UNIVERSITY OF CALGARY

Running Synthesis and Control for Monopods and Bipeds with Articulated Legs

by

Qinghong Guo

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

May, 2009

© Qinghong Guo 2009

UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Running Synthesis and Control for Monopods and Bipeds with Articulated Legs" submitted by Qinghong Guo in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY.

Supervisor, Dr. Chris J. B. Macnab
Department of Electrical and Computer Engineering

Co-Supervisor, Dr. Jeff K. Pieper
Department of Mechanical and Manufacturing Engineering

Dr. Om P. Malik
Department of Electrical and Computer Engineering

Dr. Alex Ramirez-Serrano
Department of Mechanical and Manufacturing Engineering

Dr. John Bertram
Department of Cell Biology and Anatomy

External Examiner, Dr. Chris J. Damaren
University of Toronto

May 4, 2009

Date

Abstract

This thesis focuses on offline running synthesis and online running control designs for robots with one and two articulated legs. Since no compliant parts are used in the hardware designs, the robots are hard to stabilize due to large ground reaction forces. Study of extreme configurations may provide acute understandings of legged locomotion systems.

Inspired by new results in biological sciences, biomechanical analysis, and legged robotics, a fundamental assumption is made: the energy cost of the robot in the flight phase is small, when the robot runs on flat even ground. This assumption is formulated as a static optimization problem. Solving this static optimization problem produces the initial joint velocities for the flight phase. The running gaits can then be generated by dynamic optimization. The ground reaction forces are constrained within the permitted range. The stability criterion based on the Zero-Moment Point (ZMP) serves as other nonlinear constraints.

A finite-time controller is employed in the flight phase to improve landing accuracy. In the stance phase, the controller is composed of three modules. The finite-time position-tracking module, designed with the same principle as the flight controller, prepares correct initial state for the subsequent flight phase. The force-suppression module rejects excessive external forces, preventing robot damage. The online ZMP compensator drags the ZMP closer to the center of the support range, with sacrifice of position tracking accuracy, and thus, the running stability can be sustained.

Simulations have demonstrated the effectiveness of the proposed approaches.

Acknowledgements

This dissertation was accomplished with the help from many individuals. I would like to express my sincere appreciation to all of them.

Foremost on the list are my supervisors, Dr. Chris Macnab and Dr. Jeff Pieper. When I had to pick a particular research topic for my thesis, I was lost in the deep ocean of numerous possibilities. Their insightful suggestions helped me choose a topic in legged locomotion which has become one of the most active research areas in robotics. Discussions with them about technical challenges are always enjoyable and inspiring. In particular, their optimistic spirit has been a powerful source that guides me in overcoming difficulties in all aspects.

I would like to thank Dr. Russ Tedrake and Dr. Larry Bates from the very bottom of my heart. Many years ago, I asked Dr. Russ Tedrake for help on passive dynamic walkers. He answered my questions warmheartedly, and provided me with his simulation code. In the two-hour presentation that I gave to his group, he asked me numerous questions. Although his sharp arguments of my work were hard to take, I have to admit that most of his comments were correct. His passion, tirelessness, kindness, and insights teach me a lot. Dr. Larry Bates explained to me the basic principles of the Poincaré return map. Moreover, he provided me with the source of the Lemma 5.2.1 and helped me to prove the Lemma 5.2.2.

Thanks are also given to Dr. John Bertram, Dr. Benjamin Morris, Dr. Wenyuan Liao and Dr. Richard Cushman. Dr. John Bertram confirmed the fundamental assumption (A1) in this dissertation. Communications with Dr. Benjamin Morris helped me to better understand the passive walking robots. Dr. Wenyuan Liao and

Dr. Richard Cushman helped me in understanding the Poincaré return map.

Mr. Yongdong Lü and Mrs. Li Zhang, and Mr. Gang Chang give me rides to the shopping mall every weekend, and they deserve my appreciation. One of my roommates, Mr. Yan Li, cooked many times for me in the past months so that I could focus on my thesis writing. Whenever I worked at home, my roommates stopped their entertainment activities to keep the house quiet. I am in debt to all of them: Mr. Yan Li, Mr. Guanglei Ma, Mr. Shuyu Wang, Miss Dan Xue, Mr. Jing Wang, Mr. Yuan Yuan, Miss Meng Li, and Miss Tianyuan Ju.

Special thanks are due to the beautiful and smart lady, Jin, who has had to raise my son alone, far away from Calgary. I know how difficult it is to take care of a little boy while working hard as a design engineer for the Intel Corporation. Her understanding and support drive me forward steadily. Meanwhile, I thank my son, Jeffrey, for his inspiration at the beginning of this project. He started to run two days after he could stand up! This observation changed my blind belief that walking should proceed running.

Finally, I would like to present my sincere gratitude to my parents for their trust, encouragement, and love. Although they are not well-educated and they cannot understand what I am doing, they always believe that I am doing something significant for the world. I know I am not that good, but I will keep moving forward for their pride and honor.

To my family

For their understanding, inspiration, encouragement, support, and love.

Table of Contents

Approval Page ii

Abstract iii

Acknowledgements iv

Table of Contents vii

List of Tables x

List of Figures xi

List of Acronyms xiv

List of Symbols xv

1 Introduction 1
- 1.1 Scope of the thesis . . . 3
  - 1.1.1 Tasks of the thesis . . . 4
  - 1.1.2 Objective of the thesis . . . 4
- 1.2 Motivations . . . 5
- 1.3 Outline and contributions . . . 8

2 Review of Related Background 11
- 2.1 Gait stability . . . 11
  - 2.1.1 Static stability . . . 12
  - 2.1.2 Dynamic stability . . . 13
- 2.2 One-legged hopping . . . 20
  - 2.2.1 The SLIP models . . . 20
  - 2.2.2 Realistic SLIP hoppers . . . 23
  - 2.2.3 Hopping with an articulated leg . . . 26
- 2.3 Bipedal running . . . 29
  - 2.3.1 Running with two telescopic legs . . . 31
  - 2.3.2 Running with two articulated legs . . . 34
- 2.4 Summary . . . 36

3 Hopping Synthesis for a Rigid Articulated Leg **37**
3.1 Optimal gait synthesis for legged robots 38
3.2 Modeling of the hopper 41
3.2.1 The flight phase 43
3.2.2 Collision between the foot and the ground 51
3.2.3 The stance phase 51
3.3 Hopping optimization 55
3.3.1 Simple rules for choosing the boundary joint angles 55
3.3.2 Single-phase optimization 56
3.3.3 Complete-cycle optimization 58
3.3.4 A five step procedure for optimizing the hopping cycle 59
3.3.5 A performance index 60
3.4 Simulation results and discussions 61
3.4.1 Hopping on even ground 63
3.4.2 Hopping up stairs 67
3.4.3 A complete hopping sequence 74
3.4.4 Some remarks 74
3.5 Summary 78

4 Running Synthesis for a Biped with Rigid Articulated Legs **80**
4.1 Modeling of the biped 80
4.1.1 The flight phase 82
4.1.2 Collision between the foot and the ground 88
4.1.3 The stance phase 88
4.2 Optimization of bipedal running 91
4.2.1 Single-phase optimization 92
4.2.2 Optimization of an entire half running cycle 94
4.2.3 The second half cycle 94
4.2.4 A performance index 95
4.3 Simulation results and discussions 96
4.3.1 Running on even ground 96
4.3.2 Running up stairs 101
4.3.3 Some remarks 105
4.4 Summary 106

5 Hybrid Finite-Time Control Designs for the One-Legged Hopper **107**
5.1 Finite-time stability 111
5.2 The control algorithm 115
5.2.1 Mathematical preliminaries 115
5.2.2 The hopping model 117

5.2.3 The flight phase . . . 118
5.2.4 The stance phase . . . 126
5.3 Simulations . . . 135
5.3.1 Simulation strategies . . . 135
5.3.2 The flight phase . . . 136
5.3.3 The stance phase . . . 137
5.4 Summary . . . 142

6 Hybrid Finite-Time Control Designs for Bipedal Running 148
6.1 The control algorithm . . . 149
6.1.1 The bipedal model . . . 149
6.1.2 The flight phase . . . 150
6.1.3 The stance phase . . . 153
6.2 Remarks on the control algorithm . . . 158
6.3 Simulation results . . . 159
6.3.1 The flight phase . . . 159
6.3.2 The stance phase . . . 160
6.4 Summary . . . 164

7 Conclusions and Future Work Directions 174
7.1 Concluding remarks . . . 174
7.2 Future research directions . . . 176

Bibliography 179

Appendices

A Model details of the hopper 202
A.1 System constants . . . 202
A.2 Model details of the flight phase . . . 202
A.3 Model details of the stance phase . . . 205
A.3.1 The stance model . . . 205
A.3.2 ZMP formulation . . . 208

B Model of the biped 209
B.1 System constants . . . 209
B.2 Model details of the flight phase . . . 210
B.3 Model details of the stance phase . . . 214
B.3.1 The stance model . . . 215
B.3.2 ZMP formulation . . . 220

List of Tables

3.1 Typical costs of running 61
3.2 The constants of the hopper 62
3.3 The parameters for hopping on even ground 63
3.4 The parameters for hopping up stairs 71

4.1 The constants in the simulations 96
4.2 The parameters for running on even ground 97
4.3 The parameters for running up stairs 102

5.1 Parameters of the controller in the flight phase 137
5.2 Parameters of the controller in the stance phase 141

6.1 Parameters of the controller in the flight phase 160
6.2 Parameters of the controller in the stance phase 164

List of Figures

1.1 A possible control subsystem for the legged robots 4

2.1 A definition of Zero-Moment Point 14
2.2 The ZMP stability criterion, (a) a stable gait, (b) an unstable gait . . 14
2.3 A 3-DOF planar SLIP model 22

