions and Control Design Constraints

Several assumptions constitute the basis for our modeling and analysis of the flipping behavior.

Assumption 3.1. The flipping behavior is primarily planar.

The controller described in the previous section operates contralateral pairs of legs in synchrony. The robot’s response lies almost entirely in the sagittal plane, and departures are rare enough to be negligible.

Assumption 3.2. The leg masses are negligible in flight.

We assume that the leg masses are sufficiently small so that their effect on the body dynamics when they are not in contact with the ground is negligible. However,

we still need to consider the masses of the stance legs in formulating the dynamics in order to avoid peculiar singularities arising from the nonzero hip torque actuation.

Assumption 3.3. The tail of the body should keep contact with the ground throughout the flipping action.

This assumption is motivated by a number of observations gathered during the empirical flipping experiments presented above. First, during the initial thrust phases, the front legs provide most of the torque. The configurations where the tail endpoint of the body is in contact with the ground yield the longest duration of contact for the front leg, harvesting the most possible benefit from the associated actuator until it leaves the ground.

It is also clear that one would not want to go through the vertical configuration of the body when the tail endpoint is not in contact with the ground as such configurations require overcoming a higher potential energy barrier and would be less likely to succeed.

Finally, collisions of the body with the ground, which introduce significant losses due to the high damping in the body structure designed to absorb environmental shocks, can be avoided by preserving contact with the ground throughout the flipping action.

In light of these assumptions, our controller designs face two major constraints: keeping the tail endpoint of the body on the ground and respecting the torque limitations of the actuators.

3.3 The Planar Flipping Model

3.3.1 A Generic Planar Model

Based on A3.3, our analysis will be largely confined to configurations where the tail of the body is constrained to lie on the ground. In this section, we describe a

as characterized by Coulomb friction with dynamic coefficient μ and viscous friction with damping constant k_d .

3.3.2 Contact States and Constraints

In this section, we introduce a framework for modeling the interaction of the body and the legs with the ground. Five binary flags, — a pair for the body end points (the nose and the tail), and a triple for the legs, denoted, respectively, as s_n, s_t, s_1, s_2, s_3 — are sufficient to encode the contact configurations of the system within the contact state spaces \mathcal{H}_b and \mathcal{H}_l ,

$$[(s_n, s_t), (s_1, s_2, s_3)] \in \mathcal{H}_b \times \mathcal{H}_l$$

$$\mathcal{H}_b := \{0, 1\} \times \{0, 1\},$$

$$\mathcal{H}_l := \{0, 1\} \times \{0, 1\} \times \{0, 1\}.$$

For any $s \in \mathcal{H}_l$, we use s_i to denote the corresponding contact state for the i^{th} leg for $i = 1, 2, 3$. In the sequel, we will find it useful to introduce a partial order on \mathcal{H}_l imposed by the following relation,

Definition 3.1. Let $p, r \in \mathcal{H}_l$. We denote by the symbol \geq , the following relation

$$(p \geq r) \Leftrightarrow ((\forall i \ p_i = 0 \implies r_i = 0)) \tag{3.1}$$

Lemma 3.1. *The relation \geq is a partial order on \mathcal{H}_l .*

Proof. \geq is clearly reflexive and transitive. Also, if $p \geq r$ and $r \geq p$, then we have $\forall i, p_i = r_i$, which implies that $p = r$. Hence, \geq is also antisymmetric. \square

The morphology of our model requires that the endpoints of the body be above the ground,

$$b_z > d \sin |\alpha|, \tag{3.2}$$

and that a leg must reach the ground

$$d_i \sin \alpha > l - b_z \tag{3.3}$$

d	d_1	d_2	d_3	l	m	I	m_t
(m)	(m)	(m)	(m)	(m)	(kg)	(kgm ²)	(kg)
0.28	-0.2	0	0.2	0.12	7.5	0.18	0.1

Table 3.2: RHex’s Kinematic and dynamic parameters.

before it can apply any torque to the body.

Note that three instances of the leg contact constraint (3.3) partition the space of valid configurations into four different regions, within which certain legs are required to be in flight (see Figure 3.3).

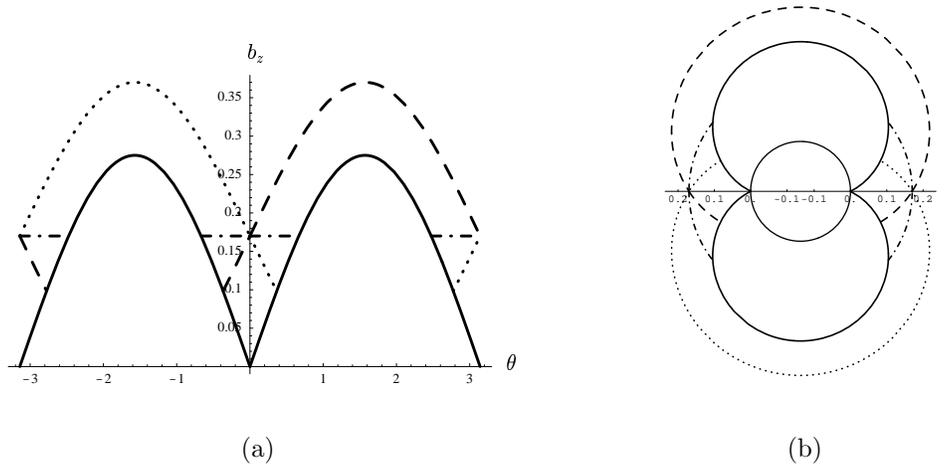


Figure 3.3: Hybrid regions in the planar flipping model in (a) the $b_z - \alpha$ plane (b) polar coordinates with angle α and radius b_z using RHex’s morphology (see Table 3.2). Solid lines indicate body ground contact for the nose ($\alpha < 0$) and the tail ($\alpha > 0$). The liftoff transitions of the front, middle and back legs are represented with dotted, dash-dot and dashed lines, respectively.

3.3.3 The 1DOF Planar Model

Our subsequent analysis mainly concerns configurations where the tail of the body is in contact with the ground and all the legs are positioned such that $0 < \gamma_i < \pi/2$.

Our controller design in Section 3.4 respects these constraints by proper choice of control inputs, reducing the system to only one degree of freedom: the body angle α with respect to the horizontal. As a convention, we assume that the tail coincides with the origin.

The foot position and leg orientation can then be expressed as functions of α ,

$$\gamma_i = \text{asin} \left[\frac{d_i + d}{l} \sin \alpha \right] \quad (3.4)$$

$$y_i = (d_i + d) \cos \alpha + l \cos \gamma_i . \quad (3.5)$$

We will also find it convenient to write the leg contact constraints of (3.3) in functional form for this low degree of freedom model, $s_c : \mathcal{C} \rightarrow \mathcal{H}_l$, with the component for the i^{th} leg specified as

$$s_c(\alpha)_i = \begin{cases} 1 & \text{if } \sin \alpha \leq l/(d + d_i) \\ 0 & \text{otherwise} \end{cases} .$$

In the context of this low degree of freedom model, we will refer to the dynamical state of the system, \mathbf{q} , as the *body state* and the discrete leg touchdown configuration as the *contact state*.

3.3.4 Continuous Dynamics

In this section, we derive the contact constraint forces and the vector field for the constrained model of Section 3.3.3, for a particular choice of contact state, assuming that the legs that are touching the ground as well as the tail of the body are vertically constrained in both directions (i.e. the ground reaction force can be negative as well as positive). We then present the final form of the equations of motion using the actual contact state $s_m(\mathbf{q}, \boldsymbol{\tau}) \leq s_c(\alpha)$, defined in Section 3.3.5 to yield the continuous dynamics for our model.

Figure 3.4 illustrates the free body diagrams for the body link as well as one of the legs in contact with the ground. Solving the force and moment balance equations

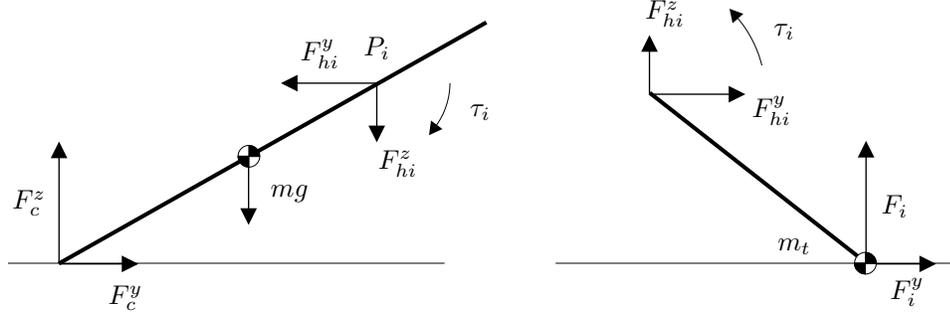


Figure 3.4: Free body diagrams for the body and one of the legs.

for leg i , we have,

$$(l \cos \gamma_i + l \bar{\mu}_i \sin \gamma_i) F_i = l m_t a_i^y \sin \gamma_i - l m_t b_i^y \ddot{\alpha} \sin \gamma_i - \tau_i + k_d \dot{y}_i, \quad (3.6)$$

where $\bar{\mu}_i := -\mu \text{sign}(\dot{y}_i)$ is defined as the effective Coulomb friction coefficient, k_d is the frictional damping constant and $\dot{y}_i = a_i^y - b_i^y \dot{\alpha}$ is obtained by differentiating (3.5) twice.

Combined with the moment balance for the body link around the tail, instances of (3.6) for each leg that can reach the ground result in a linear set of equations whose solution yields the dynamics.

The number of these equations, however, varies based on the value of $s_c(\alpha)$. The following presentation assumes that all the legs can reach the ground, i.e. $s_c(\alpha) = [1, 1, 1]$, but the readers should note that there are $2^3 = 8$ different cases for different contact states. According to the free body diagram, we may write

$$\mathbf{A}_p(\mathbf{q}) \mathbf{v} = \mathbf{b}(\mathbf{q}, \boldsymbol{\tau}) \quad (3.7)$$

where $q \in \mathcal{Q}$, $p = [p_1, p_2, p_3] \in \mathcal{H}_l$ is an arbitrary contact state such that $p \leq s_c(\alpha)$

and the arrays are specified as follows:

$$\mathbf{A}_p(\mathbf{q}) := \begin{bmatrix} f_1 & 0 & 0 & lm_t b_1^y \sin \gamma_1 \\ 0 & f_2 & 0 & lm_t b_2^y \sin \gamma_2 \\ 0 & 0 & f_3 & lm_t b_3^y \sin \gamma_3 \\ p_1 y_1 & p_2 y_2 & p_3 y_3 & -(I + md^2) \end{bmatrix} \quad (3.8)$$

$$\mathbf{b}(\mathbf{q}, \boldsymbol{\tau}) := \begin{bmatrix} lm_t a_1^y \sin \gamma_1 + k_d \dot{y}_1 - \tau_1 \\ lm_t a_2^y \sin \gamma_2 + k_d \dot{y}_2 - \tau_2 \\ lm_t a_3^y \sin \gamma_3 + k_d \dot{y}_3 - \tau_3 \\ mgd \cos \alpha \end{bmatrix}$$

$$\mathbf{v} := \begin{bmatrix} F_1 & F_2 & F_3 & \ddot{\alpha} \end{bmatrix}^T$$

$$f_i := l \cos \gamma_i + l \bar{\mu}_i \sin \gamma_i. \quad (3.9)$$

Lemma 3.2. *Suppose $0 < \alpha < \pi/2$, $\dot{\alpha} > 0$ and $0 < \gamma_i < \pi/2$ (the normal range of operation for our thrust controller). Then the matrix $\mathbf{A}_p(\mathbf{q})$ is full rank.*

Proof. We first rewrite $\mathbf{A}_p(\mathbf{q})$ in the form of four column vectors

$$\mathbf{A}_p(\mathbf{q}) = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix}$$

For the given range of states, the foot locations y_i are clearly positive. Moreover, as a consequence of (3.9), together with the assumption that $\dot{\alpha} < 0$ we have $f_i > 0$. Finally, differentiation of (3.5) also yields $b_i^y > 0$.

Now, suppose $\sum_{i=1}^{i=4} c_i w_i = 0$ for some $c_i \in \mathbb{R}$. The first three rows of this vector sum yield

$$c_i = -\frac{c_4 lm_t b_i^y \sin \gamma_i}{f_i} \quad i = 1, 2, 3.$$

Substitution in the fourth row of the vector sum reads

$$-c_4 \left(lm_t \sum_{i=1}^{i=4} \frac{b_i^y \sin \gamma_i}{f_i} p_i y_i + I + md^2 \right) = 0.$$

Clearly, the sum within the parentheses is strictly positive, yielding $c_i = 0$ for $i = 1, \dots, 4$. Consequently, the column vectors w_i are linearly independent and the matrix $\mathbf{A}_p(\mathbf{q})$ has full rank. \square

As a consequence of Lemma 3.2, the matrix $\mathbf{A}_p(\mathbf{q})$ is always invertible in the range of operation for our controller. Consequently, the solution to (3.7) yields the ground reaction forces on the legs as well as the vector field for the particular contact state choice p ,

$$\mathbf{v}_p(\mathbf{q}, \boldsymbol{\tau}) := \mathbf{A}_p(\mathbf{q})^{-1} \mathbf{b}(\mathbf{q}, \boldsymbol{\tau}) . \quad (3.10)$$

The final form of the equations of motion uses the actual contact state, $p = s_m(\mathbf{q}, \boldsymbol{\tau})$, defined in Section 3.3.5,

$$\ddot{\alpha}(\mathbf{q}, \boldsymbol{\tau}) = [0 \ 0 \ 0 \ 1] \mathbf{A}_{s_m(\mathbf{q}, \boldsymbol{\tau})}(\mathbf{q})^{-1} \mathbf{b}(\mathbf{q}, \boldsymbol{\tau}) \quad (3.11)$$

and only depends on the current body state and the torque input vector, the value of the contact state already being determined.

3.3.5 Hybrid Leg Contacts

Given the current state \mathbf{q} , we can “read off” from $s_c(\alpha)$ the number of kinematically possible leg contacts so as to determine the dimension of the square array $\mathbf{A}_p(\mathbf{q})$ in (3.7). However, only when a specific set of torques, $\boldsymbol{\tau} \in \mathbb{R}^3$, is imposed at a specified body state, $\mathbf{q} \in \mathcal{Q}$, can we determine the actual leg contact state according to the function, $s_m : \mathcal{Q} \times \mathbb{R}^3 \rightarrow \mathcal{H}_l$, and complete the specification of the dynamics in (3.11). The following definitions will be important in the computation of s_m and subsequent discussions:

Definition 3.2. $p \in \mathcal{H}_l$ is called *consistent* at a particular state \mathbf{q} and for a given control input vector $\boldsymbol{\tau}$, denoted $\text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(p)$, if and only if

$$\forall i (p_i = 1) \implies F_i(\mathbf{q}, \boldsymbol{\tau}, p) > 0$$

Definition 3.3. Let $p \in \mathcal{H}_l$ be a contact state. p is *maximal* at \mathbf{q} and $\boldsymbol{\tau}$, denoted $\text{maximal}_{[\mathbf{q}, \boldsymbol{\tau}]}(p)$, if and only if

$$\text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(p) \implies (\forall r \in \mathcal{H}_l \text{ cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(r) \implies (p \geq r)) \quad (3.12)$$

We prove the existence and uniqueness of the maximal contact state in Appendix D. The following Theorem states a very important consequence of maximality.

Theorem 3.3. *If $p \in \mathcal{H}_l$ is the maximal contact state at \mathbf{q} for a given $\boldsymbol{\tau}$, then*

$$\forall r \in \mathcal{H}_l, (r \leq p) \implies (\ddot{\alpha}_r(\mathbf{q}, \boldsymbol{\tau}) \leq \ddot{\alpha}_p(\mathbf{q}, \boldsymbol{\tau})).$$

Proof. This proof is a continuation of the existence and uniqueness proof for the maximal contact state (see Appendix D). It follows trivially from the last row of the vector equality (D.1). \square

Finally, the following assumption constitutes the basis of our hybrid leg contact model.

Assumption 3.4. The contact state of the system is the maximal contact state for its current body state \mathbf{q} and the control torque vector $\boldsymbol{\tau}$.

Based on this assumption, the following algorithm computes the maximal and hence the actual contact state for a given state \mathbf{q} and torque vector $\boldsymbol{\tau}$.

Algorithm 3.1. (Definition of $s_m : \mathcal{Q} \times \mathbb{R}^3 \rightarrow \mathcal{H}_l$)

For a given state $\mathbf{q} \in \mathcal{Q}$ and control inputs $\boldsymbol{\tau} \in \mathbb{R}^3$, this iterative algorithm determines a consistent contact state assignment which is also maximal.

1. Start with an initial assumption for the leg contact states based on the kinematic constraints.

$$p^0 = s_c(\alpha)$$

2. Using (3.7), compute the ground reaction forces $F_i(p^k)$ resulting from the current leg contact state assignment p^k .
3. If $\forall i, F_i(p^k) > 0$, p_k is the actual touchdown state, stop the iterations. Otherwise, proceed with the next step.
4. Choose the next leg touchdown states to be considered as follows.

$$p_i^{k+1} = \begin{cases} p_i^k & \text{if } F_i(p^k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, 2, 3$$

5. Go to step 2 with $k \leftarrow k + 1$

3.3.6 Leg-Ground Collisions

The flipping behavior described in Section 3.2.1 involves collisions of the front legs with the ground. In order to recover as much of the impact kinetic energy as possible before each thrust cycle, our controllers position the front leg vertically prior to impact, resulting in the radial compliance of the leg to do most of the work. The vertical placement also avoids slippage of the leg as well as friction losses and eliminates the need for the motor to apply any torque during the collision due to the kinematically singular configuration. Moreover, during the decompression of the front leg, the middle and back legs can still apply additional thrust to inject energy even during the collision. We will find it convenient to summarize the results of the entire front leg stance phase via an effective “coefficient of restitution” model whose nature we now detail.

In order to derive an accurate model of the collision, it would be possible to extend the continuous dynamics of Section 3.3.4 to incorporate compliance and other dynamical reaction forces of the front leg so as to construct a “stance phase” model that might then be integrated to obtain a more accurate prediction of the body kinetic energy returned at the next leg liftoff event. Examples of such predictive impulse

models can be found in the literature [34]. However, the accuracy of such models is still hostage to the difficulty of determining the dynamic properties of materials as well as other unmodeled effects [19].

In consequence, we have chosen to incorporate a purely algebraic collision law in our model, where a single coefficient of restitution summarizes — as a function of the configuration, α and the “average torque magnitude” during the collision stance phase — the incremental effects of leg compression/decompression and additional thrust from the middle and back legs.

The following assumptions underlie the construction of our collision law.

Assumption 3.5. If a leg is in flight, its angular velocity relative to the body is always zero ($\dot{\phi}_i = 0$), but its position can be arbitrarily specified under the constraint that the toe cannot penetrate the ground.

Assumption 3.6. A collision occurs whenever the body is falling ($\dot{\alpha} < 0$), and a leg previously in flight comes into contact with the ground either by explicit positioning through ϕ_i or by the virtue of changes in the configuration α as a result of the dynamics.

Accurate modeling of multiple simultaneous collisions is a very fragile and somewhat ill-posed problem [20]. Our flipping controller, due to the very particular sequence of leg placements that it enforces, never encounters multiple simultaneous collisions, motivating the following assumption.

Assumption 3.7. Multiple simultaneous collisions are not allowed.

In RHex, the rebounding of the body as a result of leg ground collision mainly involves the radial compression and decompression of the leg.

Assumption 3.8. During the collision, we assume that $\tau_i = 0$ and all the contact forces are due to the radial compression of the leg. As a consequence, the impulsive force at the contact point acts along the leg.

When the radial leg direction lies outside the friction cone, the required frictional impulse would be greater than the Coulomb friction force, violating one of the basic constraints of algebraic collision laws [19]. Even though the flipping controller always chooses leg angles prior to collision to satisfy this constraint, for completeness, we augment our model to use an incremental approach through the dynamics of (3.11) for collisions outside the domain of the algebraic law.

Assumption 3.9. If the leg touches the ground outside the friction cone (i.e. $|\tan(\alpha + \phi_i)| > \mu$), then the leg immediately starts slipping and transitions into stance without any impulsive collisions. The system velocities remain continuous ($\dot{\alpha}^+ = \dot{\alpha}^-$).

Under these assumptions, our collision law models the damping losses arising from the compression and decompression of the front leg as well as the additional thrust provided by the middle and back legs. We assume that these losses can be lumped into a single coefficient of restitution $-1 \leq k_r(\gamma_c)$ as a function of the toe angle at the onset of collision, γ_c ,

$$\dot{\alpha}^+ = -k_r(\gamma_c) \dot{\alpha}^- . \quad (3.13)$$

Note that as a result of Assumption 3.9, $k_r(\gamma_c) = -1$ whenever $|\cot(\gamma_c)| > \mu$.

3.3.7 Body Contact Forces During Collision

The collision model described in Section 3.3.6 implicitly assumes that the impulsive forces necessary to yield the discontinuous change in the body velocity are applied both at the toe and the body-ground contact point. However, the constraints imposed by these contact points are unilateral and only allow ground reaction forces with positive vertical components. If the necessary impulsive forces violate this constraint, the collision model becomes invalid and some unmodeled behavior (such as the tip of the body leaving the ground) will arise. In this section, we derive the ground reaction forces arising from the collision and investigate the validity of the

model as a function of kinematic robot design as well as the leg configuration at the time of collision.

The algebraic law of (3.13) can only be realized through impulsive forces both at the toe and at the body-ground contact point. Figure 3.5 illustrates the impulsive contact forces effective during the collision. Note that P acts along the leg as a result of Assumption 3.8.

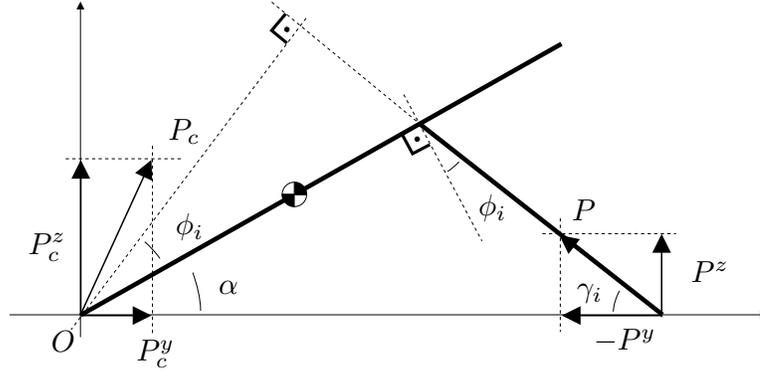


Figure 3.5: Impulsive contact forces during leg-ground collision.

Moment balance around the point O , yields the leg contact impulse P ,

$$P = \frac{(I + md^2)(\dot{\alpha}^+ - \dot{\alpha}^-)}{(d + d_i) \cos \phi_i}.$$

The step change in the translational body velocities yield the body ground contact impulses as

$$\begin{aligned} P_c^y &= -md \sin \alpha (\dot{\alpha}^+ - \dot{\alpha}^-) + P \cos \gamma_i \\ P_c^z &= md \cos \alpha (\dot{\alpha}^+ - \dot{\alpha}^-) - P \sin \gamma_i. \end{aligned}$$

Substituting the algebraic collision model of (3.13), we obtain

$$\begin{aligned} P_c^y &= \left(-md \sin \alpha + \frac{(I + md^2)}{(d + d_i) \cos \phi_i} \cos \gamma_i \right) (1 + k_r(\gamma_c)) \dot{\alpha}^- \\ P_c^z &= \left(md \cos \alpha - \frac{(I + md^2)}{(d + d_i) \cos \phi_i} \sin \gamma_i \right) (1 + k_r(\gamma_c)) \dot{\alpha}^- \end{aligned}$$

As a result of the unilateral contact constraint, our collision model is only valid when $P_c^z > 0$. Hence, the regions where this collision model is valid are given by the inequality

$$md \cos \alpha - \frac{(I + md^2)}{(d + d_i) \cos \phi_i} \sin \gamma_i > 0 .$$

We can write this inequality as a function of only the configuration of the system at the time of collision and the kinematic robot parameters, yielding

$$\frac{d_i + d}{l} \cos^2 \alpha + \sqrt{1 - \frac{(d + d_i)^2}{l^2} \sin^2 \alpha} \cos \alpha - \frac{I + md^2}{mld} > 0 . \quad (3.14)$$

3.4 An Improved Controller

3.4.1 Constraints on the Control Inputs

Given a particular contact state p , ground reaction forces on the toes can be determined using (3.6). Similarly, we can compute the contact force on the tail,

$$F_c^z = [-p^T \quad md \cos \alpha] \mathbf{v}_p(\mathbf{q}, \boldsymbol{\tau}) + mg - md \sin \alpha \dot{\alpha}^2 .$$

To preserve consistency with the assumed contact state in a physically realistic way, all of these ground reaction forces must be positive, inducing limitations on the set of input torque vectors. We capture these constraints and the practical limits on the magnitude of the input torques with the following definition.

Definition 3.4. For a particular state $\mathbf{q} \in \mathcal{Q}$ and a contact state $p \in \mathcal{H}_l$, we define the corresponding set of allowable torques, $\mathcal{T}(\mathbf{q}, p)$ as the set of all torque input vectors $\boldsymbol{\tau} \in \mathbb{R}^3$ such that

$$\begin{aligned} F_c^z(\mathbf{q}, \boldsymbol{\tau}, p) &\geq 0 \\ \forall i, F_i(\mathbf{q}, \boldsymbol{\tau}, p) &\geq 0 \\ \forall i, |\tau_i| &\leq \tau_{max} \end{aligned}$$

3.4.2 Maximal Thrust Control

In this section, we present a torque control strategy which maximizes the thrust while respecting the body contact constraint as well as limitations on the torque deliverable by the hip actuators.

The vector field (3.11) is a continuous function of the state and the input torques. As a consequence, the problem of choosing hip controls to maximize the thrust becomes a constrained optimization problem over the allowable input torque space, given the current system state and the torque limit constraints. However, different regions in the state space with different leg contact state assignments result in a vector field which is a nonlinear function of the input torques, yielding a computationally demanding nonlinear optimization problem.

Fortunately, in each of the distinct leg contact states, the optimization problem is linear in the control input torques. Hence, the problem decomposes into a small number of separate linear programming problems, from whose independent solutions may be derived a single correct torque value for the three hips. Namely, the optimal torque vector is computed for each distinct leg contact state vector compatible with the kinematic constraints. Then, putative motor torques determined for each kinematically possible leg contact, the unique physical contact state vector is determined by recourse to Algorithm 1, and the command torque vector to each motor determined. More formally, given a leg contact state vector, $p \in \mathcal{H}_l$, we pose the corresponding linear programming problem for that region of configuration state space:

$$\begin{aligned} \boldsymbol{\tau}_M(\mathbf{q}, p) &:= \operatorname{argmax}_{\boldsymbol{\tau} \in \mathcal{T}(\mathbf{q}, p)} (\ddot{\alpha}(\mathbf{q}, \boldsymbol{\tau}, p)) \\ \ddot{\alpha}_M(\mathbf{q}, p) &:= \ddot{\alpha}(\mathbf{q}, \boldsymbol{\tau}_M(\mathbf{q}, p), p) . \end{aligned}$$

The set of contact state assignments that we need to consider is determined by the kinematic constraints, $\mathcal{P}_{\mathbf{q}} := \{p \in \mathcal{H}_l \mid s_c(\alpha) \geq p\}$. The solution to the global

problem then becomes,

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{q}, p_{max}) \tag{3.15}$$

$$p_{max} := \operatorname{argmax}_{p \in \mathcal{P}_{\mathbf{q}}} (\ddot{\alpha}_M(\mathbf{q}, p)) .$$

Note that p_{max} is also maximal at the current body state and with the torque solution to the above optimization problem as a result of Claim 3.3. Consequently, the actual contact state determined by the algorithm of Section 3.3.5 will necessarily match p_{max} , that is

$$s_m(\mathbf{q}, \boldsymbol{\tau}(\mathbf{q}, p_{max})) = p_{max} .$$

3.4.3 Hybrid Energy Pumping

Depending on the frictional properties of the surface, our maximal thrust controller may or may not be enough to complete the flip. In cases where it fails to achieve the sufficient energy level in the first attempt, our controller uses the same strategy as the first generation controller presented in Section 3.2.1. Once the body starts falling, the new controller waits until the front legs collide with the ground and repeatedly applies maximal thrust following each collision.

Currently, we have very little analytical understanding of the behavior arising from this hybrid controller. As a consequence, we only explore in simulation the flipping behavior and its dependence on various surface parameters in the following sections.

3.5 Simulations

In this section, we use simulations based on the dynamical model of Sections 3.3.4 and 3.3.5 to evaluate the performance of the maximal thrust control approach combined with the hybrid energy pumping strategy.

3.5.1 Apex Return Maps

Figure 3.6 illustrates an example simulation of a successful flip with multiple hops. Note that in the last cycle, as the robot “escapes” from the basin of attraction of the initial configuration, the point of inflection at $t \approx 4.5$ corresponds to the configuration where the downward torque from gravity is just balanced by the rear leg thrusting torque.

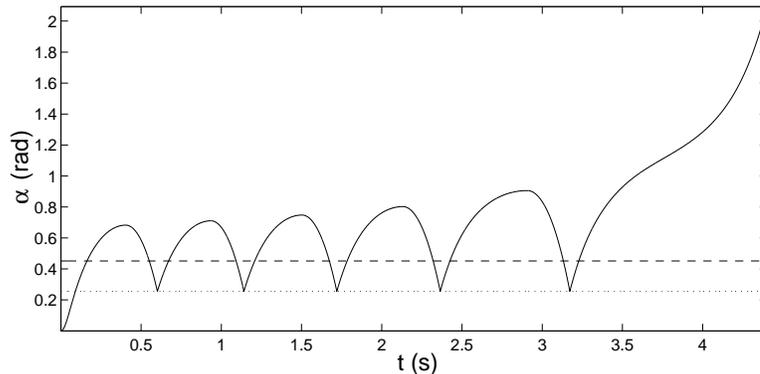


Figure 3.6: An example simulation of (3.11) with RHex parameters (see Table 3.2) and $\mu = 0.8$, $k_d = 11$ and $k_c = 0.75$; using the maximal thrust feedback controller (3.15). Dotted and dashed lines indicate the kinematic liftoff constraints for the front and middle legs, respectively.

In presenting the properties of the hybrid pumping strategy, we will find it most convenient to sample the α trajectory at its highest point (apex) during each cycle. This will result in a one dimensional return map, characterizing the behavior of the energy pumping strategy under the maximal thrust actuation. This section explores on the basis of numerical simulation some of the properties of this return map and their dependence on various surface parameters and the coefficient of restitution. Formal analysis of this model is presently in progress.

Figure 3.7 illustrates different types of return maps resulting from different choices of the surface parameters μ , k_c and k_d . This collection of return maps appears to capture all the possible types of phenomena that arise from our hybrid controller.

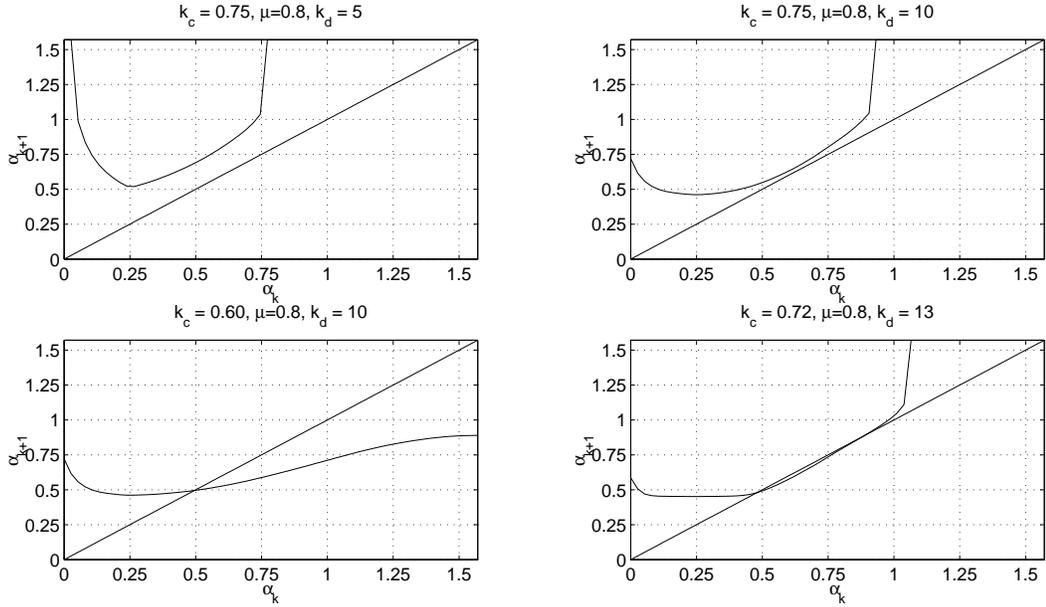


Figure 3.7: The predicted range of physical behaviors based upon numerical return maps computed for (3.11) with representative surface parameter settings using the maximal thrust feedback controller (3.15).

The upper left case has low ground friction and hence the initial thrust is sufficient to flip the robot body over. Moreover, there is no fixed point, so even if the robot were to start off the ground, it would always be able to flip. In contrast, the upper right case has enough friction to make flipping in one thrust impossible, but still has no fixed point, yielding successful flipping after several hops.

The bottom two cases have qualitatively different behavior. In contrast to the previous cases, they have stable fixed points, trapping the robot at a small angle. For the bottom left case, there is no other fixed point, which makes it impossible for the robot to flip. The bottom right case, however, also has another unstable fixed point, making a successful flip possible for initial conditions above the fixed point.

3.5.2 Maximal Thrust vs PD Control

Results of simulation runs for a range of surface friction parameters are illustrated in Figures 3.8 and 3.9, for $k_c = 0.75$ and $k_c = 0.9$, respectively. The simulations were run over a range of surface friction properties, until either the robot flipped over or at the end of 50 hops. A particular attempt was considered a successful flip if the body angle reached $\pi/2$ before the end of the simulation (labeled *single thrust* and *multiple hops* in the plots), or the sequence of apex heights kept increasing even in the last hops (indicating a weakly unstable fixed point in the return map, labeled *expected flip* in the plots). All other runs were considered failed flip attempts.

One of the reasons for the choice of such high coefficients of restitution is the active nature of the collisions we briefly describe in Section 3.3.6. In this context, higher coefficient of restitutions reflect the additional thrust occurring exerted during the decompression of the front leg. On RHex, we observed the duration of the collision to be significant, increasing the effect of this active phase of the collision. Actual values of the coefficient of restitution still remain to be experimentally verified.

These results demonstrate that maximal thrust control yields considerably better flipping performance than the PD control in all cases. For smaller k_c , where the “active” collision is not properly modeled, the PD controller never succeeds with multiple hops and only has a chance when the first thrust is sufficient. When the effects of the active collision are incorporated through the coefficient of restitution, the maximal thrust controller is still successful in a very large range of surface conditions and yields strictly better results than the PD control.