3.1 The model of the articulated hopper 42
3.2 Initial velocities vs. γ 64
3.3 Boundary rotational kinietic energy vs. γ 65
3.4 Performance vs. γ 66
3.5 Hopping on even ground generated by use of (3.31) 67
3.6 Hopping on even ground generated by use of (3.34) 68
3.7 Comparison of joint angles 68
3.8 Comparison of joint velocities 69
3.9 Comparison of joint torques 69
3.10 The ZMP, acceleration components of the CoM, and the normal GRF resulted from (3.31) 70
3.11 The ZMP, acceleration components of the CoM, and the normal GRF resulted from (3.34) 70
3.12 Hopping up stairs by using (3.31) 72
3.13 Hopping up stairs by using (3.34) 72
3.14 Comparison of joint angles when hopping up stairs 73
3.15 Comparison of joint velocities when hopping up stairs 73
3.16 Comparison of joint torques when hopping up stairs 74
3.17 The complete hopping sequence 75
3.18 The joint angles of the complete sequence 75
3.19 The joint velocities of the complete sequence 76
3.20 The joint torques of the complete sequence 76
3.21 The ZMP, acceleration of the robot's robot, and the normal GRF of the complete hopping sequence 77

4.1 The model of the planar biped 81
4.2 A running cycle on even ground. The solid lines indicate right leg and upper body. The dotted lines indicate left leg. The curve formed by "*" indicates CoM trajectory. 98
4.3 Joint angles for running on even ground 98
4.4 Joint velocities for running on even ground 99
4.5 Control signals for running on even ground 99

4.6 The ZMP trajectory (top), acceleration components of the CoM (middle), and the vertical GRF (bottom) for running on even ground. For the middle graph: horizontal component, solid line, vertical component, dashed line. 100
4.7 A gait cycle for running up stairs. The solid lines indicate right leg and upper body. The dotted lines indicate left leg. The curve formed with "*" indicates CoM trajectory. 102
4.8 Joint angles for running up stairs 103
4.9 Joint velocities for running up stairs 103
4.10 Control signals for running up stairs 104
4.11 The ZMP trajectory, acceleration components of the CoM, and the vertical GRF for running up stairs 104

5.1 The model of the articulated hopper 117
5.2 Joint angles in the flight phase 138
5.3 Joint velocities in the flight phase 138
5.4 Joint errors in the flight phase 139
5.5 Joint torques in the flight phase 139
5.6 Joint torques in the flight phase when a traditional sliding mode is used 140
5.7 Comparison of the hip trajectories 143
5.8 Comparison of the knee trajectories 143
5.9 Comparison of the ankle trajectories 144
5.10 Comparison of the angle errors 144
5.11 Comparison of the velocity errors 145
5.12 Comparison of the joint torques 145
5.13 Comparison of the ZMP trajectories 146
5.14 Comparison of the GRF trajectories 146

6.1 The model of the planar biped 149
6.2 θ_1 and $\dot{\theta}_1$ in the flight phase 160
6.3 Tracking errors $\tilde{\theta}_1$ and $\dot{\tilde{\theta}}_1$ in the flight phase 161
6.4 θ_2 and $\dot{\theta}_2$ in the flight phase 161
6.5 Tracking errors $\tilde{\theta}_2$ and $\dot{\tilde{\theta}}_2$ in the flight phase 162
6.6 Joint torques in the flight phase 162
6.7 θ_1 and $\dot{\theta}_1$ in the stance phase by using Scheme-1 165
6.8 θ_1 and $\dot{\theta}_1$ in the stance phase by using Scheme-2 165
6.9 θ_1 and $\dot{\theta}_1$ in the stance phase by using Scheme-3 166
6.10 Tracking errors of θ_1 and $\dot{\theta}_1$ in the stance phase 166
6.11 θ_2 and $\dot{\theta}_2$ in the stance phase by using Scheme-1 167
6.12 θ_2 and $\dot{\theta}_2$ in the stance phase by using Scheme-2 167

6.13 θ_2 and $\dot{\theta}_2$ in the stance phase by using Scheme-3 168
6.14 Tracking errors of θ_2 and $\dot{\theta}_2$ in the stance phase 168
6.15 θ_3 and $\dot{\theta}_3$ in the stance phase by using Scheme-1 169
6.16 θ_3 and $\dot{\theta}_3$ in the stance phase by using Scheme-2 169
6.17 θ_3 and $\dot{\theta}_3$ in the stance phase by using Scheme-3 170
6.18 Tracking errors of θ_3 and $\dot{\theta}_3$ in the stance phase 170
6.19 GRFs in the stance phase 171
6.20 ZMP trajectories in the stance phase 171
6.21 Torques of the hips in the stance phase 172
6.22 Torques of the knees in the stance phase 172
6.23 Torques of the ankles in the stance phase 173

List of Acronyms

ASLIP	Asymmetric spring-loaded inverted pendulum
CoM	Center of mass
DOF	Degree of freedom
GRF	Ground reaction force
LHS	Left hand side
RHS	Right hand side
SLIP	Spring-loaded inverted pendulum
TPBVP	Two-point boundary value problem
ZMP	Zero-Moment Point

List of Symbols

Symbols

B	Coefficient matrix of torque vector τ
D	Inertia matrix
F	Force
G	Vector of gravity torques
H	Matrix containing the Centrifugal and Coriolis terms
I	Moment of inertia about the center of mass
$I_{\mathrm{n\times n}}$	Identity matrix with dimension of $\mathrm{n} \times \mathrm{n}$
J	Jacobian matrix
K	Gain matrix
L	Length or distance (constant)
M	Mass
N	Number of discretized intervals
P	$(X, Y)^{\mathrm{T}}$, Cartesian position (Chapters 3 and 4)
P	Parameter vector (Chapters 5 and 6)
S	Switching matrix
T	Time (constant)
U	Decision vector
V	$(\dot{X}, \dot{Y})^{\mathrm{T}}$, Cartesian velocity (Chapters 3 and 4)
V	Lyapunov function (Chapters 5 and 6)
W	Work
X	Horizontal position
Y	Vertical position
c	System constant
g	Gravity acceleration
h	Height
k	Positive constant
l	Link length
q	Generalized coordinate vector
r	Position indicator of the CoM
s	Sliding surface
t	Time (variable)
u	Control input
v	Desired error trajectory
x	System state

α Relative angle of the knee
ϵ Cost of transport
η Bound (in most cases)
γ Trade-off factor (Chapters 3 and 4)
γ Power of the finite-time function (Chapters 5 and 6)
θ Joint angle
μ Static friction coefficient
τ Joint torque
ω Angular velocity
Θ Vector containing joint angles
Λ Gain matrix
Φ Energy
Γ Angular momentum
$\mathcal{C}$ Cost functional
$\mathcal{L}$ Lagrangian
$\mathcal{O}$ Objective function
$\mathcal{Y}$ Regressor matrix

Superscripts

T	Transpose of a matrix or a vector
cc	Complete cycle
f	Flight phase
s	Stance phase
+	immediately after the foot/ground collision
-	immediately before the foot/ground collision

Subscripts

F	Force
a	Ankle
c	Convergence (time) or complex (bound)
d	Desired values (in most cases)
dis	Disturbance
f	Foot (Chapters 3 and 4)
f	Viscous friction (Chapters 5 and 6)
f1	Front part of the foot
f2	Rear part of the foot
final	Final values
g	Center of mass
initial	Initial values
k	Kinetic energy
max	Maximum
min	Minimum
p	Potential (energy) (Chapters 3 and 4)
p	Position (controller) (Chapters 5 and 6)
rot	Rotational (energy)
s	Stride (length) (Chapters 3 and 4)
s	Settling (time) (Chapters 5 and 6)
sw	Ankle of the swing leg or the swing leg itself
t	Total (mass *et al.*)
td	Touchdown
to	Take-off
x	Horizontal direction
y	Vertical direction
zmp	Zero-Moment Point
α	Proportional (gain)
β	Integral (gain)

Chapter 1

Introduction

Locomotion usually refers to the ability of a body to move from one place to another, and it takes diverse forms [151]. Typical locomotion forms include swimming with fins, flying with wings, walking and running on feet, side-winding on belly, and so on. It is interesting that flying creatures (*e.g.* birds and flies) and amphibians (such as frogs and salamanders) also have legs, implying that *legged locomotion* may be the most fundamental locomotion form.

The principles of legged locomotion have not been completely understood. Some radical questions remain to be answered in systematic ways. For instance, at a given forward speed, animals immediately choose important gait parameters, such as initial posture, stride length, and stride frequency [29, 72]. For legged robots, this may be very difficult. The need for better understanding of the mechanisms behind legged locomotion has been stimulating and inspiring the development of legged robots [118].

Legged robots have been one of the most active research topics in robotics for decades. Compared with wheeled and tracked vehicles, the legged robots possess better mobility [118]. An ideal legged robot has the potential to maneuver isolated step stones that wheeled and tracked vehicles cannot travel. Besides, legged machines, especially those with one or two legs, usually occupy small terrain areas, and thus, it is possible for legged robots to co-operate with humans in crowded environment. Moreover, legged robots can be more versatile than their wheeled and tracked

counterparts. By making use of other tools, legged robots, or more generally limbed robots, are capable of extending their functions. Riding on a unicycle, a two-legged robot becomes a wheeled vehicle [131].

Due to these remarkable features, legged robots are expected to accomplish military missions in unexplored outdoors, hazardous inspection and repair tasks in nuclear and chemical industries, service work in crowded hospitals, companion and entertainment jobs at home, and much more. A wide spectrum of applications, with a huge potential of commercial profit, also drives forward the development of legged robots.

This dissertation focuses on running synthesis and running control designs for robots with one or two articulated legs in the sagittal plane. Throughout this thesis, *running* is defined as a movement on foot so that all feet leave the ground for a portion of each stride. It relates to alternations of the two consecutive phases: the *flight phase* and the *stance phase.* In the flight (aerial) phase, the body flies in the air with all parts off the ground, and its center of mass (CoM) follows a ballistic trajectory. In the stance (support) phase, one foot is in contact with the ground, and the corresponding leg is the *support leg.* For a two-legged system, the leg other than the support leg is the *swing leg.* The movement of the swing leg that prepares for the next flight phase is the *recovery.* The two running phases switch according to events. When a flight phase finishes, one leg begins to collide with the ground. This event is the *touchdown.* When a stance finishes, the body starts to fly into the air. This event is called the *take-off* (also *lift-off*). Running of a one-legged system is usually termed "*hopping*". A complete *hopping cycle* contains a flight phase, a touchdown, a stance phase, and a take-off. Starting from a left foot take-off, a half bipedal

running cycle consists of a left foot take-off, a flight phase, a left foot touchdown, and a stance phase. The other half bipedal running cycle is skew-symmetric with the first half cycle, with roles of the feet switched. In fact, all bouncing gaits, such as hopping, running, jumping, leaping, etc., can be defined similarly, and thus, they are interchangeable in the latter chapters.

For comparisons, *Walking* is a movement on foot with at least one foot in contact with the ground for all times. Clearly, walking cannot be done by one-legged systems. It is characterized by alternations of the two consecutive phases: the *single support phase* and the *double support phase.* In the single support phase, only one leg supports the body, and the other swings. Naturally, the leg that supports the body is the *support leg*, and the leg that swings about the hip is the *swing leg.* In the double support phase, both feet support the body.

The one-legged robot will often be called the one-legged hopper. Other aliases for the one-legged robot include one-legged machine, hopping machine, hopping robot, and monopod. The two-legged robot will also be called the two-legged machine or biped for short. Actually, a biped is not necessarily a robot. It may be a human, or an animal with two legs.

1.1 Scope of the thesis

The ultimate objective of the undergoing research is to build a versatile biped that can walk and run. To achieve this objective, many tasks need to be accomplished. This thesis aims to address control-related issues for running robots with one or two articulated legs.

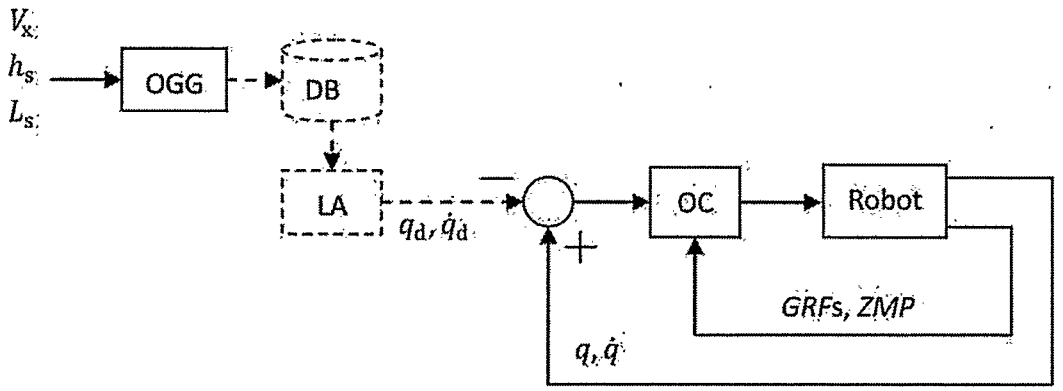

Figure 1.1: A possible control subsystem for the legged robots

1.1.1 Tasks of the thesis

A possible control subsystem is shown in Fig. 1.1. The commanded inputs are the average forward speed V_x, the landing height h_s, and the stride length L_s. "OGG" is the offline gait generator, "DB" is the database containing a large number of gaits corresponding to different ground conditions and different commanded parameters, "LA" is the learning algorithm that generates gaits in real time based on the data generated offline, and "OC" is the online controller. $(q^T, \dot{q}^T)^T$ and $(q_d^T, \dot{q}_d^T)^T$ represent the actual and desired states of the robot, respectively. "GRFs" means the ground reaction forces to the robot, and "ZMP" stands for the Zero-Moment Point. The concepts of GRF and ZMP will be reviewed in Chapter 2.

The solid blocks in Fig. 1.1 are covered in this thesis.

1.1.2 Objective of the thesis

The objectives of this thesis are:

1. to provide solutions to the problem of building practical robots that can run

with one or two articulated legs. Offline gait generators and online control subsystems, respectively for the monopod and the biped, will be discussed..

2. to advance the understanding of legged locomotion via running analysis. The data captured from running motions of humans or animals can be reasonably explained.

3. to obtain insight into the design details, such as power supplies and choice of actuators, by adding running functions to the robots originally developed for walking.

1.2 Motivations

Most existing biped robots are primarily designed to walk. It has been shown that walking and running are the most favorite gaits for bipeds [138], suggesting that running should be paid more attention. To distinguish running from walking, crucial comparisons between them are made as follows:

1. Running has the flight phase with no parts touching the ground. Walking contains a double support phase where both feet stand on the ground simultaneously.

2. Running can achieve a faster maximum speed and a larger range of speed. The maximum speed of bipedal walking is bounded by $\sqrt{gl_0}$ with g being the gravity acceleration and l_0 being the length of the leg [5]. Running speed is constrained by the permitted ground reaction forces (GRFs) and capability of the actuators.

3. In human running, the vertical GRF usually appears to be Bell-shaped with the opening downwards. In contrast, the vertical GRF of walking is typically M-shaped, with two peaks and a valley between [37].

4. In running, the CoM reaches the lowest position at the mid-stance when the hip of the stance leg passes over the ankle, whereas the CoM climbs to the highest position at the mid-stance in walking [19].

These differences imply that the techniques for walking robots may not be effective for running robots, and thus running robots need more investigation.

Some practices in running robots with articulated legs have motivated the work in this thesis. In experiments carried out by Vermeulen *et al.*, a kneed hopper, named OLIE (for "one leg is enough"), failed many times to track the pre-planned joint trajectories [145]. In the experiment reported by Morris *et al.*, the well-known bipedal robot, RABBIT, could only run for six consecutive steps [93]. After the sixth step, the experiment was automatically terminated since the tracking error at one knee exceeded the limit, 0.3 radians. Although the HRP-2LR, a humanoid robot with a height of $1.27m$, has succeeded in running for several years, its forward speed is only $0.16ms^{-1}$ and the duration of the flight phase is only $0.06s$ [96]. Running synthesis and running control are far from "almost solved", as was declaimed in [114].

Compared to the great success of Raibert's hoppers and their descendants [118, 117, 41, 2, 16], the running machines with articulated legs seem to be more difficult to control. For running of rigid bipeds, the joint accelerations and the impulsive impact from the ground are usually large. The magnitudes of the GRFs may reach 5 times the robot weight [133]. To stabilize the robot while preventing it from damage,

special considerations in control design must be taken.

Since the stance phase of running is similar as the single support phase of bipedal walking, the control algorithms that work for the stance phase of running can certainly be applied for the single support phase of walking. Conversely, a control algorithm stabilizing the single support phase of walking may not be able to work for the stance phase of running. Studies of running offer new ideas to investigate bipedal walking.

Another technical challenge is the stabilization of the flight phase. Once the robot leaves the ground, its angular momentum and the CoM trajectory cannot be manipulated. Even for humans, the flight phase is vulnerable to external disturbances. Moreover, since a convincing "flight stability" concept does not exist, the control objective in the flight phase is not completely clear. Solutions to this issue require better understanding of running principles.

Desire for a unified biped model that can execute both walking and running have also motivated this research. A promising model for this purpose was pioneered by Seyfarth and his coworkers. This model contains a point-mass torso and two telescopic springy legs [37, 60]. It successfully predicted both walking and running gaits. However, the resulting walking and running gaits are separated by a large speed gap [37], implying that slow running is impossible. Legs with variable stiffness offer a possible solution [56, 45]. Roughly speaking, legs with high stiffness are suitable for walking and with low stiffness for running. Similarly, legs with variable damping ratio may also work [63], assuming spring-damper pairs are employed in the biped model (e.g., see [144, 91]). Intuitively, change of the stiffness or damping ratio in real time is not costless [92]. It is unclear whether the energy-efficiency of

the robots with such actuators is satisfying.

Examination of a simple rigid biped model is a good starting point, since such a robot can walk and run. Consequently, it is possible to compare running with walking in some crucial aspects, such as energy efficiency and permitted joint torques. Thus, valuable information can be provided for robot designs.

Due to the great potentials for practical use, it is worthy revisiting robots with articulated legs, in the context of running synthesis and running control.

1.3 Outline and contributions

The remainder of this thesis is organized as follows:

Chapter 2 briefly reviews selected background material, including the most-frequently applied gait stability concepts and the corresponding stability margins, and existing one- and two- legged running models and running robots.