3.6 Conclusion

In this chapter, we have developed a family of controllers to implement self-righting behavior on RHex, which is perhaps the simplest instance of self-manipulation other

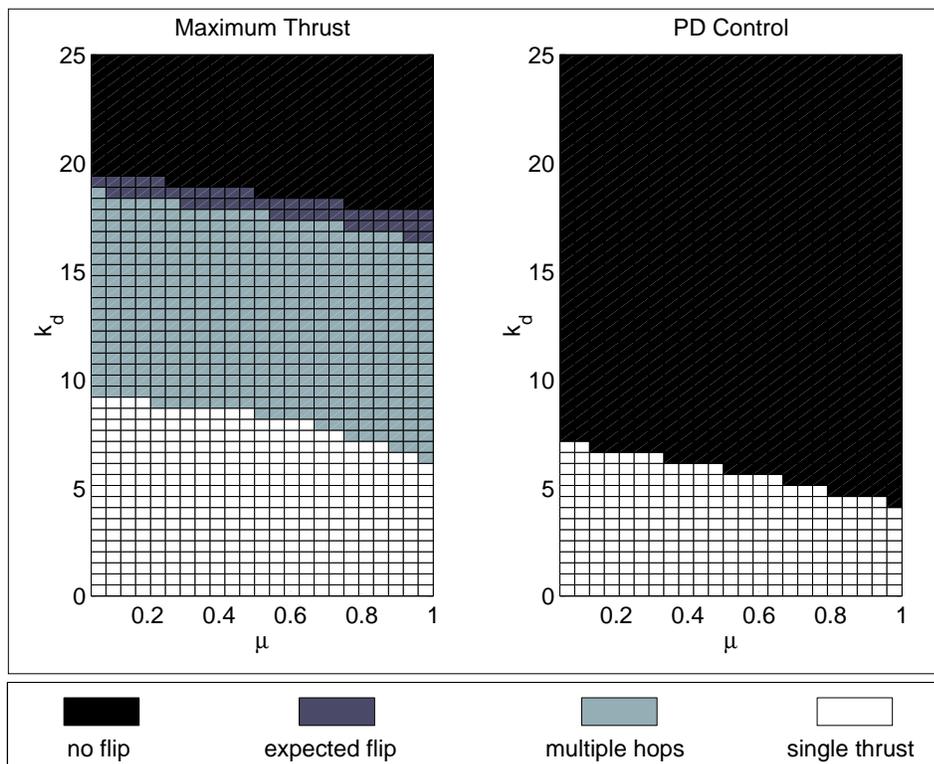


Figure 3.8: Outcomes for flipping attempts with lower coefficient of restitution, $k_c = 0.75$.

than locomotion itself. In doing so, we have also demonstrated a simple instance of a template dynamical system, which greatly facilitated the characterization of controller stability and performance. This first example use of templates and their embedding demonstrates the advantages of the decomposition of the task coordinates from the extra degrees of freedom.

In addition to the concepts presented in this chapter, there is substantial future work to be done for the flipping behavior. Most importantly, more formal analysis of the preliminary model we have described is of great interest. Extensions of the flipping behavior such as uninterrupted rolling or handstands will require a much better analytical understanding of the model as well as modifications such as relaxing the friction constraint on the body. We believe that, such extensions to the behavioral

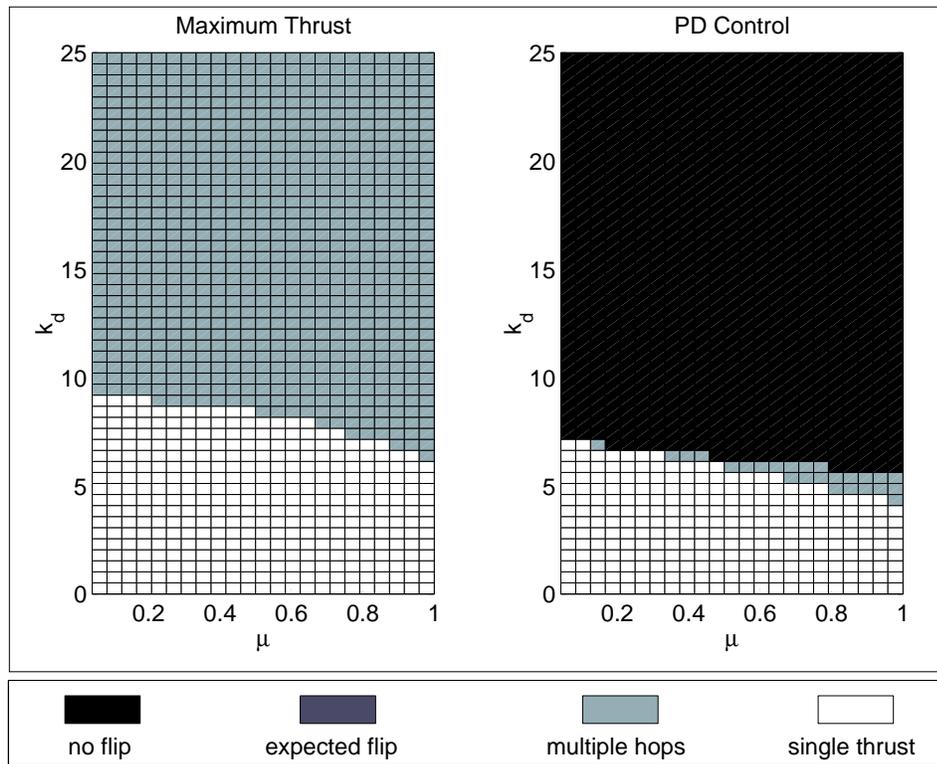


Figure 3.9: Outcomes for flipping attempts with higher coefficient of restitution, $k_c = 0.9$.

suite of a morphology as limited as RHex, is the best way to address the shortcomings of contemporary actuation and energy storage technology while continuing to press ahead in the development of practically useful robots.

CHAPTER 4

The Basic Planar Spring-Loaded Inverted Pendulum

4.1 Motivation

Biomechanists have gained great leverage in understanding basic principles of locomotion in creatures as diverse as humans and cockroaches by considering the basic Spring-Loaded Inverted Pendulum (SLIP) model shown in Figure 4.1 as a metaphor for running and hopping [2, 10, 11, 24]. While simple to the biomechanist, even this model presents difficulties to the engineer wishing to pursue formal analysis and control since it is a hybrid system with nonlinear stance dynamics which are not closed-form integrable. Even so, previous work provides approximate functional relationships for the SLIP dynamics, enabling a consideration of control via established techniques [71, 70]. With the help of such formal models, growing biological evidence also suggests that the SLIP model may be more than just a descriptive model [69], but rather a literal control target for a wide range of running animals. In this chapter, we will pursue this idea in the context of robotics and present ways in which the descriptive power of this model can be extended to enable synthesis of effective running controllers for robots.

The first work in the control of SLIP runners was the successful implementation

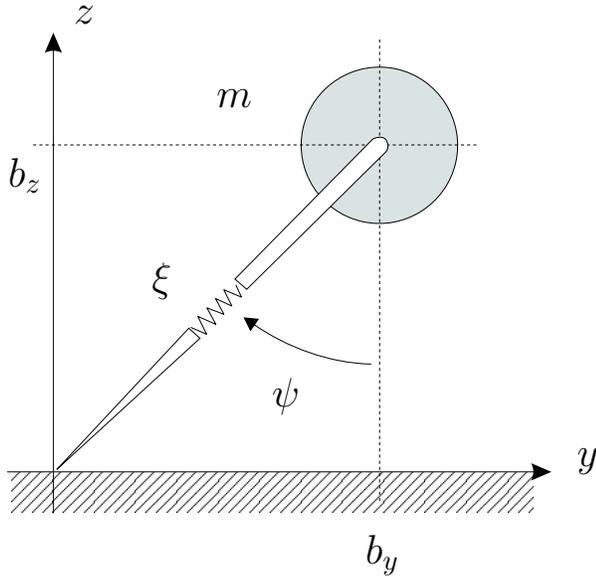


Figure 4.1: The basic Spring Loaded Inverted Pendulum(SLIP) leg model.

by Raibert and his students [60] of simple, roughly decoupled controllers to independently control the hopping height and forward velocity of their robots. This stunning success motivated a series of papers [44, 49, 70, 77] characterizing the stability of these decoupled controllers.

Along these lines, the first part of this chapter presents a new coupled approximate deadbeat controller for a SLIP runner, having a “special” spring potential model which makes a simplified version of the stance dynamics closed-form integrable. The resulting discrete closed loop system yields a dynamically informed task interface and forms the template dynamics that we use as a task level representation of planar running. This interface consists of three parameters: forward velocity, hopping height and duty factor, which are intuitive and very useful when considering higher level control problems such as foot placement on irregular terrain [41].

For the special SLIP runner under consideration, a return map giving the next apex state (hopping height, forward velocity and ratio of flight to stance times) is derived in terms of the previous apex state and the control inputs. Our control

strategy is to solve the deadbeat problem - that is invert the return map to give the control necessary to achieve a desired next apex state, and add integral compensation to improve robustness. Even though there are currently no analytical stability results for the deadbeat controller, in this chapter, we explore its performance and stability properties via simulation.

The second part of the chapter concentrates on the control of locomotion for a more complicated leg morphology — a four degree of freedom planar leg — using the basic SLIP controller as its task level command interface. Structurally, our controller is very similar to Raibert’s work on controlling complex legged machines and models through basic SLIP controllers [61]. Our design also follows the decompositional nature of the template based control paradigm and attempts to isolate the task degrees of freedom from the “shape” variables. To this end, we introduce a “posture principle” as a low dimensional attracting manifold in the overall state space wherein the SLIP dynamics are embedded. As a means of evaluating the overall performance of the resulting locomotion controller, comparative simulation studies are also presented. Most of the work presented in this chapter was also published in a conference paper [65].

4.2 The Basic SLIP

4.2.1 System Model and Assumptions

The basic SLIP model considered in this chapter is shown in Figure 4.1. The leg is assumed massless and the body a point mass at the hip joint. During stance the leg is free to rotate around its toe and the mass is acted upon by a radial spring with potential $U(\xi)$. In flight, the mass is considered as a projectile acted upon by gravity. We assume there are no losses in either the stance or flight phases.

During locomotion, the system alternates through two phases for each stride: stance phase, when the foot is on the ground and the flight phase when the leg is

airborne. The stance phase can be further decomposed into two parts, compression and decompression. Similarly, the flight phase can be decomposed into the ascent and descent subphases. Four important events mark the transitions between these subphases: Touchdown, the instant when the leg comes into contact with the ground; Bottom, the instant when the leg reaches maximal compression during stance; Liftoff, the moment when the leg reaches maximum extension at the end of stance and finally, Apex, the instant when the body reaches its maximum height during flight.

The control inputs available to the SLIP runner are discrete in nature. Similar to the control strategies used in Raibert’s hoppers [60], the first control input is the leg angle at touchdown, ψ_t . We assume that during flight we are able to swing the leg to any desired angle relative to the ground. The remaining control inputs arise from the ability to tune the spring potential. We choose to change the spring potential via choice of the stance compression and decompression spring constants, k_c and k_d , respectively, assuming that an instantaneous change can be achieved at bottom¹.

4.2.2 Parametrization of Equilibrium Gaits

The SLIP model is a relatively simple and low dimensional model, established in the biomechanics literature to capture the essence of running behavior of diverse animals [2, 10, 24, 26]. As a consequence, from an engineering perspective, it is the most obvious candidate in encoding the task of running as a control objective. However, the model’s utility as a formal representation requires an explicit parametrization to the set of equilibrium trajectories, not readily provided by the dynamics alone.

The notion of equilibrium gaits was recognized by Raibert [60], who identified “neutral orbits” in running animals with symmetrical stance trajectories. Recently, a more formal treatment of the SLIP model, together with a complete characterization

¹Clearly, this scheme is somewhat unrealistic for practical applications. More realistic alternatives such as the use of leg precompression at touchdown as well as forced leg liftoff to affect the total energy level are possible and will be introduced in later chapters.

of all possible neutral gaits was done by Schwind [69]. In this section, I will give an overview of this work and its application to my work.

Achieving symmetrical orbits is only possible with trajectories that go through a vertical leg angle at bottom, the time of maximal leg compression. Moreover, these trajectories also are the fixed points of an associated “return map” from one bottom point to the next, characterized by the leg compression, ξ_b , the angular velocity of the mass relative to the fixed toe, $\dot{\psi}_b$, at bottom, as well as one of the control parameters, the leg spring constant². This particular parametrization defines what is termed internal gait description, denoted by $p_b \in \mathcal{P}_b$.

The internal gait parametrization is well suited for the characterization of equilibrium gaits as well as the analysis of the stance dynamics. However, from a robotics point of view, we would like a representation to capture externally observable attributes of running, such as forward speed and duty factor. Fortunately, a different parametrization of neutral orbits through a transverse section at the apex point yields such coordinates. The manifest gait parameters $p_a \in \mathcal{P}_a$ are hence defined to be the horizontal velocity \dot{b}_{ya} and the vertical height b_{za} at apex, together with the duty factor ϕ , the ratio of the duration of stance to the stride period.

Previous work has shown that these parameter sets are indeed suitable coordinates for the analysis and control of SLIP locomotion [69]. In the following sections, we will also find it useful to introduce two more coordinate sets: touchdown and liftoff. The touchdown parameter set $p_t \in \mathcal{P}_t$ consists of the leg angle ψ_t as well as the radial and angular velocities $\dot{\xi}_t$ and $\dot{\psi}_t$ at touchdown. Similarly, the liftoff parameter set $p_l \in \mathcal{P}_l$ consists of leg angle ψ_l as well as the radial and angular velocities $\dot{\xi}_l$ and $\dot{\psi}_l$ at liftoff.

²Note that symmetric trajectories require $k = k_c = k_d$.

4.3 SLIP Controllers

4.3.1 The Control Objective

In formulating the control problem, it is natural to work with the manifest gait parameters of the previous section. In order to capture the transient behavior of the system, we also define a set of apex states,

$$\mathcal{X}_a = \{X_a \mid X_a = [b_{za}, \dot{b}_{ya}, \phi]^T\}$$

Given this perspective, the next step is to introduce the apex return map,

$$f_a : \mathcal{X}_a \times \mathcal{U} \mapsto \mathcal{X}_a$$

where the set of control inputs is defined as

$$\mathcal{U} = \{u \mid u = [\psi_t, k_c, k_d]^T\}.$$

Suppose, now, that we want to achieve the desired apex state (control objective),

$$X_a^* = [b_{za}^*, \dot{b}_{ya}^*, \phi^*]^T. \quad (4.1)$$

The control problem, then, is to identify the sequence of discrete control inputs $\{u_i\}_{i=0}^n$ to asymptotically converge to the desired apex state.

4.3.2 The Approximate Deadbeat Controller

One possible solution to the control problem stated in Section 4.3.1 is the deadbeat control, that is, determining the control input $u^* = [\psi_t^*, k_c^*, k_d^*]^T$ such that $X_a^* = f_a(X_a, u^*)$, effectively taking the current apex state X_a to the desired state X_a^* in one stride. Unfortunately, despite the structural simplicity of the basic SLIP, its stance dynamics are not integrable. As a consequence, exact analytical inversion of the apex return map is not possible. Moreover, the dimension of the return map renders numerical inverses impractical for an online algorithm.

Nevertheless, we can introduce certain approximations to obtain tractable solutions to the inversion problem. Specifically, in our formal consideration, we eliminate gravity from the stance dynamics yielding a simple central force problem wherein energy and angular momentum are both constants of motion and can be used to integrate the stance dynamics. Moreover, the structure of the integrals suggest certain forms for the spring law which are physically realistic and also admit closed form integration [71, 70]. In particular, as in [70], we have chosen to work with the compressed air spring $U_A(\xi) := k/2(1/\xi^2 - 1/\xi_0^2)$. The procedure outlined in Appendix B gives an open loop approximate deadbeat controller for the ideal case where the plant exactly matches the SLIP model with the compressed air spring. The resulting control law is the approximate deadbeat controller.

4.3.3 The Modified Raibert Controller and Integral Compensation

For the purposes of comparison, we also introduce a decoupled alternative to the deadbeat strategy based on Raibert’s original control ideas. First, the forward velocity control is achieved by approximating a neutral leg placement and adjusting it with a proportional error term, yielding

$$\psi_t = \text{asin} \left(\frac{\dot{y}t_s}{2\xi_0} + k_y(\dot{b}_{ya}^* - \dot{b}_{ya}) \right) \quad (4.2)$$

where k_y and the choice of \dot{y} are controller parameters³. Next, we implement a Raibert-like hopping height controller by supplying the appropriate energy at bottom, via a change in spring constant $\Delta E_U = U_{k_d}(\xi_b) - U_{k_c}(\xi_b)$, in order to provide the energy difference between two successive apex points. In the absence of an estimate for ξ_b , we use measurements from previous strides. Similar to \dot{x} and k_y , this is an estimation parameter which requires careful tuning for best performance.

³In Raibert’s work, the definition of forward speed is somewhat ambiguous. Different possibilities such as the average speed over a stride or the velocity at bottom result in different performance. In our simulations, we always use the velocity at bottom, which we have found to work best.

Since both controllers, by their nature, will have tracking errors, we use integral feedback compensation, yielding a discrete closed loop system of the form

$$\begin{aligned} X_a[k+1] &= f_a(X_a[k], u_c(X_a[k], X_a^*[k+1] + e[k])) \\ e[k+1] &= e[k] + \frac{1}{c_i} (X_a^*[k] - X_a[k]) \end{aligned}$$

where $e[k]$ is the integral of the apex state error, $X_a^*[k]$ is the “reference” trace and $u_c(X_a, X_a^*)$ is a particular gait-level controller, in this case, either the deadbeat or the modified Raibert controller.

4.3.4 Simulation Studies

In this section, we investigate in simulation, the performance of the deadbeat controller in comparison to the Raibert controller which has widely and successfully been used in the literature for the control of legged locomotion. In our simulation study, we consider two families of waveforms we wish the apex velocity trace to track: one of step references and another of sinusoid references.

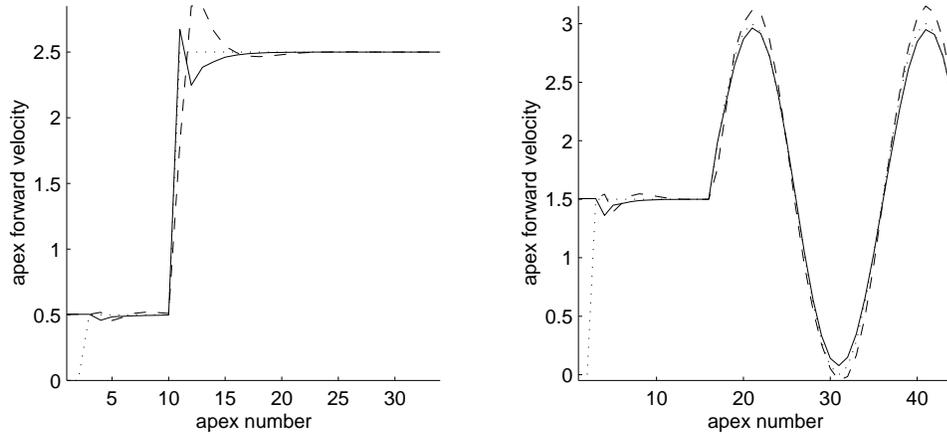


Figure 4.2: Sample runs of the deadbeat controller(solid lines) and modified Raibert(dashed lines) controller applied to the basic SLIP leg for step and sinusoid references over 35 strides. Dotted lines represent the reference trace.

Examples of both are shown in Figure 4.2. In each case, the hopper stabilizes

around an initial running speed and the desired reference waveform is introduced at the end of 15 gait cycles. In this study, we use two dimensional families of step references parametrized by their initial value and amplitude; as well as sinusoid references parametrized by their amplitude and frequency.

Simulations are run over a range of these two dimensional parameter spaces. For a particular reference command, we summarize the control performance by the mean square error (MSE),

$$\text{MSE} = \frac{1}{N} \sum_{k=15}^N \| \dot{b}_{ya}^*[k] - \dot{b}_{ya}[k] \|^2$$

where N is the total number of strides.

Furthermore, responses to these step and sinusoid reference commands are presented by collapsing the initial velocity and sinusoid amplitude dimensions by averaging. In each case, 10 data points in these dimensions are chosen such that the forward velocity command always remains in the range $[0, 3]m/s$.

Figure 4.3, summarizes the simulation results for step and sinusoid reference commands in forward velocity where we fix $b_{za}^* = 1.2m$ and $\phi^* = 3$. The plots show the mean and variance of MSE for both controllers as a function of step amplitude(left) and sinusoid frequency(right). The results show that for this plant, the deadbeat controller provides better tracking than a modified Raibert controller.

Simulations with sinusoid reference commands reveal another property of the deadbeat controller. Due to its long settling time, the tracking error of the decoupled controller increases significantly for high frequency reference commands. The deadbeat controller, however, has shorter settling times — it ideally reaches the desired trajectory in one cycle — and consequently displays better tracking over a wide range of frequencies.

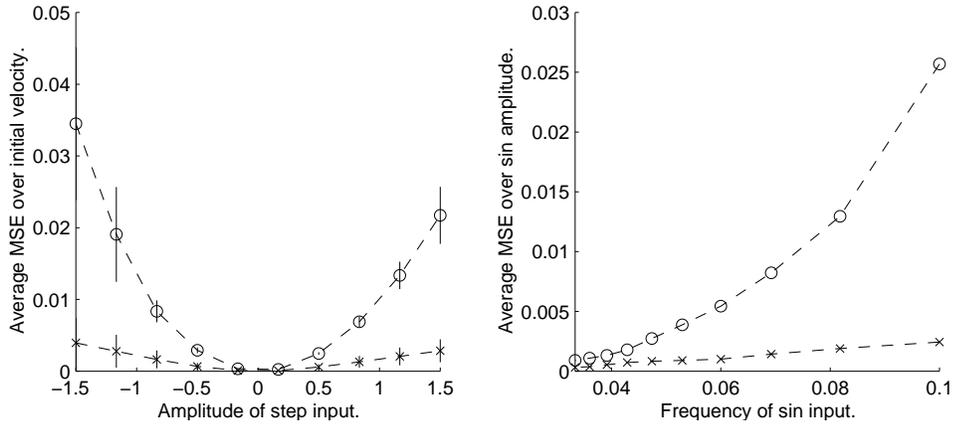


Figure 4.3: Simulation results for step(left) and sinusoid(right) references: The mean and variance of MSE as a function of the step amplitude(left) and sinusoid frequency(right), for the deadbeat(x) and modified Raibert(o) controllers. For this plant, $m = 50.48kg$, $b_{za}^* = 1.2m$, $\phi^* = 3$.

4.4 Template Control Example: Planar 4DOF Leg

In this section, we introduce a more complex dynamical leg structure, the four degree of freedom ankle-knee-hip(AKH) model (Figure 4.4). In this context, we present a new locomotion controller as an example of the template based control paradigm. We use simulation studies to demonstrate the utility of the natural decomposition arising from the paradigm, addressing two of the most characteristic problems in the control of legged locomotion: underactuated dynamics and redundancy with respect to the task coordinates.

4.4.1 The AKH Leg Model

The planar AKH model, depicted in Figure 4.4, is a four jointed planar leg with point masses on each joint. All revolute degrees of freedom except the the toe, q_f , are torque actuated and the leg is free to rotate around the toe.

Similar to the basic SLIP model, the system has two distinct phases. During the

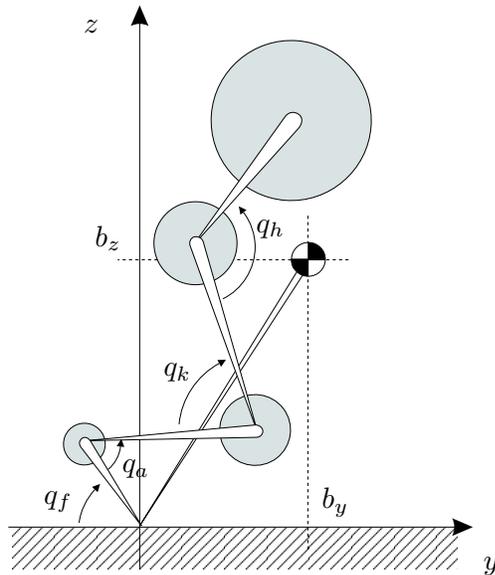


Figure 4.4: The Ankle-Knee-Hip(AKH) leg model.

stance phase, the toe is in contact with the ground, and the dynamics of the system are those of a torque actuated open kinematic chain. In contrast, during flight, we assume that the joint angles can be arbitrarily positioned and the center of mass is considered to be a projectile acted upon by gravity⁴.

4.4.2 Embedding the SLIP Template

Our ultimate goal is to be able to use controllers designed for the basic SLIP on the AKH leg without any modifications. For this purpose, we will design an *embedding* controller, which, in conjunction with the AKH system, will mimic the behavior of an ideal SLIP. This is the central idea behind the template based control paradigm, namely, reducing the dynamics of a complex system into those of a lower degree of freedom and simpler system by using the mechanics of the original system together

⁴Even though this assumption is rather unrealistic, the full dynamics of the flight phase and the required controllers to position the leg are not instrumental in our investigation of the template embedding. Our goal here is not to build a practical controller for a planar 4 DOF monopod, but rather to illustrate the use of the SLIP template as a control target.

with a properly designed embedding controller.

Towards this end, we start by defining a virtual leg connecting the toe of AKH to its center of mass. This establishes a natural association between the SLIP template and the AKH plant through identification of their centers of mass. Moreover, based on this decomposition, we can now use the control objectives of Section 4.3.1 — the achievement of desired apex height, forward velocity and duty factor — in the context of the 4 DOF leg.

Having defined the task coordinates, it still remains to map the control inputs of the SLIP template, to the torque inputs at the actuated joints of the AKH. The major difficulty in this mapping arises from the difference in the dimensions of the control affordance available to both systems. The virtual leg coordinates and the associated SLIP dynamics result in two dimensions worth of constraints, whereas there are three degrees of actuation freedom in the AKH. We define the “anchor” to be the torque level controller which achieves the desired collapse in dimension and the embedding of the template.

We address the first component in the problem – enforcing center of mass dynamics close that of a SLIP – by equalizing the mechanical work done by the virtual leg force and the joint torques. Namely, we choose the control inputs $\boldsymbol{\tau} := [\tau_a, \tau_k, \tau_h]^T$ to satisfy

$$\mathbf{F}^T \dot{\mathbf{b}} = \boldsymbol{\tau}^T \dot{\mathbf{q}}. \quad (4.3)$$

where \mathbf{F} and $\dot{\mathbf{b}}$ are the virtual spring force and the center of mass velocities, respectively. Note that this is substantially different from forcing the center of mass to follow a prespecified target trajectory. The actual stance trajectory is still governed by AKH dynamics.

Finally, we eliminate the remaining degrees of freedom by introducing a set of linear constraints on the joint angles,

$$L\mathbf{q} = l,$$

by means of which, a lower dimensional manifold in the overall state space is established. The resulting *posture principle* encodes a set of “preferred” postures, dynamically enforced by the embedding controller. In the AKH controller, this is accomplished by using PD joint controllers to minimize the posture error $L\mathbf{q} - l$, while satisfying the work constraint of (4.3). The posture principle is also used in choosing the kinematic joint angle configuration during flight, particularly important in achieving the desired virtual leg angle at each touchdown. Appendix C presents the details of this controller design as well as alternative approaches.

4.4.3 Simulation Studies

In this section, we present simulation results to investigate the performance of the embedding controller in combination with the deadbeat and Raibert task level controllers. For this purpose, we consider two different AKH legs, one human like and one kangaroo like, in conjunction with posture principles specific to each type. In particular, we use posture principles of the form

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & \beta & -1 & 0 \end{bmatrix} q = \begin{bmatrix} -\gamma \\ 0 \end{bmatrix}, \quad (4.4)$$

where β and γ are fixed parameters for any particular locomotor. Intuitively, Equation 4.4 constrains the body link angle with respect to the ground to be γ , and the knee angle to be proportional to the ankle angle. In our simulations, the human-like leg has $\gamma = \pi/2$ and $\beta = 1$ and the kangaroo-like leg, has $\gamma = \pi/4$ and $\beta = 1$. The kinematic parameters of each type of leg are given in Table 4.1.

As in Section 4.3.4 we issue step and sinusoid reference forward velocity commands and measure the tracking performance with the results shown in Figures 4.5 and 4.6, respectively. They support the validity of two of our major assumptions. First, they confirm that the SLIP model for running is applicable to significantly different kinematics and dynamics. Second, they suggest that, the connection between the

	$[m_a, m_k, m_h, m_b]$	$[l_a, l_k, l_h, l_b]$
human	[26.4, 19.3, 3.5, 1.28]	[0.15, 0.35, 0.40, 0.35]
kangaroo	[30, 30, 5, 4]	[0.5, 0.7, 0.6, 0.5]

Table 4.1: Structural simulation parameters for human-like and kangaroo-like four degree of freedom legs [40].

SLIP model and the four-jointed complex model we consider does not rely on the particular “target pose”.

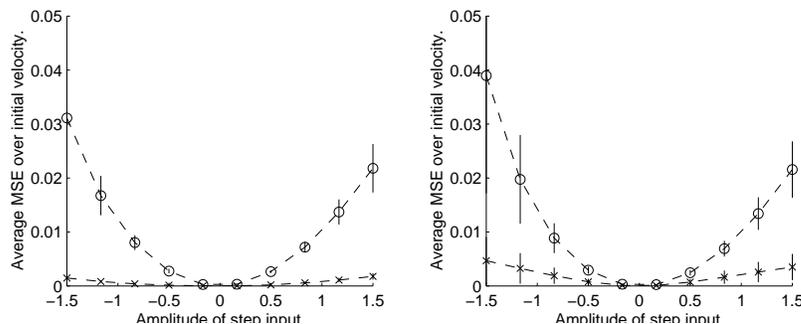


Figure 4.5: Step Reference: The mean and variance of MSE for human-like (left) and kangaroo-like (right) legs as a function of the step amplitude with the deadbeat(x) and modified Raibert(o) controllers. For this plant, $b_{za}^* = 1.2m$, $\phi^* = 3$.

4.5 Conclusion

The results presented in this chapter confirm the foresight of both those in the biomechanics community and those in the engineering community, such as Raibert, who have insisted on the utility of the SLIP model in thinking about dynamic locomotion. For not only this model useful in describing the COM behavior of a multi-jointed monopod runner as the biomechanists have claimed, but also for prescribing the control needed to achieve some of the desired behavior as Raibert originally intuited.

The extension the 2 DOF SLIP deadbeat control to the higher degree of freedom

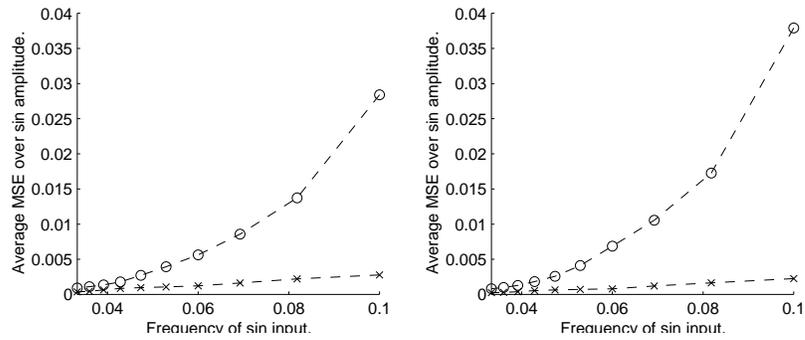


Figure 4.6: Sinusoid Reference: The mean of MSE for human-like (left) and kangaroo-like (right) legs as a function of the sinusoid frequency with the deadbeat(x) and modified Raibert(o) controllers. For this plant, $b_{za}^* = 1.2m$, $\phi^* = 3$.

AKH leg is among first successful instantiations of the dynamical template control idea within the context of legged locomotion. Both the performance of the resulting composite controller and the transparency of its task level interface demonstrate the feasibility and utility of using the simple SLIP model as a literal control target for complex locomotion tasks.

CHAPTER 5

Template Control of Planar Hexapedal Running

5.1 Motivation

This chapter presents the design of a template based locomotion controller for a planar hexapod and consists of two major parts. The first part develops the Bipedal SLIP template, which forms the task specification for both the planar and the spatial hexapods. The second part presents the embedding of the bipedal SLIP in the planar hexapod as a simple example wherein most of the key details of the controller design for the spatial case are developed.

Research on biomechanics of shows that hexapedal insects prefer the alternating tripod gaits at high speeds¹. It is characterized by the presence of two tripods, each formed by the front and back legs of one side and the middle leg of the opposite side. The tripods operate out of phase with each other, whereas the legs within a tripod are synchronized with each other. The resulting pattern is essentially a bipedal gait, where the actions of the two tripods correspond to the two legs of a biped. This gait is also the only hexapedal gait that admits duty factors lower than 50% and hence locomotion with significant aerial phases, yielding a potentially significant increase in energy efficiency. In light of these observations, we have chosen to work with

¹Even though at extremely high speeds, some cockroach species are known to adopt bipedal gaits [29], the alternating tripod is still by far the most commonly observed gait.

controllers for the planar and spatial hexapods that explicitly enforce the alternating tripod gait.

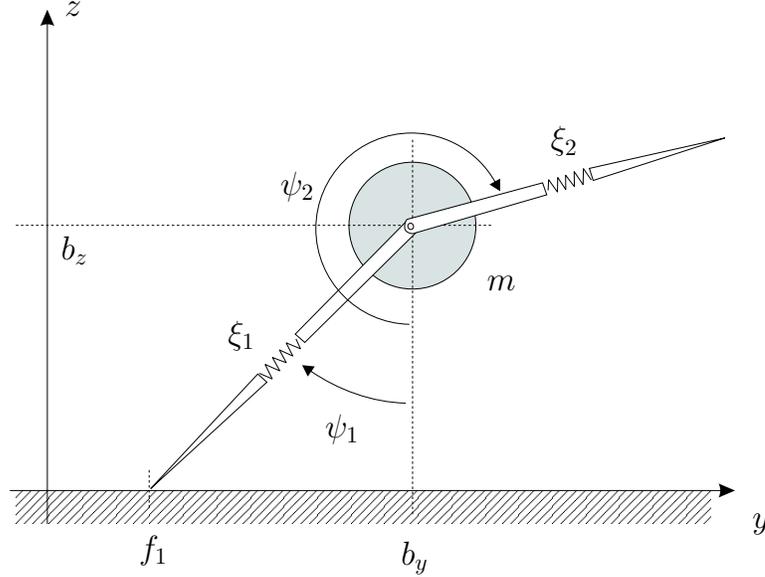


Figure 5.1: The Bipedal Spring-Loaded Inverted Pendulum(BSLIP) Model.

In an attempt to capture the characteristic features of the alternating tripod gait, the first part of this section introduces the Bipedal Spring-Loaded Inverted Pendulum (BSLIP) model, illustrated in Figure 5.1. Furthermore, it also presents an associated deadbeat gait controller, and leads towards the development of a dynamical template to be used in designing subsequent hexapedal locomotion controllers. Unlike most models in the existing literature which concentrate exclusively on the stance leg, BSLIP is designed to also admit the study of the swing leg and its impact on the overall behavior. Especially within the context of RHex, whose morphology constrains the swing legs to recirculate in order to prepare for the next stride, it supports the design of coordination algorithms to properly control the swing leg dynamics.