Chapter 3 explains the offline generator of hopping gaits. Observations from biological science, biomechanical analysis, and robotics directly inspire the algorithm. For running gaits of many animals, "the metabolic cost of swinging the limbs is negligible compared with the cost of supporting the body weight" [120]. This result is formulated as a static optimization procedure. The initial joint velocities of the flight phase can be automatically searched, provided that the initial joint angles are given. Then the flight phase can be generated by dynamic optimization. With a simple collision model, the states immediately after the foot/ground collision can be predicted. The stance phase is then treated as a typical two point boundary value problem (TPBVP). The entire hopping cycle can finally be optimized as a whole. In

the stance phase, the GRFs and the ZMP criterion serve as nonlinear constraints. Hopping gaits on even ground, and up stairs are simulated. One main contribution of the thesis is the formulation of the fundamental assumption inspired by the biomechanical observations. Another highlight of the algorithm is the universality of the formulations.

Chapter 4 presents the offline generator of bipedal running gaits. This project is a natural extension of the offline hopping synthesis, with minor modifications. In the stance phase, the recovery motion of the swing leg is constrained such that scuffing with ground can be avoided. Role switching of the two legs is treated by a switching matrix.

Chapter 5 explicates a novel control algorithm for the one-legged hopping robot. The finite-time control theory constitutes the basis of the algorithm. The sliding mode starts from the initial states of the hopping phase, and hence the reaching mode is not needed. In the flight phase, a finite-time controller is applied. By appropriately setting the settling time, the tracking errors between the actual states and the desired values converge to zero in finite time in a smooth fashion, producing an elegant flight phase. The finite-time control is robust to system uncertainties and disturbances at the joints. The control subsystem accepts the joint trajectories generated by the algorithm discussed in Chapter 3 as inputs.

To prepare the correct states at the take-off for the next flight phase, the desired final states of current stance phase must be reached. For this purpose, a finite-time controller is also applied in the stance phase.

In practice, the actual GRFs often go beyond the maximum permitted values due to the noisy joint accelerations and external force disturbances, setting the robot

under risk of damage. With the help of a force-suppression module, the actual GRFs can be kept within the maximum permitted values, and the damage due to large GRFs can be avoided.

The noisy joint accelerations and external force disturbances also influence the actual ZMP trajectory. When the ZMP goes out of the support range, the robot may fall. An online ZMP compensator is proposed. As long as the sensory system detects abnormal ZMP values, the ZMP compensator is switched on to modify the desired hip state. As a consequence, the actual system trajectories deviate from the synthesized trajectories which are the original references of the control loop. Temporarily, the robot may not be able to track the commanded forward speed correctly, since the actual initial states of the following flight phase may be away from the desired values.

The designs of the finite-time controllers, the force-suppression module, and the online ZMP compensator are other contributions of this thesis.

Chapter 6 extends the finite-time control algorithm presented in Chapter 5 to stabilize the bipedal running. The inputs of the control subsystem are the joint trajectories generated by the algorithm presented in Chapter 4.

Chapter 7 concludes the thesis. Future research directions are also suggested.

Chapter 2

Review of Related Background

This chapter reviews selected background material, including the most popular definitions of gait stability and the corresponding stability margins, the typical one-legged hopping models, and the influential bipedal models. The review is not intended to be exhaustive, but to present the development flow of running robots with one and two legs.

2.1 Gait stability

It is a common sense that a gait executed by a legged locomotion system should be stable. Otherwise, the gait may not appear elegant. In extreme situations, the system falls. Ironically, to date, all existing definitions for gait stability are controversial. Pratt and Tedrake argued that a reasonable gait stability margin should satisfy the following conditions [116]:

1. Necessary: if this condition is violated, the legged system is unstable.

2. Sufficient: if this condition is satisfied, the legged system is stable.

3. Comparable: the performance of two legged systems can be compared by using this condition.

4. Measurable and Computable: all quantities related to the stability margin can be measured in real time, the stability margin itself can be computed online,

and thus the control algorithm is applicable.

5. Meaningful: The stability margin has understandable physical meanings.

Here, the most-frequently used definitions of the gait stability and stability margins are briefly reviewed.

2.1.1 Static stability

When a legged locomotion system is moving, if the horizontal projection of its CoM is within the support polygon, the gait is said to be statically stable or statically balanced [85, 136, 36, 152]. With point feet, a robot achieving a statically stable gait must have at least four legs such that when one leg leaves the ground, the other feet can still form a support area. Bipeds with flat feet can also perform statically stable walking gaits. According to this definition, running-like gaits, which contain flight phases where all feet are off the ground, cannot be statically stable. The statically stable gaits usually have to be slow. Due to the effects of the inertia and accelerations, the horizontal projection of the CoM of a fast robot may go outside of the support region.

The static stability margin can then be defined as the minimal distance between the horizontal projection of the CoM and the boundary of the support polygon. [36, 152].

The static stability concepts cannot be applied to those extreme cases where the contact points between the robot and the terrains are not in the same plane [152].

2.1.2 Dynamic stability

If a legged locomotion system maintains movement while violating the static stability condition, the gait is said to be dynamically stable or dynamically balanced [119, 145]. To achieve a dynamically stable gait, the system has to adopt compensations for the tipping motion over time [119, 145]. A dynamically stable gait can be very fast, and the projection of the CoM often goes outside of the support polygon. What follows explains the main criteria for dynamically stable gaits.

ZMP

The concept of ZMP was first introduced by Vukobratović and Juričić in 1968, although it was not officially named at that time [148]. It has been interpreted in many ways [49, 140, 7, 50, 39, 147, 125, 111]. For conceptual simplicity, the ZMP may be defined as the point on the ground at which the net moment due to inertial and gravitational forces has no component along the horizontal axes [111]. In Fig. 2.1, r_{a}, r_{g}, and r_{zmp} are the Cartesian locations of the ankle, the robot's CoM, and the ZMP, M_{a} and F_{a} stand for the moment and force acted on the ankle by the links other than the support foot, F_{zmp} is the force applied at the ZMP by the ground, m is the robot's mass, and g is the gravitational acceleration. Clearly, mg is the robot's weight. When the support foot does not rotate about horizontal axes, the following equation must be satisfied:

$$(r_{\mathrm{a}} \times F_{\mathrm{a}} + r_{\mathrm{g}} \times mg + r_{\mathrm{zmp}} \times F_{\mathrm{zmp}} + M_{\mathrm{a}})_{\mathrm{h}} = 0, \tag{2.1}$$

where the subscript "h" means the horizontal components of the overall moment inside the parentheses. Solutions to (2.1) give the Cartesian coordinates of the ZMP.

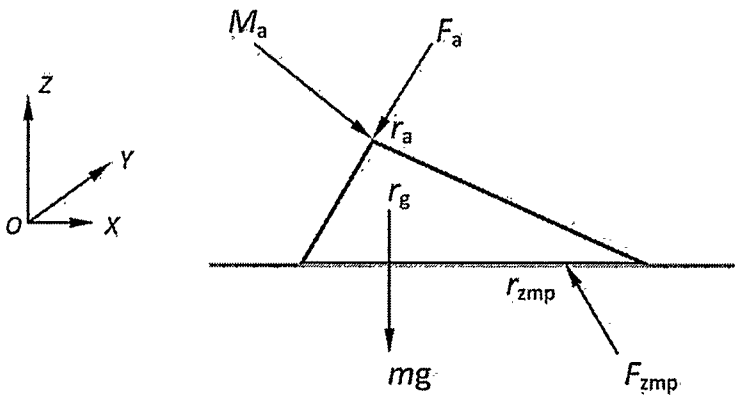

Figure 2.1: A definition of Zero-Moment Point

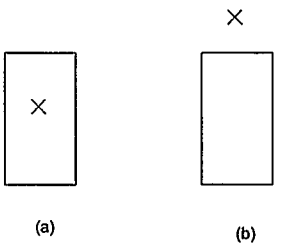

Figure 2.2: The ZMP stability criterion, (a) a stable gait, (b) an unstable gait

The ZMP stability criterion says that, if the ZMP is within the support polygon, the gait is dynamically stable. Moreover, the stability margin is the shortest distance from the ZMP to the boundary of support polygon. Fig. 2.2 illustrates the ZMP stability criterion. Each rectangle represents a support foot. The cross indicates the ZMP location. A stable gait (left graph) and an unstable gait (right graph) are shown.

If the contacts between the robot legs and the terrains are not in the same plane, the ZMP definition, as well as the ZMP stability criterion, cannot be applied. For bipedal locomotion, if the two feet are in contact with two non-planar terrains,

the terrains are projected vertically onto the horizontal surface, forming the virtual ground. On the virtual ground, there exists a virtual ZMP such that the horizontal components of the moment of the total inertia forces are zero. The projections of the two feet onto the virtual ground form the virtual support polygon. Correspondingly, if the virtual ZMP is within the virtual support polygon, the gait is dynamically stable, and the stability margin is the shortest distance from the virtual ZMP to the boundary of the virtual support polygon.

For the flight phase in running-like gaits, since all feet are off the ground, support polygon and virtual support polygon are not defined. Consequently, the ZMP stability criterion does not work.

Flight stability

The flight stability was proposed by Kwon and Park recently [74]. During the flight phase, if the angular momentum about the CoM of the system maintains within a limited range around zero, the flight phase is said to be dynamically stable. During the flight phase the angular momentum about the CoM of the system cannot be manipulated. Thus, to prevent the system from rotating about its CoM in the air, the angular momentum about its CoM should be zero [33]. However, maintaining the angular momentum about the CoM is not sufficient to achieve "stable flight". If the upper body and the leg(s) rotate about the hip simultaneously with large angular velocities in opposite directions, the net angular momentum about the CoM is possibly zero, but the overall energy consumption is high, and the posture in the air is neither natural nor elegant.

Once the variation range of the magnitude of the angular momentum is deter-

mined, the flight stability margin may be defined as the difference between the range limit and the magnitude of the angular momentum. If this quantity is positive, then the flight is stable, and otherwise unstable. A larger value of this quantity indicates higher stability. Since the variation range of the magnitude of the angular momentum about the CoM is difficult to determine, it is not easy to apply the flight stability concept and flight stability criterion to running synthesis and running control at current stage. A more flexible method is to force the magnitude of the angular momentum, with respect to the system's CoM, to be as small as possible by some means, such as optimization techniques [110].

Orbital stability

Qualitatively speaking, a system is orbitally stable if in presence of bounded disturbances, a phase trajectory of an autonomous system shifts to another nearby phase trajectory with similar shape [54, 40]. Mathematically, the phase trajectory of an autonomous system, denoted by Π, is said to be orbitally stable if given an $\epsilon > 0$ there is a $\delta > 0$ such that if P', a representative point on another phase trajectory Π', is within a distance δ of Π at time t_0 then P' remains within a distance ϵ of Π for all time $t \geq 0$ [48, 40]. If no such a δ exists, the phase trajectory Π is orbitally unstable. The orbital stability requires that the two phase trajectories Π and Π' remain close if their initial distance is small. Analogous to the definition of asymptotic stability in the sense of Lyapunov, the phase trajectory Π is said to be asymptotically orbitally stable, if Π is orbitally stable and in addition, the distance between P' and Π converges to zero as time approaches to infinity [40]. The asymptotic orbital stability requires that the two phase trajectories Π and Π' converge finally if their

initial distance is small.

The difference between the orbital stability and the well-known Lyapunov stability lies in that the orbital stability does not describe the characteristics of the equilibrium that the Lyapunov stability deals with.

With the controller applied, the closed-loop legged locomotion system becomes autonomous. The legged locomotion system is hybrid, since the gait cycle contains continuous dynamics in different gait phases and discrete events such as the take-offs and touchdowns. The orbital stability of the gait cycle can be examined by using the Poincaré return map of the gait. A Poincaré section can be chosen as the one dimensional lower subspace of the phase space of the gait cycle at a particular event, such as a touchdown. If the phase trajectory intersects the Poincaré section at the fixed point, denoted by x^*, in one gait cycle, it comes back to the neighborhood of x^* in the next gait cycle, if the gait is orbitally stable. Moreover, the phase trajectory comes back to x^* itself, if the gait is periodic.

In the literature, the gaits are always expected to be periodic, for ease of analysis. For a periodic gait, the phase trajectory is a limit cycle, and $x^* = F(x^*)$, where $F : \Re^n \mapsto \Re^n$ is the discrete Poincaré map. Suppose a phase trajectory starts at $x^* + \Delta x^*$, a little away from the fixed point x^*, due to bounded perturbations. In the next cycle, the phase trajectory intersects the Poincaré section at $F(x^* + \Delta x^*)$, the first return map of $x^* + \Delta x^*$. The map can be expanded by using the Taylor Series as:

$$F(x^* + \Delta x^*) \approx F(x^*) + (\nabla F)\Delta x^* = x^* + (\nabla F)\Delta x^*,$$

where ∇F is the gradient of F with respect to the states x^*. If the periodic gait is orbitally stable, the limit cycle is attractive. This means that the magnitude of the

eigenvalues of ∇F evaluated at x^* is strictly less than one.

For many non-smooth systems, it is impractical to obtain the closed forms of the map F and of the gradient ∇F. Each state can be perturbed once a time by a small number. The first return map corresponding to this perturbation can be obtained. This process repeats until all states are perturbed. All of the perturbations form a diagonal matrix, denoted by v here, and all of the first return maps form a square matrix, denoted by u. The gradient of the map, ∇F, can be calculated with

$$\nabla F = uv^{-1}.$$

The orbital stability margin may be defined as the difference between 1 and the largest magnitude of the eigenvalues of the sensitivity matrix ∇F [116]. According to this definition, a larger orbital stability margin means a faster speed with which the perturbed phase trajectory converges to the unperturbed limit cycle.

The orbital stability concept offers a powerful mathematical analysis tool. By using this tool, stability of the periodic gaits performed by a legged robot can be strictly proved, in spite of the hybrid nature of the system. Also, this tool helps to pick the suitable initial values for the system such that the synthesized gaits are orbitally stable [84, 40, 38].

The Poincaré map has been widely applied in gait analysis for legged locomotion systems. In the latter sections of this chapter, this point will be frequently discussed whenever the running models with one or two legs are investigated by using this technique. However, it may not be convenient to apply this technique to a legged locomotion system with high degrees of freedom (DOFs), since construction of the Poincaré return map for such systems is usually difficult. Further, since this skill

assumes a periodic gait, and hence a limit cycle, it cannot be used to analyze non-periodic gaits that a legged locomotion system performs in most times. Due to this, Pratt and Tedrake argued that the orbital stability is not a good concept in general [116]. Moreover, the Poincaré return map only handles autonomous systems, it implicitly assumes that the form of the controller has been determined, and the control gains can then be chosen via stability analysis [92].

Other definitions of gait stability

Pratt and Tedrake defined the gait stability for a biped as [116]:

"A biped is stable if and only if the state of the robot is not inside the Basin of Fall, where a Fall means that a point on the robot, other than a point in the feet, touches the ground, and the Basin of Fall is a subset of the state space that leads to a fall". To be more specific, "a Time-Limited Basin of Fall is defined as a subset of the state space that leads to a fall within a finite time".

Based on this stability definition, they proposed several stability margins, emphasizing the retrieval[1] capability from a trend of fall (see [116] for more details).

Wieber defined a Viable Kernel that is a union of all Viable states from which a fall can be avoided, and out of which a fall is unavoidable [152]. In essence, the Viable Kernal is a complement of the Basin of Fall [116]. Then, the gait stability margin, the viability margin in Wieber's terminology, can be defined as the distance from the state of the system to the closest non-viable state. Conceptually, these definitions endow with insightful understandings about legged locomotion. Unfortunately, at current stage, it is neither feasible to accurately predict the recovery capability of

[1]The word "recovery" was used in the original paper. In this thesis, "recovery" means the motion of the swing leg from hind to front to prepare the next flight.

the robot that is going to fall, nor is possible to compute the Viable Kernel of the system. Applications of these stability margins have not been found.

More dynamic stability margins, rarely applied to legged locomotion, can be found in a survey paper [36].

2.2 One-legged hopping

The existing one-legged hopping robots can approximately be classified into two categories: hopping with a telescopic springy leg or with an articulated leg. In an articulated leg, the hip-knee or hip-knee-ankle combination behaves like a spring. Therefore, all hoppers can be modeled as a spring-loaded inverted pendulum (SLIP) system. In this section, different SLIP models are presented first, the SLIP hoppers are then reviewed, and finally the articulated hoppers are discussed.

2.2.1 The SLIP models

The basic parts of the SLIP models consist of a massive upper body (torso) and a massless, telescopic, undamped, springy leg. The two links connect together at the hip. Since the leg is assumed massless, no energy is lost when the foot collides with the ground. This simplifies analysis and control designs. For different purposes, the models vary with different configurations. The hip is usually, but not necessarily, located at the CoM of the torso. The spring in the leg may be linear or nonlinear. According to the number of the degrees of freedom (DOFs) in the stance phase, the SLIP models may possess one, two, three, or more than three DOFs. The SLIP models can be utilized to analyze running-like motions [35, 129], or facilitate to

design control laws for running machines [118, 32, 112].

The 1-DOF SLIP model

In the 1-DOF SLIP model, the torso is a point mass, the "hip" is fixed, and the leg length is the only state variable. Such a system can only perform vertical bouncing motions. If the leg is passive, this model can be used to analyze vertical hopping of animals and humans [14, 87]. For vertical hopping synthesis for realistic robots, the leg needs to be actuated [143, 28].

The 2-DOF planar SLIP models

In the 2-DOF SLIP models, the torso is a point mass, and thus the inertia of the torso is ignored. The rotary hip is frictionless. The leg length and the leg angle are usually chosen as the generalized variables.

When the hip and the leg are passive, the 2-DOF SLIP model accurately predicts the GRF patterns, energy fluctuations, and other important relationships in running-like gaits, with different initial states provided [14, 87, 127].

A periodic forward hopping gait can be obtained by applying a radial thrust force along the leg to the torso and choosing an appropriate foot location at the touchdown [86].

The 3-DOF planar SLIP models

By adding the pitch angle of the torso as the third generalized variable, the 2-DOF planar SLIP models become the 3-DOF planar SLIP models. A typical configuration of this model is shown in Fig. 2.3. The three variables are the leg length l, the leg angle ϕ, and the torso pitch φ. In [6], the hip of the model coincides with the CoM

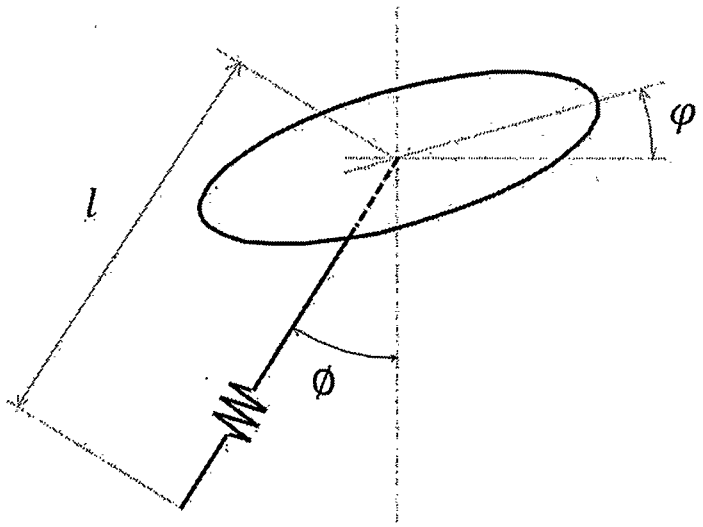

Figure 2.3: A 3-DOF planar SLIP model

of the torso. In [38], the hip is intentionally a little higher than the CoM of the torso when the pitch angle of the torso is zero. For these two configurations, periodic running gaits are found by using the Poincaré return map. Poulakakis and Grizzle proposed a ASLIP (i.e. Asymmetric SLIP) model where the hip joint is located below the torso's CoM [112]. To maintain the torso upright, active controls are required. Note that the hip and the leg may or may not be passive, and a torsional spring may be fitted at the hip.