Later in the chapter, detailed simulation studies characterize the bipedal SLIP template's stability properties under a new deadbeat controller. Furthermore, a new set of control inputs are introduced, together with constraint parameters which are

found to have significant impact on the stability of the embedded versions of the controller. Once again, we investigate through simulation, their impact on the template performance.

The second part of the chapter concerns the embedding of the BSLIP template within Slimpod, a planar hexapod model. Even though the association between the tripods and the biped legs seems conceptually trivial, there are several issues arising from different possible contact configurations within tripods. In this context, we design a collection of continuous controllers for both the stance and flight tripods. The active embedding of the SLIP dynamics involves the stance tripod and yields the proper center of mass dynamics. On the other hand, the flight tripod mirrors the recirculation of the BSLIP flight leg and prepares its legs for the next touchdown and the following stride.

The resulting Slimpod controller is a combination of the BSLIP template and the Slimpod embedding algorithms and yields stable running over a wide range of speeds. However, in foresight of the simulation studies pertaining to the spatial hexapod, we do not present any detailed simulation results for the Slimpod. The embedding example of this section is primarily intended to build a framework toward the development of the full spatial controller and a detailed treatment of the Slimpod model is not of much interest.

5.2 The Bipedal SLIP Template

5.2.1 System Model and Dynamics

Figure 5.1 illustrates the Bipedal Spring-Loaded Inverted Pendulum (BSLIP) model, which consists of a point mass m , attached to two compliant massless legs – labeled as the left(1) and right(2) legs – that can freely rotate around the hip joint. Both legs incorporate passive springs with potential $U(\xi)$, as well as viscous damping

with constant d_s . The mass is constrained to remain in the saggital plane, and is acted upon by gravity. Each leg has two possible discrete modes: *stance* and *swing*, which are further divided into compression, decompression, ascent and descent subphases as in Section 4.2.1.

Throughout the stance phase of a leg, its toe is fixed on the ground and the body is acted upon by the associated spring and damping forces. When the legs are in their swing phase, however, they do not affect the body dynamics. Their length and angle is governed by fully actuated first order dynamics, through which the touchdown angle and precompression can be controlled.

The overall system has four modes: *left stance*, *right stance*, *double stance* and *flight*, determined by which legs are in contact with the ground. Based on the formalism of [37] we define the set of modes for the biped as

$$\mathcal{M}_b := \{ \text{fl, ls, rs, ds} \} .$$

During flight, the dynamics of the system are those of a point mass acted upon by gravity, in conjunction with first order dynamics for both legs. Let

$$\mathcal{C}_b := \{ \mathbf{b} \mid \mathbf{b} = (b_y, b_z) \in \mathbb{R}^2 \}$$

denote the configuration of the body. Moreover, let the set of leg states be defined as

$$\mathcal{L}_i := \{ \mathbf{l}_i \mid \mathbf{l}_i = (\xi_i, \psi_i) \in S^1 \times \mathbb{R} \} \quad \text{with } i = 1, 2 .$$

Then, we define the flight state space as

$$\mathcal{Q}_{\text{fl}} := \left\{ \mathbf{q}_{\text{fl}} \mid \mathbf{q}_{\text{fl}} = [\mathbf{b}, \dot{\mathbf{b}}, \mathbf{l}_1, \mathbf{l}_2]^T \right\} .$$

The associated vector field is given by

$$\begin{aligned} \ddot{\mathbf{b}} &= \begin{bmatrix} 0 \\ -g \end{bmatrix} \\ \dot{\psi}_i &= w_i \\ \dot{\xi}_i &= v_i , \end{aligned}$$

where w_i and v_i are control inputs, and $i \in \{1, 2\}$ indicates the left or right leg.

In contrast, during left and right stances, the dynamics of the system are those of a basic SLIP, together with the first order dynamics of the swing leg. The corresponding state spaces are defined as

$$\begin{aligned}\mathcal{Q}_{\text{ls}} &:= \left\{ \mathbf{q}_{\text{ls}} \mid \mathbf{q}_{\text{ls}} = [\mathbf{l}_1, \dot{\mathbf{l}}_1, \mathbf{l}_2]^T \right\} \\ \mathcal{Q}_{\text{rs}} &:= \left\{ \mathbf{q}_{\text{rs}} \mid \mathbf{q}_{\text{rs}} = [\mathbf{l}_2, \dot{\mathbf{l}}_2, \mathbf{l}_1]^T \right\} .\end{aligned}$$

In both cases, the dynamics take the form

$$\begin{aligned}\ddot{\xi}_i &= \xi_i \dot{\psi}_i^2 - g \cos \psi_i - \frac{DU(\xi_i) + d_s \dot{\xi}}{m} \\ \ddot{\psi}_i &= \frac{-2\dot{\xi}_i \dot{\psi}_i + g \sin \psi_i}{\xi_i} \\ \dot{\psi}_{1-i} &= w_{1-i} \\ \dot{\xi}_{1-i} &= v_{1-i} .\end{aligned}\tag{5.1}$$

Finally, The state vector during double stance is defined as

$$\mathcal{Q}_{\text{ds}} := \left\{ \mathbf{q}_{\text{ds}} \mid \mathbf{q}_{\text{ds}} = [\mathbf{b}, \dot{\mathbf{b}}] \right\}$$

and the vector field can be derived as

$$\ddot{\mathbf{b}} = - \sum_{i \in \{1, 2\}} \frac{DU(\xi_i) + d_s \dot{\xi}}{m \xi_i} \begin{bmatrix} b_y - f_i \\ b_z \end{bmatrix} + \begin{bmatrix} 0 \\ -g \end{bmatrix} .$$

Transitions between modes are governed by the transition diagram illustrated in Figure 5.2. Once again, we use the hybrid system formalism in [37], and use *threshold functions* to specify the conditions that trigger these transitions. For this purpose, we first define the functions

$$\begin{aligned}h_i^t &:= b_z - \xi_i \cos \psi_i \\ h_i^l &:= \xi_0 - \xi_i ,\end{aligned}$$

whose zero crossings determine the touchdown and liftoff transitions for the i^{th} leg. Using these definitions, Table 5.1 lists the threshold functions for each possible mode transition of the diagram 5.2.

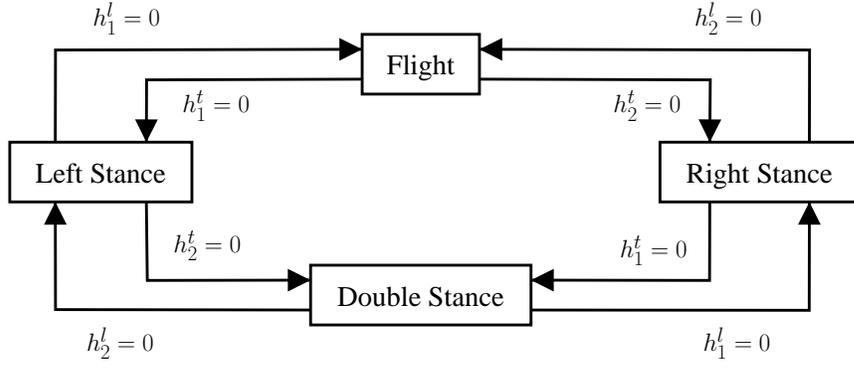


Figure 5.2: Mode transitions in the Bipedal SLIP model.

	Flight	Left Stance	Right Stance	Double Stance
Flight	-	h_1^t	h_2^t	-
Left Stance	h_1^l	-	-	h_2^t
Right Stance	h_2^l	-	-	h_1^t
Double Stance	-	h_2^l	h_1^l	-

Table 5.1: Bipedal SLIP threshold functions for each possible mode transition.

In the following sections, we will also find it useful to combine the state spaces of all the modes into a single set. We hence define the *hybrid state space* as

$$\mathcal{Q} := \{\mathbf{q} \mid \mathbf{q} = (\Omega, \mathbf{q}_\Omega); \Omega \in \mathcal{M}_b\} .$$

5.3 A Bipedal SLIP Controller

In this section, we describe an approximate deadbeat controller for the bipedal SLIP, based on the basic SLIP template. This controller is primarily designed for running rather than walking and does not support nonzero double stance phases. It represents an important intermediate step towards the design of running controllers for the planar and spatial hexapods as it addresses many of the underlying issues such as the coordination of stance and swing legs as well as the stability of the higher

level gait regulation.

5.3.1 The Structure of the Controller

Similar in spirit to Raibert’s bipedal controllers, our approach attempts to treat the biped as a single monopod, where the left and right legs alternately act as the stance leg. This abstraction yields the ability to use the gait characterization for monopeds and considerably simplifies the design of the bipedal controller.

We achieve the alternation between the two legs using the finite state machine (FSM) illustrated in Figure 5.3, which also modulates the behavior of the continuous recirculation control for the swing legs.

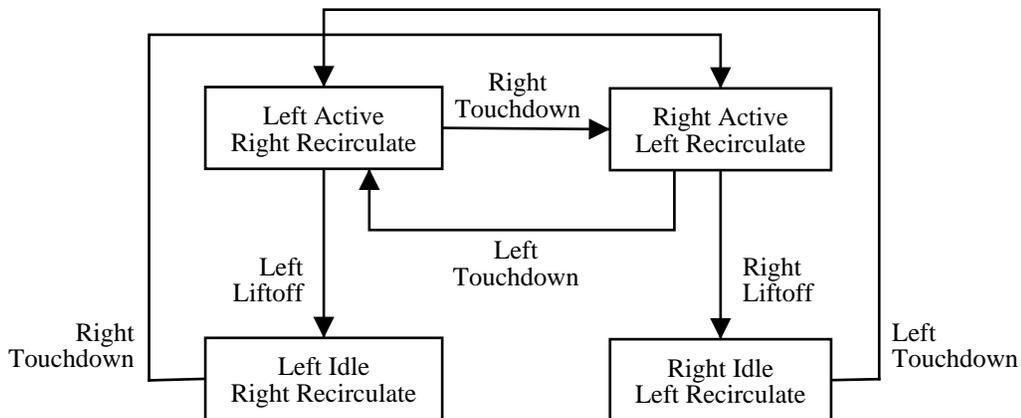


Figure 5.3: The finite state machine governing leg alternation in BSLIP.

In this scheme, each leg assumes one of three roles: *active*, *idle* or *recirculate*. At any time, one of the legs is labeled as the recirculation leg and is controlled by the coordination mechanism described in Section 5.3.5. The remaining leg is either active, in which case it is in contact with the ground, or idle, which occurs when the overall system is in flight. Although in the context of the bipedal SLIP, the active leg is not explicitly controlled, it will be the primary means by which the controllers for the planar and spatial hexapods perform active embedding of the SLIP template.

Throughout the remainder of the thesis, the symbols ψ_r and ξ_r denote the states of the recirculating leg, whereas ψ and ξ refer to the states of the remaining leg.

An important consequence of using a discrete state machine within the controller is the resulting increase in the number of modes in the resulting closed loop hybrid system. Similar to controllers with internal dynamics, such as observers, or periodic excitation, this changes the dimension and the structure of the system. Even though the scope of this thesis does not cover a formal treatment of problems that may arise from such combinations, the structural clarity that they admit greatly benefits the controller designs for both the planar and spatial hexapods.

5.3.2 BSLIP Control Inputs

The swing leg is the primary means of control for BSLIP. During recirculation, the length and the angle of the swing leg determine two of the control inputs to the system: the touchdown angle ψ_t and the touchdown leg precompression ξ_t . Once these control inputs are determined by the gait level controller, continuous control and coordination of the swing leg through the recirculation controller of Section 5.3.5 ensures that the proper swing leg states are achieved in a timely manner. Finally, a third control input comes from the ability to force a premature liftoff of the stance leg when its length reaches ξ_t .

The choice of the leg precompression and liftoff leg length as control parameters offers several advantages. Most importantly, it admits a passive stance phase, allowing much simpler and more robust mechanical designs. A literal instantiation of this control scheme is Zeglin’s bow leg hopper [81, 82], where a curved leg made of composite material, together with a tunable extension limit on the leg yield a completely passive stance phase. Similarly, our hexapedal platform incorporates only passive radial compliance in the legs, and hence requires controllers which do not use active tuning during stance. In this context, we achieve the desired control inputs through

the concept of a *virtual toe*, whose relative placement with respect to the center of mass allows the precompression of the associated *virtual leg*.

Another advantage of this approach is its low actuator bandwidth requirements. Especially for bipeds, the relatively long swing phase allows the use of lower power actuators as the injection of energy is spread through the longer time period of recirculation. Moreover, the lack of active tuning during stance admits more accurate dynamical models and increases the performance of model based controllers.

In light of these observations, we redefine the set of control inputs as

$$\mathcal{U}_b = \{u_b \mid u_b = [\psi_t, \xi_t, \xi_l]^T\} .$$

Moreover, as in Section 4.3.1, we will be working with the manifest gait parameters. The set of apex states are similarly defined as

$$\mathcal{X}_a = \{X_a \mid X_a = [b_{za}, \dot{b}_{ya}, \phi]^T\}$$

yielding the apex return map,

$$f_a : \mathcal{X}_a \times \mathcal{U}_b \mapsto \mathcal{X}_a . \tag{5.2}$$

Once again, the control objective is to identify the sequence of control inputs $\{(u_b)_i\}_{i=0}^n$ to asymptotically converge to the desired apex state $X_a^* = [b_{za}^*, \dot{b}_{ya}^*, \phi^*]^T$.

5.3.3 An Approximate Apex Return Map

The return maps and the controllers of Section 4.3 are not suitable for the set of control inputs that we have chosen to work with for the BSLIP model. Consequently, in this section, we present the derivation an approximate apex return map using the new set of inputs, as well as the associated deadbeat control strategy. The derivations for this return map were done in collaboration with Richard Altendorfer.

A simplifying assumption that we make in the design of this deadbeat controller is the omission of the duty factor ϕ from the set of goal states. This is mainly

due to the relatively small affordance granted by the physically feasible range of the precompressed and liftoff leg lengths ξ_t and ξ_l . Moreover, the apex duty factor is not instrumental enough in our illustration of the template control paradigm to warrant the substantial complexity it introduces in the inversion of the apex return map. Consequently, both the derivation of the return map and the design of the controller do not take the apex duty factor into consideration. As such, the value of the duty factor will not be explicitly controlled, but rather, will vary according to the speed and height of locomotion.

Descent Phase Map

Suppose we are given an apex state X_a and control inputs u_b . Moreover, assume that the active leg is positioned at the commanded angle and length. Simple projectile motion yields the touchdown state as

$$\begin{bmatrix} b_{zt} \\ \dot{b}_{yt} \\ \dot{b}_{zt} \end{bmatrix} = \begin{bmatrix} \xi_t \cos \psi_t \\ \dot{b}_{ya} \\ \sqrt{2g(b_{za} - b_{zt})} \end{bmatrix}$$

We will also find it convenient to compute the magnitude of the velocity v_t , the attack angle δ_t and the touchdown angular momentum l_t ,

$$\begin{aligned} v_t &= \sqrt{\dot{b}_{ya}^2 + \dot{b}_{zt}^2} \\ \delta_t &= \text{atan}(-\dot{b}_{ya}/\dot{b}_{zt}) \\ l_t &= m\xi_t v_t \sin(\delta_t + \psi_t) . \end{aligned}$$

Finally, the total energy at touchdown, including the potential energy stored in the active leg spring, is

$$E_s = \frac{mv_t^2}{2} + \widehat{U}_g(\xi_t) .$$

Approximate Stance Phase Map

Even though the stance dynamics of SLIP with gravity are not exactly integrable, various approximations are available [69]. In particular, we will be using the linearized gravity approximation where the true potential

$$U_g(\xi, \psi) := U(\xi) + mg\xi \cos \psi$$

is approximated by

$$\widehat{U}_g(\xi) := U(\xi) + mg\xi .$$

In addition to this simplification, one further approximation is needed to reduce the system to an equivalent of the classical central force problem. We assume that the damping losses can be neglected and do not dramatically change the trajectories of the system. Even though this approximation introduces tracking bias at the gait level, it is always possible to use integral compensation to correct the resulting error.

Following these simplifying assumptions, the angular momentum and the energy of the resulting system are conserved, yielding exactly integrable dynamics[35]. The angular displacement of the leg as a function of the leg compression is given by

$$\Delta\psi(\xi_1, \xi_2) := \int_{\xi_1}^{\xi_2} \frac{\text{sign}(\xi - \xi_0)l_t}{\xi \sqrt{2m(E_s - \widehat{U}_g(\xi))\xi^2 - l_t^2}} d\xi . \quad (5.3)$$

The total angular displacement during stance can be computed by individually considering the compression and decompression subphases, separated by the bottom point, where the leg is maximally compressed. The leg angle at liftoff can then be computed as

$$\psi_l = \psi_t + \Delta\psi(\xi_t, \xi_b) + \Delta\psi(\xi_b, \xi_l) . \quad (5.4)$$

The maximal leg compression occurs at the turning point of the trajectory, where $\dot{\xi} = 0$. The energy balance at the bottom point hence becomes

$$E_s = \frac{l_t^2}{2m\xi_b^2} + \widehat{U}_g(\xi_b) . \quad (5.5)$$

For a linear Hooke's spring model $U(\xi) = k(\xi - \xi_0)^2/2$, (5.5) can be written as the quartic polynomial

$$\frac{k}{2}\xi_b^4 + (mg - k\xi_0)\xi_b^3 + \left(\frac{k\xi_0^2}{2} - E_s\right)\xi_b^2 + \frac{l_t^2}{2m} = 0, \quad (5.6)$$

which can be solved in closed form.

This polynomial always has two real roots. Moreover, one of the roots is always in the range $[0, \xi_t]$ and corresponds to the maximum compression of the leg and is the solution to be used for ξ_b . The second root is in the range $[\xi_t, +\infty]$ and corresponds to the maximum extension ξ_{max} of the leg if the toe were pinned to the ground to avoid liftoff. The significance of the second root is in determining the existence of the second integral in (5.4). If $\xi_{max} < \xi_l$, the leg never lifts off and the system gets trapped in the stance phase. In those cases, our hybrid transition model becomes invalid and the apex return map cannot be defined. Consequently, our controller design must respect this constraint and choose appropriate controls to satisfy $\xi_{max} > \xi_l$.

Once the maximum leg compression is identified, we only need to evaluate the elliptic integrals in (5.4) to compute the angular span of the stance phase. However, the deadbeat controller requires an online numerical inversion of the apex return map, and the exact computation of these integrals quickly becomes intractable. Consequently, we adopt a special version of the mean value theorem [72] to approximate the integral in (5.3), yielding

$$\Delta\psi(\xi_1, \xi_2) \cong \frac{|\xi_2 - \xi_1|l_t}{\widehat{\xi}\sqrt{2m(E_s - \widehat{U}_g(\widehat{\xi}))\widehat{\xi}^2 - l_t^2}}$$

where $\widehat{\xi} := \xi_1 + (\xi_2 - \xi_1)/4$.

Once the liftoff angle is determined, the remaining states can be easily computed using the conservation of energy and angular momentum during stance, yielding

$$\begin{aligned} b_{zl} &= \xi_l \cos \psi_l \\ \widehat{v}_l &= \sqrt{v_t^2 + 2(\widehat{U}_g(\xi_t) - \widehat{U}_g(\xi_l))/m} \\ \delta_l &= \pi + \text{asin}\left(\frac{l_t}{m\xi_0\widehat{v}_l}\right) - \psi_l. \end{aligned}$$

Finally, we correct the liftoff velocity to account for the actual change in potential energy arising from the height difference of the touchdown and liftoff points. The remaining liftoff states then take the form

$$\begin{aligned} v_l &= \sqrt{v_t^2 + 2(U(\xi_t) - U(\xi_l))/m + 2g(b_{zt} - b_{zl})} \\ \dot{b}_{yl} &= v_l \sin \delta_l \\ \dot{b}_{zl} &= v_l \cos \delta_l . \end{aligned}$$

Ascent Phase Map

The final component of the apex return map is the period from liftoff to the next apex point. Simple projectile dynamics govern the behavior, yielding the apex states as

$$\begin{aligned} \dot{b}_{ya} &= \dot{b}_{yl} \\ b_{za} &= \xi_l \cos \psi_l + \frac{\dot{b}_{zl}^2}{2g} . \end{aligned}$$

The combination of these descent, stance and ascent maps yields the approximate apex return map \hat{f}_a , whose inversion in the following sections constitute the basis of our deadbeat controller.

5.3.4 The Bipedal Deadbeat Gait Controller

As in the basic SLIP, the most obvious way to achieve the control objective of Section 5.3.1 is to compute the inverse of the apex return map. Fortunately, it is relatively straightforward to obtain an approximate solution to this problem when only two of the desired apex states, the forward velocity \dot{b}_{ya} and the hopping height b_{za} are considered. As in Section 5.3.3, we do not take the duty factor parameter into consideration.

First, we can compute the total energy difference between the current and desired

apex states as

$$\Delta E = \frac{1}{2}m(\dot{b}_{ya}^{*2} - \dot{b}_{ya}^2) + mg(b_{za}^* - b_{za}) .$$

In cases where a positive energy change is required, the controller will inject the difference through the leg precompression ξ_t . Fixing the liftoff length to the rest length of the spring, we have

$$\begin{aligned} \xi_t &= \xi_0 - \sqrt{\frac{2\Delta E}{k}} \\ \xi_l &= \xi_0 . \end{aligned}$$

In contrast, when the required change in the energy level is negative, the difference must be taken out of the system through a premature liftoff at length ξ_l . In this case, we choose the touchdown leg length to be the rest length of the spring, yielding

$$\begin{aligned} \xi_t &= \xi_0 \\ \xi_l &= \xi_0 - \sqrt{-\frac{2\Delta E}{k}} . \end{aligned}$$

Once the control inputs ξ_t and ξ_l are determined through this deadbeat energy balance, the apex return map can be written solely as a function of the touchdown angle. As a consequence, the remaining control input can be determined by numerically solving for ψ_t , the resulting one dimensional equation

$$\dot{b}_{ya}^* = (\pi_{\dot{b}_y} \circ \hat{f}_a)(\psi_t) \tag{5.7}$$

where $\pi_{\dot{b}_y}$ retrieves the forward velocity component of the approximate apex return map.

Unfortunately, (5.7) does not always have an exact solution. As a uniform solution, we adopt a slightly general framework which minimizes the error between the desired goal and the attainable apex states, yielding

$$\psi_t = \operatorname{argmin}_{-\frac{\pi}{2} < \psi < \frac{\pi}{2}} \left(\dot{b}_{ya}^* - (\pi_{\dot{b}_y} \circ \hat{f}_a)(\psi) \right)^2 . \tag{5.8}$$

Due to their deadbeat nature, the solutions to (5.8) often involve dramatic changes in the system state. In the presence of our modeling approximations and the resulting noise, the attempt to achieve the goal state in a single stride often impairs the ability of the controller to stabilize the system from bad initial conditions. As a consequence, the final component in the design of the deadbeat controller is a reference governor to avoid target states which require dramatic changes in the system state.

There are a variety of approaches in the literature towards dealing with constraints on the inputs and states of discrete time systems under noise [7, 31, 32, 33]. Some of these can guarantee convergence properties under the action of the reference governor constructions. In the context of the present problem, however, simple solutions based on the structure of the system turn out to be sufficient to significantly extend the domain of attraction. Therefore, the scope of this thesis does not cover the application of more general reference governor designs to the control problem.

In particular, we limit the leg touchdown angle command to stay within a certain neighborhood of the neutral touchdown angle — the angle command to yield the same apex state in the next stride. The approximate inversion of the apex return map that we have presented yields an estimate of the neutral command vector as

$$u_n := [\psi_n, \xi_{tn}, \xi_{tn}]^T = f_a^{-1}(X_a, X_a) . \quad (5.9)$$

Hence, the solution of (5.8) is clipped to lie within the range $[\psi_n - \Delta_\psi, \psi_n + \Delta_\psi]$. Note that this construction for the reference governor does not have an internal state, but only constrains the control inputs based on the measured system state.

In later sections, we will find it useful to define the functional form of the reference governor

$$g : \mathcal{X}_a \times \mathcal{U}_b \mapsto \mathcal{U}_b .$$

5.3.5 Coordination of Swing and Stance Legs

As outlined in Section 5.3.1, the recirculation of the swing leg is the primary means of achieving the control inputs chosen by the gait level controller. In particular, the touchdown angle ψ_t and the leg precompression ξ_t both correspond to the states of the swing leg at touchdown. In this section, we describe a coordination strategy for the control of the swing leg, ensuring that it achieves the desired state prior to coming into contact with the ground.

The steady-state trajectories that we are interested in are those that follow the characteristic mode sequence for running: left stance, flight, right stance, flight. For such trajectories, we will use the term *step* to refer to the stance phase of one of the legs, together with the flight phase immediately after, until the touchdown of the other leg. In contrast, we define the term *stride* to refer to the combination of all the four modes, starting from the touchdown of a particular leg. Note that the strides starting with the left and right legs overlap.

In this context, we define the *stride phase* associated with each leg as a measure of the time elapsed within the stride,

$$\Phi_i : \mathcal{Q} \mapsto S^1 .$$

In subsequent discussions, we will use a simple coordinatization of S^1 , mapping it onto the real interval² $[0, 1[$. Stride phase maps are required to satisfy the following constraints

1. Φ_i is continuous.
2. $\Phi_i(\mathbf{q}) = 0$ whenever \mathbf{q} is the touchdown state of leg i .
3. $\Phi_i(\mathbf{q}) = 0.5$ whenever \mathbf{q} is the touchdown state of leg $1 - i$.

²It is implicitly assumed that another chart is also defined to cover the singularity associated with the endpoints of this interval.

There are an infinite number of ways to define such stride maps. However, we are particularly interested in those that do not rely on approximate estimation of the future state evolution of the system. Proper coordination of the biped legs depends on the accuracy in which the second and third constraints are satisfied. As a consequence, the ability to compute the stride phase purely based on the measured state rather than approximate model based estimates is critical for online implementations of our controller.

The particular stride phase we use for the bipedal SLIP is defined as

$$\Phi_i(\mathbf{q}) = \begin{cases} \frac{1}{4} - \frac{1}{4} \frac{\text{atan}(e_\xi(\xi_l - \xi_i), -\dot{\xi}_i)}{\text{atan}(e_\xi(\xi_l - \xi_t), -\dot{\xi}_t)} & \text{if leg } i \text{ is active} \\ \frac{1}{2} - \frac{1}{4} \frac{\text{atan}(e_z(b_{zt} - b_z), \dot{b}_z)}{\text{atan}(e_z(b_{zt} - b_{zl}), \dot{b}_{zl})} & \text{if leg } i \text{ is idle} \\ \frac{3}{4} - \frac{1}{4} \frac{\text{atan}(e_\xi(\xi_l - \xi_{1-i}), -\dot{\xi}_{1-i})}{\text{atan}(e_\xi(\xi_l - \xi_t), -\dot{\xi}_t)} & \text{if leg } 1 - i \text{ is active} \\ 1 - \frac{1}{4} \frac{\text{atan}(e_z(b_{zt} - b_z), \dot{b}_z)}{\text{atan}(e_z(b_{zt} - b_{zl}), \dot{b}_{zl})} & \text{if leg } 1 - i \text{ is idle} \\ 1 & \text{otherwise} \end{cases} \quad (5.10)$$

where $0 \leq \text{atan}(y, x) < 2\pi$ is the four-quadrant arctangent function, and, e_ξ and e_z are scaling constants to adjust the relative ranges for the position and velocity variables in each case.

Note that this construction assigns equal phase slices to the single stance and flight, which is not always realistic for gaits whose flight phases are very short. To address such cases, the phase map would need to have a prior estimate of the flight duration, together with means of *exactly* satisfying the constraints in the definition of the phase map. Unfortunately, we have not been able to identify such a map and will be using the definition of (5.10) in all the bipedal SLIP simulations as well as the template controllers for the planar and spatial hexapods.

The definition of this stride map admits the representation of the stable periodic trajectories of the overall system as limit cycles on the two torus $S^1 \times S^1$. The map we have defined in (5.10) supports a very particular set of limit cycles, where the time trajectories over both S^1 instances are identical 180° out of phase. Figure 5.4 illustrates this construction.

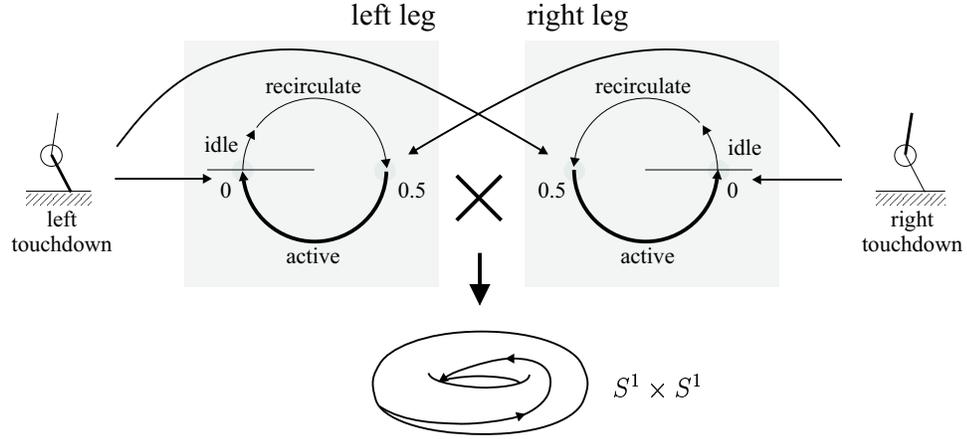


Figure 5.4: Evolution of the stride phase values for left and right legs of the bipedal SLIP, together with an illustration of the limit cycle on the torus.

The coordination strategy we use is a *mirror law* – a controller without any internal state, purely based on feedback [16]. Its design is based on determining the target angle for the recirculation of the swing leg based on the current stride phase of the active leg. This target, by design, spans the range from liftoff to the touchdown angle command ψ_t and takes the form

$$\psi_i^*(\mathbf{q}) = \psi_{i0} + (\Phi_i(\mathbf{q}) - 0.5)(\psi_t - \psi_{i0} + 2\pi\text{sign}(\psi_t))$$

where ψ_{i0} is the start angle for the recirculation of leg i , and should be sufficiently large to clear the ground. Note that the direction of the recirculation depends on the direction of travel through the sign of the desired touchdown angle.

By definition, the recirculation leg is always in flight. As a consequence, the control input consists of its angular velocity. A feedback law on the first order dynamics yields a continuous swing controller to track the target angle, and is specified as

$$w_s = -K_\psi(\psi_r - \psi_r^*(\mathbf{q}))$$

where K_s is the proportional feedback gain.

Similarly, the swing leg length is also controlled through linear feedback to achieve

the desired precompression in time for touchdown. The resulting control input reads

$$v_s = -K_\xi(\xi_r - \xi_t)$$

and is independent of the stride phase.

Before we can conclude the presentation of the recirculation controller, our choice of a state based strict mirror law needs to be justified. The most important reason for this choice is the impact of the recirculation accuracy on the performance of the gait level controller. The design of Section 5.3.4 implicitly assumes that the touchdown angle and precompression commands will be exactly achieved prior to touchdown, ensuring the validity of the return map, previously derived in Section 5.3.3.

It would have been possible to introduce coordination strategies in the spirit of [43], introducing internal clocks associated with each leg phase, together with coupling laws to ensure convergence to the desired limit cycle on the torus. This and similar approaches have the advantage of being much more practical and robust due to their limited dependence on state feedback. On the other hand, the overall performance the controller is likely to be impaired by the discrepancy of the actual touchdown states from those commanded by the deadbeat controller. In the scope of this thesis, we have chosen to characterize in isolation, the stability properties of the model based deadbeat controller, rather than alternative approaches which are more practical, but stand further from the return map derivations.

5.3.6 Stability with Limited Precompression

This section investigates in simulation, regions of the BSLIP gait parameter space that are reachable through asymptotically stable trajectories, using the deadbeat controller design of Section 5.3.4. In this context, the effects of further constraints on the control inputs are also studied. In particular, it will be important to consider a lower limit on leg precompression to constrain the maximum energy injection into the system as well as a limit on the maximum deviation of the touchdown angle command

m (kg)	ξ_0 (m)	k_b (N/m)	d_s (Ns/m)	Δ_ψ (rad)	c_{ij}	c_z
7	0.17	5000	3	0.15	3	10

Table 5.2: Structural and controller parameters for all BSLIP simulations.

from its neutral value.

In later chapters, both of these constraints will be instrumental in ensuring the overall stability of spatial hexapod locomotion. As a result of the approximate nature of the template embedding algorithms, the energy gained through the precompression of the template usually spills over to other degrees of freedom in the system, particularly the body pitch, and roll. In such cases, oscillations in the orientation of the body lead to unexpected changes in the hybrid configuration of the system, and usually decrease the number of legs in contact with the ground. The control affordance on the overall dynamics is hence greatly reduced, leading to overall instability of the system.

The two main reasons for the approximate nature of the embedding algorithms are the limited torque affordance of hip actuators and the singularities arising from the morphology of the hexapedal models. Not surprisingly, the rate of energy injection through leg precompression has a significant effect on the torque requirements, and hence indirectly determines the quality of the approximations. As a consequence, this section explores in simulation, the dependence of the task level BSLIP stability on the lower bound ξ_{min} for the leg precompression control.

Table 5.2 gives the structural and controller parameters for all BSLIP simulations of this section. The values are chosen to match the dynamic parameters of RHex as closely as possible. However, in some cases, particularly for the damping coefficient d_s , the match is only qualitative and no experimental verification has been done. Similarly, the tuning of the controller parameters was done manually, using simulations with the given morphology.

The simulations of this section consist of sweeps over the forward velocity target within the range $[0, 3]m/s$ and the hopping height target over the range $[0.13, 0.3]m$, for different values of the precompression limit $c_m := \xi_{min}/\xi_0$. For any given goal, the simulations were started from an initial condition close to the fixed point. The biped was run for 100 strides and the stability of the goal point was determined by looking at the periodicity of the last 10 apex points with a tolerance margin of 10^{-5} . Furthermore, the uniqueness of the fixed point was also verified for each run. Figure 5.5 illustrates the results, where shaded regions illustrate the stable reachable goal points.