A 3D SLIP model

The planar 2- and 3- DOF SLIP models move in the sagittal plane. A SLIP model can also run in 3 dimensional space. The torso in the 3D SLIP model reported in [65] is simplified as a point mass, and thus no torso pitching is considered. When the foot is in touch with the ground, the system can be described by three variables: the leg length and the leg angles with respect to the sagittal plane and the lateral

plane. Therefore, the 3D SLIP model also possesses 3 DOFs in the stance phase. Seipel and Holmes examined this model by constructing the Poincaré return map [128]. They concluded that no 3D running gaits are stable if the leg and the hip are passive, and thus, active controls are needed, as in [65].

2.2.2 Realistic SLIP hoppers

In the early 1980s, Raibert and his colleagues built a couple of one-legged robots that can hop in 2D or 3D [118]. In these designs, the hip was driven by pneumatic or hydraulic actuators, and the leg was formed by a pneumatic cylinder with a light-weighted rod sliding inside. The leg was attached to the torso at the torso's CoM. The forward speed, hopping height, and the leg angle could be controlled independently. A finite-state machine, essentially a top-level supervisory controller, helped to switch from one phase to the other. Raibert's hoppers have demonstrated remarkable robustness to ground changes and external disturbances, and have been regarded excellent implementations of the theoretical 3-DOF planar SLIP models and the 3D SLIP model [15, 38, 128, 126].

To improve the energy efficiency, Thompson and Raibert simulated a "passive dynamic running" model [141]. This SLIP-like model contains a massive leg. A torsional spring was adopted at the hip. As the name implied, the hip and the leg are passive. Periodic hopping gaits were found by optimization with suitable initial values.

Alexander compared the Raibert's hopping robots with the leg mechanisms of animals, and suggested uses of springs in legged robots [4]. With this in mind, and also inspired by the success of Raibert's hoppers, Buehler and his team constructed

the Monopods I and II in the early 1990s [117, 41]. A DC motor was used to drive the hip, and a real spring was used in the leg. Especially, in Monopod II, springs connect in series with the hip actuator. Experiments exhibited that the energy efficiency of the Monopod II, in the sense of cost of transport [120, 26], is surprisingly high [2]. To investigate the stability and robustness of Raibert's hoppers, M'Closkey and Burdick studied a 2-DOF planar SLIP model, with a torque and a thrust force applied to the hip and the leg, respectively. With Raibert's control laws, they observed that the forward dynamics did not affect the vertical dynamics [86], verifying the reasonability of Raibert's decoupled control laws. Schwind and Koditschek examined extensively the 2-DOF planar SLIP models with different actuations and control laws, and found locally stable running gaits by using Poincaré return map [127]. Brown and Zeglin developed the Bow Leg hopper [16, 161]. The leg was made of light-weighted fiberglass material with a string connecting the toe and the hip. This design meets well the basic assumption of the 3-DOF planar SLIP models, and the energy consumption is very low. Due to success of Raibert-type hoppers, some researchers felt optimistic and declared that the problems of running had almost been solved by Raibert [114].

Francois and Samson introduced a linear control framework for a SLIP-like model which resembles the passive dynamic running model proposed by Thompson and Raibert, but with active hip and leg [32]. They simplified the nonlinear "complete model" of the hopper and obtained a "nominal" model which was linear and integrable. For the linear nominal model, they proposed two control algorithms. Impulsive or piecewise-constant control signals were injected into the actuators. Linear state feedbacks were also employed. The gains of the feedbacks were obtained by

using linear optimal control method. An integral control term was added to the linear control rules to remove residual velocity errors. For the same model, Hyon and Emura proposed the so-called "energy-preserving control" [57]. The researchers did not use any target dynamics, rather, they designed the control algorithm directly based on the complete nonlinear dynamics. In the flight phase, the hip joint is controlled by "once-switching" of two constant inputs. Quasi-periodic hopping gaits are obtained without touchdown dissipation. Then by adaptation of the touchdown angle, the quasi-period hopping gaits asymptotically converge to periodic ones. Throughout the whole hopping cycle, the energy of the robot maintains constant.

Mombaur et al. proposed another SLIP-like model that runs in the sagittal plane with a massive leg [91]. A torsional spring-damper pair at the hip was used in parallel with a rotary actuator. Another prismatic spring-damper pair was used in series with a "series elastic actuator" (SEA in [115]). A two-level optimization technique was applied to search for the periodic hopping gaits. The outer level optimization sought the appropriate initial values and system parameters with system stability being the objective function. The inner level optimization searched the periodic hopping gaits with the initial values and the system parameters provided by the outer level optimization. The hopper could hop stably without use of feedback. Open-loop control laws were able to stabilize well the system and the system appeared to be robust to different parameter uncertainties and some external disturbances. Unfortunately, since all solutions were found by using numerical methods, more characteristics of such a system could not be analyzed directly. The energy efficiency of the system was not analyzed. On the one hand, the dampers consume energy. But on the other hand, absence of feedbacks may save energy. It would be interesting to

see the overall performance in the sense of energy cost.

Although the SLIP hoppers accomplish stable hopping, they cannot be used in practical work. More versatile running robots are needed. Is the knowledge of the SLIP hoppers helpful for building more complex running robots, such as a bipedal runner? Raibert answered this question positively by comparing the single leg hopping and bipedal running, and combining two SLIP hoppers into one bipedal running machine [118]. However, some researchers thought the SLIP models and the SLIP hoppers are overly-simplified [22, 1], and may not be useful for development of articulated leg systems.

2.2.3 Hopping with an articulated leg

Before Raibert built the celebrated SLIP hoppers, he sketched an articulated leg which was supposed to execute 3D hopping [118]. In this conceptual sketch, the hopper was composed of three links (a torso, a thigh, and a shank), and two actuated joints. The hip could rotate in both sagittal and lateral planes, the knee was powered in sagittal plane only. This sketch is probably the first articulated leg model in formal publications. Although this idea was not really implemented, it stimulated the construction of the articulated hopper - Monopod [75]. Monopod consists of an upper body, a leg, and a foot. The hip joint, driven by a hydraulic actuator, is offset from the CoM of the upper body. The ankle joint is actuated by an inelastic tendon connected to a hydraulic actuator mounted at the hip. A fiberglass leaf spring makes the foot. The maximum hopping speed is $2.3ms^{-1}$.

Zeglin built another articulated leg - Uniroo [160]. As the name hints, it morphologically looks like a one-legged kangaroo with three links (torso, thigh, and shank).

The three rotary joints (hip, knee, and ankle) are driven by hydraulic actuators. The hip is away from the torso's CoM. The leg is massive.

Zhang *et al.* simulated a couple of articulated legs, which were called unipeds in [163]. By applying neural-fuzzy controllers, several jump-like gaits were achieved.

Saranli *et al.* proposed a 4-DOF articulated leg model consisting of a torso, a thigh, a shank, and a foot (called "AKH" model in [124], with "A" standing for ankle, "K" for knee, and "H" for hip). The hip, the knee, and the ankle are active, and the toe is passive. They also simulated a couple of control laws. First, a virtual 2-DOF planar SLIP model was created by connecting the torso's CoM and the toe to form a virtual springy leg. The stiffness of the leg is manually tuned, depending on the desired behavior. A deadbeat controller and a Raibert-type controller were applied to the 4-DOF AKH model, respectively. The control parameters were chosen such that the dynamic behavior of the closed-loop system was as close as possible to those of the target 2-DOF planar SLIP model, with the assumption that the work done by the 4-DOF AKH model being roughly equal to that of the 2-DOF target.

Berkemeier and Fearing commanded an acrobat model, with two rigid links and a revolute actuator in between [11]. Miyazaki et al. advanced Berkemeier and Fearing's work by removing the sliding motion accompanying the hopping [90]. Both groups applied the well-known feedback linearization techniques.

Ikeda *et al.* constructed an articulated hopper, named "Mono-leg" [62]. Mimicking kangaroos, Mono-leg contains five links and four rotary joints. The researchers first captured hopping data of kangaroos, then identified the mathematical model, such that the kangaroo's hopping could be described by a series of 2^{nd} order ordinary differential equations. Finally, the control laws derived by inverse dynamics

were applied such that the closed-loop behavior followed the kangaroo's model.

Hyon and Mita developed a hopper, "Kenken", that appears like a dog's hind limb [59]. The hopper contains four links - a torso, a thigh, a shank, and a foot. The hip, located a little behind the torso's CoM, and the knee are actuated by hydraulics. The passive ankle is driven by a linear spring that connects the thigh and the heel in parallel with the shank. In the stance phase, the hip and the knee are controlled by simple rules, and the robot rotates about the toe. In the flight phase, the leg angle at touchdown is controlled by simple proportional laws.