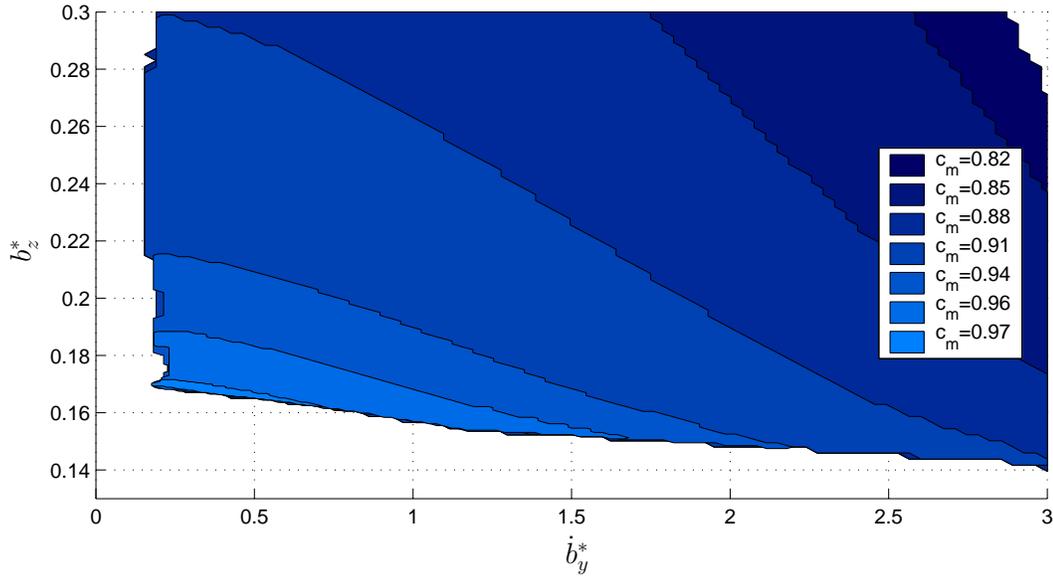


Figure 5.5: Stable reachable goal regions for the Bipedal SLIP model for different precompression limits. $c_m := \xi_{min}/\xi_0$.

Once again, for reasons outlined in Section 5.3.3, the duty factor dimension of the neutral gait space is not considered separately. Each of the goal points with a particular forward velocity and hopping height yields a unique associated duty factor, which arises from the constrained choice of the actuation strategy rather than active deadbeat control. As a consequence, possible goal states for the deadbeat controller

lie on a two dimensional submanifold in the neutral gait space. The projection of this manifold onto the forward velocity vs. hopping height plane, illustrated in Figure 5.5 is hence sufficient for presenting the goal seeking performance of the closed loop system.

The most striking aspect of these results is the dependence of the reachable goal set on the precompression limit. As the maximum energy that can be injected to the system decreases through the limitation of the precompression, the maximum speed and height that can be sustained in a stable manner decrease as well.

Interestingly, when the goal point is above the energy limit that can be sustained, the system usually still maintains asymptotic stability. However, the final stable orbit ends up being different than the target point. As a consequence, stable locomotion at the upper boundary of each region in Figure 5.5 is obtained when any goal above that boundary is requested.

Two other interesting outcomes of these simulations are the unreachable band around zero velocity, also mirrored for negative velocities, and the inability to reach hopping heights below a certain threshold. The latter is a natural consequence of the kinematics at touchdown, which imposes a lower bound on the hopping height. The unreachable band around the zero velocity goal, however, is somewhat more surprising and requires further attention.

For goals in this band above the kinematic height constraint, the hopper always stabilizes into a period two orbit, usually hopping backward and forward in alternation. Figure 5.6 illustrates an example run with such a period two stable orbit.

This odd behavior seems to be an outcome of the error introduced by the linearized gravity approximation of the stance map of Section 5.3.3. The key aspect of this approximation is that it is most accurate when the stance phase is symmetric around the vertical, in which case the moment generated by the gravity is zero on average through the stance phase. However, the goals that are close to zero usually involve transients wherein the stance phases are strongly asymmetrical. As a consequence,

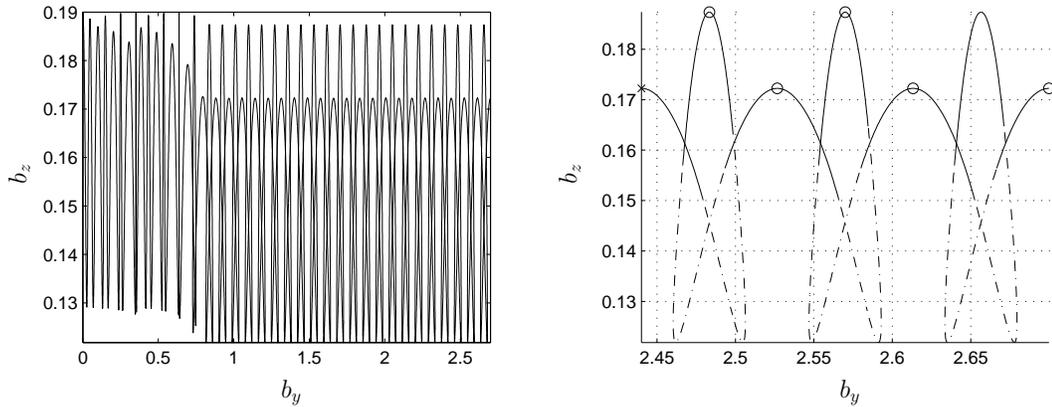


Figure 5.6: Sample period two BSLIP run with $\dot{b}_y^* = 0.2m/s$ and $b_z^* = 0.19m$. The left figure illustrates the overall trajectory for 60 steps, whereas the right figure shows the last 6 steps of the same run. Flight phases are depicted with solid lines, whereas stance phases are shown as dashed lines.

it is not surprising that the additional gravitational moment results in unexpected behavior.

It would certainly be possible to augment the deadbeat controller to more accurately deal with such situations. However, in the ranges of operation we are most interested in, those with high speed goals, the simple linearized gravity approximation yields effective and stable locomotion. Therefore, we will not present more careful studies of goal regions with lower speeds.

5.3.7 Basins of Attraction

In this section, we investigate the basins of attraction for the Bipedal SLIP deadbeat controller with $c_m = 0.96$, corresponding to six goal points in the gait parameter space, illustrated in Figure 5.7. This particular choice of c_m was motivated by preliminary simulations of our hexapedal controllers and was found to yield a good compromise between the size of the reachable goal set, the settling time to the goal and the overall stability of the system.

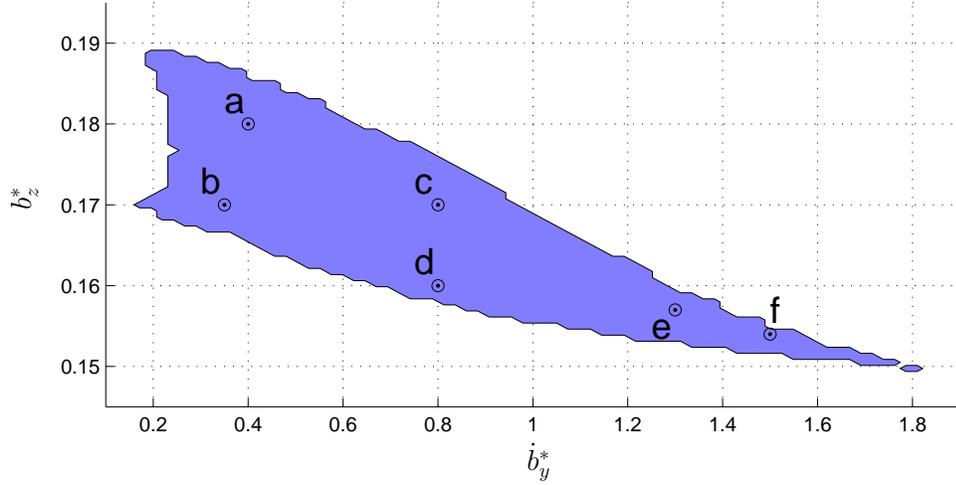


Figure 5.7: Goal points within the reachable apex goal set of the Bipedal SLIP, using $c_m = 0.96$. The labeled points indicate five different goal points for which the basins of attraction are illustrated in subsequent figures.

The control input at each step $u_b[k]$ is computed as a combination of the inverse of the apex return map and the reference governor construction, yielding

$$u_b[k] = g \circ \hat{f}_a^{-1}(X_a[k])$$

where the apex goal is implicit in the inverse of the apex map and g captures the reference governor introduced in Section 5.3.4. The resulting discrete closed loop apex return map takes the form

$$X_a[k+1] = \overline{f}_a(X_a[k]). \quad (5.11)$$

Figure 5.8 illustrates the basins of attraction for this closed loop system for the six different apex goal points of Figure 5.7. The simulations were run over 100 steps, starting from a two dimensional range of apex initial conditions. The actual goal point as seen by the controller was adjusted offline for each set of simulations and compensated for the steady state error. A particular run was found to be stable when the last 10 apex states were identical up to a tolerance of 10^{-5} and equal to the single fixed point for the set.

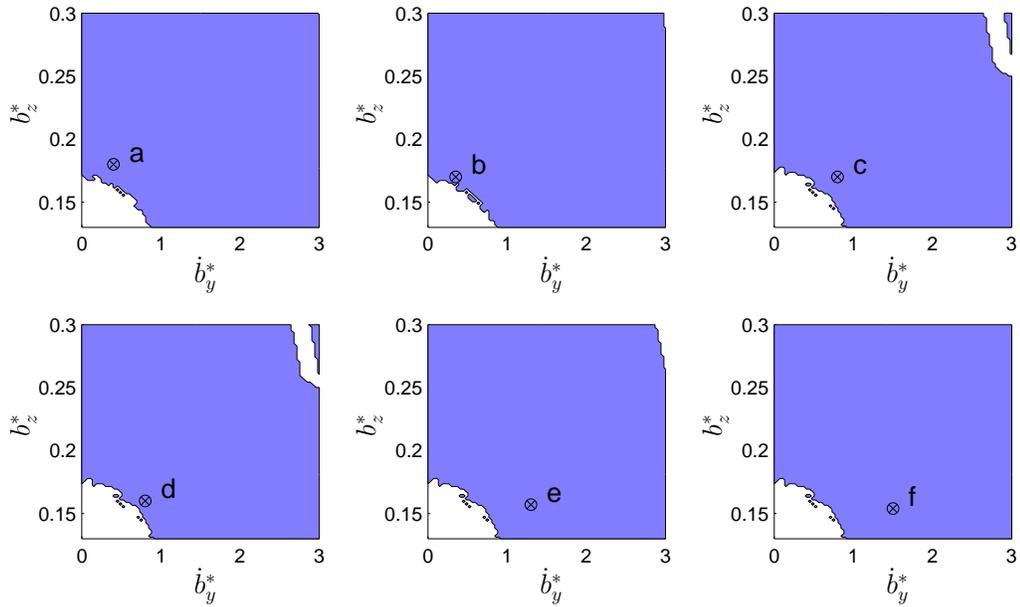


Figure 5.8: Basins of attraction of the Bipedal SLIP with the deadbeat controller for six different apex goals. The graphs illustrate the stable regions in the space of initial apex states. For each graph, the labeled point indicates the goal setting and also shows its correspondence to Figure 5.7.

The results demonstrate that the basins of attraction for the controller in each case cover most of the state space. Once again, the unstable initial conditions at the bottom left corner in each case are primarily a result of kinematic constraints on the configuration as a result of the leg length. In some cases — c and d, in particular — the instability at the top right corner results from the violation of the required mode sequence. In general, however, the closed loop system under the deadbeat controller has satisfactory stability properties.

The Bipedal SLIP with deadbeat will be employed as the gait level controller for the spatial hexapod. As a consequence, its stability properties will have significant impact on the performance of the embedding controller for the hexapod. Fortunately, simulation results suggest that the deadbeat based gait level controller design has sufficiently large basins of attraction and will not impact the performance of the

embedding controllers.

5.4 Slimpod: A Planar Hexapod Model

5.4.1 System Model and Assumptions

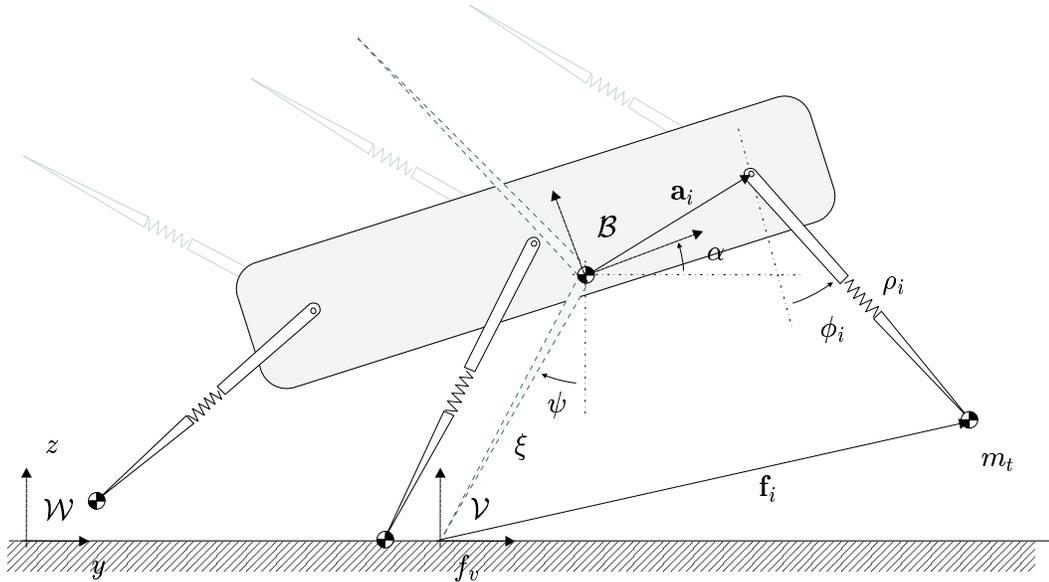


Figure 5.9: Slimpod: The planar compliant hexapod model.

Figure 5.9 illustrates the planar hexapod model that we consider in this chapter. First, we define three reference frames: \mathcal{W} as the fixed inertial world frame, \mathcal{V} as the virtual toe frame with the same orientation as \mathcal{W} , but located at the foot of a virtual leg, and finally \mathcal{B} as the body frame, affixed to the center of mass of the system.

The model consists of a rigid body with six compliant legs. The attachment points of the legs are fixed at positions \mathbf{a}_i in the body frame. The rigid body has mass m and inertia I . The orientation of the body frame with respect to the world frame determines the body pitch α and is also expressed by the rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$, following the standard notation in robotics [21]. The position of the body in \mathcal{V} is denoted by \mathbf{b} .

Very small masses $m_t \ll m$ at each toe are introduced to capture the flight dynamics of the legs. Moreover, all legs have radial springs with potential $V_i(\rho_i)$ and radial viscous dampers with coefficients d_i . Each hip is also independently actuated with torque τ_i , the primary means of control in the system. Similar to the hybrid models of Chapter 4, each leg has two possible modes: stance and flight. Once again, different contact configurations for the legs yield different dynamics for the overall system.

For legs in stance, the toe positions \mathbf{f}_i are fixed on the ground and the rigid body is directly acted upon by the leg forces. In contrast, legs that are in flight do not exert forces on the body. Instead, the motion of the leg is governed by the dynamics of the associated toe mass under the influence of the leg forces. Moreover, for the legs in flight, the position and velocity of their toe masses become independent coordinates of the overall dynamics.

5.4.2 The Hybrid State Space and Related Notation

In this section, we introduce a framework for modeling the interaction of the legs with the ground. Six binary flags $s_i \in \{0, 1\}$, — one for each leg — are sufficient to encode the contact configuration of the system. In each of the possible 64 contact states, the system has different dynamics as well as a state space with possibly a different structure. In the sequel, as in Section 5.2.1, we use the hybrid system formalism of [37].

Let the mode index set representing all possible contact states be defined by

$$\mathcal{M}_h := \{ \Omega \mid \Omega = (s_1, \dots, s_6) \} .$$

The following projection matrix will be useful in masking various leg force vectors based on the current contact state

$$\mathbf{S} := \text{diag}(s_1, \dots, s_6) .$$

Considering the configuration of the rigid body in conjunction with the configurations of each toe yields a uniform definition for the configuration space as

$$\mathcal{C}_h := \{ \mathbf{c}_h \mid \mathbf{c}_h = [\mathbf{b}, \alpha, \mathbf{f}_1, \dots, \mathbf{f}_6]^T \} .$$

Throughout the rest of the chapter, we will also refer to an index set for all the legs, defined as

$$\mathcal{I} := \{ 1, 2, 3, 4, 5, 6 \} .$$

Similarly, we define the index sets \mathcal{I}_1 and \mathcal{I}_2 for the left and right tripods, respectively, with the following properties

$$\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}$$

$$\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$$

$$|\mathcal{I}_1| = |\mathcal{I}_2| = 3 .$$

Finally, the aliases \mathcal{I}_a , \mathcal{I}_i and \mathcal{I}_r are defined for the active, idle and recirculation tripods, respectively.

5.4.3 Polar Leg Coordinates and Leg Forces

Through their interaction with the ground, the six legs of the planar hexapod are the only means of exerting external force on the rigid body. In this context, it is much more natural to work with the positions and orientations of the legs with respect to the body frame as these are the coordinates where compliance and actuation models are specified.

Given the current configuration of the system $\mathbf{c}_h \in \mathcal{C}_h$, leg vectors in the body frame can be easily computed as

$$\begin{aligned} \mathbf{l}_i &= R^T(\alpha) (\mathbf{f}_i - \mathbf{b}) - \mathbf{a}_i \\ \dot{\mathbf{l}}_i &= D_\alpha R^T(\alpha) (\mathbf{f}_i - \mathbf{b}) \dot{\alpha} + R^T(\alpha) (\dot{\mathbf{f}}_i - \dot{\mathbf{b}}) , \end{aligned} \tag{5.12}$$

where $R(\alpha)$ denotes the standard planar rotation matrix

$$R(\alpha) := \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}. \quad (5.13)$$

We will also find it useful to express these vectors in polar coordinates, $\bar{\mathbf{l}}_i := [\rho_i, \phi_i]$, under the coordinate transformations

$$\begin{aligned} \rho_i &= \sqrt{\mathbf{l}_{iy}^2 + \mathbf{l}_{iz}^2} & \phi_i &= \text{atan}(-\mathbf{l}_{iz}, \mathbf{l}_{iy}) \\ \dot{\rho}_i &= \frac{\mathbf{l}_{iy}\dot{\mathbf{l}}_{iy} + \mathbf{l}_{iz}\dot{\mathbf{l}}_{iz}}{\rho_i} & \dot{\phi}_i &= \frac{\mathbf{l}_{iy}\dot{\mathbf{l}}_{iz} - \mathbf{l}_{iz}\dot{\mathbf{l}}_{iy}}{\rho_i^2}. \end{aligned} \quad (5.14)$$

The forces generated by the legs arise from their radial compliance and damping as well as the hip torques. Based on the leg model described in Section 5.4.1, the radial force component can be written as

$$F_{ri} = -DV_i(\rho_i) - d_i\dot{\rho}_i.$$

Consequently, the force exerted on the ground by the i^{th} leg can be written as

$$\mathbf{F}_i = R(\alpha + \phi_i) \begin{bmatrix} \tau_i/\rho_i \\ -F_{ri} \end{bmatrix}.$$

Note that for the legs that are in stance, the ground exerts on the system, external reaction forces equal to the opposite of the corresponding leg forces. In contrast, for the legs in their swing phase, only the toe masses feel the effects of the leg force and no external ground reaction forces are present.

In the sequel, we will find it useful to define vector forms of the leg forces,

$$\begin{aligned} \boldsymbol{\tau} &:= [\tau_1, \dots, \tau_6] \\ \mathbf{F}_r &:= [F_{r1}, \dots, F_{r6}]. \end{aligned}$$

5.4.4 The Continuous Dynamics

Within a particular mode, the continuous dynamics of the planar hexapod model are simply those of a planar rigid body under the influence of external forces generated

by the legs. Consequently, the equations of motion take the form

$$\begin{aligned} m\ddot{\mathbf{b}} &= \sum_{i=1}^6 s_i \mathbf{F}_i \\ I\ddot{\boldsymbol{\alpha}} &= \sum_{i=1}^6 s_i (\mathbf{f}_i - \mathbf{b}) \times \mathbf{F}_i \\ \ddot{\mathbf{f}}_i &= (1 - s_i) \mathbf{F}_i . \end{aligned}$$

It is important to note that, through the use of the individual leg contact states s_i , only the stance legs affect the motion of the rigid body. Moreover, we assume that the collisions of the toe masses with the ground are plastic, such that their velocity becomes zero at the beginning of each stance. As a consequence, stance toes remain stationary until they liftoff.

5.4.5 Mode Transitions

Throughout the evolution of Slimpod trajectories over time, there are discrete changes in leg contact states. In order to formally specify the points in the trajectory where such changes occur, we need to define threshold functions for each mode pair. Fortunately, it is possible to independently consider the touchdown and liftoff conditions for all the legs. Defining

$$\begin{aligned} h_i^t &:= b_z - \rho_i \cos \phi_i \\ h_i^l &:= -\pi_z(\mathbf{F}_i) , \end{aligned}$$

where $\pi_z(\mathbf{v})$ is the projection of the vector \mathbf{v} onto the z coordinate axis. These conditions ensure that swing legs go into stance when their toes come in contact with the ground. Moreover, the transition from stance to swing is determined by the magnitude of the z component of the ground reaction force on the toe.

Finally, the threshold function for a particular mode pair can be constructed by combining the appropriate touchdown or liftoff conditions for all the legs.

5.5 Template Control of Slimpod Locomotion

5.5.1 Structure of the Embedding Controller

The design principle of the template based controller for the planar hexapod is to choose proper hip controls τ to bring the motion of the body center of mass as close as possible to the trajectories of the bipedal SLIP. Towards this end, our approach entails four separate components:

1. Imposing an alternating tripod gait to identify the left and right tripods with the left and right legs of the BSLIP template.
2. Appropriately placing the virtual foot to achieve the touchdown control inputs of the gait level bipedal SLIP controller.
3. Determining the appropriate torque controls for the stance tripod to tune the dynamics of the body center of mass.
4. The coordination and control of the swing tripod.

The following section describe in detail each of these components and how they fit together to achieve stable locomotion.

5.5.2 Imposing the Alternating Tripod Gait

Most hexapedal insects use the alternating tripod gait for locomoting at higher speeds. It offers a higher stability margin resulting from the tripod of support, as well as a longer period of time for the protraction of the swing tripod. In many ways, it is the closest hexapedal gait to bipedal locomotion, and inherits all of the advantages outlined in Section 5.1.

As a consequence, the mechanism that we use to impose the tripod gait is remarkably similar to the bipedal controller presented in Section 5.3.1. In fact, the finite

state machine that modulates the continuous controllers for each tripod has the exact same structure as its bipedal counterpart as illustrated in Figure 5.3.

As in the bipedal controller, each tripod can be in one of three phases: active, idle and recirculate. Each of these phases entails a different control strategy for the legs of the associated tripod. The following sections describe these controllers in detail. However, we now give a summary of each phase and its role in the final controller.

At any given time, one and only one of the tripods is in its recirculation phase. This corresponds to the time period when all three legs of the tripod rotate forward to prepare for their next touchdown. In order to achieve the proper touchdown state prior to the next stance phase, we will use the coordination controller of Section 5.3.5 together with extensions specific to the hexapod morphology. In particular, the concept of the virtual toe introduced in Section 5.5.3 will be instrumental in implementing the touchdown precompression of the virtual leg. The rest of the modifications to the recirculation controller are then presented in Section 5.5.4.

The phase of the non-recirculating tripod is either idle or active. The idle phase occurs when both virtual legs of the template biped are in flight. The associated controller attempts to place the tripod in a position where it will not prematurely touch the ground, allowing the recirculating tripod to reach the ground first and start the next stride. The details of this controller are described in Section 5.5.5.

Finally, a tripod in its active phase is assumed to be in stance. Unlike BSLIP where the stance leg is unactuated, the control of the active tripod is the key component in the design of the embedding controller. In particular, it is the primary means by which the dynamics of the body center of mass are tuned to mimic SLIP dynamics. In this context, Section 5.5.6 presents a special form of the dynamics on the virtual toe frame, followed by Section 5.5.7 where the active tripod controller is developed.

5.5.3 Virtual Toe Coordinates and Foot Placement

In order to establish the interface between the planar hexapod and the bipedal SLIP, it will be necessary to introduce an imaginary leg between connecting center of mass to the origin of the frame \mathcal{V} (see Figure 5.9). Together with an imaginary swing leg, also attached to the center of mass of the body, a virtual biped is formed, providing the link between the gait controllers of Chapter 4 and the hexapedal embedding controllers.

The gait control of the bipedal SLIP is accomplished through proper choice of the leg angle and length at every touchdown. For an actual biped, this is as simple as appropriately positioning the swing leg and waiting for the toe to come in contact with the ground. However, the virtual biped that constitutes the template for the planar hexapod does not have a physical swing leg that can come in contact with the ground. As a consequence, its toe has to be “placed” explicitly by the embedding controller when the swing tripod touches the ground, marking the transition to the next stance phase.

In general, the instant that the swing tripod becomes the active tripod (i.e. at least two of its legs touch the ground) may not correspond to the time that the virtual swing leg would have come in contact with the ground. As a consequence, the touchdown state of the imaginary leg will be different than the touchdown controls imposed by the gait level controller. In such cases, it turns out to be best to attempt to preserve the touchdown leg length at the expense of achieving the desired touchdown angle. Hence, at the beginning of each stance, our controller places the virtual toe at

$$f_v[k+1] = f_v[k] + b_y + \bar{\xi}_t \sin \bar{\psi}_t$$

where the adjusted touchdown angle and length are determined as

$$\begin{aligned}\overline{\psi}_t &= \begin{cases} \text{acos}(b_z/\xi_t) & \text{if } \xi_t \geq b_z \\ \psi_t & \text{otherwise} \end{cases} \\ \overline{\xi}_t &= \begin{cases} \xi_t & \text{if } \xi_t \geq b_z \\ b_z/\cos \psi_t & \text{otherwise} \end{cases} .\end{aligned}$$

In the first cases, the virtual leg can reach the ground and the toe is placed to yield the desired length and whichever angle results from the kinematics at that instant. In contrast, the second case occurs when the body height is too large to allow the virtual leg to reach the ground with the desired length. In such cases, the toe is placed to yield the desired virtual leg angle, but the length is computed by solving the kinematics. This second case is only introduced to gracefully handle situations that are far away from steady state equilibrium.

Following the toe placement, we define the *virtual toe* coordinate system as polar coordinates in \mathcal{V} , together with the body pitch. The corresponding configuration space is defined as

$$\mathcal{C} := \{ \mathbf{c} \mid \mathbf{c} = [\xi, \psi, \alpha]^T \} .$$

5.5.4 Coordination and Control of the Recirculating Tripod

The gait level controller of Section 5.3.4 commands discrete control inputs, specifying the touchdown and liftoff states of the virtual leg at each stride. Similar to the bipedal SLIP model, the recirculation controller is primarily responsible for achieving these gait level controls through proper coordination of the swing tripod. The main difference in the Slimpod model, however, comes from the lack of radial actuation in the legs, making it impossible to literally perform precompression of the legs prior to touchdown.

Fortunately, the idea of an explicitly placed virtual toe admits the precompression of the virtual leg, and obviates the need for radial actuation in the physical legs. The

goal of the recirculation controller is hence to position the recirculation tripod legs such that the virtual toe is appropriately placed at touchdown, and yields the desired virtual leg angle and precompression commands.

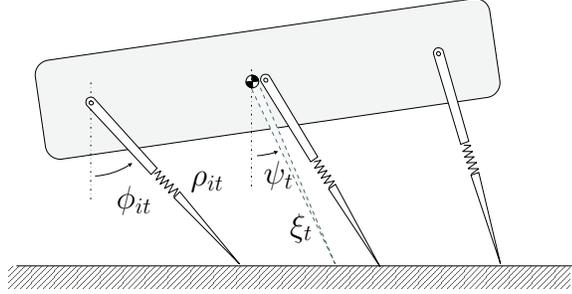


Figure 5.10: Touchdown kinematics of the recirculating tripod of the Slimpod model.

This can be accomplished by determining individual target angles for all legs through the solution of the kinematics at touchdown, illustrated in Figure 5.10. For a particular leg of the recirculation tripod $i \in \mathcal{I}_r$, the location of the hip at touchdown, \mathbf{p}_i can be computed as

$$\mathbf{p}_i = \xi_t \begin{bmatrix} \sin \psi_t \\ \cos \psi_t \end{bmatrix} + R(\alpha_t) \mathbf{a}_i . \quad (5.15)$$

Consequently, the target angles for each of the recirculation tripod legs are

$$\phi_{it}^* = \begin{cases} \text{acos}(p_{iz}/\rho_0) & \text{if } \rho_0^2 - p_{iz}^2 > 0 \\ 0 & \text{otherwise} \end{cases} . \quad (5.16)$$

The difficulty in using (5.16) in the continuous coordination controller arises from its dependence on the knowledge of the body pitch at touchdown, α_t . An accurate estimate of this quantity requires integration of the 1dynamics, which is not feasible for online implementation. Moreover, inaccuracies in the estimation will result in the premature touchdown of the tripod, affecting the performance of the SLIP embedding.

In order to address these problems, it is possible to modify (5.15) to use the *current* pitch measurement rather than an estimate of the touchdown pitch. As a

consequence, the target angle for the recirculating tripod legs becomes a continuous function of the current pitch, yielding

$$\phi_{it}^*(\alpha) = \begin{cases} \text{acos}(p_{iz}(\alpha)/\rho_0) & \text{if } \rho_0^2 - p_{iz}(\alpha)^2 > 0 \\ 0 & \text{otherwise} \end{cases} .$$

Having thus defined the touchdown target for the recirculation legs, the continuous control of the legs is very similar to the coordination strategy as Section 5.3.5. For this model, we define the stride phase as

$$\Phi(\mathbf{c}, \dot{\mathbf{c}}) = \begin{cases} \frac{1}{2} - \frac{\text{atan}(e_\xi(\xi_l - \xi), -\dot{\xi})}{2\text{atan}(e_\xi(\xi_l - \xi_t), -\dot{\xi})} & \text{for } \Omega_v \in \{\text{ls}, \text{rs}\} \\ 1 - \frac{\text{atan}(e_z(b_{zt} - b_z), \dot{b}_z)}{2\text{atan}(e_z(b_{zt} - b_{zl}), \dot{b}_z)} & \text{for } \Omega_v = \text{fl} \\ 1 & \text{otherwise} \end{cases} \quad (5.17)$$

where Ω_v denotes the mode of the virtual biped, determined by the finite state machine of Section 5.5.2. The continuous targets for the recirculating legs hence become

$$\phi_i^*(\mathbf{c}, \dot{\mathbf{c}}) = \phi_{i0} + \Phi(\mathbf{c}, \dot{\mathbf{c}})(\phi_{it}(\alpha) - \phi_{i0})$$

where ϕ_{i0} denotes the initial leg angles when the tripod first starts recirculation. The torque controls for the legs are determined through a simple PD law tracking the desired leg angles, taking the form

$$\tau_i = -K_\phi(\phi_i - \phi_i^*) - K_{\dot{\phi}}\dot{\phi}_i .$$

5.5.5 Control of the Idle Tripod

The only role of the idle tripod is to get out of the way to avoid touching the ground before the recirculating tripod reaches its target. However, it is not always possible to accomplish this without perturbing the flight dynamics of the body. Usually, some of the legs are still in contact with the ground when the active tripod transitions into its idle phase. This is a consequence of the transition condition, which is based on the length of the virtual leg, rather than the actual legs. Hence, the idle tripod controller

incorporates a torque control strategy to minimize its impact on the dynamics of the body. In particular, for all the legs $i \in \mathcal{I}_i$, we have

$$\tau_i = \begin{cases} -K_\phi(\phi_i - \pi/2) - K_\dot{\phi}\dot{\phi}_i & \text{if } s_i = 0 \\ 0 & \text{if } s_i = 1 \end{cases}$$

forcing the free legs to retract to the angle $\phi_i = \pi/2$.

5.5.6 Dynamics in the Virtual Toe Coordinates

The control of the active tripod requires active embedding of BSLIP dynamics within the Slimpod model. In this context, we will find it more convenient to work with the equations of motion in virtual toe coordinates.

Assuming $m \ll m_t$, the kinetic and potential energies of the system due to the rigid body can be written as

$$\begin{aligned} K &= \frac{1}{2}m(\xi^2\dot{\psi}^2 + \dot{\xi}^2) + \frac{1}{2}I\dot{\alpha}^2 \\ V &= mg\xi \cos \psi . \end{aligned}$$

The equations of motion then take the form

$$\begin{aligned} \ddot{\xi} &= \xi\dot{\psi}^2 - g \cos \psi + K_\xi/m \\ \ddot{\psi} &= \frac{-2\dot{\xi}\dot{\psi} + g \sin \psi}{\xi} + K_\psi/(m\xi^2) \\ \ddot{\alpha} &= K_\alpha/I , \end{aligned} \tag{5.18}$$

where $\mathbf{K} := [K_\xi, K_\psi, K_\alpha]^T$ is the forcing vector arising from radial leg forces and hip torques. This vector takes the form

$$\mathbf{K} := [K_\xi, K_\psi, K_\alpha]^T = (D_{\mathbf{c}}\boldsymbol{\phi}) \mathbf{S}\boldsymbol{\tau} + (D_{\mathbf{c}}\boldsymbol{\rho}) \mathbf{S}\mathbf{F}_r . \tag{5.19}$$

where $D_{\mathbf{c}}\boldsymbol{\phi}$ and $D_{\mathbf{c}}\boldsymbol{\rho}$ denote the jacobian matrices of the hip angles and lengths of all the legs with respect to the virtual leg state.

In subsequent discussions, we denote by $(\mathbf{M})_{\mathcal{I}}$ the matrix constructed by choosing the columns of an $m \times n$ matrix \mathbf{M} , corresponding to the integers in the index set \mathcal{I}

in the same order that they appear in \mathbf{M} . Furthermore, letting $\mathbf{J} := (D_{\mathbf{c}}\phi)_{\mathcal{I}_a}$, we use the row vector \mathbf{J}_ψ to denote the row of \mathbf{J} associated with ψ . Similarly, $\mathbf{J}_{\psi,\alpha}$ denotes the submatrix of \mathbf{J} with only the rows corresponding to ψ and α . Other combinations are defined accordingly.

5.5.7 Control of the Active Tripod

The active tripod controller must choose hip torque controls for the stance legs to yield rigid body center of mass dynamics as close as possible to the SLIP dynamics of (5.2). The ideal SLIP dynamics are characterized by a simple central force, generated with a conservative potential. Consequently, perfect embedding of the template can be achieved by

$$\mathbf{K}^* = [U^*(\xi), 0, M_\alpha^*]^T$$

where $U^*(\xi)$ is the target radial potential and M_α^* is an arbitrary pitching torque.

Two central components of the embedding problem are also identified by this construction: achieving the template behavior through the SLIP length and angle coordinates and the stabilization of the pitch as an independent degree of freedom.

In this context, when all the legs are in contact with the ground, the most obvious candidate for a solution is through inversion of the jacobian. Assuming that inversion is possible, the torque controls take the form

$$\boldsymbol{\tau} = \mathbf{J}^{-1} (\mathbf{K}^* - \mathbf{B}) ,$$

where we define

$$\mathbf{B} := (D_{\mathbf{c}}\boldsymbol{\rho})_{\mathcal{I}_a} \mathbf{F}_r$$

$$\mathbf{J} := (D_{\mathbf{c}}\phi)_{\mathcal{I}_a} .$$

Unfortunately, due to a number of constraints inherent to the morphology, \mathbf{J} usually ends up being rank deficient in the configurations that are characteristic of

alternating tripod running (see Appendix E). As a consequence, solution of the problem through exact inversion is not possible.