De Man and his research group built the OLIE [145], which has been mentioned in Section 1.2. The hopper is composed of three links (a torso, a thigh, and a shank) and two rotary joints (a hip and a knee). The hip is located at the CoM of the torso and is actuated by a DC motor. The knee is actuated by a DC motor, together with two torsional springs. The joint trajectories are generated in real time by fitting 5^{th} order polynomials. This group also examined another legged model that directly copies the structure of a human leg. Different from the OLIE, the hip connects the two ends of the torso and the thigh. A flat foot is used to improve gait stability in the stance phase.

Recently, a creative articulated leg prototype was reported in [55]. The leg has three links, including a torso, a thigh, and a shank. The hip is driven by a DC motor, and the knee is actuated by a DC motor combined with springs. Steel cables are used as transmission mechanisms, and thus no backlash is introduced. The stiffness of the flexible knee can be adjusted in real time by software. This unique design makes the leg versatile: it has the potential to execute both walking and running gaits. A multi-loop nonlinear controller, with the hybrid zero dynamics (HZD) of

the ASLIP model being the target, has been simulated to stabilize the robot [112].

Compared to the SLIP hoppers, the articulated legs can be more compact, since the links can be folded about the rotary joints. As a consequence, the swing leg during the flight phase retracts easily, and hence it avoids scuffing with the ground in the middle of the flight without much effort. Also, it is easier to construct and control an articulated leg, due to use of rotary joints. Natural legs in animals imply that the articulated legs can be energy efficient, with compliant elements properly installed.

2.3 Bipedal running

The monopedal hopping and bipedal running share some common characteristics [118]. For either of the runners, only one leg supports the body at a time, only one leg recovers by swinging leg forward at a time, and the two running phases alternate strictly. Therefore, synthesis of bipedal running may borrow some ideas from the successful experience of the hopping machines.

Certainly, bipedal running has its own features different from those of the monopedal hopping. Important comparisons are listed as follows:

1. For two-legged running, the torso's pitching can be substantially slight in the flight phase, since the angular momenta of the two legs can be partly or completely compensated by each other with leg swinging in opposite directions. As a contrast, in the one-legged system with a massive leg, the leg recovery introduces torso's pitching. The one-legged hopper with a massive leg is less efficient than the biped running, since the torso's pitching consumes large amount of

energy. To improve energy efficiency, the leg of the hopper, *e.g.* the bow leg [16], should be very light-weighted.

2. The biped can run faster than the monopod. For a one-legged hopper, the leg recovery takes time. If the forward speed is high, the recovery duration is too short for the hopper to finish in time, unless the actuators are very powerful. For bipedal running, recovery motion of the swing leg generally starts at the beginning of the stance phase, and hence, it finishes before the end of the flight phase with small and energy-saving actuators. However, the swing leg must have some retraction mechanisms to avoid scuffing with the ground during the stance phase.

3. For a 3D biped moving in the sagittal plane, the motions of the legs, assuming not in phase, generate a yaw moment on the torso, and they must be compensated by some special designs. In slow gaits, this could be done by swinging an arm in a proper way. In fast gaits, the next stride generates a counter yaw moment to compensate for the yaw moment in the current stride. The hoppers do not have such a problem, since the leg swings in the plane that contains the CoM.

4. The biped has more DOFs than the monopod, and thus it is easier to find a solution to bipedal running than that to the one-legged hopping. In other words, bipeds are intrinsically easier to balance [149].

2.3.1 Running with two telescopic legs

After building the one-legged hoppers, Raibert's team constructed a bipedal running machine that runs in the sagittal plane [118]. The bipedal running machine contains two one-legged hoppers. It runs in such a way that the two hoppers hop out of phase. During the stance phase, the swing leg retracts quickly to avoid scuffing with the ground. The architecture of the two level control schemes for the one-legged hoppers maintains, with more states and state transitions in the finite state machine.

Hyon *et al.* extended their "energy-preserving control" (see Section 2.2.2) to a planar bipedal running model [58]. The model consists of a massive torso and two massive, telescopic, springy legs. The torso's pitching is considered, and the hips are located below the torso's CoM. To hold the torso upright, the rotary hips have to be controlled. Thus, the energy of the whole system is not preserved. The same bipedal model was later examined by Abdallah and Waldron with different assumptions [1]. Throughout the entire running cycle, the torso is always upright. During the stance phase, the vertical GRF is assumed to be constant, and the horizontal friction force is assumed large enough to avoid slipping. In periodic running, the two legs are always symmetric about the vertical. In acceleration or deceleration, the leg angles at take-off and touchdown are identical but the leg lengths are different. Stable running gaits were realized in simulations.

McGeer simulated a passive bipedal running model in the beginning of 1990s [83]. The model is composed of a point-mass torso, two massive, telescopic, springy legs, and two semi-circular feet. A torsional spring connects the hip and the swing leg when the support foot is in contact with the ground. Like the well-known passive

walking model in [84], the passive running biped is not powered. It runs down along a slight incline, driven by gravity. The pendulum effects and the torque produced by the torsinonal spring help the swing leg to recover in time. During the stance phase, the support leg is suppressed and the swing leg shortens automatically to avoid scuffing with the ground, without changing its moment of inertia and the location of its CoM. The periodic bipedal running gaits were found by using the Poincaré return map. Other running bipeds without feedbacks were also reported in [109, 91]. It is not clear whether the *self-stabilizing* characteristics exist in all running robots.

Unified biped models which can walk and run have also been investigated. Aiming to add running functions to the humanoid robot HRP1, Kajita *et al.* proposed a running synthesis algorithm making use of a simple bipedal model [67]. The model contains a point-mass torso, attached by two telescopic massless legs at the hips. No compliant elements are used. The controllable telescopic legs act like two springs. During the stance phase, the desired vertical trajectory of the hip is assumed to be known in advance. The vertical GRF and the time durations of the two running phases can easily be calculated. By assuming that the net torque about the support foot is zero, the horizontal dynamics of the torso is formulated. The forward speed in the flight phase is adjusted by the foot placement at touchdown, following Raibert's idea. The resolved trajectory of the hip then helps to find the joint trajectories by using inverse kinematics. Unfortunately, simulation results showed that the generated running gaits are not realizable, since the required joint torques are too large, and the energy efficiency is extraordinarily low. A "resolved momentum control" was then proposed to overcome these shortcomings [97, 64]. The goal of the resolved momentum control is to track the desired linear and angular momentum about the

hip, rather than to track the joint trajectories directly. By this way, the planned joint trajectories can be modified online. Experimental results verified the effectiveness of the resolved momentum control law [66]. Srinivasan and Ruina utilized the same biped model to search for the natural choices of bipeds, with respect to different forward velocities [138]. They found that walking and running are the most favorite gaits for bipedal locomotion systems.

Alexander investigated a simple bipedal model that walks and runs in the sagittal plane [5]. This model consists of a point-mass torso and two telescopic springy legs. The two primastic legs and the two rotary hips are actuated. The location of the leg's CoM is assumed unchanged when the leg lengthens and shortens. The vertical GRF is modeled as a truncated Fourier series with kernels being cosine functions of time. The model predicts walking and running gaits, corresponding to different stiffness of the leg springs. Geyer *et al.* investigated a simple bipedal model that contains a point-mass torso and two telescopic massless springy legs [37]. By using the Poincaré return map of a single step, stable periodic gaits are searched. Surprisingly, this simple model predicts both stable running and stable walking gaits. There exists a large speed gap between low-speed walking and high-speed running, and thus, according to their results, low-speed running may not be feasible. This apparently disagrees with humans' experience. Since a spring-mass model is traditionally used to characterize the running motion, these new results have greatly influenced the research community, and have been stimulating new research efforts. Similar work was also reported in [101]. The model follows McGeer's passive bipedal running model with the two semi-circular feet removed. A variety of spring stiffness were tested. Running, walking, and skipping gaits were observed. The running gait is the

favorite choice among the three. These successful practices suggest that a versatile biped can be compliant.

2.3.2 Running with two articulated legs

From late 2003 to late 2004, Qrio (developed by Sony Corp.), HRP-2LR (developed by Kajita *et al.*), and Asimo (developed by Honda Motor) realized running function. As a fact, research activities on running synthesis for articulated biped robots became very active after the turn of the century.

Chevallereau *et al.* synthesized running gaits for a planar biped [23]. The rigid robot contains five links (a torso, two thighs, and two shanks) and four revolute joints (two hips and two knees). Each joint angle is represented by a 4^{th} order polynomial. The five coefficients of the polynomial, and hence the running gaits, were obtained by dynamically optimizing the forward velocity, the joint torques, and the overall energy consumption. With the same model, Fujimoto reported an offline running generator [34], based on the Pontryagin's Minimum Value Principle. The initial states of the robot have to be chosen manually, depending on design intuition and experience. It is not clear whether or not the generated running gaits are energy efficient. Chevallereau *et al.* then developed a hybrid control algorithm for the biped model [24]. This algorithm requires defining a set of output functions equal in number to the controls. The control tasks are embedded into the output functions in such a way that zeroing the outputs ensures that the tracking errors converge to zero. Technically, this is equivalent to designing indirectly a reduced order hybrid zero dynamics for the robot. The Poincaré return map proves that the periodic running gaits can be asymptotically stable. Morris *et al.* tried to verify

this algorithm on a 5-link rigid biped, the RABBIT, and six running steps were performed [93]. This control algorithm was also applied to a biped with 7 links and 6 actuators [132].