Another complication arises from the familiar practical limitations of conventional actuators. For instance, the maximum torque that can be delivered by the actuators that are used in RHex is $7Nm$, and decreases with the rotational speed of the motor. This limit easily exceeded by controllers that adopt inverse dynamics type approaches, especially near singular configurations. In order to ensure practical applicability, the design of the active tripod controller needs to incorporate an explicit mechanism to satisfy practical torque limitations.

Our approach is to consider individual components in the forcing vector of (5.19). Doing so yields the ability to identify and overcome singularities with graceful degradation in performance. Moreover, the separation of the problem into its components related to the embedding of the SLIP and the stabilization of the pitch, admits prioritization of these objectives in face of reduced control affordance.

The singularity analysis in Appendix E demonstrates that, for configurations close to the desired alternating tripod running trajectories \mathbf{J} has at most rank 2, with isolated points where the rank is reduced to 1. In particular, for RHex’s morphology and configurations where the legs are parallel, $K_\xi \cong 0$. As a consequence, in the design of the embedding controller, we will not consider the ξ component of (5.19). Fortunately, the morphology of the system, together with the particular set of steady state trajectories we are interested in, the natural dynamics in this direction are very close to those of passive SLIP. In fact, for trajectories with $\alpha = 0$ and having all the legs parallel, the correspondence becomes exact and the dynamics of the center of mass are those of a passive SLIP with potential $U(\xi) = \sum_{i \in \mathcal{L}_a} V_i(\xi)$.

Consequently, the ψ component of the forcing vector becomes critical in accurate embedding of the SLIP template. The constraint $K_\psi = 0$ reduces the system to a central force problem, which will closely approximate the desired template dynamics as a result of the natural dynamics in the radial direction.

In particular, assuming that all three legs of the tripod are in contact with the ground, the set of torques that satisfy this constraint is defined as

$$\mathcal{T}_\psi := \{ \boldsymbol{\tau} \mid \mathbf{J}_\psi \boldsymbol{\tau} + B_\psi = 0 \} .$$

The elements of this set can be written as a sum of two torque components $\boldsymbol{\tau} = \boldsymbol{\tau}_\psi + \boldsymbol{\tau}_\perp$ such that

$$\begin{aligned} \boldsymbol{\tau}_\psi &= \mathbf{J}_\psi^T \frac{B_\psi}{\mathbf{J}_\psi \mathbf{J}_\psi^T} \\ \boldsymbol{\tau}_\perp &\in \ker(\mathbf{J}_\psi) . \end{aligned}$$

It is important to note that \mathcal{T}_ψ represents an affine subspace of two dimensions, within the free torque space. All control torques in this subspace satisfy the SLIP embedding constraint and the resulting forces on the center of mass will be purely radial. Therefore, our controller construction will always attempt to choose torques from within \mathcal{T}_ψ .

Our secondary goal is the stabilization of the pitch. For this purpose, we attempt to enforce second order linear feedback dynamics on the pitch, taking the form

$$M_\alpha^* = -K_\alpha \alpha - K_{\dot{\alpha}} \dot{\alpha}$$

The solution that yields the desired pitch torque while satisfying the ψ component can be computed as

$$\boldsymbol{\tau}_{\psi,\alpha} = \mathbf{J}_{\psi,\alpha}^T (\mathbf{J}_{\psi,\alpha} \mathbf{J}_{\psi,\alpha}^T)^{-1} ([0, M_\alpha^*]^T - \mathbf{B}_{\psi,\alpha}) .$$

Ideally, if there were no constraints on the torques that can be delivered by the actuator, this solution would be directly usable to accomplish the desired embedding. However, for any practical application, there will be a limit to the maximum torque that can be delivered by physical actuators. Combined with other constraints such as finite friction between the feet and the ground, a more careful consideration of the

torque limitations becomes necessary. We hence define the set of allowable torques as

$$\bar{\mathcal{T}} := \{ \boldsymbol{\tau} \mid \tau_{i,min} \leq \tau_i \leq \tau_{i,max} \} .$$

Our goal is then to find the torque vector satisfying these constraints that is closest to the desired vector. Namely, the solution takes the form

$$\boldsymbol{\tau}_s = \underset{\boldsymbol{\tau} \in (\bar{\mathcal{T}} \cap \mathcal{T}_\psi)}{\operatorname{argmin}} \|\boldsymbol{\tau} - \boldsymbol{\tau}_{\psi,\alpha}\| .$$

In cases where no solution can be found, that is, whenever $\bar{\mathcal{T}} \cap \mathcal{T}_\psi = \emptyset$, then we relax the embedding constraint to yield the more approximate solution

$$\boldsymbol{\tau}_s = \underset{\boldsymbol{\tau} \in \bar{\mathcal{T}}}{\operatorname{argmin}} \frac{\langle \boldsymbol{\tau}_\psi \mid \boldsymbol{\tau} - \boldsymbol{\tau}_\psi \rangle}{\|\boldsymbol{\tau}_\psi\|} ,$$

which is the torque vector in the allowable space that is closest to the embedding subspace.

5.6 Conclusion

In this chapter, we introduced the Bipedal Spring-Loaded Inverted Pendulum (BSLIP) model as a template for hexapedal locomotion as well as its embedding in the simple planar hexapod(Slimpod) model towards its ultimate instantiation in the spatial hexapod.

The BSLIP model incorporates specific extensions to the basic SLIP to capture the coordination and control issues associated with the alternating tripod gait. Most importantly, through the introduction of a second leg, it addresses the control of flight leg recirculation and dynamically instantiates SLIP control inputs that were otherwise commanded through unrealistic assumptions.

In this context, this chapter also presented a new deadbeat control strategy for BSLIP that is capable of regulating the forward speed and hopping height of the template. Simulation studies demonstrated that, even under a variety of constraints on

the control inputs, the closed loop template system under the action of this controller has very large basins of attraction for different task level goals.

Finally, the Slimpod model represents an intermediate step towards the development of a full template embedding controller for a spatial hexapod. It captures most of the fundamental issues associated with the spatial hexapod. In this chapter, methods were developed within Slimpod for negotiating the limitations imposed by the kinematic singularities in the system, as well as for respecting actuator torque limitations within an inverse dynamics style controller design. All of these methods will also be employed in the spatial hexapod controller with minor adjustments.

CHAPTER 6

Running with a Spatial Hexapod

6.1 Motivation

In this chapter, we introduce a spatial hexapod model that captures the morphology of RHex and approximates its continuous dynamics. In this context, we extend the template based controllers for the Slimpod to the spatial model and investigate in simulation, the performance and stability properties of the resulting closed loop dynamical system.

The hexapedal locomotion controller as it is described in this chapter is very close to what a template embedding controller for RHex would look like. Even though there are still several practical issues to be addressed, such as the omission of leg slippage, this controller represents in principle, the alternative extreme to pure open loop locomotion controllers of Chapter 2. The combination of the unactuated dynamics and the present controller design form a completely autonomous dynamical system devoid of all explicit clocks. Surprisingly, despite the presence of severe model mismatches due to limited torque affordance and kinematic singularities, these dynamics exhibit asymptotically stable limit cycles and yield stable locomotion for a fairly large range of speeds. Furthermore, the closed loop system achieves the desired task level decomposition and admits a much more natural task interface.

Nevertheless, all the practical problems that we have sketched in preceding chapters still prevail. In particular, the lack of an realistic ground friction model as well as the reliance on high bandwidth accurate state feedback are significant barriers towards any real implementation. As we have noted before, the ultimate solution in the control of hexapedal solution will not solely be in either of these extreme approaches. In this context, we emphasize once again the importance of the present work primarily as a demonstration of methods on each extreme as well as the associated characteristic features.

6.2 The Compliant Hexapod Model

6.2.1 System Model and Assumptions

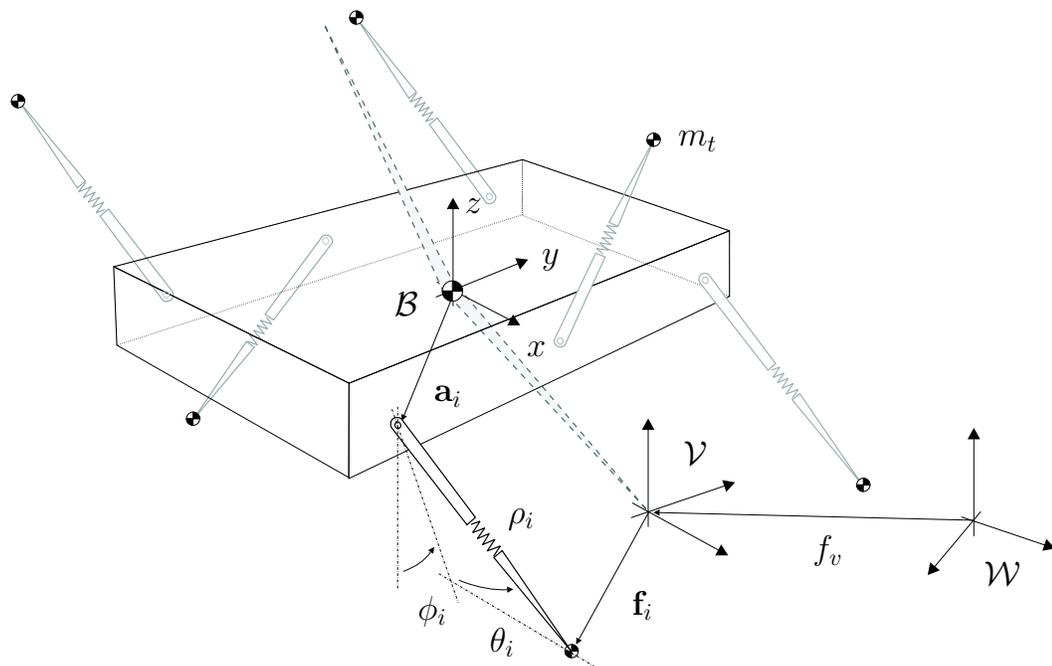


Figure 6.1: The compliant hexapod model.

Figure 6.1 illustrates the spatial hexapod model. Similar to Slimpod, three ref-

reference frames are defined: \mathcal{W} as the fixed inertial world frame, \mathcal{V} as the virtual toe frame, located at the foot of a “virtual leg” and finally \mathcal{B} as the body frame, affixed to the center of mass of the system. \mathcal{V} and \mathcal{W} have the same orientation except a yaw rotation around the z axis.

The model consists of a rigid body with six compliant legs whose attachment points are fixed at positions \mathbf{a}_i in the body frame. The body has mass m and inertia matrix \mathbf{I}_0 with respect to the body frame. Following the standard notation in robotics [21], the orientation of the body frame with respect to the world frame is expressed by the rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$. This also determines the yaw γ , pitch α and the roll β angular degrees of freedom of the body. Similarly, the position of the body in \mathcal{V} is denoted by \mathbf{b} .

Each leg has an associated mass $m_t \ll m$ introduced to capture its flight dynamics. These masses introduce three additional degrees of freedom per leg: the radial extension ρ_i , the hip angle ϕ_i and the sideways leg angle θ_i . Each leg incorporates radial springs with potential $V_{\rho_i}(\rho_i)$ and radial viscous dampers with coefficients d_{ρ_i} . Similarly, the sideways leg degrees of freedom also have associated torsional springs with potential $V_{\theta_i}(\theta_i)$ and viscous dampers with coefficients d_{θ_i} .

For legs in stance, the toe positions \mathbf{f}_i are fixed on the ground and the rigid body is directly acted upon by the leg forces. In contrast, legs that are in flight do not exert forces on the body. Instead, the motion of the leg is governed by the dynamics of the associated toe mass under the influence of the leg forces. Moreover, the position and velocity of toe masses in flight become independent coordinates of the overall dynamics.

Finally, the hybrid modes and transitions of this system are identical to those of Slimpod as described in Section 5.4.2, except an augmented state space to incorporate the additional degrees of freedom. The configuration space for the spatial hexapod

hence is defined as

$$\mathcal{C}_h := \{ \mathbf{c}_h \mid \mathbf{c}_h = [\mathbf{b}, \alpha, \beta, \gamma, \mathbf{f}_1, \dots, \mathbf{f}_6]^T \} . \quad (6.1)$$

6.2.2 The Continuous Dynamics

As in Section 5.4.3, the state of the rigid body and the positions of each toe are sufficient to encode the overall state of the body. Given the current configuration of the system $\mathbf{c}_h \in \mathcal{C}_h$, leg vectors in the body frame can be easily computed as

$$\begin{aligned} \mathbf{l}_i &= \mathbf{R}^T (\mathbf{f}_i - \mathbf{b}) - \mathbf{a}_i \\ \dot{\mathbf{l}}_i &= \frac{d}{dt} (\mathbf{R}^T) (\mathbf{f}_i - \mathbf{b}) + \mathbf{R}^T (\dot{\mathbf{f}}_i - \dot{\mathbf{b}}) , \end{aligned} \quad (6.2)$$

where \mathbf{R} denotes $\mathcal{Y}\mathbf{R}$. Similarly, the legs states in polar coordinates, $\bar{\mathbf{l}}_i := [\rho_i, \theta_i, \phi_i]$, can be computed using the coordinate transformations

$$\begin{aligned} \rho_i &= \sqrt{ \mathbf{l}_{ix}^2 + \mathbf{l}_{iy}^2 + \mathbf{l}_{iz}^2 } & \dot{\rho}_i &= \frac{ \mathbf{l}_{ix} \dot{\mathbf{l}}_{ix} + \mathbf{l}_{iy} \dot{\mathbf{l}}_{iy} + \mathbf{l}_{iz} \dot{\mathbf{l}}_{iz} }{ \rho_i } \\ \phi_i &= \text{atan}(-\mathbf{l}_{iz}, \mathbf{l}_{iy}) & \dot{\phi}_i &= \frac{ \mathbf{l}_{iy} \dot{\mathbf{l}}_{iz} - \mathbf{l}_{iz} \dot{\mathbf{l}}_{iy} }{ \mathbf{l}_{iy}^2 + \mathbf{l}_{iz}^2 } \\ \theta_i &= \text{asin}(\mathbf{l}_{ix} / \rho_i) & \dot{\theta}_i &= \frac{ \rho_i \dot{\mathbf{l}}_{ix} - \dot{\rho}_i \mathbf{l}_{ix} }{ \sqrt{ \mathbf{l}_{iy}^2 + \mathbf{l}_{iz}^2 } } . \end{aligned} \quad (6.3)$$

The forces generated by the legs arise from both their radial and sideways compliance and damping, as well as the hip torques. Its components associated with radial and sideways degrees of freedom hence can be written as

$$\begin{aligned} F_{ri} &= -DV_{\rho_i}(\rho_i) - d_{\rho_i} \dot{\rho}_i \\ \tau_{\theta_i} &= -DV_{\theta_i}(\theta_i) - d_{\theta_i} \dot{\theta}_i . \end{aligned}$$

Consequently, the force and torque exerted on the body by the i^{th} leg are

$$\begin{aligned} \mathbf{F}_i &= \begin{bmatrix} \sin \theta_i & \cos \theta_i & 0 \\ \cos \theta_i \sin \phi_i & -\sin \theta_i \sin \phi_i & \cos \phi_i \\ -\cos \theta_i \cos \phi_i & \sin \theta_i \cos \phi_i & \sin \phi_i \end{bmatrix} \begin{bmatrix} F_{ri} \\ \tau_{\theta_i} / \rho_i \\ \tau_i / (\rho_i \cos \phi_i) \end{bmatrix} \\ \boldsymbol{\kappa}_i &= (\mathbf{l}_i + \mathbf{a}_i) \times \mathbf{F}_i . \end{aligned}$$

Finally, the continuous dynamics within any of the modes are classical rigid body dynamics under the effect of external force and torques, yielding

$$\begin{aligned} m\ddot{\mathbf{b}} &= -[0, 0, mg]^T + \mathbf{R} \sum_{i=1}^6 s_i \mathbf{F}_i \\ \mathbf{R}\mathbf{I}_0\mathbf{R}^T\dot{\mathbf{w}} &= -J(\mathbf{w})\mathbf{R}\mathbf{I}_0\mathbf{R}^T\mathbf{w} + \mathbf{R} \sum_{i=1}^6 s_i \boldsymbol{\kappa}_i \\ \dot{\mathbf{R}} &= J(\mathbf{w})\mathbf{R}, \end{aligned}$$

where we define

$$J([w_x, w_y, w_z]^T) := \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}.$$

6.3 Spatial Extensions to the BSLIP Template

The Bipedal SLIP model of Chapter 5 is constrained to move in the saggital plane. As a result, it cannot capture components of the motion in the horizontal plane and hence is insufficient as a template for spatial locomotion. However, our embedding controller needs a concise representation of characteristic spatial motions, in particular for steering control. Consequently, this section introduces a horizontal plane template as an extension to the BSLIP template, to represent body yaw and sideways body translation.

Unlike the saggital plane BSLIP template, the horizontal plane template is much less informed by the biomechanics of locomotion in animals and insects. Instead, it is mainly inspired from RHex's morphology and the properties of the spatial hexapod model. Its presence is more of necessity than as a descriptive model.

There are a number of physically motivated horizontal plane locomotion models in the literature for hexapedal insects [67, 68]. The applicability of these models in the design of a complete spatial controller, however, is limited by the fact that they are developed in isolation from the saggital plane SLIP template. In contrast, the

extensions of this section effectively encode the coupling between the horizontal and saggital components of motion.

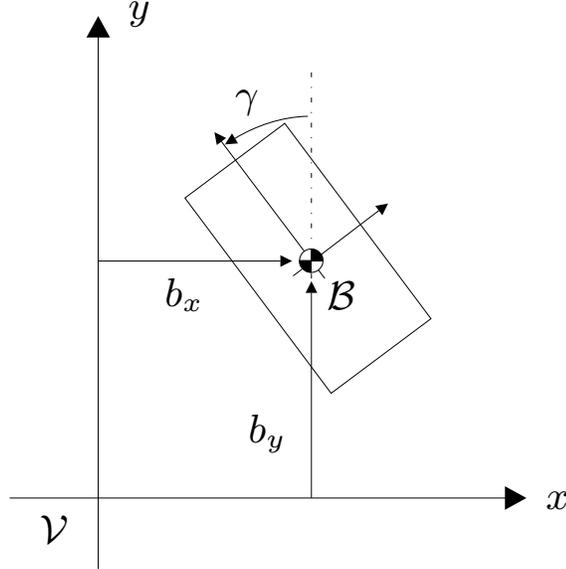


Figure 6.2: Horizontal plane extensions to the Bipedal SLIP template.

Figure 6.2 illustrates the horizontal component of the extended template. The saggital plane component is identical to the BSLIP template of Section 5.2, instantiated on the $y - z$ plane of the virtual foot frame \mathcal{V} . The dynamics of the saggital plane are assumed to be completely decoupled from the rest of the template.

As part of the horizontal plane extensions, we introduce two new variables, the yaw angle γ as the angle between the body frame \mathcal{B} and the virtual foot frame \mathcal{V} and body position b_x along the x axis. Both of these degrees of freedom have restoring forces towards their neutral positions, $\gamma = 0$ and $b_x = 0$, as well as linear damping. Moreover, the yaw degree of freedom has controllable torque actuation.

The horizontal dynamics also have two different modes, stance and flight, inherited from the saggital plane template. During stance, the dynamics in these degrees of

freedom are decoupled and take the form

$$\begin{aligned}\ddot{\gamma} &= -K_\gamma\gamma - K_{\dot{\gamma}}\dot{\gamma} - \tau_\gamma \\ \ddot{b}_x &= -K_x b_x - K_{\dot{x}}\dot{b}_x .\end{aligned}$$

In contrast, the flight dynamics are unactuated and take the form

$$\begin{aligned}\ddot{\gamma} &= 0 \\ \ddot{b}_x &= 0 .\end{aligned}$$

Transitions between stance and flight modes are governed by the saggital plane dynamics. The horizontal plane dynamics do not introduce any new modes and hence do not have separate transition rules.

The decoupling of the saggital plane from the horizontal plane makes it possible to solve the extended template by first considering the SLIP dynamics. Once the durations of individual modes are computed, the horizontal plane dynamics can be considered, yielding an extended apex return map. In later sections of this chapter, we will introduce a torque control strategy for the yaw degree of freedom, motivated by the structural constraints of the spatial hexapod model.

6.4 Template Control of Hexapod Locomotion

6.4.1 Virtual Leg Coordinates and Euler Angle Definitions

As in the Slimpod, the design of the embedding controller is most conveniently done in virtual leg coordinates, illustrated in Figure 6.3. Two of the coordinates, ψ and ξ correspond to their counterparts in the Bipedal SLIP. The third coordinate, η augments the basic model with a sideways angle.

In addition to these spatial SLIP coordinates, we will also use Euler angles to encode the orientation of the body [35]. Let γ , α and β as the yaw, pitch and roll of the body, defined as successive rotations first around the z axis, then around the

defined as

$$\mathcal{C} := \{ \mathbf{c} \mid \mathbf{c} = [\xi, \psi, \eta, \gamma, \alpha, \beta] \}$$

We are now in a position to derive the translational equations of motion

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m\xi^2 & 0 \\ 0 & 0 & m\xi^2 \end{bmatrix} \ddot{\mathbf{c}}_p = \begin{bmatrix} m\xi\dot{\psi}^2 + m\xi\dot{\eta}^2 - mg \cos \psi \\ -2m\xi\dot{\xi}\dot{\psi} + mg\xi \sin \psi \sin \eta \\ -2m\xi\dot{\xi}\dot{\eta} - mg\xi \cos \psi \cos \eta \end{bmatrix} + \mathbf{K}_p$$

as well as the rotational equations of motion

$$\begin{bmatrix} \mathbf{I}_y \sin^2 \alpha + \cos^2 \alpha (\mathbf{I}_z \cos^2 \beta + \mathbf{I}_x \sin^2 \beta) & (\mathbf{I}_z - \mathbf{I}_x) \cos \alpha \cos \beta \sin \beta & \mathbf{I}_y \sin \alpha \\ (\mathbf{I}_z - \mathbf{I}_x) \cos \alpha \cos \beta \sin \beta & \mathbf{I}_x \cos^2 \beta + \mathbf{I}_z \sin^2 \beta & 0 \\ \mathbf{I}_y \sin \alpha & 0 & \mathbf{I}_y \end{bmatrix} \ddot{\mathbf{c}}_r = f(\mathbf{c}_r, \dot{\mathbf{c}}_r) + \mathbf{K}_r .$$

where $\mathbf{K}_p := [K_\xi, K_\psi, K_\eta]^T$ and $\mathbf{K}_r := [K_\gamma, K_\alpha, K_\beta]^T$ are the effective external forces and $f(\mathbf{c}_r, \dot{\mathbf{c}}_r)$ captures the free rigid body dynamics in Euler angle coordinates¹.

The overall forcing vector is defined as

$$\mathbf{K} := \begin{bmatrix} \mathbf{K}_p \\ \mathbf{K}_r \end{bmatrix} = D_{\mathbf{c}} \boldsymbol{\rho} \mathbf{F}_r + D_{\mathbf{c}} \boldsymbol{\theta} \boldsymbol{\tau}_\theta + D_{\mathbf{c}} \boldsymbol{\phi} \boldsymbol{\tau} .$$

The derivation details of the components of \mathbf{K} as well as a singularity analysis of the jacobian corresponding to the active hip actuation is presented in Appendix E.

6.4.3 Structure of the New Spatial Controller

The locomotion controller for the spatial hexapod duplicates the components outlined in Section 5.5.1, with two exceptions. The first difference is in the placement of the virtual foot which, in this case, also incorporates the yaw degree of freedom. The second difference is the control of the active tripod.

¹In designing the embedding controller, we will assume that the angular velocity of the rigid body remains relatively small and $f(\mathbf{c}_r, \dot{\mathbf{c}}_r)$ is negligible.

For both the Slimpod and the hexapod, the beginning of each step is marked by the touchdown of at least two of any active tripod legs. In the beginning of each such step, the controller also chooses a new position and orientation for \mathcal{V} based on the gait level control commands. As in the Slimpod, the position of the new frame is placed at the tip of the virtual leg as commanded by the gait level BSLIP controller, yielding

$$f_v = \mathbf{r}_B + \bar{\xi}_t \begin{bmatrix} -\sin \bar{\psi}_t \sin \gamma_t \\ \sin \bar{\psi}_t \cos \gamma_t \\ -\cos \bar{\psi}_t \end{bmatrix}$$

where γ_t is the body yaw at touchdown and the adjusted touchdown angle and length are determined as

$$\bar{\psi}_t = \begin{cases} \text{acos}(b_z/\xi_t) & \text{if } \xi_t \geq b_z \\ \psi_t & \text{otherwise} \end{cases}$$

$$\bar{\xi}_t = \begin{cases} \xi_t & \text{if } \xi_t \geq b_z \\ b_z/\cos \psi_t & \text{otherwise} \end{cases}.$$

Furthermore, in the spatial case, the orientation of the new virtual toe frame is chosen to be the body yaw angle at the beginning of the step.

6.4.4 Control of the Active Tripod

Ideally, we would like the embedding controller to achieve exact targets for all components of the state, yielding

$$\mathbf{K}^* = \left[U^*(\xi), 0, 0, M_\gamma^*, M_\alpha^*, M_\beta^* \right]. \quad (6.4)$$

However, unlike the Slimpod, the spatial hexapod is usually underactuated, with only the active tripod in contact with the ground. As a consequence, even in the absence of singularities, it is impossible to exactly satisfy all the coordinates in (6.4). Fortunately, the structure of the jacobian $D_c\phi$, presented in Appendix E, suggests possible reductions that substantially simplify the problem.

First of all, the morphology of RHex and hence the hexapod model does not allow much affordance over the radial component of (6.4). In fact, at the singular configuration where all the legs and the virtual SLIP are parallel, there is no affordance in that direction. Consequently, we are forced to drop the radial component from the inversion of the jacobian and assume that the resulting natural dynamics will be close to desired as a result of the passive leg compliance.

A similar argument also holds for η , the sideways angle of the spatial SLIP. In this case, however, the natural dynamics are quite different than what would be required for a passive virtual SLIP. The passive sideways compliance and sprawl in the legs forms a restoring force towards $\eta = 0$ and naturally attracts the trajectories towards the saggital plane determined by the equilibrium state of the stance tripod. Even though we do not have any analytical evidence on the nature of these natural dynamics, we believe that this sideways restoring force is instrumental in properly embedding saggital plane SLIP dynamics in the spatial hexapod.

We are still left with four of the virtual toe coordinates: the saggital SLIP angle ψ and the body Euler angles γ , α and β . Among these, ψ is critical in achieving the target dynamics and directly determines the quality of the embedding. Our controller hence always attempts satisfying the target vector field associated with ψ .

Unfortunately, we only have two degrees of actuation freedom left to achieve the vector field components corresponding to the three Euler angles. To overcome this problem, we adopt a hybrid control strategy where the affordance is timeshared among subsets of the orientation coordinates.

In this context, we use once again, the structure of the jacobian $D_{\mathbf{c}}\phi$ (see Appendix E). Around the singular configuration with all legs vertical and a neutral body orientation, there is no affordance over the roll. Moreover, the ψ and α coordinates are coupled and cannot be independently controlled. The only coordinate that can effectively be controlled is the yaw angle. In contrast, configurations away from the singularity admit independent affordance over both thre pitch α and the roll β .

Our controller design hence considers two sets of different coordinates for two different parts of the stance phase. The first set incorporates the saggital SLIP angle, the body pitch and the body roll, defining $\mathbf{c}_1 := [\psi, \alpha, \beta]^T$. This set is used for configurations away from the singular vertical posture. The second set, used for configurations close to the singularity, is defined as $\mathbf{c}_2 := [\psi, \gamma]^T$, incorporating only the saggital SLIP angle and the body yaw.

The continuous controllers associated with each of these sets are identical to the constructions of Section 5.5.7, using two submatrices of the jacobian corresponding to the coordinates in \mathbf{c}_1 and \mathbf{c}_2 :

$$\begin{aligned} \mathbf{J}_1 &:= (D_{\mathbf{c}_1}\phi)_{\mathcal{I}_a} \\ \mathbf{J}_2 &:= (D_{\mathbf{c}_2}\phi)_{\mathcal{I}_a} . \end{aligned}$$

Based on these jacobians, we also define formally, the set of singular configurations;

$$\mathcal{C}_s := \{ \mathbf{c} \mid \det(\mathbf{J}_1 |_{\mathbf{c}}) < d_{min} \} .$$

Typically, the trajectories of the hexapod go through \mathcal{C}_s in midstance, where all three legs are close to vertical and the body orientation is somewhat neutral. The choice of d_{min} is done through manual tuning, and affects the peak torque requirements as well as the performance of the yaw control.

Finally, we are in a position to specify the particular torque choices corresponding to the orientational coordinates. In this thesis, we choose very simple linear feedback laws and use

$$\begin{aligned} M_\gamma^* &= -K_\gamma\gamma - K_{\dot{\gamma}}\dot{\gamma} \\ M_\alpha^* &= -K_\alpha\alpha - K_{\dot{\alpha}}\dot{\alpha} \\ M_\beta^* &= -K_\beta\beta - K_{\dot{\beta}}\dot{\beta} , \end{aligned}$$

completing the design of the active tripod controller.

6.5 Simulation Studies

In this section, we investigate in simulation, the existence and stability of attracting limit cycles for spatial hexapod locomotion under the embedding controller. As a result of the BSLIP template’s nature, the controller affords explicit affordance over the location of the fixed point, yielding the ability to control the speed of locomotion, while maintaining asymptotic stability of the corresponding fixed points. In order to characterize the performance of the closed loop system, stability studies over a variety of speeds are presented.

6.5.1 Numerical Environment and Model Parameters

The simulation environment that we used to generate the results presented in the following sections is *SimSect* [66], a hybrid dynamical simulation environment. It was originally created for the forward integration of the dynamical system described in Section 6.2.1. SimSect is now sufficiently general to perform the integration of arbitrary hybrid dynamical systems and was also used to support the Bipedal SLIP simulation studies of Section 5.3.6.

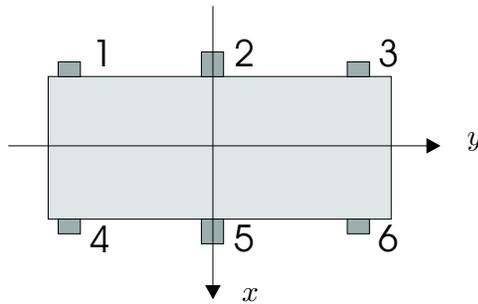


Figure 6.4: Numbering convention for the spatial hexapod legs.

The results of the following sections were obtained using dynamical parameters of the spatial hexapod model that match RHex’s morphology as closely as possible. RHex’s body is roughly symmetric with respect to all of the three major planes. With

\mathbf{a}_1	\mathbf{a}_2	\mathbf{a}_3	\mathbf{a}_4	\mathbf{a}_5	\mathbf{a}_6
$[-0.1, -0.2, 0]$	$[-0.15, 0, 0]$	$[-0.1, 0.2, 0]$	$[0.1, -0.2, 0]$	$[0.1, 0, 0]$	$[0.1, 0.2, 0]$

Table 6.1: Leg attachment points for the spatial hexapod. Units are in meters.

m	m_t	I_x	I_y	I_z
$7kg$	$0.05kg$	$0.1kgm^2$	$0.029kgm^2$	$0.11kgm^2$

Table 6.2: Dynamical parameters of the spatial hexapod model

the leg index assignments illustrated in Figure 6.4, Table 6.1 gives the attachment locations of all six legs.

RHex’s body is also roughly rectangular along all the body axes and measures $20cm$, $53cm$ and $15cm$ along the x , y and z axes, respectively. As a consequence, we assume that the inertia matrix of the body is diagonal, $\mathbf{I}_0 = \text{diag}(I_x, I_y, I_z)$, and compute the diagonal elements using these dimensions and a uniform mass distribution. Table 6.2 gives the computed values, together with the mass parameters.

In order to complete the metric specification of the model, the structural and dynamic parameters of the legs also need to be specified. Normally, this would need to be done through experimental characterization of RHex’s various leg morphologies to identify their material properties as well as specific stiffness and damping models. Unfortunately, dynamic characterization and modeling of composite materials and structures is a difficult problem and remains outside the scope of this thesis. As a consequence, we have chosen to use a simple linear spring model. The stiffness parameters used in our simulation were obtained from a recent, preliminary characterization of various 4-bar leg designs for RHex [53].

In this context, however, there is an important problem that requires attention. All the experiments performed with RHex so far used identical legs mounted to all

ρ_0 (m)	θ_0 (rad)	$k_{\rho a}$ (N/m)	$k_{\rho b}$ (N/m)	$k_{\theta a}$ (Nm/rad)
0.17	0	1500	2000	300
$k_{\theta b}$ (Nm/rad)	$d_{\rho a}$ (Ns/m)	$d_{\rho b}$ (Ns/m)	$d_{\theta a}$ (Nms/rad)	$d_{\theta b}$ (Nms/rad)
400	12	16	1.2	1.6

Table 6.3: Leg model parameters for the spatial hexapod. Spring constants k and damping coefficients d with the subscript b denote those of the middle legs, whereas parameters with the subscript a correspond to the front and back legs.

six hips. However, this arrangement introduces an inevitable lateral asymmetry as a consequence of the structure of the tripods, with only the middle leg providing support on one side to balance the two legs of the opposite side. Unfortunately, for the embedding controller, this asymmetry substantially impairs the roll stability during locomotion. Often, this destroys the asymptotically stable fixed points and the system can only retain bounded stability. In order to address this issue, we have chosen to work with stiffer middle legs to balance the moment arms of the legs on the opposite side. The implementation of this difference is still possible in practice as leg designs with stiffer material properties are already present for RHex.

The final and perhaps the most problematic parameter is the damping in the legs. Because there are currently no empirical measurements of this parameter, we have chosen to work with a simple linear damping model tuned through qualitative matching of experiments and simulation using the open-loop locomotion controller. Even though this is sufficient for the demonstration of the controller design concept, a much more careful modeling effort would clearly be needed for any physical instantiation. Table 6.3 gives the leg model parameters.

6.5.2 Presence and Character of Stable Orbits

The template embedding controller described in Section 6.4 is capable of producing asymptotically stable locomotion for a reasonable range of speeds. This section describes an additional constraint we impose on the controller to significantly extend its stability, as well as the character of the resulting attracting limit cycles.

For reasons outlined in Section 5.3.6, the precompression of the virtual leg at touchdown has a significant effect on the stability of the body orientation for the hexapod. Most importantly, when precompressions smaller than a certain value are commanded, the approximate nature of the embedding controller results in part of the injected energy spilling over to body roll and pitch, eventually destabilizing the system. On the other hand, limits on the precompression that are too restricting destroy the limit cycles and stable locomotion cannot be obtained. Furthermore, simulation studies show that the appropriate value of ξ_{min} to avoid both of these problems depends on the goal speed, which introduces a further complication.

Within these narrow ranges of admissible precompressions, the energy based dead-beat height control of Section 5.3.4 does not perform well. The effects of both the inaccurate nature of SLIP embedding and the additional energy of body angular velocity that is unknown to the BSLIP template significantly impair the stability of the height control.

For these reasons, we modify the hexapedal controller to use a fixed touchdown compression ξ_t corresponding to each velocity setting. Together with other gains in the controller, this parameter is tuned offline and is closest in spirit to the constant thrust control of Raibert's hoppers. Previous research demonstrates that this approach can yield asymptotic convergence to a fixed height in the presence of damping [44] and also admits low-bandwidth height control through slow evolution of the fixed precompression. It is also important to note that the final height value lies on the upper boundary of the regions illustrated in Figure 5.5.

For speeds in the range $1m/s$ to $2m/s$, the closed loop system resulting from the action of this controller on the hexapod has an asymptotically stable limit cycle, whose size and shape depends on the forward speed goal at the task level. Based on simulation studies with appropriate tuning of the body orientation gains and the precompression limit, this limit cycle seems to have the same structure in all cases.

The projection of these limit cycles on the task coordinate space — the four dimensional state space of the virtual BSLIP — always yields period one orbits. This also represents a fixed point in the apex gait parametrization space, obtained through a transverse section of the BSLIP state space at the highest point of the flight phase. Figure 6.5 illustrates the trajectories of an example run converge on the limit cycle.

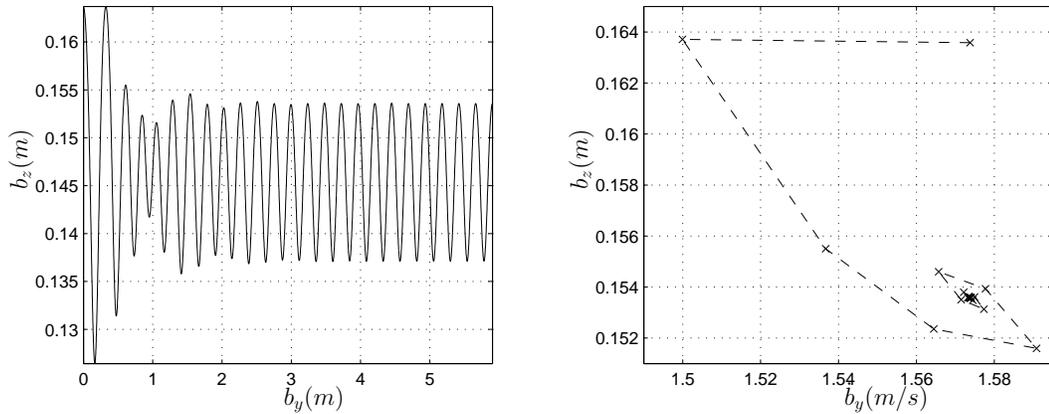


Figure 6.5: Illustration of the attracting limit cycle for an example run with the spatial hexapod for $\dot{b}_y^* = 1.6m/s$. The leftmost figure shows the sagittal plane position of the robot while the middle figure shows the progression of the gait level apex state towards the fixed point in the $\dot{b}_y - b_z$ plane.

On the other hand, on the overall, the limit cycle exhibits a period two behavior with respect to the apex transverse section as a result of the degrees of freedom in the body orientation. Figure 6.6 illustrates the the yaw, pitch and roll trajectories for the same example run as Figure 6.5. The alternation of the left and right tripods and the resulting period two roll and yaw behavior are clearly seen in the trajectories.

Note, however that due to the foreaft symmetry of the tripods, the periodicity of the pitch degree of freedom is still one.

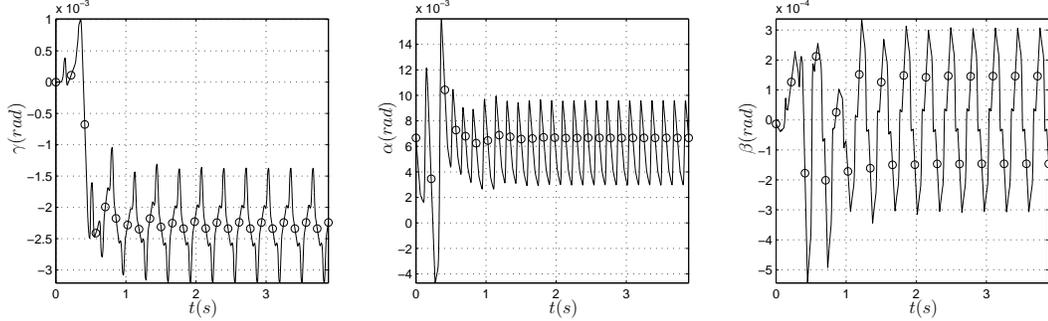


Figure 6.6: Yaw, pitch and roll trajectories over time for an example run with the spatial hexapod for $\dot{b}_y^* = 1.6m/s$. The circles indicate the apex points throughout the trajectories.

This structure of the limit cycles induced by the embedding controller is the same across all the simulations that are presented in subsequent sections. Naturally, the tuning of the gains in the system impacts both the presence and character of these limit cycles. However, a detailed account of different types of stable orbits that can be achieved remains outside the scope of this thesis.

6.5.3 Stability and Basins of Attraction

The action of the embedding controller described in preceding chapters on the spatial hexapod model results in an autonomous, closed loop dynamical system. In Section 6.5.2, we have identified stable fixed points of this system, which depend on the gait level forward speed command to the locomotion controller. This section investigates in simulation, the domain of attraction of this dynamical system for six different goal speed settings spanning the range of speeds that can be achieved by the controller.

Unfortunately, the identification of the basins of attraction through simulations

over the whole space of initial conditions is not feasible. The configuration space of the hexapod, as defined in (6.1) is 24 dimensional, yielding a 48 dimensional state space to be explored. This is clearly not a practical goal.

Nevertheless, it is possible to exploit symmetries in the morphology as well as some properties of the controller to significantly reduce this space to a much more practical size.

First of all, we constrain our initial conditions to those at the apex point of the center of mass trajectory. This corresponds to states where $\dot{b}_z = 0$ and all the legs are in flight. Furthermore, we assume that the initial leg states are chosen to be exactly at their target location based on the commands of the embedding controller. These assumptions reduce the dimension of the initial state space to 11, which are all the states of the rigid body except the vertical velocity.

Similarly, under these assumptions, the spatial hexapod model described in Section 6.2.1 is symmetric with respect to the horizontal plane position and orientation coordinates. This results in further reduction of the relevant state space to a dimension of 8, consisting of body height b_z , body velocities \dot{b}_x, \dot{b}_y , the initial body pitch α and roll β as well as the angular velocity components $\dot{\gamma}, \dot{\alpha}$ and $\dot{\beta}$.

Even this reduced space of initial conditions to be considered in the identification of the domain of attraction is not practical for an exhaustive set of simulations. Even a reasonable grid with 10 intervals along each axis would require 10^8 simulations, which would take around 570 years to complete with an optimized C implementation of the simulations and the computational resources currently available to me.

Consequently, this section only presents projections of the theoretical basins of attraction onto the $\dot{b}_y - b_z, \alpha - \beta$ and $\dot{b}_x - \dot{b}_y$ planes. By means of several isolated sample runs for different arbitrary initial conditions, I have found these pairs of coordinates to have the most impact on the stability of the system compared to the rest of the space.

Figure 6.7 illustrates a cross section of the basin of attraction by the initial $\dot{b}_y - b_z$

plane. Two aspects of the graphs need clarification. Firstly, the blank regions at the bottom left corner of each graph represent invalid initial conditions rather than truly unstable points. These points correspond to configurations where one or more of the legs are initially in contact with the ground, and are hence invalid initial conditions for our investigation.

The second observation is the presence of several scattered, isolated patches of stable initial conditions. This type of behavior is characteristic of hybrid systems, where even very small difference in the initial condition can quickly lead to dramatic changes in the system trajectories due to discrete jumps in the mode sequence. For example, a small difference in the body orientation may lead to one of the legs lifting off, which in turn aggravates the deviation from the trajectory that would have been observed otherwise. A natural consequence of this is that the basin of attraction loses its connectedness.

In addition to these two aspects, the basins of attraction for each speed are surprisingly large. These results demonstrate that, within the context of the model and its assumptions, it is possible to obtain stable regulation of forward speed and recover from disturbances of fairly large size.

Figures 6.8 and 6.9 illustrate the cross sections of the basin of attraction by the $\alpha - \beta$ and $\dot{b}_x - \dot{b}_y$ planes, respectively. Both of these sets of simulations confirm that the template embedding controller is capable of achieving reasonable stability in achieving the target fixed point.

An aspect of the fixed points which is clear from these graphs is the diminishing size of the basin with increasing goal speeds. Especially at $\dot{b}_y^* = 2.0m/s$, the stability with respect to body orientation as well as the initial saggital plane states is substantially smaller. This is a somewhat expected result of the increase speed during stance, which also decreases its duration and makes the stabilization of the body orientation more fragile.

Figure 6.8 also demonstrates that the controller is more sensitive to the roll degree

of freedom compared to the pitch. As a result of the kinematics of the tripods, even relatively small roll angles cause legs on one side to lose contact with the ground, destroying the control affordance. Furthermore, the embedding controller for two legs currently only attempts to stabilize the pitch with its diminished actuation affordance. As such, it becomes much harder to recover from disturbances in the roll than the pitch.

Overall, the simulations presented in this section suggest that the spatial hexapod under the action of the hexapod controller is reasonably stable for a large range of speed goal settings.

6.6 Conclusion

This chapter presents the extension of the BSLIP embedding controller design for the planar hexapod to the more complicated setting of the spatial hexapod. Towards this end, we introduced a rudimentary template for the components of the motion on the horizontal plane. Furthermore, we have extended most of the components for the planar controller, including the placement of the virtual toe, the stabilization of the body orientation as well as the control of the active tripod to address problems specific to the spatial setting.

In this context, simulation studies identified that the closed loop system that arises from the action of the embedding controller on the spatial hexapod model has asymptotically stable fixed points with reasonably large basins of attraction. Furthermore, the location of these fixed points can be tuned through the goal speed setting at the task level, yielding effective control over the gait parameters.

Even though the underlying model still deviates significantly from RHex both in its ground contact model and its correspondance to the dynamical parameters of RHex, the presence of asymptotically stable fixed points suggests that a physical instantiation of the control strategy on RHex is a strong possibility.

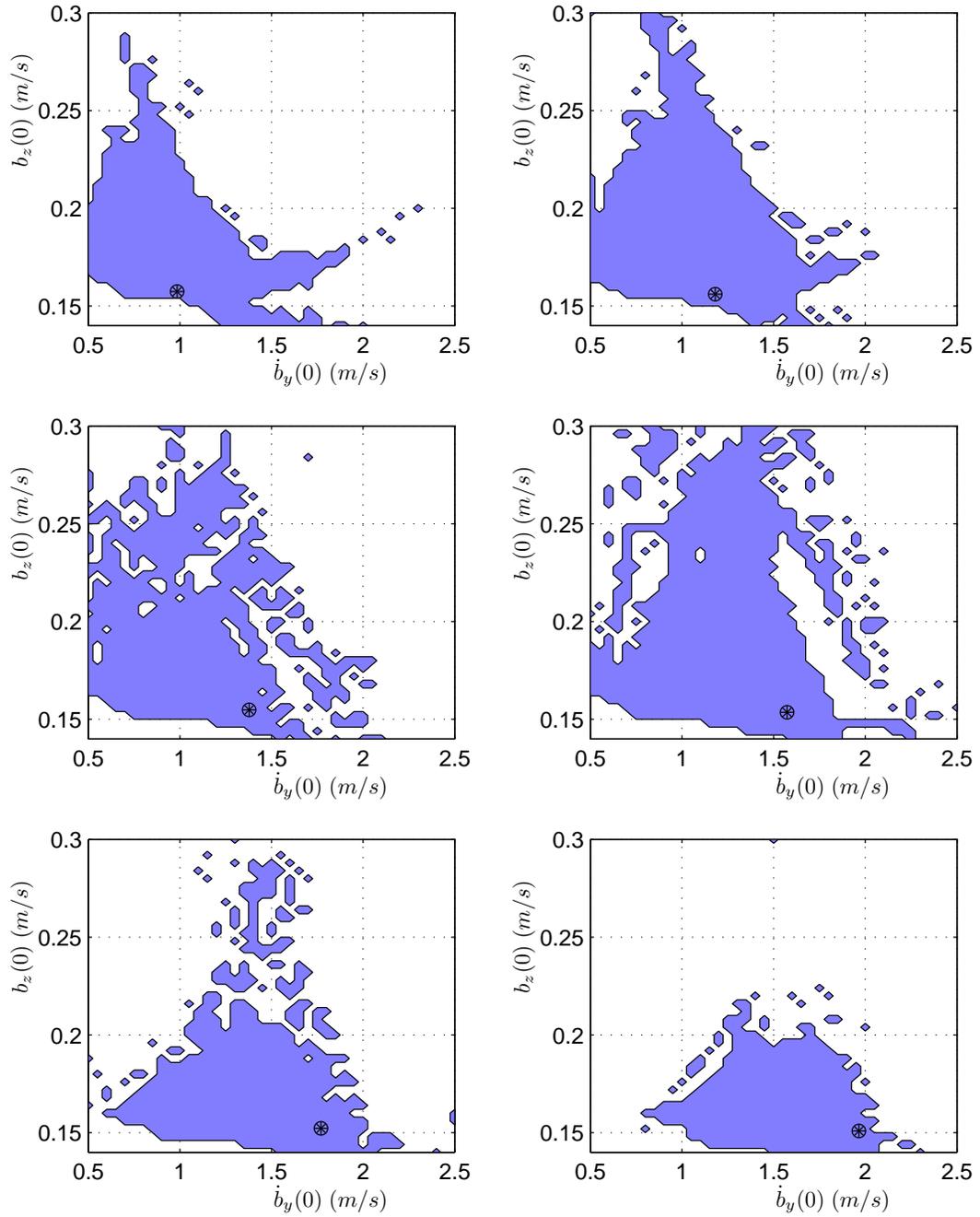


Figure 6.7: Projection of the basin of attraction of the hexapedal controller for 6 different speed goals $\dot{b}_y^* \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}m/s$ onto BSLIP apex coordinates. For each goal setting, the filled circle indicates the stable fixed point.

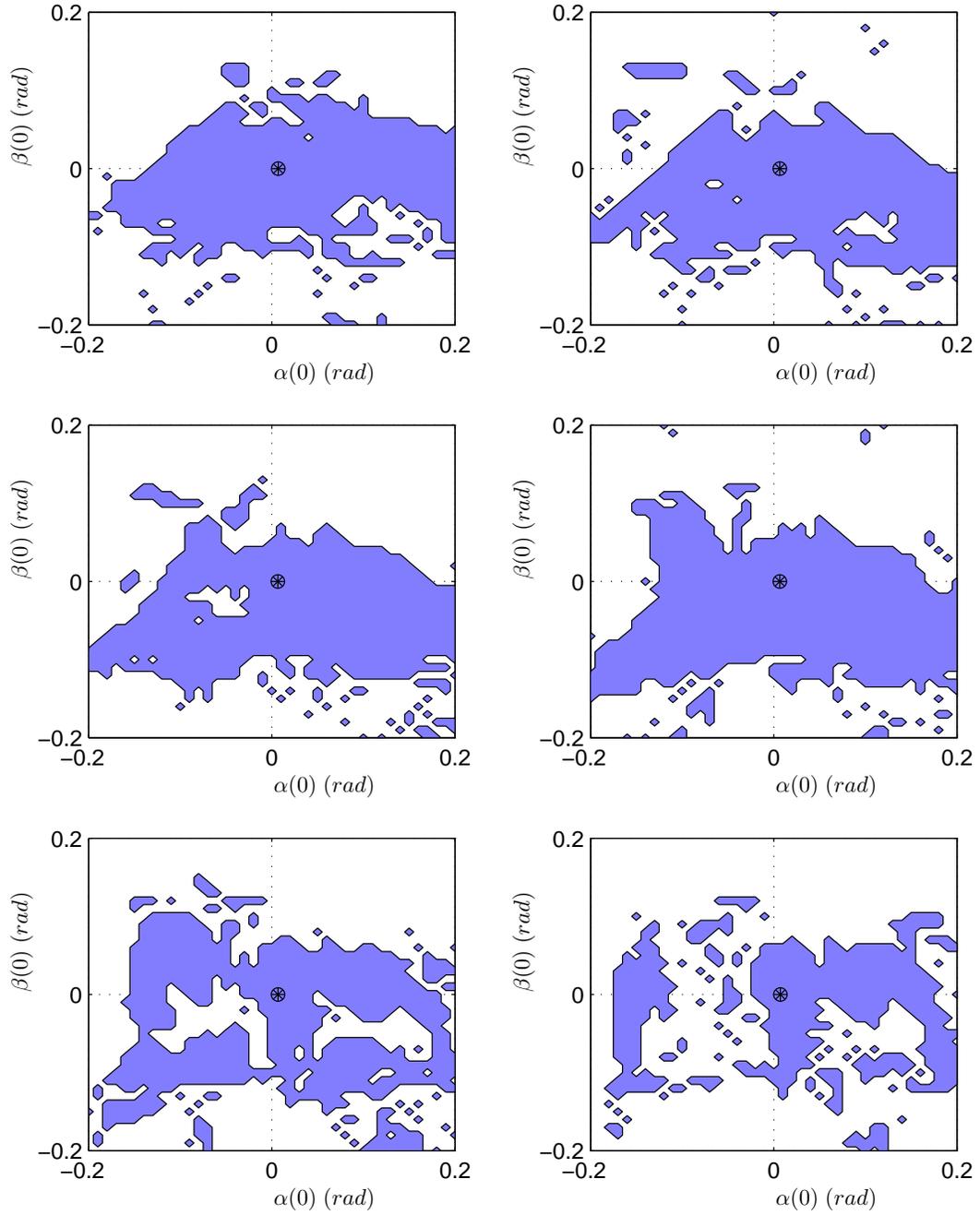


Figure 6.8: Projection of the basin of attraction of the hexapedal controller for 6 different speed goals $b_y^* \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}m/s$ onto pitch and roll coordinates. For each goal setting, the filled circle indicates the stable fixed point.

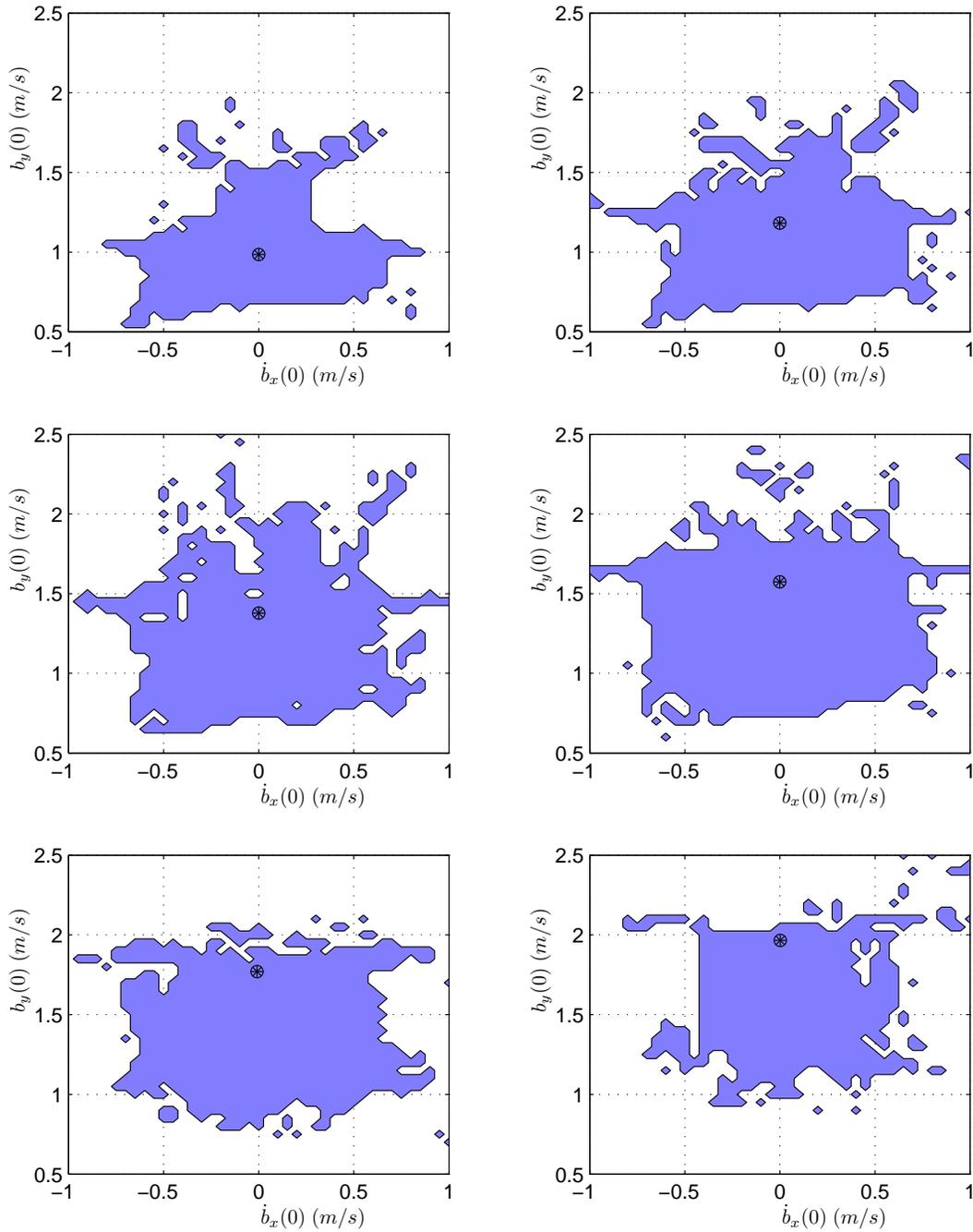


Figure 6.9: Projection of the basin of attraction of the hexapedal controller for 6 different speed goals $\dot{b}_y^* \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}m/s$ onto horizontal plane velocity coordinates. For each goal setting, the filled circle indicates the stable fixed point.

CHAPTER 7

Conclusion

This thesis concerns the design of RHex, an autonomous hexapedal robot, together with controllers to achieve dynamical locomotion of underactuated hexapedal robots. In this context, we presented two extreme solutions: a family of purely open-loop strategies that lack any task level feedback, and a dynamical template based controller design which relies on high bandwidth feedback to reduce the controlled dynamics to a simple and easily controllable system.

The most natural next step is the experimental implementation of the template based controllers on RHex. In this section, we summarize the outstanding issues that need to be resolved before this goal can be achieved, as well as potential future directions for our research.

The most striking inaccuracy in our hexapedal models is the absence of a physically realistic ground friction model. Clearly, our assumption that toes always remain stationary once they come in contact with the ground is far from the truth. In fact, leg slippage is one of the most common failure modes of even the open-loop controllers at higher speeds. Even though for initial experimental validation of our results, it would be possible to engineer experiments to yield sufficiently high coefficients of friction between the toes and the ground, more robust implementations will require physically realistic ground contact models.

Similarly, we presently do not have any accurate, experimentally verified models for RHex’s legs. The simple linear compliance and damping models that we have employed in this thesis promise to capture some of the essential aspects of the morphology. However, both the models themselves and the associated parameters need to be refined and verified for a successful experimental implementation of the controller.

The second important set of challenges arises from the high bandwidth feedback required by our inverse dynamics style controller. RHex is currently unable to meet either the computational demands of the control laws or the range of sensory feedback that they require. Fortunately, recent developments promise to extend RHex’s sensing capabilities towards accurate estimation of its body and leg states. In this context, it would be critical to characterize of the performance of our controller under various levels of sensor noise and identify which sensory components need to be most accurate. The design and implementation of a sufficient sensory suite to satisfy these requirements as well as the additional limitations that would be imposed by realistic frictional models, are outstanding problems.

It is also important to consider that such an extreme solution with critical reliance on feedback may not be best for a practical implementation. In fact, biological evidence suggests that high level of sensory feedback is not necessary in achieving basic tasks of locomotion. For many animals, such as cats and cockroaches, the cyclic movements of limbs during locomotion are controlled by groups of neurons, capable of autonomously sustaining their excitation patterns, modulated by low bandwidth sensory input [57]. Consequently, these neural groups, called Central Pattern Generators (CPG), are able to control the periodic movements of the limbs to achieve cyclic locomotion tasks even when the sensory bandwidth is relatively small. Experiments show that walking behavior can be observed in cats lacking higher levels of the nervous system hence having no inertial or visual sensory inputs.

Pure open-loop control, however, is also an extreme solution. The lack of feedback, as usual, has considerable impact on robustness, stability and controllability of the

resulting locomotion. For instance, even though passive feedback incorporated by RHex’s mechanical design through sprawled posture and leg compliance yields stable and robust locomotion on flat terrain, it fails to perform well on various other tasks such as climbing slopes or running with an aerial phase at higher speeds. Furthermore, the energetics of locomotion, especially on rough terrain suffer from lack of response to environmental disturbances.

It is hence clear that controllers of higher performance are likely to be found in the range between these two extremes, combining feedback with open-loop components. Several recent developments within the RHex project [46, 43] seem to suggest that such combinations are feasible and indeed result in increased performance and extended behaviors.

The implementation of template based controllers on an experimental platform promises to lead to a wide range of future directions for research. In particular, it becomes much more feasible to perform planning, navigation and mapping through the simple task level interface offered by the BSLIP template. Naturally, a much better understanding of the horizontal plane motions, leading to a careful performance characterization and refinement of the associated template would be necessary before such efforts can proceed much further. Nevertheless, we believe that the stability and the control affordance granted by the basic embedding controller will be instrumental towards effective higher level control of legged locomotion.

Independently, the energetic performance granted by our new control strategy still remains to be characterized. As a result of energy recycling through passive leg compliance, as well as the relatively long aerial phases characteristic of the embedding controllers, we suspect that there will be significant improvements in RHex’s energetic efficiency. Such improvement promise to offer increased runtime and robustness resulting from lower torque requirements, all of which contribute towards a fully autonomous legged platform.

In summary, there is still a long research path to our ultimate goal of building

a fully autonomous legged platform capable of surviving a large range of outdoor environments for extended periods of time. However, we believe that our work represents an important step in this direction and begins to develop some of the tools and concepts that are necessary to achieve this goal.

APPENDICES

APPENDIX A

Details of the RHex Experiments, Experimental Setup and Failure Modes

A.1 Running Experiments

We ran the robot over carpet, linoleum, grass and gravel. The carpet and linoleum surfaces were standard office floors found close to the lab. The grass was wet on the day of the experiment and showed height variations of about 2 cm. The gravel patch contained fairly large gravel pieces (see Figure 2.1) between three and eight cm diameter. For all the experiments, the robot was driven over a test stretch of 2 m. In order to obtain precise timing and to synchronize the data logging with the test stretch, a switch was mounted in the front of the robot, which was triggered as the robot ran into a Styrofoam panel held at the beginning and the end of the test stretch. The runs over each surface were repeated until ten successful runs were obtained. The average velocity and power consumption for each run was then computed with the available data.

Ten successive experiments were run for the carpet surfaces with no failures. One run on the Linoleum floor was discarded, since the robot deviated too much from the straight line. A total of 16 runs on grass were necessary, with six runs discarded. In five runs, the operator failed to align the start or stop trigger panel properly, and

the front legs pushed it aside, preventing the switch to be actuated. One run was abandoned due to R/C noise in the remote control command input. Gravel was more challenging - of the 25 runs, five were discarded because the robot deviated too much from a straight line, five due to the operator missing the trigger switch, one due to remote control noise, and four because the front switch broke on impact with the trigger panel.

A.2 Turning Experiments

The turning experiments were run on carpet. In order to reduce the data processing for this set of experiments, only six runs were processed in this fashion for each forward velocity, instead of the usual ten. Only few runs were discarded due to noise in the remote control which interfered with the velocity and/or the steering command.

A.3 Obstacle Crossing

The first obstacle was a 1.22 m long strip of 3" (7.62cm) thick Styrofoam board, a standard insulating construction material, cut to 15 cm height. This represents 80% of the robot's leg length and exceeds its 10.5 cm ground clearance by 4.6 cm, or almost 50%. The Styrofoam was chosen for for this experiment and the random obstacle course described below for its ready availability, low cost, and ease of cutting. It is softer than wood, yet hard enough that the robot does not deform it. The second obstacle was built from construction lumber and consisted of a 10 cm high and 63 cm wide base (as viewn in the sagittal plane) on top of which a 8.5 cm high and 3.5 cm wide block was mounted at a distance of 25 cm from the front and a 12.5 cm high and 8.5 cm wide second block was mounted at a distance of 50 cm from the front. In both experiments the control parameters were the same as in the walking

experiments above. All data shown was obtained by the visual tracking procedure described in Section 2.4.2, with the camera oriented for a perpendicular view of the sagittal plane. The average forward velocity of each run was obtained before, over, and after the obstacle.

Fourteen successive experiments were required and logged for both obstacles. From the runs over the first obstacle, the robot failed to surmount it only once, but the vision post processing algorithm failed to extract reliable position data for three successful runs, when the robot's direction after obstacles deviated significantly from straight path. From the fourteen successive runs over the composite obstacle, the robot failed twice to surmount the obstacle, and the vision post processing failed to extract data from the post-obstacle portion of two runs.

A.4 Obstacle Course

The experimental setup for the obstacle course consisted of ten randomly spaced obstacles of 12.2 cm height. The free spaces between the ten 3" (7.62cm) wide Styrofoam blocks were 1.07, 0.47, 0.78, 0.68, 1.02, 0.91, 0.66, 0.29 and 0.96 m, selected between 0.5 and 2 body lengths from a uniform random distribution. Thus the total obstacle course extended over 8.13 m, which also includes one half body length before and after the course. The time between start and finish was measured via a stopwatch. During these experiments an operator attempted to keep the robot on course using the limited directional control described above. Nevertheless, the directional disturbances due to the obstacles caused the robot at times to veer towards the lateral limits of the 1.2 m wide course. In those instances, operators who followed the robot along the course, placed a Styrofoam panel along the lateral limits to make up for the lack of side walls. When the collision angle with these walls was sufficiently small, the robot re-aligned itself with the course.

Due to the large number of runs required for this experiment, and the high power

requirements, we made an exception and ran the robot from higher capacity external batteries via an umbilicus. This greatly reduced the experimental effort by eliminating the need to recharge and exchange the on-board batteries. However, no performance improvement resulted from this arrangement, compared to running off freshly charged on-board batteries. The on-board batteries were kept in place to maintain the total robot mass.

A total of 26 successive experiments on the obstacle course were recorded. Of these, 16 were discarded for the following reasons: The robot turned itself sideways beyond quick recovery (2), shut itself off (1), required operator intervention through the R/C unit, such as turning in place or short reversal of direction to complete the course (3), turned itself on its back either by climbing up against the side walls (1) or the obstacle (3), wrapped the power cord around the legs (1), ends up "sitting" aligned with and on top of an obstacle, unable to reach the ground (3), or burned electrical circuits (1). The remaining eleven runs were used to calculate the velocity, power and specific resistance data shown in Figure 2.9.

A.5 Rough Surface

In order to re-create Full's rough surface [27], we compared the height distribution of his environment [25] to checkerboard arrays of randomly uniformly distributed block heights. When scaled to RHex's dimensions, we decided that a height variation of between 4" (10.16 cm) and 12" (30.48 cm), or 1.16 leg lengths was a good match (Figure A.1). To simplify cutting by a local lumber yard, the block heights were discretized to 1" (2.54 cm) increments. The block width of 7" (17.78 cm) permitted the use of four standard 3.5"x3.5" (8.89cm x 8.89 cm) cross section lumber per checkerboard block. The total surface consisted of 72 blocks (6 by 12) thus requiring 288 individually cut lumber sections. The robot was run in the direction of the 12 block length of the surface, with wall panels on each side. Its cartesian position

	1	2	3	4	5	6	7	8	9	10	11	12
1	12"	8"	4"	7"	11"	12"	9"	6"	6"	7"	7"	8"
2	9"	4"	11"	8"	8"	10"	11"	5"	12"	6"	9"	12"
3	9"	8"	10"	9"	6"	10"	8"	5"	4"	7"	5"	6"
4	10"	6"	12"	10"	9"	11"	8"	8"	9"	6"	5"	8"
5	5"	11"	10"	10"	10"	12"	5"	11"	8"	10"	6"	8"
6	4"	9"	7"	8"	10"	8"	5"	6"	9"	12"	9"	11"

Figure A.1: Height distribution over the rough surface.

and orientation, projected onto the horizontal plane, was measured with the visual tracking setup described in the steering experiment, above.

We carried out 32 experiments on this surface, with a success rate of 50%. During the unsuccessful runs, the robot either ran head-on into a side wall or into one of the isolated posts (typically the isolated high block with coordinates 2,9 in Figure A.1 (3), broke a leg (twice), hit one of the walls (3), had to back up and continue forward (4). Also, four of the experiments did not complete due of R/C failure. From the 16 experiments that were successfully recorded, we have used 10 with the cleanest vision data to facilitate the post-processing.

APPENDIX B

Derivation of the Deadbeat Control Laws for the Basic SLIP

Recall the SLIP plant model of Section 4.3.1 $f_a : \mathcal{X}_a \times \mathcal{U} \mapsto \mathcal{X}_a$ where

$$\mathcal{U} = \{u \mid u = [\psi_t, k_c, k_d]^T\}$$

is the set of control inputs. The control problem at hand is to identify the sequence of discrete control inputs $\{u_i\}_{i=0}^n$ to asymptotically converge to the desired apex state.

The most direct way to find the deadbeat control u^* would be to invert the map f_a . However, the control inputs appear in the apex return map in a complicated manner making a direct computation of the inverse map difficult. In consequence, we introduce a new coordinate system, which affords an almost completely closed form inverse to an approximate return map.

We want the ability to control the SLIP hopper to achieve a goal state,¹

$$Z_l^* = [\psi_l, E_l^*, \delta_l^*, \phi^*]^T$$

We are looking for the the deadbeat control, \bar{u}^* , such that

$$Z_l^* = f_l(Z_l, \bar{u}^*)$$

¹As in Section 4.3.1 we can only choose three independent control objectives, here we select E_l , δ_l and ϕ

B.1 The Liftoff Return Map

Consider the new state and control sets,

$$\mathcal{Z}_l := \{Z_l \mid Z_l = [\psi_l, E_l, \delta_l, \phi]^T\}$$

$$\bar{\mathcal{U}} := \{\bar{u} \mid \bar{u} = [\psi_t, a_1, \alpha]^T\}$$

where E_l is the energy at liftoff, δ_l is the ratio of forward velocity to vertical velocity at liftoff and

$$a_i^2 := \frac{\xi_t^2 \dot{\psi}_t^2 + k_i / (m \xi_t^2)}{\dot{\xi}_t^2}; \quad i = 1, 2 \quad (\text{B.1})$$

$$\alpha^2 := \frac{a_2^2}{a_1^2} = \frac{\xi_t^2 \dot{\psi}_t^2 + k_c / (m \xi_t^2)}{\xi_t^2 \dot{\psi}_t^2 + k_d / (m \xi_t^2)} \quad (\text{B.2})$$

Assuming $\xi_l = \xi_t = \xi_0$, the liftoff return map $f_l : \mathcal{Z}_l \times \bar{\mathcal{U}} \mapsto \mathcal{Z}_l$ can be written as

$$f_l(Z_l, \bar{u}) := \begin{bmatrix} \vartheta_l \\ E_l + \Delta E_U + \Delta E_g \\ \mathbf{t}_{(1, -\vartheta_l)} \circ \mathbf{t}_{(\frac{-1}{\alpha}, \psi_t)}(\delta_t) \\ \frac{t_f}{t_s} \end{bmatrix} \quad (\text{B.3})$$

where

$$\vartheta_l := \psi_t - \mathbf{t}_{(1, \psi_t)}(\delta_t) \left(\frac{\alpha + 1}{\alpha} \right) \frac{1}{a_1} \text{acot}(a_1)$$

$$\Delta E_U := U(k_d, \xi) - U(k_c, \xi) = \frac{1}{2} m \dot{\xi}_t^2 (\alpha^2 - 1)$$

$$\Delta E_g := mg(\xi_l \cos \vartheta_l - \xi_0 \cos \psi_t),$$

$$t_f := \frac{1}{g} \sqrt{\frac{2}{m} \left(\frac{1}{\delta_l^2 + 1} \right)} \left(\sqrt{E_l - mg \xi_l \cos \psi_l} - \sqrt{E_l + \delta_l^2 mg \xi_l \cos \psi_l - (1 + \delta_l^2) mg \xi_0 \cos \psi_t} \right)$$

$$t_s := \sqrt{\frac{\xi_0^2}{\dot{\xi}_t^2}} \left(\frac{1}{1 + a_1^2} \right) \left(\frac{\alpha + 1}{\alpha} \right)$$

and we define the following two parameter family of functions,

$$\mathbf{t}_{(\sigma_1, \sigma_2)}(\chi) := \tan(\sigma_1 \text{atan}(\chi) + \sigma_2).$$

Notice that apart from certain values of the parameters (e.g. $\sigma_1 = 1$ and $\sigma_2 = 0$) this family cannot be expressed in terms of a single elementary function. Finally note that both δ_t and $\dot{\xi}_t$, which appear in (B.3) can be expressed in terms of Z_l and ψ_t .

B.2 Inverting the Return Map to Find Deadbeat Control

The simple form of the liftoff return map makes it possible, under a reasonable assumption, to reduce the inversion of f_l to the solution of a single equation in a single variable. The assumption that makes this possible is

$$\Delta E_g \equiv 0$$

This assumption is reasonable in practice since ΔE_g appears in (B.3) only as a result of the unnatural energy discontinuities at touchdown and liftoff due to our no-gravity stance model, and does not appear in the stance dynamics with gravity.

Given this assumption, solution of the E_l and ϕ equations of (B.3) yields

$$\alpha^2(Z_l, Z_l^*, \psi_t) = \frac{2}{m} \frac{E_l^* - E_l}{\dot{\xi}_t^2(Z_l, \psi_t)} + 1 \quad (\text{B.4})$$

$$a_1^2(Z_l, Z_l^*, \psi_t) = \sqrt{\frac{\xi_t^2}{\dot{\xi}_t^2(Z_l, \psi_t)}} \left(\frac{\alpha(Z_l, Z_l^*, \psi_t) + 1}{\alpha(Z_l, Z_l^*, \psi_t)} \right) \frac{\phi^*}{t_f(Z_l, \psi_t)} - 1 \quad (\text{B.5})$$

We then substitute both (B.4) and (B.5) into the δ_l equation of (B.3) to arrive at a single equation in a single unknown variable, ψ_t . Namely the equation

$$\delta_l^* = \mathbf{t}_{(1, -\vartheta_l(Z_l, Z_l^*, \psi_t))} \circ \mathbf{t}_{\left(\frac{-1}{\alpha(Z_l, Z_l^*, \psi_t)}, \psi_t\right)}(\delta_t(Z_l, \psi_t)) \quad (\text{B.6})$$

The function of ψ_t on the right hand side of the equation behaves nicely (e.g. it is monotone for most choices of Z_l, Z_l^*) and can be easily solved using numerical methods. After solving for ψ_t from (B.6), we substitute the result into (B.4) and (B.5) to obtain α and a_1 . From here, it is trivial to go back to k_c and k_d , completing the inversion.

Finally, we can express the desired liftoff state, Z_l^* in terms of X_a^* and the control inputs. Substituting the appropriate relationships, (B.6) becomes

$$\mathbf{t}_{(1, \vartheta_l(X_a, X_a^*, \psi_t))}(\delta_l^*(X_a, X_a^*, \psi_t)) = \mathbf{t}_{(\frac{-1}{\alpha(X_a, X_a^*, \psi_t)}, \psi_t)}(\delta_t(X_a, \psi_t)) \quad (\text{B.7})$$

Equation (B.7) can be used to solve for ψ_t numerically (since no closed form expression involving elementary functions is available). This is in turn used to find k_c and k_d using the closed form expressions (B.1), (B.2), (B.5) and (B.4).

APPENDIX C

Preliminary Analysis for the Template Based Control of the Ankle-Knee-Hip Model

This appendix concerns details on the template based control of the AKH model of Figure 4.4 as well as preliminary analysis of both the virtual work based embedding and a more exact embedding strategy based on the inversion of the plant model.

C.1 Problem Statement

C.1.1 Target Dynamics

Our objective is to control the AKH system such that its center of mass mimics the dynamics of a SLIP. Formally, this takes the form of a two dimensional constraint

$$m\ddot{\mathbf{b}} = \mathbf{F} , \tag{C.1}$$

which ensures that the acceleration of the center of mass equals that of a virtual SLIP.

C.1.2 The Posture Principle

Once (C.1) is satisfied, there are two remaining degrees of freedom in the AKH that need to be actively controlled. In order to pin this freedom down, we intro-

duce the idea of a *posture principle*, which designates a two dimensional “preferred” posture manifold in the configuration space. The embedding controllers will then be responsible from ensuring convergence onto this manifold.

We define the posture manifold as the zero level set of the function

$$p : \mathcal{Q} \mapsto \mathbb{R}^2 .$$

We also define the center of mass (COM) map $g : \mathcal{Q} \mapsto \mathbb{R}^2$ of the AKH as

$$\mathbf{b} = g(\mathbf{q}) . \tag{C.2}$$

Note that the posture principle also induces a pseudo-inverse of this COM map, $g_p^\dagger : \mathbb{R}^2 \mapsto \mathcal{Q}$, with the properties

$$\begin{aligned} g \circ g_p^\dagger &= i_d \\ p \circ g_p^\dagger &= 0 . \end{aligned}$$

C.2 The Virtual Work Approach

C.2.1 An Approximate Embedding Strategy

The controller design of Section 4.4.2 achieves an approximate embedding of the SLIP template within AKH using torque commands respecting the virtual work constraint

$$\dot{\mathbf{b}}^T \mathbf{F} = \dot{\mathbf{q}}^T \boldsymbol{\tau} .$$

Torque vectors satisfying this constraint take the form

$$\boldsymbol{\tau} = Dg(\mathbf{q})^T \mathbf{F} + \left(\mathbf{I}_4 - \frac{\dot{\mathbf{q}} \dot{\mathbf{q}}^T}{\|\dot{\mathbf{q}}\|^2} \right) \boldsymbol{\tau}_d \tag{C.3}$$

where $\boldsymbol{\tau}_d$ captures the remaining actuation freedom.

Another critical constraint comes from the underactuation of the toe joint. The control outputs of the controller must respect this constraint to ensure their practical

applicability. Fortunately, the virtual work constraint alone does not require nonzero torque at the toe.

Proposition C.1. *For all configurations $\mathbf{q} \in \mathcal{Q}$, $D_{q_f} g(\mathbf{q})^T g(\mathbf{q}) = 0$.*

Proof. The open kinematic chain of Figure 4.4 has rotational symmetry with respect to its toe joint. As a consequence, the center of mass map takes the special form

$$g(\mathbf{q}) = R(q_f) f(q_a, q_k, q_h) ,$$

where R is the standard planar rotation matrix as defined in (5.13). Differentiation with respect to q_f yields

$$D_{q_f} g(\mathbf{q}) = R'(q_f) f(q_a, q_k, q_h) .$$

We then have

$$\begin{aligned} D_{q_f} g(\mathbf{q})^T g(\mathbf{q}) &= f(q_a, q_k, q_h)^T R'(q_f)^T R(q_f) f(q_a, q_k, q_h) \\ &= f(q_a, q_k, q_h)^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} f(q_a, q_k, q_h) \\ &= 0 \end{aligned}$$

independent of the configuration or structural parameters. □

C.2.2 Stabilization of the Posture

We achieve the stabilization of the posture through the use of the remaining freedom in $\boldsymbol{\tau}_d$. Our construction is specific to the particular family of linear posture principle maps we defined in Section 4.4.3. Specifically, we define

$$\begin{aligned} p(\mathbf{q}) &:= \mathbf{L}\mathbf{q} - \mathbf{l} && \text{(C.4)} \\ \mathbf{L} &:= \begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & \beta & -1 & 0 \end{bmatrix} \\ \mathbf{l} &:= \begin{bmatrix} -\gamma \\ 0 \end{bmatrix} . \end{aligned}$$

We then use a simple proportional derivative torque law to determine the desired torque vector for posture stabilization, yielding

$$\boldsymbol{\tau}_d = \begin{bmatrix} 0 & 0 \\ 0 & -k_a \\ 0 & 0 \\ -k_h & 0 \end{bmatrix} (\mathbf{L}\mathbf{q} - \mathbf{1}) + \begin{bmatrix} 0 & 0 \\ 0 & -k_{ad} \\ 0 & 0 \\ -k_{hd} & 0 \end{bmatrix} \mathbf{L}\dot{\mathbf{q}} .$$

Note that this controller does not in any way guarantee convergence to the posture manifold as a result of both the hybrid nature of AKH locomotion and the projections of (C.3).

C.3 Exact Embedding Through Plant Inversion

In this section, we describe our preliminary efforts to achieve exact embedding of SLIP within AKH. Even though we have not been successful in this regard, some of our results in the process are useful in understanding the associated problems.

C.3.1 Achieving the Target Dynamics

The virtual work solution of the previous sections will usually fail to achieve exact SLIP dynamics on the center of mass. In order to accomplish such an exact embedding, we introduce an inverse dynamics style controller.

The equations of motion for AKH take the usual generic form

$$M(\mathbf{q})\ddot{\mathbf{q}} + V(\mathbf{q}, \dot{\mathbf{q}}) + B(\mathbf{q}) = \boldsymbol{\tau} . \quad (\text{C.5})$$

Differentiation of (C.2) yields the velocity and the acceleration of the COM as a function of the joint states as

$$\begin{aligned} \dot{\mathbf{b}} &= \frac{d}{dt} (g(\mathbf{q})) = Dg(\mathbf{q})\dot{\mathbf{q}} \\ \ddot{\mathbf{b}} &= \frac{d^2}{dt^2} (g(\mathbf{q})) = [(\dot{\mathbf{q}}^T \otimes I_n)D_{\mathbf{q}}(Dg^S(\mathbf{q}))]\dot{\mathbf{q}} + Dg(\mathbf{q})\ddot{\mathbf{q}} . \end{aligned}$$

The acceleration of the center of mass then takes the form

$$\ddot{\mathbf{b}} = [(\dot{\mathbf{q}}^T \otimes I_n)D_{\mathbf{q}}(Dg^S(\mathbf{q}))]\dot{\mathbf{q}} + Dg(\mathbf{q})M^{-1}(\mathbf{q})[-V(\mathbf{q}, \dot{\mathbf{q}}) - B(\mathbf{q}) + \boldsymbol{\tau}] .$$

As in Section C.2.1, the torque solutions need to satisfy the constraints regarding both the SLIP embedding and the underactuation at the toe. In order to uniformly capture both of these, we define

$$G := \begin{bmatrix} Dg(\mathbf{q})M^{-1}(\mathbf{q}) \\ e_1 \end{bmatrix}$$

$$v := \begin{bmatrix} \frac{\mathbf{F}(\mathbf{q})}{m} - [(\dot{\mathbf{q}}^T \otimes I_n)D_{\mathbf{q}}(Dg^S(\mathbf{q}))]\dot{\mathbf{q}} + Dg(\mathbf{q})M^{-1}(\mathbf{q})[V(\mathbf{q}, \dot{\mathbf{q}}) + B(\mathbf{q})] \\ 0 \end{bmatrix} ,$$

where e_1 is the first component of the canonical basis. The constraints then become

$$G\boldsymbol{\tau} = v ,$$

yielding solutions of the form

$$\boldsymbol{\tau} = G^T(GG^T)^{-1}v + Hu$$

$$\text{range}(H) = \ker(G) .$$

C.3.2 Stabilization of the Posture

Stabilization of posture requires that two conditions are satisfied. Firstly, the posture manifold must be invariant under the action of the controlled dynamics. That is, we must have

$$\frac{d}{dt} \begin{bmatrix} p(\mathbf{q}) \\ Dp(\mathbf{q})\dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} Dp(\mathbf{q}) & 0 \\ (\dot{\mathbf{q}}^T \otimes I_2)D_q(Dp(\mathbf{q})^S) & Dp(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = 0 .$$

Secondly, we require that the trajectories outside the manifold asymptotically converge to it. This can be verified by defining a posture error function,

$$\epsilon(\mathbf{q}, \dot{\mathbf{q}}) := \| p(\mathbf{q}) \|^2 + \| Dp(\mathbf{q})\dot{\mathbf{q}} \|^2 .$$

Consequently, a sufficient condition for converge takes the form

$$\frac{d}{dt}\epsilon \circ \mathcal{L}^t(\mathbf{q}, \dot{\mathbf{q}}) = D\epsilon \cdot \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) < 0$$

where we define

$$D\epsilon(\mathbf{q}, \dot{\mathbf{q}}) := \begin{bmatrix} 2p^T(\mathbf{q})Dp(\mathbf{q}) + \dot{\mathbf{q}}^T(\dot{\mathbf{q}}^T \otimes I_n)D_{\mathbf{q}}(Dp^T(\mathbf{q})p(\mathbf{q}))^S & 2\dot{\mathbf{q}}^T Dp^T(\mathbf{q})Dp(\mathbf{q}) \end{bmatrix}$$

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) := \begin{bmatrix} \dot{\mathbf{q}} \\ M^{-1}(\mathbf{q})(-V(\mathbf{q}, \dot{\mathbf{q}}) - B(\mathbf{q}) + \boldsymbol{\tau}) \end{bmatrix}.$$

One possible approach for achieving this stabilization is to obtain error dynamics of the form

$$\dot{\epsilon} = -K\epsilon.$$

For the linear posture principle of (C.4), an inverse dynamics solution to this problem yields the equation

$$2(\mathbf{L}\mathbf{q} - \mathbf{l})^T \mathbf{L}\dot{\mathbf{q}} + 2\dot{\mathbf{q}}^T \mathbf{L}^T \mathbf{L}M^{-1}(-V - B + G^T(GG^T)^{-1}v + Hu) = -K\epsilon.$$

Based on this equation, we have attempted to use the obvious but problematic solution

$$u = \frac{H^T M^{-1} \mathbf{L}^T \mathbf{L} \dot{\mathbf{q}}}{\|H^T M^{-1} \mathbf{L}^T \mathbf{L} \dot{\mathbf{q}}\|^2} (-K\epsilon - 2(\mathbf{L}\mathbf{q} - \mathbf{l})^T \mathbf{L}\dot{\mathbf{q}} + 2\dot{\mathbf{q}}^T \mathbf{L}^T \mathbf{L}M^{-1}(V + B - G^T(GG^T)^{-1}v)),$$

which attempts a direct inversion of (C.6). Unfortunately, this solution encounters a singularity at the zero level set of $\|H^T M^{-1} \mathbf{L}^T \mathbf{L} \dot{\mathbf{q}}\|$ and does not yield stability. In the scope of this thesis, we have not been able to identify alternative solutions that are free of singularities, capable of ensuring stability of the posture manifold.

APPENDIX D

Existence and Uniqueness of the Maximal Contact State for the Flipping Model

First, we will need to state some of the properties of consistent contact state assignments and the partial order defined in Section 3.3.2.

Lemma D.1. *At any given state \mathbf{q} and for any given control input vector $\boldsymbol{\tau}$, there is at least one consistent leg contact state assignment.*

Proof. $p \in \mathcal{H}_l$ such that $\forall i p_i = 0$ is always consistent. □

Theorem D.2. *Let $p, r \in \mathcal{H}_l$. The following statement holds*

$$((p \geq r) \wedge \text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(p)) \implies \text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(r)$$

Proof. If $p = r$, then the statement holds trivially. So, suppose $p \geq r$, $p \neq r$ and that p is consistent. Let $\mathbf{v}_p := \mathbf{A}_p(\mathbf{q})^{-1} \mathbf{b}(\mathbf{q}, \boldsymbol{\tau})$ and $\mathbf{v}_r := \mathbf{A}_r(\mathbf{q})^{-1} \mathbf{b}(\mathbf{q}, \boldsymbol{\tau})$ denote the solutions of (3.7), for both leg touchdown state vectors p and r , respectively. Noting that $\mathbf{A}_r(\mathbf{q}) \mathbf{v}_r = \mathbf{b}(\mathbf{q}, \boldsymbol{\tau}) = \mathbf{A}_p(\mathbf{q}) \mathbf{v}_p$, we have

$$\mathbf{A}_r(\mathbf{q})(\mathbf{v}_p - \mathbf{v}_r) = -(\mathbf{A}_p(\mathbf{q}) - \mathbf{A}_r(\mathbf{q})) \mathbf{v}_p.$$

The matrices $\mathbf{A}_p(\mathbf{q})$ and $\mathbf{A}_r(\mathbf{q})$ have the same elements except the last row, where the components of p and r differ. Let J be the set of indices for those legs j such that

$p_j = 1$ and $r_j = 0$. J is always nonempty because $p \neq r$ and $p \geq r$. Consequently, we can write

$$-(\mathbf{A}_p(\mathbf{q}) - \mathbf{A}_r(\mathbf{q}))\mathbf{v}_p = [0, 0, 0, -\sum_{j \in J} p_j F_i(p) y_i]^T.$$

Rewriting $\mathbf{A}_r(\mathbf{q})$ as a block matrix, we have

$$\mathbf{A}_r(\mathbf{q})^{-1} = \begin{bmatrix} \Delta F_{(3x3)} & B_{(3x1)} \\ X_{(1x3)} & -D_{(1x1)} \end{bmatrix}^{-1} = \begin{bmatrix} \Delta G_{(3x3)} & W_{(3x1)} \\ Y_{(1x3)} & -Z_{(1x1)} \end{bmatrix}$$

where $\Delta G := (\Delta F + BD^{-1}X)^{-1}$, $Z := (D + X\Delta F^{-1}B)^{-1}$, $W := \Delta F^{-1}BZ$, $Y := D^{-1}X\Delta G$. In Lemma 3.2, we have shown that for the normal operating domain of our model, $f_i > 0$. Inspection of (3.8) then shows that the elements of the diagonal matrix ΔF and its inverse as well as B , X are positive. Moreover, $D > 0$, yielding $D^{-1} > 0$ and $Z > 0$ as well. Finally, being defined as products of matrices with positive elements, W and Z also have all positive elements. As a consequence, we obtain the following difference in the solution vectors for p and r

$$\mathbf{v}_p - \mathbf{v}_r = -\mathbf{A}_r(\mathbf{q})^{-1}(\mathbf{A}_p(\mathbf{q}) - \mathbf{A}_r(\mathbf{q}))\mathbf{v}_p = \begin{bmatrix} -W_{(3x1)} \\ Z_{(1x1)} \end{bmatrix} \sum_{j \in J} p_j F_i(p) y_i. \quad (\text{D.1})$$

By definition of consistency, we have $F_i(p) > 0$. Hence, (D.1) yields $\forall i$, $F_i(p) - F_i(r) < 0$. We now have

$$\forall i \ r_i = 1 \implies p_i = 1 \implies F_i(p) > 0 \implies F_i(r) > 0$$

which, by definition, implies the consistency of r . □

Theorem D.3. *Let $p, r, s \in \mathcal{H}_l$. The following statement holds*

$$\begin{aligned} & \text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(p) \wedge \text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(r) \\ & \implies (p \geq r) \vee (r \geq p) \vee (\exists s \in \mathcal{H}_l \text{ cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(s) \wedge (s \geq p) \wedge (s \geq r)) \end{aligned}$$

Theorem D.4. *(Existence and uniqueness of a maximal contact state)*

In any given state \mathbf{q} and for a given control input vector $\boldsymbol{\tau}$, there is always a unique consistent leg contact state assignment which is also maximal.

Proof. Suppose that there are no maximal leg contact state assignments. As a consequence, for all leg contact assignments, the opposite of (3.12) must hold, namely

$$\forall p \in \mathcal{H}_l \neg \text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(p) \vee (\exists r \in \mathcal{H}_l \text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(r) \wedge \neg(p \geq r)) \quad (\text{D.2})$$

Now, consider the following inductive construction. Pick p_0 to be a consistent leg contact state assignment. By Theorem D.1, such an assignment can always be found. For the inductive step, suppose $\text{cons}_{[\mathbf{q}, \boldsymbol{\tau}]}(p_k)$ holds. By (D.2), we can always find $r \in \mathcal{H}_l$, which is consistent such that either $r > p_k$ holds, or neither $p \geq r$ nor $r \geq p$ hold. If $r > p_k$, then choose $p_{k+1} = r$. Otherwise, using Theorem D.3, we can always find $s \in \mathcal{H}_l$ such that s is consistent, and it is greater than both p and r . Note that $(s \neq p) \wedge (s \neq r)$ because any other case would imply that r and p could be ordered. We can now choose $p_{k+1} = s$.

This inductive construction results in an infinite ordered sequence $\{p_k\}$ such that $\forall k \in \mathbb{N}, (p_{k+1} \geq p_k) \wedge (p_{k+1} \neq p_k)$. However, we know that \mathcal{H}_l is finite, therefore, there exists at least one maximal leg contact state assignment for any given state and control input vector. \square

APPENDIX E

Derivation and Analysis of the Jacobians for the Planar and Spatial Hexapod Models

E.1 The Slimpod Model

E.1.1 Derivation of the Jacobians

Recall from Section 5.5.6 that the forcing vector that appears in the dynamics, expressed in the virtual toe frame reads

$$\mathbf{K} := [K_\xi, K_\psi, K_\alpha]^T = (D_{\mathbf{c}}\boldsymbol{\phi}) \mathbf{S}\boldsymbol{\tau} + (D_{\mathbf{c}}\boldsymbol{\rho}) \mathbf{S}\mathbf{F}_r .$$

The jacobian matrices $D_{\mathbf{c}}\boldsymbol{\phi}$ and $D_{\mathbf{c}}\boldsymbol{\rho}$ are defined as

$$D_{\mathbf{c}}\boldsymbol{\phi} := \begin{bmatrix} \frac{\delta\phi_1}{\delta\xi} & \cdots & \frac{\delta\phi_6}{\delta\xi} \\ \frac{\delta\phi_1}{\delta\psi} & \cdots & \frac{\delta\phi_6}{\delta\psi} \\ \frac{\delta\phi_1}{\delta\alpha} & \cdots & \frac{\delta\phi_6}{\delta\alpha} \end{bmatrix}, \quad D_{\mathbf{c}}\boldsymbol{\rho} := \begin{bmatrix} \frac{\delta\rho_1}{\delta\xi} & \cdots & \frac{\delta\rho_6}{\delta\xi} \\ \frac{\delta\rho_1}{\delta\psi} & \cdots & \frac{\delta\rho_6}{\delta\psi} \\ \frac{\delta\rho_1}{\delta\alpha} & \cdots & \frac{\delta\rho_6}{\delta\alpha} \end{bmatrix} .$$

Recalling definitions of leg vectors in the body frame in (5.12) as well as the polar leg states of (5.14), components of the jacobians can then be computed as

$$\frac{\delta\rho_i}{\delta x} = \frac{1}{\rho_i} \mathbf{1}_i^T \frac{\delta\mathbf{l}_i}{\delta x}, \quad \frac{\delta\phi_i}{\delta x} = \frac{1}{\rho_i^2} \mathbf{1}_i^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \frac{\delta\mathbf{l}_i}{\delta x},$$

where x denotes one of the virtual frame state variables ξ , ψ or α and we have

$$\begin{aligned} \frac{\delta \mathbf{l}_i}{\delta \xi} &= -R^T(\alpha) \begin{bmatrix} \sin \psi \\ \cos \psi \end{bmatrix} & \frac{\delta \mathbf{l}_i}{\delta \psi} &= R^T(\alpha) \xi \begin{bmatrix} -\cos \psi \\ \sin \psi \end{bmatrix} \\ \frac{\delta \mathbf{l}_i}{\delta \alpha} &= D_\alpha R^T(\alpha) \left(\mathbf{f}_i - \xi \begin{bmatrix} \sin \psi \\ \cos \psi \end{bmatrix} \right). \end{aligned}$$

E.1.2 Singularity Analysis

In this section, we investigate the affordance granted by the structure of the jacobian through linearization. The Slimpod locomotion controllers presented in this thesis always result in trajectories that go through the singular configuration where the body orientation is neutral and all the stance legs are vertical. The associated coordinates are given as $\xi = 0.1$, $\psi = 0$ and $\alpha = 0$ as well as the leg coordinates $\rho_i = 0.1$ and $\phi_i = 0$.

In the sequel, we assume that only the legs of the left tripod — i.e. legs 1, 3 and 5 are in contact with the ground and the kinematic parameters of the system are close to those of RHex, the planar projections of those given in Table 6.1. Linearization of the jacobian $D_c \phi$ around the singular configuration yields

$$(D_c \phi)_{I_1} \approx \begin{bmatrix} 0 & 0 & 0 \\ 1 + 2\alpha & 1 & 1 - 2\alpha \\ -1 - 2\alpha + 2\psi & -1 & -1 + 2\alpha - 2\psi \end{bmatrix}.$$

Two conclusions can be drawn from this form of the jacobian. First of all, when the robot is close to the singular configuration, there is no affordance over the radial component of the virtual SLIP. The consequence of this is the inability to exactly tune the radial dynamics of the SLIP as a result of RHex's limited actuation affordance.

The second observation is that, whenever $\psi \approx 0$, the second and third columns become linearly dependent. As a consequence, when the legs are sufficiently close

to the vertical, the SLIP angle and pitch cannot be independently actuated. In this context, our embedding controller designs favor the SLIP angle coordinate over the pitch and always ensure that the robot center of mass feels a central force. It is also worthwhile to note that, as the robot proceeds through its stance phase, ψ departs from zero, yielding individual control over both ψ and the pitch α .

E.2 The Spatial Hexapod Model

E.2.1 Derivation of the Jacobians

The forcing vector for the spatial hexapod was defined in Section 6.4.2 as

$$\mathbf{K} := [K_\xi, K_\psi, K_\eta, K_\gamma, K_\alpha, K_\beta] = (D_{\mathbf{c}\phi}) \mathbf{S}\boldsymbol{\tau} + (D_{\mathbf{c}\theta}) \mathbf{S}\boldsymbol{\tau}_\theta + (D_{\mathbf{c}\rho}) \mathbf{S}\mathbf{F}_r .$$

The jacobian matrices $D_{\mathbf{c}\phi}$, $D_{\mathbf{c}\theta}$ and $D_{\mathbf{c}\rho}$ are defined as

$$D_{\mathbf{c}\phi} := \begin{bmatrix} \frac{\delta\phi_1}{\delta\xi} & \cdots & \frac{\delta\phi_6}{\delta\xi} \\ \frac{\delta\phi_1}{\delta\psi} & \cdots & \frac{\delta\phi_6}{\delta\psi} \\ \frac{\delta\phi_1}{\delta\eta} & \cdots & \frac{\delta\phi_6}{\delta\eta} \\ \frac{\delta\phi_1}{\delta\gamma} & \cdots & \frac{\delta\phi_6}{\delta\gamma} \\ \frac{\delta\phi_1}{\delta\alpha} & \cdots & \frac{\delta\phi_6}{\delta\alpha} \\ \frac{\delta\phi_1}{\delta\beta} & \cdots & \frac{\delta\phi_6}{\delta\beta} \end{bmatrix}, \quad D_{\mathbf{c}\theta} := \begin{bmatrix} \frac{\delta\theta_1}{\delta\xi} & \cdots & \frac{\delta\theta_6}{\delta\xi} \\ \frac{\delta\theta_1}{\delta\psi} & \cdots & \frac{\delta\theta_6}{\delta\psi} \\ \frac{\delta\theta_1}{\delta\eta} & \cdots & \frac{\delta\theta_6}{\delta\eta} \\ \frac{\delta\theta_1}{\delta\gamma} & \cdots & \frac{\delta\theta_6}{\delta\gamma} \\ \frac{\delta\theta_1}{\delta\alpha} & \cdots & \frac{\delta\theta_6}{\delta\alpha} \\ \frac{\delta\theta_1}{\delta\beta} & \cdots & \frac{\delta\theta_6}{\delta\beta} \end{bmatrix}, \quad D_{\mathbf{c}\rho} := \begin{bmatrix} \frac{\delta\rho_1}{\delta\xi} & \cdots & \frac{\delta\rho_6}{\delta\xi} \\ \frac{\delta\rho_1}{\delta\psi} & \cdots & \frac{\delta\rho_6}{\delta\psi} \\ \frac{\delta\rho_1}{\delta\eta} & \cdots & \frac{\delta\rho_6}{\delta\eta} \\ \frac{\delta\rho_1}{\delta\gamma} & \cdots & \frac{\delta\rho_6}{\delta\gamma} \\ \frac{\delta\rho_1}{\delta\alpha} & \cdots & \frac{\delta\rho_6}{\delta\alpha} \\ \frac{\delta\rho_1}{\delta\beta} & \cdots & \frac{\delta\rho_6}{\delta\beta} \end{bmatrix} .$$

Once again, recalling the cartesian leg vectors of (6.2) and the polar leg states of (6.3) we have,

$$\frac{\delta\rho_i}{\delta x} = \frac{1}{\rho_i} \mathbf{l}_i^T \frac{\delta\mathbf{l}_i}{\delta x} \quad \frac{\delta\theta_i}{\delta x} = \frac{1}{\rho_i^2 \sqrt{\rho_i^2 - \mathbf{l}_{ix}^2}} \begin{bmatrix} \rho_i^2 - \mathbf{l}_{ix}^2 \\ -\mathbf{l}_{ix}\mathbf{l}_{iy} \\ -\mathbf{l}_{ix}\mathbf{l}_{iz} \end{bmatrix}^T \frac{\delta\mathbf{l}_i}{\delta x}$$

$$\frac{\delta\phi_i}{\delta x} = \frac{1}{\rho_i^2 - \mathbf{l}_{ix}^2} \mathbf{l}_i^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \frac{\delta\mathbf{l}_i}{\delta x}$$

where x denotes one of the virtual state variables $\xi, \psi, \eta, \gamma, \alpha$ or β ; and we have

$$\begin{aligned}
\frac{\delta \mathbf{l}_i}{\delta \xi} &= \mathbf{R}^T \begin{bmatrix} -\cos \psi \cos \eta \\ \sin \psi \\ -\cos \psi \sin \eta \end{bmatrix} & \frac{\delta \mathbf{l}_i}{\delta \psi} &= \mathbf{R}^T \xi \begin{bmatrix} \sin \psi \cos \eta \\ \cos \psi \\ \sin \psi \sin \eta \end{bmatrix} \\
\frac{\delta \mathbf{l}_i}{\delta \psi} &= \mathbf{R}^T \xi \begin{bmatrix} \sin \psi \cos \eta \\ \cos \psi \\ \sin \psi \sin \eta \end{bmatrix} & \frac{\delta \mathbf{l}_i}{\delta \gamma} &= \frac{\delta \mathbf{R}^T}{\delta \gamma} (\mathbf{f}_i - \xi \begin{bmatrix} \cos \psi \cos \eta \\ -\sin \psi \\ \cos \psi \sin \eta \end{bmatrix}) \\
\frac{\delta \mathbf{l}_i}{\delta \alpha} &= \frac{\delta \mathbf{R}^T}{\delta \alpha} (\mathbf{f}_i - \xi \begin{bmatrix} \cos \psi \cos \eta \\ -\sin \psi \\ \cos \psi \sin \eta \end{bmatrix}) & \frac{\delta \mathbf{l}_i}{\delta \beta} &= \frac{\delta \mathbf{R}^T}{\delta \beta} (\mathbf{f}_i - \xi \begin{bmatrix} \cos \psi \cos \eta \\ -\sin \psi \\ \cos \psi \sin \eta \end{bmatrix})
\end{aligned}$$

E.2.2 Singularity Analysis

In this section, we investigate the affordance granted by the structure of the jacobian through linearization. The hexapedal locomotion controllers presented in this thesis always result in trajectories that go through the singular configuration of the hexapod where the body orientation is neutral and all the stance legs are vertical. The associated coordinates are given as $\xi = 0.1$, $\psi = 0$, $\eta = \pi/2$, $\gamma = \alpha = \beta = 0$ as well as the leg coordinates $\rho_i = 0.1$, $\phi_i = 0$ and $\theta_i = \theta_0$.

In the sequel, we assume that only the legs of the left tripod — i.e. legs 1, 3 and 5 of Figure 6.4 — are in contact with the ground and the kinematic parameters of the system are close to those of RHex, as given in Table 6.1. Linearization of the jacobian

$D_{\mathbf{c}}\phi$ around the singular configuration yields

$$(D_{\mathbf{c}}\phi)_{\mathcal{I}_1} \approx \begin{bmatrix} -10\gamma & 15\gamma & -10\gamma \\ 1 + 2\alpha - \beta & 1 + 1.5\beta & 1 - 2\alpha - \beta \\ -\gamma & -\gamma & -\gamma \\ 3.57 + 2\gamma + 2\alpha - \eta - \beta - 10\xi & -1.43 - \eta - 2.25\beta + 15\xi & 3.57 - 2\gamma - 2\alpha - \eta - \beta - 10\xi \\ -1 + 2\gamma - 2\alpha + \beta + 2\psi & -1 - 1.5\beta & -1 - 2\gamma + 2\alpha + \beta - 2\psi \\ -\gamma + \alpha - \psi & -2.25\gamma - 1.5\alpha + 1.5\psi & -\gamma + \alpha - \psi \end{bmatrix}$$

where the numerical values were truncated to two digits after the decimal point for space considerations.

Two things are immediately visible from the linearized jacobian: For small angles with respect to the virtual foot frame, there is no substantial affordance over neither ξ not η degrees of freedom. Furthermore, as a result of the stiff springs along the sideways extension of the legs, the yaw angle does not deviate significantly from zero. The same argument holds for the η coordinate. Consequently, assuming $\gamma \approx 0$ and $\eta \approx 0$, the linearized jacobian becomes

$$(D_{\mathbf{c}}\phi)_{\mathcal{I}_1} \approx \begin{bmatrix} 0 & 0 & 0 \\ 1 + 2\alpha - \beta & 1 + 1.5\beta & 1 - 2\alpha - \beta \\ 0 & 0 & 0 \\ 3.57 + 2\alpha - \beta - 10\xi & -1.43 - 2.25\beta + 15\xi & 3.57 - 2\alpha - \beta - 10\xi \\ -1 - 2\alpha + \beta + 2\psi & -1 - 1.5\beta & -1 + 2\alpha + \beta - 2\psi \\ \alpha - \psi & -1.5\alpha + 1.5\psi & \alpha - \psi \end{bmatrix}.$$

The second important observation is that, when $\psi \approx 0$, the saggital SLIP angle and pitch degrees of freedom are strongly coupled as second and fifth rows of the jacobian become linearly independent. As a consequence, near the singularity, both of these coordinates cannot be independently actuated. However, as ψ deviates from zero as the virtual leg progresses through its stance phase, these coordinates become decoupled, admitting their independent actuation.

Finally, this form of the jacobian reveals that near the singularity, there is no affordance over the roll but the yaw degree of freedom can always be actuated independently of all the others.

APPENDIX F

Projections onto the Limited Torque Space

The design of the embedding controllers for both the planar and the spatial hexapods are constrained by limitations of actuation, as well as torque constraints imposed by ground friction considerations. In all cases, individual leg torques are required to lie within closed real intervals and the constraints are independent across legs. Consequently, when three legs are in contact with the ground, the allowable torque space takes the form of a rectangular volume in \mathbb{R}^3 , whereas for only two contact legs, it becomes a rectangular area in \mathbb{R}^2 .

In both of these configurations, the desired torque that will yield the ideal embedding of the template is point in the unconstrained torque space. However, this point usually violates aforementioned constraints and necessitates a method for gradual degradation of the embedding while satisfying inherent limitations. For this purpose, we allow the torque to lie within an affine torque subspace, what we call the *embedding subspace*, capturing some of the key components of the embedding.

Formally, the problem can be specified as follows. We are given a unconstrained torque space \mathcal{T} , which is either \mathbb{R}^2 or \mathbb{R}^3 depending on the number of legs in contact with the ground. Also given is a vector $\boldsymbol{\tau}_s \in \mathcal{T}$, specifying an affine subspace as

$$\mathcal{T}_s := \{ \boldsymbol{\tau} \mid \langle \boldsymbol{\tau}_s \mid \boldsymbol{\tau} - \boldsymbol{\tau}_s \rangle = 0 \} .$$

Moreover, there are constraints on individual leg torques, leading to a restricted,

allowable torque space

$$\overline{\mathcal{T}} := \{ \boldsymbol{\tau} \mid \tau_{i,min} \leq \tau_i \leq \tau_{i,max} \} .$$

Finally, a desired torque vector $\boldsymbol{\tau}^* \in \mathcal{T}$ is also specified as the ultimate goal to be achieved.

The problem is then to find the torque vector within $\overline{\mathcal{T}}$ that is closest to $\boldsymbol{\tau}^*$, but is also within the subspace \mathcal{T}_s if possible. Formally, we seek the solution

$$\hat{\boldsymbol{\tau}}^* = \underset{\boldsymbol{\tau} \in (\overline{\mathcal{T}} \cap \mathcal{T}_s)}{\operatorname{argmin}} \|\boldsymbol{\tau} - \boldsymbol{\tau}^*\| , \quad (\text{F.1})$$

whenever possible. If, however, $\overline{\mathcal{T}} \cap \mathcal{T}_s = \emptyset$, then (F.1) does not have a solution and we need to relax the constraints and pick the point which is closest to the embedding subspace \mathcal{T}_s , yielding

$$\hat{\boldsymbol{\tau}}^* = \underset{\boldsymbol{\tau} \in \overline{\mathcal{T}}}{\operatorname{argmin}} \frac{\langle \boldsymbol{\tau}_s \mid \boldsymbol{\tau} - \boldsymbol{\tau}_s \rangle}{\|\boldsymbol{\tau}_s\|} . \quad (\text{F.2})$$

In subsequent sections of this appendix, we will present algorithms to compute these solutions, both for two and three leg stance configurations.

F.1 Respecting Torque Limits with Two Legs

When only two legs of the active tripod are in contact with the ground, the projection problem becomes two dimensional and we have $\mathcal{T} = \mathbb{R}^2$. Moreover, the allowable torque space $\overline{\mathcal{T}}$ becomes a rectangular area, and the subspace defining the desired projection, \mathcal{T}_s is simply a perpendicular line passing through $\boldsymbol{\tau}_s$. Figure F.1 illustrates all the relevant possible arrangements for the problem in the two dimensional case.

The first arrangement occurs when the desired torque vector is within the allowable region, that is, $\boldsymbol{\tau}^* \in \overline{\mathcal{T}}$. In this case, exact solution is possible and we have

$$\hat{\boldsymbol{\tau}}^* = \boldsymbol{\tau}^* . \quad (\text{F.3})$$

The second possible arrangement occurs when $\boldsymbol{\tau}^* \notin \overline{\mathcal{T}}$, but the desired projection subspace intersects the allowable region, that is, $\mathcal{T}_s \cap \overline{\mathcal{T}} \neq \emptyset$. In this case, we can

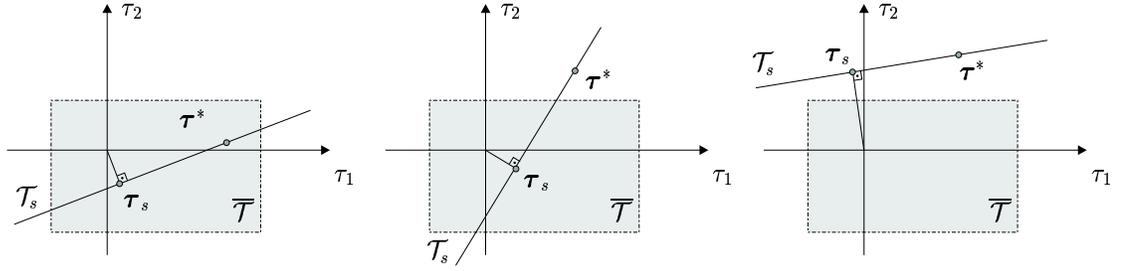


Figure F.1: All of the three different possible arrangements for the allowable torque space, the affine embedding subspace \mathcal{T}_s and the desired exact solution τ^* .

to identify the solution to (F.1) by projecting τ^* onto the boundary of $\overline{\mathcal{T}}$ along the subspace \mathcal{T}_s .

For this purpose, we first need to identify the intersection set $\mathbf{A} := \mathcal{T}_s \cap \delta\overline{\mathcal{T}}$. There are numerous standard geometric algorithms to compute intersections of lines with rectangular regions on the plane [8]. In general, the result will take the form of a line segment, identified by its endpoints τ_a and τ_b . In this case, the solution to the original problem is reduced to

$$\hat{\tau}^* = \operatorname{argmin}_{\tau \in \{\tau_a, \tau_b\}} \|\tau - \tau^*\|, \quad (\text{F.4})$$

which can easily be computed by evaluation for individual elements.

The final arrangement occurs when the projection subspace \mathcal{T}_s does not intersect the allowable torque space at all. In this case, (F.2) must be used and the solution will not exactly satisfy the embedding requirement. Let \mathbf{C} denote the set of all four corners of the allowable torque rectangle. The solution then takes the form,

$$\hat{\tau}^* = \operatorname{argmin}_{\tau \in \mathbf{C}} \frac{\langle \tau_s | \tau - \tau_s \rangle}{\|\tau_s\|}. \quad (\text{F.5})$$

which can, once again, be solved by evaluation for individual elements in \mathbf{C} .

The combination of (F.3), (F.4) and (F.5) yields the general solution for our projection problem in two dimensions.

F.2 Respecting Torque Limits with Three Legs

The structure of the problem when three legs are in contact with the ground is exactly the same as the case with two legs. The only differences are in the dimensions of various spaces involved. In particular, the overall torque space, which becomes $\mathcal{T} = \mathbb{R}^3$. Moreover, the embedding subspace in this case is a plane in three dimensions, once again, passing through, and defined by $\tau_s \in \mathcal{T}$. Finally, the space of allowable torques becomes a rectangular volume.

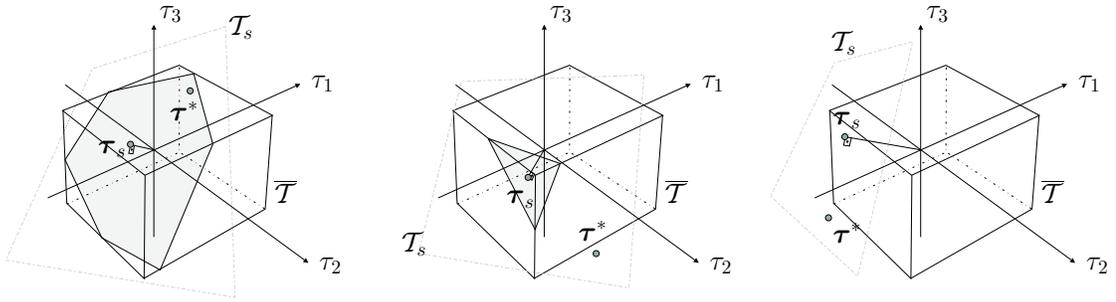


Figure F.2: All of the three different possible arrangements for the allowable torque space, the affine embedding subspace \mathcal{T}_s and the desired exact solution τ^* .

Figure F.2 illustrates three relevant cases for the three dimensional case, which are structurally identical to those presented in the previous section. The solutions for the first and the third cases are identical to those of the two dimensional problem. In particular, whenever $\tau^* \in \overline{\mathcal{T}}$, the solution is

$$\hat{\tau}^* = \tau^* , \quad (\text{F.6})$$

and whenever $\mathcal{T}_s \cap \overline{\mathcal{T}} = \emptyset$, the solution takes the form

$$\hat{\tau}^* = \operatorname{argmin}_{\tau \in \mathbf{C}} \|\tau - \tau^*\| , \quad (\text{F.7})$$

where \mathbf{C} is defined as the set of corner points of the rectangular allowable torque volume.

Finally, the second arrangement for the three dimensional problem requires identifying the intersection of the embedding plane with the rectangular allowable torque volume. In general, such intersections yield a polygon of up to six vertices, and can easily be determined using standard algorithms for computing intersections of line segments with a plane. Once the polygon is determined, the solution is identified to be the closest point on the polygon to the desired torque point $\boldsymbol{\tau}^*$.

Once again, the combination of the individual solutions for all three cases of Figure F.2 yields the global solution to the problem.

APPENDIX G

RHexLib: A Real-Time Control Software Library

G.1 Motivation and Goals

The first generation of RHex's control software was written to run on the DOS operating system and was severely limited on its speed as well as memory storage capability. Moreover, the design of the software was unsuitable for collaborative development on the robot as a result of its monolithic architecture.

In the months following the initial prototype version of RHex, I have designed an improved version of the software, which was capable to overcome the limitations of the underlying operating system. However, this iteration still lacked the modularity necessitated by the increasing collaborative development efforts, requiring the integration of different algorithms and controllers within the same software architecture. RHexLib was born as a result of these architectural demands. The original design and implementation was done by myself and Eric Klavins in the beginning of year 2001.

RHexLib is a library of software components and modules for specific tasks, together with a simple scheduler. Its design mainly emphasizes modularity to ease development and integration of independent, different algorithms and controllers. On the other hand, it also seeks to provide real-time performance on necessary compo-

nents such as lower level control loops and hardware access. This appendix provides an overview of the general structure of the library as well as some of the design details on the core components of the library.

G.2 The Structure of the Library

At the core of RHexLib is the concept of a Module, which encapsulates tasks that have to be performed periodically, such as the reading of encoders and motor control computations. Most of the RHexLib components are implemented as modules, each providing specialized extensions to a standard module interface.

Complementary to the module concept in RHexLib is a simple scheduler, the Module Manager. This scheduler manages the execution, access control and registry of all the modules in the system through a standard interface that they are all required to provide. As a consequence, all of the important tasks encoded within RHexLib have a uniform structure, yielding a convenient framework for their integration and individual development. Section G.3 presents the design details of modules and the Module Manager.

A third and final core component of the library is the abstract hardware interface. This component provides a uniform interface through which all of the low level hardware access is performed. The existence of several RHex robots with usually small, but sometimes significant hardware differences necessitates an abstraction layer to enable development of algorithms and controllers independent of the particular hardware platform instantiation. The design of the hardware layer is described in Section G.4.

G.3 The Module Manager

G.3.1 The Module Class

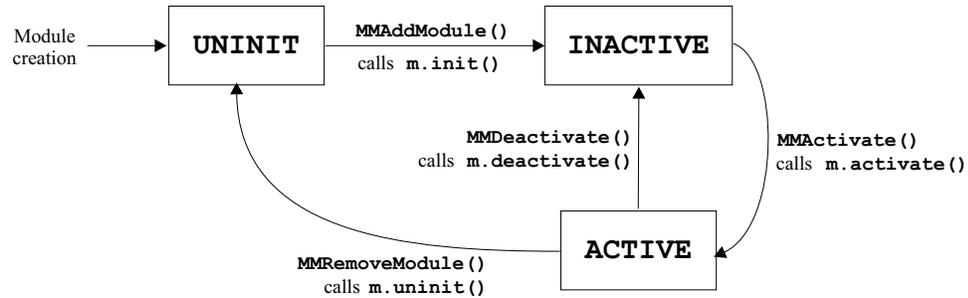


Figure G.1: The life cycle of a Module *m* in RHexLib. The transitions are triggered through the module manager’s utility functions.

Modules in RHexLib are constructs that encapsulate tasks that are performed periodically, and that require supervisory access and execution control. Each module is uniquely identified with a string *name* and a numeric *index*. Furthermore, at any time during execution, they can be in one of three basic states: *uninit*, *inactive* and *active*. Figure G.1 illustrates the associated state transition diagram.

The module’s functionality is determined by five virtual methods, provided by the instantiations of the abstract base class. These methods are normally called by the module manager, which supervises the access control and execution of all modules. The module manager also maintains a global list of modules in the system, ensuring uniqueness within its name space.

The following list summarizes the module methods and their functionality.

- `virtual void init(void)`

This method is called when the module is first added to the module manager’s list. It usually performs internal initialization and locates or adds other modules that are needed.

- `virtual void uninit(void)`

This method complements the `init()` method and is called when the module is removed from the system.

- `virtual void activate(void)`

This method is called when a request for activation of the corresponding module comes to the module manager. Following activation, the module manager starts calling the module's `update()` method periodically. As a consequence, this method is responsible from preparing everything that is necessary for the functionality of the `update()` method. This usually involves setup and activation of other modules as well as internal initialization.

- `virtual void deactivate(void)`

This method complements `activate()` and is responsible from releasing all the resources held by the module. Following deactivation, the module manager ceases to call the `update()` method.

- `virtual void update(void)`

This method is where the periodic task that characterizes a module's functionality is performed. When a module is in its active state, the module manager periodically calls its `update()` method. There are no restrictions on its contents with the exception that its execution time must be sufficiently lower than its scheduling period to allow proper operation of other system components. RHexLib's scheduler is non-preemptive and hence relies on proper design of the update methods of all modules.

In addition to these virtual methods, the `Module` class also provides several methods to access a variety of common tasks. Detailed descriptions of associated functionality can be found in RHexLib's user manual [63].

G.3.2 Scheduling of Module Updates

RHexLib adopts a non-preemptive, static scheduling scheme to linearize the execution of the `update()` methods of all the active modules in the module manager's list. By non-preemptive, we mean that the execution of the `update()` methods are not interrupted once called. By static, we mean that the execution order of all the modules is fixed at initialization and does not change dynamically.

Three parameters are specified for each module at the time they are initialized and determine their static schedule: *period*, *offset* and *order*. The module manager has a fixed scheduling period which divides the execution into time slots of equal duration. These slots form the basic time unit for scheduling purposes and are called *cycles*.

The *period* parameter specifies how many cycles there are between two successive updates of a module. So, a module with *period* = 1 is updated every cycle, *period* = 2 is updated every other cycle and so on.

The *offset* parameter specifies the offset of the very first update of a module from the beginning of execution after startup. This parameter can be used to interleave the updates of modules whose periodicity is larger than one, to ensure uniform CPU utilization.

Finally, the *order* parameter determines the calling order in which the `update()` methods of all the modules that are scheduled to execute within the same cycle. This is very important to ensure the proper ordering of critical tasks such as the reading of encoders, motor control computations and the update of motor commands.

The scheduling mechanism is accessed through the module manager function `MMMainLoop()`. Once called, this function never returns.

G.3.3 Managing Modules

RHexLib has a multitude of functions that are used to manage the modules in the system and their behavior. The following list briefly outlines some of the most important ones. Once again, the RHexLib manual is the best reference for the details [63].

- `void MMAddModule(Module *m, int period, int offset, int order)`

This function adds the specified module to the system's global list. It also calls the `init()` method of the module and puts it in the inactive state. The name and index are specified by the Module object and must be unique. An error is generated if a module with the same name and index combination had already been added to the global list. All modules must be added through this function to enable the rest of their functionality.

- `void MMRemoveModule(Module *m)`

This function removes the module from the global list. It also calls the associated `uninit()` method and puts the module in its uninitialized state.

- `Module *MMFindModule(char *name, int index)`

This function locates a module in the global list by its name and index.

- `void MMActivateModule(Module *m)`

This function puts the module in its active state, but does not perform any access control. It calls the associated `activate()` function and puts it in the list of modules to be periodically updated.

- `void MMDeactivateModule(Module *m)`

This function puts the module in its inactive state calls its `deactivate()` method and removes it from the list of modules to be updated. It also does not perform any access control.

- `void MMGrabModule(Module *m, Module *asker)`

This function is the preferred way to activate a module and should be used instead of `MMActivateModule()` whenever possible. It performs two kinds of access and execution control.

Firstly, for modules that utilize critical resources and can only be used only by a single entity, this function ensures that the module can be grabbed only once and generates an error for further grab attempts until the module is released.

Secondly, for modules that can be used by multiple entities, it maintains a grab counter. When the module is grabbed for the first time, it gets activated. When it is released as many times as it was grabbed (through the `MMReleaseModule()` function, it gets deactivated.

The combination of these two features creates a sufficiently strict access control mechanism which ensures proper usage of critical resources and avoids unnecessary activation of modules.

- `void MMReleaseModule(Module *m, Module *asker)`

This function complements `MMGrabModule()` and releases the module and the associated resources. If the grab count reaches zero, then the module is deactivated.

G.4 The Hardware Class Interface

One of the most important features of RHexLib is its transparency with respect to minor changes in the underlying hardware components. Even though for significant structural differences, it does not provide the same level of convenience, for the most part, it succeeds in transparently dealing with the inevitable albeit minor divergence of hardware components in multiple copies of the same robotic platform. Within the

RHex project, which brings together multiple institutions with substantially different research goals and methodologies, the ability to cope with such situations has been very critical. This was made possible within RHexLib through its hardware abstraction layer, which I will briefly describe in this section.

G.4.1 The Hardware Base Class

The hardware interface is defined through the Hardware abstract base class, whose instantiations provide access to lower level hardware functionality. This class itself does not contain any access methods. Instead, it is a placeholder for pointers to component objects, which are themselves instantiations of component classes.

All RHexLib modules that access any of the hardware components first get a hold of the associated component instantiation through the pointers in the Hardware class. Once they get a hold of this access handle, the individual component interfaces, which are also standard across all the different hardware instantiations, provide the necessary means of performing the desired task.

G.4.2 Hardware Components

In this section, I will briefly describe the current set of hardware components supported by RHexLib. These components also take the form of abstract base classes whose instantiations must implement an associated set of member functions. The following list outlines the components and their functionality.

- `class EncoderHW`

This component provides access to motor shaft encoder counters in the system. It performs all necessary unit conversions and provides direct access to motor shaft absolute angle after calibration as well as their angular velocities.

- `class AnalogHW`

This component is an access point for the analog inputs and outputs in the underlying hardware. The details of the hardware implementation including the quantization, unit conversion and analog acquisition are automatically performed and are hidden behind this interface.

- `class DigitalHW`

This component presents the digital inputs and outputs of the underlying hardware.

- `class TimerHW`

This component interfaces to Intel 82C54 type timer counters in the system. These timers have a multitude of functional capabilities including generating periodic interrupts, measuring pulse width of external signals etc. Please consult the associated product datasheet for details of operation.

- `class AccelHW`

This component supports linear solid state accelerometers for all three axes of the robot body. The unit conversions and the low level acquisition is done below the hardware layer.

- `class GyroHW`

Similar to the accelerometer interface, the gyro component interfaces with rate gyros for all three rotational axes.

- `class DCMotorHW`

This hardware component is a rather important one and is responsible from interfacing with DC motors in the system. It supports critical functions such as sending voltage or current commands to the motors as well as measuring or estimating various quantities such as motor voltage, current, back EMF voltage

and motor temperature. No low level control tasks are performed below the hardware layer.

G.5 Conclusion

RHexLib is a real-time robot control library which started as a specific purpose application to control the RHex robot. However, over the course of its development, it evolved to a somewhat more general purpose library, which is simple to use and greatly facilitates collaborative development on robotic control software.

The concepts that RHexLib introduces are not new. Rather than any new software methods for robotic control, the main contribution of this library is to facilitate collaborative research and development on programming behaviors within a physically situated robotic platform.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] M. Ahmadi and M. Buehler. The ARL Monopod II Running Robot: Control and Energetics. In *IEEE Int. Conf. Robotics and Automation*, pages 1689–1694, Detroit, MI, May 1999.
- [2] R. McN. Alexander. Three uses for springs in legged locomotion. *International Journal of Robotics Research*, 9(2):53–61, 1990.
- [3] R. McN. Alexander and A. S. Jayes. Vertical movement in walking and running. *Journal of Zoology, London*, 185:27–40, 1978.
- [4] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Brown Jr., D. McMordie, U. Saranli, R. J. Full, and D. E. Koditschek. RHex: A Biologically Inspired Hexapod Runner. *Autonomous Robots*, 11:207–213, 2001.
- [5] R. J. Bachman, G. M. Nelson, W. C. Flannigan, R. D. Quinn, J. T. Watson, A. K. Tryba, and R. E. Ritzmann. Construction of a Cockroach-like hexapod robot. In *Eleventh VPI & SU Symposium on Structural Dynamics and Control*, pages 647–654, Blacksburg, VA, May 1997.
- [6] J. E. Bares and D. S. Wettergreen. Dante II: Technical Description, Results and Lessons Learned. *International Journal of Robotics Research*, 18(7):1–29, July 1999.
- [7] A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):415–19, March 1998.
- [8] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2000.
- [9] N. Bernstein. *Coordination and Regulation of Movements*. New York: Pergamon, 1935.
- [10] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22:1217–1227, 1989.
- [11] R. Blickhan and R. J. Full. Similarity in multilegged locomotion: Bouncing like a monopode. *Journal of Comparative Physiology, A*. 173:509–517, 1993.

- [12] R. A. Brooks. A Robot That Walks; Emergent Behaviors from a Carefully Evolved Network. Technical Report Memo 1091, MIT AI Lab, February 1989.
- [13] I. E. Brown and G. E. Loeb. A reductionist approach to creating and using neuromusculoskeletal models. In J. M. Winters and P. E. Crago, editors, *Biomechanics and Neural Control of Posture and Movement*. Springer-Verlag, New York, 2000.
- [14] M. Buehler, R. Battaglia, A. Cocosco, G. Hawker, J. Sarkis, and K. Yamazaki. SCOUT: A simple quadruped that walks, climbs and runs. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 1707–1712, Leuven, Belgium, May 1998.
- [15] M. Buehler, A. Cocosco, K. Yamazaki, and R. Battaglia. Stable open loop walking in quadruped robots with stick legs. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 2348–2353, Detroit, Michigan, May 1999.
- [16] M. Buehler, D. E. Kodischek, and P. J. Kindlmann. Planning and Control of Robotic Juggling and Catching Tasks. *International Journal of Robotics Research*, 13(2):101–118, April 1994.
- [17] G. A. Cavagna, N. C. Heglund, and C. R. Taylor. Mechanical work in terrestrial locomotion: Two basic mechanisms for minimizing energy expenditure. *American Journal of Physiology*, 233(5):R243–R261, 1977.
- [18] G. A. Cavagna, F. P. Saibene, and R. Margaria. Mechanical work in running. *Journal of Applied Physiology*, 19(2):249–256, 1963.
- [19] A. Chatterjee. *Rigid Body Collisions: Some General Considerations, New Collision Laws, and Some Experimental Data*. PhD thesis, Cornell University, 1997.
- [20] A. Chatterjee and A. Ruina. A new algebraic rigid body collision law based on impulse space considerations. *Journal of Applied Mechanics*, 65(4):894–900, December 1998.
- [21] J. J. Craig. *Introduction to Robotics Mechanics and Control*. Addison-Wesley, Reading, Massachusetts, 1986.
- [22] K. S. Espenschied, R. D. Quinn, H. J. Chiel, and R. D. Beer. Biologically-Inspired Hexapod Robot Control. In *Proceedings of the 5th International Symposium on Robotics and Manufacturing: Research, Education and Applications (ISRAM' 94)*, Maui, Hawaii, August 1994.
- [23] C. T. Farley, R. Blickhan, J. Saito, and C. R. Taylor. Hopping frequency in humans: A test of how springs set stride frequency in bouncing gaits. *Journal of Applied Physiology*, 71(6):2127–2132, 1991.
- [24] C. T. Farley, J. Glasheen, and T. A. McMahon. Running springs: Speed and animal size. *Journal of Biomechanics*, 185:71–86, 1993.

- [25] R. J. Full. Personal communication.
- [26] R. J. Full. Integration of individual leg dynamics with whole body movement in arthropod locomotion. In R. D. Beer, R. E. Ritzmann, and T. McKenna, editors, *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, pages 3–20. Academic Press, San Diego, CA, 1993.
- [27] R. J. Full, K. Autumn, J. I. Chung, and A. Ahn. Rapid negotiation of rough terrain by the death-head cockroach. *American Zoologist*, 38:81A, 1998.
- [28] R. J. Full and D. E. Koditschek. Templates and Anchors: Neuromechanical Hypotheses of Legged Locomotion on Land. *Journal of Experimental Biology*, 202:3325–3332, 1999.
- [29] R. J. Full and M. A. R. Koehl. Drag and lift on running insects. *The Journal of Experimental Biology*, 176:89–101, 1992.
- [30] G. Gabrielli and T. H. von Karman. What price speed? *Mechanical Engineering*, 72(10):775–781, 1950.
- [31] E. G. Gilbert and I. Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *I. J. Robust Nonlinear Control*, 9(15):1117–41, 30 December 1999.
- [32] E. G. Gilbert and I. Kolmanovsky. A generalized reference governor for nonlinear systems. In *Proceedings of the IEEE Conference On Decision and Control*, volume 5, pages 4222–7, Orlando, FL, December 2001.
- [33] E. G. Gilbert, I. Kolmanovsky, and K. Tin Tan. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *I. J. Robust Nonlinear Control*, 5(5):487–504, August 1995.
- [34] W. Goldsmith. *Impact: The theory and physical behavior of colliding solids*. Edward Arnold, Ltd., London, 1960.
- [35] H. Goldstein. *Classical Mechnanics*. Addison-Wesley, 1980.
- [36] P. Gregorio, M. Ahmadi, and M. Buehler. Design, control and energetics of an electrically actuated legged robot. *IEEE Trans. Systems, Man, and Cybernetics*, 27B(4):626–634, Aug 1997.
- [37] J. Guckenheimer and S. Johnson. Planar hybrid systems. In *Hybrid Systems II, Lecture Notes in Computer Science*, pages 202–225. Springer-Verlag, 1999.
- [38] E. Hale, N. Schara, J. W. Burdick, and P. Fiorini. A Minimally Actuated Hopping Rover for Exploration of Celestial Bodies. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 420–7, San Francisco, CA, April 2000.

- [39] S. Hirose. A study of design and control of a quadruped walking machine. *International Journal of Robotics Research*, 3(2):113–133, 1984.
- [40] Jessica K. Hodgins and Nancy S. Pollard. Typical human and kangaroo leg characteristics. Personal Communication.
- [41] Jessica Kate Hodgins. *Legged Robots on Rough Terrain: Experiments in Adjusting Step Length*. PhD thesis, Carnegie Mellon University, November 1989. CMU-CS-89-151.
- [42] Interelectric AG, Sachseln, Switzerland. *Maxon Motor Catalog*, 1997/98, www.maxon.com.
- [43] E. Klavins. *Decentralized Phase Regulation of Cyclic Robotic Systems*. PhD thesis, Computer Science and Engineering. The University of Michigan, 2001.
- [44] Daniel E. Koditschek and Martin Bühler. Analysis of a simplified hopping robot. *International Journal of Robotics Research*, 10(6):587–605, December 1991.
- [45] H. Komsuoglu and D. E. Koditschek. Preliminary analysis of a biologically inspired 1-dof ‘clock’ stabilized hopper. In *World Multiconference on Systemics, Cybernetics and Informatics (SCI2000)*, volume IX, pages 670–675, Orlando, FL, 2000.
- [46] H. Komsuoglu, D. McMordie, U. Saranli, N. Moore, M. Buehler, and D. E. Koditschek. Proprioception based behavioral advances in a hexapod robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, volume 4, pages 3650–5, Seoul, South Korea, 2001.
- [47] L. Matthies et al. A Portable, Autonomous, Urban Reconnaissance Robot. In *Proceedings of The 6th International Conference on Intelligent Autonomous Systems*, Venice, Italy, July 2000.
- [48] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.
- [49] R. T. M’Closkey and J. W. Burdick. Periodic motions of a hopping robot with vertical and forward motion. *International Journal of Robotics Research*, 12(3):197–218, 1993.
- [50] T. A. McMahon and G. C. Cheng. The mechanics of running: How does stiffness couple with speed. *Journal of Biomechanics*, 23(Suppl. 1):65–78, 1990.
- [51] D. McMordie and M. Buehler. Towards pronking with a hexapod robot. In *Proceedings of 4th Int. Conf. on Climbing and Walking Robots*, Karlsruhe, Germany, September 2001.
- [52] H. Miura and I. Shimoyama. Dynamic walk of a biped. *International Journal of Robotics Research*, 3(2):60–74, 1984.

- [53] E. Z. Moore. Leg design and stair climbing control for the rhex robotic hexapod. Master's thesis, Dept. of Mechanical Engineering, McGill University, November 2001.
- [54] J. Nakanishi, T. Fukuda, and D. E. Koditschek. A brachiating robot controller. *IEEE Transactions on Robotics and Automation*, 16(2):109–123, 2000.
- [55] G. M. Nelson and R. D. Quinn. Posture Control of a Cockroach-like Robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 157–162, Leuven, Belgium, May 1998.
- [56] NSF Institute for Mathematics and Its Applications Spring 1998 Workshop on Animal Locomotion and Robotics., June 1-5 1998. <http://www.ima.umn.edu/dynsys/spring/dynsys10.html>.
- [57] K. Pearson. The control of walking. *Scientific American*, 235(6):77–86, December 1976.
- [58] J. Pratt, P. Dilworth, and G. Pratt. Virtual model control of a bipedal walking robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 193–198, Albuquerque, NM, April 1997.
- [59] J. Pratt and G. Pratt. Intuitive control of a planar bipedal walking robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 2014–2021, Leuven, Belgium, May 1998.
- [60] M. H. Raibert. *Legged robots that balance*. MIT Press, Cambridge MA, 1986.
- [61] M. H. Raibert and J. K. Hodgins. Animation of dynamic legged locomotion. *Computer Graphics*, 25(4):349–58, July 1991.
- [62] A. A. Rizzi, L. L. Whitcomb, and D. E. Koditschek. Distributed real-time control of a spatial robot juggler. *IEEE Computer Special Issue on Computer Architectures for Intelligent Systems*, 25(5), May 1992.
- [63] U. Saranli. *RHexLib Programmer's Manual*. The University of Michigan, 2001. In preparation.
- [64] U. Saranli, M. Buehler, and D. E. Koditschek. RHex: A Simple and Highly Mobile Robot. *International Journal of Robotics Research*, 20(7):616–631, July 2001.
- [65] U. Saranli, W. J. Schwind, and D. E. Koditschek. Toward the Control of a Multi-Jointed, Monoped Runner. In *Proceedings of the IEEE International Conference On Robotics and Automation*, volume 3, pages 2676–82, Leuven, Belgium, May 1998.
- [66] Uluc Saranli. SimSect Hybrid Dynamical Simulation Environment. Technical Report CSE-TR-436-00, UM, Ann Arbor, MI, 2000.

- [67] J. Schmitt and P. Holmes. Mechanical models for insect locomotion: Dynamics and stability in the horizontal plane - i. theory. *Biological Cybernetics*, 83:501–515, April 2000.
- [68] J. Schmitt and P. Holmes. Mechanical models for insect locomotion: Dynamics and stability in the horizontal plane - ii. application. *Biological Cybernetics*, 83:517–527, April 2000.
- [69] W. J. Schwind. *Spring Loaded Inverted Pendulum Running: A Plant Model*. PhD thesis, Electrical Engineering: Systems. The University of Michigan, 1998.
- [70] W. J. Schwind and D. E. Koditschek. Control of forward velocity for a simplified planar hopping robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, Nagoya, Japan, May 1995.
- [71] W. J. Schwind and D. E. Koditschek. Characterization of monopod equilibrium gaits. In *Proceedings of the IEEE International Conference On Robotics and Automation*, Albuquerque, NM, April 1997.
- [72] W. J. Schwind and D. E. Koditschek. Approximating the stance map of a 2-dof monopod runner. *Journal of Nonlinear Science*, 10:533–568, 2000.
- [73] S. M. Song and K. J. Waldron. *Machines that walk: The Adaptive Suspension Vehicle*. MIT Press, Cambridge MA, 1989.
- [74] I. E. Sutherland and M. K. Ullner. Footprints in the asphalt. *International Journal of Robotics Research*, 3(2):113–133, 1984.
- [75] L. H. Ting, R. Blickhan, and R. J. Full. Dynamics and static stability in hexapedal runners. *The journal of Experimental Biology*, 197:251–269, 1994.
- [76] E. Tunstel. Evolution of Autonomous Self-Righting Behaviors for Articulated Nanorovers. In *Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 341–6, Noordwijk, The Netherlands, June 1999.
- [77] A. F. Vakakis, J. W. Burdick, and T. K. Caughey. An ‘interesting’ strange attractor in the dynamics of a hopping robot. *International Journal of Robotics Research*, 10(6):606–618, December 1991.
- [78] K. J. Waldron and V. J. Vohnout. Configuration design of the Adaptive Suspension Vehicle. *International Journal of Robotics Research*, 3(2):37–48, 1984.
- [79] E. Westervelt and J. Grizzle. Design of asymptotically stable walking for a 5-link planar biped walker via optimization. In *Proceedings of the IEEE International Conference On Robotics and Automation*, Washington D.C., May 2002.
- [80] E. Westervelt, J. Grizzle, and D. Koditschek. Hybrid zero dynamics of planar biped walkers. *Accepted for publication in IEEE Transactions on Automatic Control*, April 2002.

- [81] G. Z. Zeglin and B. Brown. The bow leg hopping robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, volume 1, pages 781–786, Leuven, Belgium, May 1998.
- [82] G. Z. Zeglin and B. Brown. Control of a bow leg hopping robot. In *Proceedings of the IEEE International Conference On Robotics and Automation*, volume 1, pages 793–798, Leuven, Belgium, May 1998.