Park employed the impedance control algorithm [51] to control a 12-DOF rigid humanoid robot such that it walks stably [102]. Each massive leg has 6 DOFs, 2 at the ankle, 1 at the knee, and 3 at the hip. The impedance control algorithm limited the GRFs while tracking the pre-planned joint trajectories. By inserting the desired impedances of the limbs into the dynamic models, joint torques were solved via inverse dynamics. This algorithm was then applied to a 19-DOF humanoid robot to achieve stable running [105, 73]. The stance phase was assumed to be symmetric about the vertical. In these works, the ZMP stability criterion was used to generate the desired joint trajectories. Very recently, Kwon and Park updated this algorithm and applied it to the 19-DOF humanoid robot again [74]. Asymmetric stance phases were assumed. They also proposed a novel stability criterion for the flight phase. The flight phase is said to be stable if the overall angular momentum about the robot's CoM is within a narrow range of zero.

Inspired by Geyer's studies on the compliant bipedal model [37], Iida *et al.* examined a more realistic bipedal model that aims for practical robotic applications. This bipedal model is composed of seven links (a torso, two thighs, two shanks, and two feet), and six rotary joints (two hips, two knees, and two ankles). The two hips are actively actuated, and other joints are passively driven by antagonistic arrangements of springs. By simulations, stable human-like walking was observed regardless of the initial conditions. A real bipedal robot mimicking this model was then developed. The experimental results verified that walking with compliant legs is plausible. Sey-

farth *et al.* built the JenaWalker II that consists of compliant legs [130]. The biped with seven segments and six revolute joints resembles the Iida's bipedal robot. The two hip joints are driven by the central pattern generators (CPG). The passive knees and ankles are driven by antagonistic spring pairs, respectively. The elastic structure is configurable, and the compliance of the joints is tunable. Walking and running experiments demonstrated that the hip torques corresponding to different forward speed exhibited within a reasonable range. More recently, Iida *et al.* further validated the versatility of the compliant bipedal model [61]. They found that the leg angle at the touchdown can be self-tuned by the internal model dynamics. Walking and running gaits can be performed without or with very little feedback controls.

2.4 Summary

This chapter briefly reviewed the literature related to this thesis. Main definitions of gait stability and the corresponding stability margins were presented first. The ZMP stability and the orbital stability were emphasized. Theoretical running models and realistic running robots with one and two legs were then discussed. Despite stunning success of the SLIP hoppers, the running robots need more study. For versatility, the running robots should be able to walk as well, resulting in booming of unified bipedal robots. Rigid bipeds can certainly realize this goal. New observations in biomechanical analysis and robotics suggest that multiple gaits can also be accomplished by compliant bipeds.

Chapter 3

Hopping Synthesis for a Rigid Articulated Leg

Aiming for advancing the theoretical knowledge of hopping and running, an original offline gait generator for an articulated hopping leg without elastic elements, based on dynamic optimization, is presented in this chapter. Recent results from the biological sciences directly inspire our approach. "The metabolic cost of swinging the limbs is negligible compared with the cost of supporting the body weight" for running gaits of many animals [120]. Measured human torque curves at the ankle, the knee, and the hip show relatively small torques during the flight phase compared to the stance phase [88]. Synthesized bipedal running gaits also display this characteristic [97, 34]. Thus, based on these results found in biomechanical analysis and robotics, the following assumption is made:

Assumption (A1): *If the robot is hopping or running on even level ground, the energy consumed by the robot in the flight phase is much less than that in the stance phase.*

The assumption (A1) is the cornerstone of running synthesis reported in this thesis. It is formulated as a simple constrained static optimization problem. Solving this static optimization problem produces solutions for the initial joint velocities in the flight phase. This differs significantly from the approach in [34] which requires the initial velocities to be given, although both approaches assume known initial joint angles. Ultimately, after a dynamic optimization procedure, the generated gait

serves to validate the assumption (A1). The same strategy can generate hopping and running gaits where even ground conditions do not hold, *e.g.* going up stairs.

The commanded parameters are the desired horizontal step length, the average forward velocity, and the landing height. A numerical technique called *the direct single shooting method* performs the optimization, generating the gait. Section 3.1 provides an overview of the fundamental procedures in optimal gait synthesis for legged robots. The dynamic model of the robot under investigation appears in Section 3.2, including an analysis of the angular momentum and total energy. Two novel objective functions formulate the fundamental assumption (A1), and help to choose the initial joint velocities for the flight phase without much effort. Details of the computer algorithm implementation follow in Section 3.3 and the simulated robot performs its hopping in Section 3.4.

3.1 Optimal gait synthesis for legged robots

Suppose a planar legged robot contains rigid links and rotational joints. The general form for the dynamics is [137]

$$D(q)\ddot{q} + H(q,\dot{q})\dot{q} + G(q) = u, \tag{3.1}$$

where the vector $q(t) \in \Re^n$ contains the n generalized coordinates (including both Cartesian and angular). The terms $D(q) \in \Re^{n\times n}$, $H(q,\dot{q}) \in \Re^{n\times n}$, and $G(q) \in \Re^n$ are the inertia matrix, the matrix containing the centrifugal and Coriolis terms, and the gravitational torque vector, respectively. The input vector $u(t) \in \Re^n$ is a linear combination of the (control) input joint torques. In state space, (3.1) becomes

$$\dot{x} = f(x,u), \tag{3.2}$$

where $x(t) = (q^{\mathrm{T}}, \dot{q}^{\mathrm{T}})^{\mathrm{T}}$ is the state vector of the system. To determine an appropriate control, one can minimize the cost functional

$$C = \int_{t_0}^{t_f} \mathcal{L}(x, u, t)dt, \tag{3.3}$$

where $\mathcal{L}(x, u, t)$ is the Lagrangian determined according to the control objectives.

During the minimization, the state variables $x(t)$ must remain within certain bounds due to physical constraints. The equality constraints $\xi(x)$ constrain the boundary states of a motion phase and the inequality constraints $\zeta(x)$ constrain the motion, and take the general form

$$\xi(x) = 0, \tag{3.4}$$

$$\zeta(x) \leq 0. \tag{3.5}$$

The physical saturations of the i^{th} actuator limit the joint torque such that

$$\tau_{\mathrm{min,i}} \leq \tau_i(t) \leq \tau_{\mathrm{max,i}}. \tag{3.6}$$

Finding the optimal joint trajectory $x_*(t)$ with associated control inputs $u_*(t)$ means ensuring the cost functional (3.3) is minimized subject to the state constraints (3.4) and (3.5) as well as the torque saturations (3.6). This is a typical constrained nonlinear optimization problem, requiring a numerical solution.

An approximation technique usually generates the ideal gait for a legged robot. In (3.2), the state vector x and the control vector u are coupled. Knowing one, the other follows. A common approach is to approximate the state vector x with a weighted sum of basis functions. Finite Fourier series [18], polynomials [20], [21], [23], splines [46], or some mixture of them [156] can all serve as appropriate basis

functions. A nonlinear combination of the chosen basis functions can express the input vector u. Optimizing the chosen cost functional determines the coefficients of the basis functions, allowing approximate computation of x and u. Nonlinear programming [156], sequential quadratic programming [9], Pontryagin's Maximum Principle [121, 12], genetic algorithm [18], simulated annealing [18] can all perform a satisfactory search for the coefficients of the basis functions.

Alternatively, one may construct the profile of the input vector u first. The state vector x depends on a given input vector u and initial state x_0. Choosing a piecewise constant profile for u reduces computational complexity. Let us define a decision vector $U \in \Re^{N \cdot n}$ that gathers all the input vector sequences of all joints as

$$U = \left[u(1)^{\mathrm{T}}, u(2)^{\mathrm{T}}, \cdots, u(N)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{3.7}$$

where N is the number of time intervals, and the time interval is $\Delta t = (t_{\mathrm{f}} - t_0)/N$. Accordingly, (3.6) becomes

$$U_{\min} \leq U \leq U_{\max}, \tag{3.8}$$

where the bounds $U_{\min}$ and $U_{\max}$ define the admissible region of the decision vector.

Given an initial estimate of U, denoted by U_0, integrating (3.2) supplies the joint trajectories x. Then the cost functional $\mathcal{C}$, the equality constraints $\xi(x)$, and the inequality constraints $\zeta(x)$ can be evaluated. If the solution violates one of the constraints, the procedure repeats with a new U_0 . This trial-and-update procedure of U_0, *the direct single shooting method* [122, 10], halts when it finds a local minimum value of the cost functional. Note that the decision vector (3.7) is composed of discrete-time values, but the system model, the cost functional, and the constraints are expressed in continuous time. In practice, discretizing these expressions achieves

better computational efficiency. Hence, the discretized version of the cost functional C is usually used, rather than C itself.

Dynamic optimization has been widely applied to walking synthesis for bipedal robots. Applying this method to a one-legged hopper is a small contribution of the thesis.

3.2 Modeling of the hopper

The planar articulated hopper studied here consists of 4 links and 3 frictionless pin joints (Fig. 3.1). Without loss of generality, four assumptions are made.

1. The mass and inertia of the foot are negligible compared to other links.
2. The height of the foot is negligible, and the ankle joint is assumed to contact the ground. Due to this assumption, *foot* in this thesis can be replaced with *ankle.*
3. During the stance phase, the foot firmly grips the ground, with no slip, bounce, or rotation.
4. During the flight phase, the foot link remains parallel to the ground.

The first three links have length l_i and masses M_{i} ($\mathrm{i} = 1, 2, 3$). The total mass is $M_{\mathrm{t}} = \sum_{i=1}^{3} M_{\mathrm{i}}$. Each moment of inertia I_{i} is taken about the CoM of the i^{th} massive link. The joint angles θ_{i} completely determine the robot's posture. The joint torques are τ_{i}. The counter-clockwise direction is positive.

The CoM and the foot of the robot are related by

$$P_{\mathrm{g}} = P_{\mathrm{f}} + f(\Theta), \tag{3.9}$$

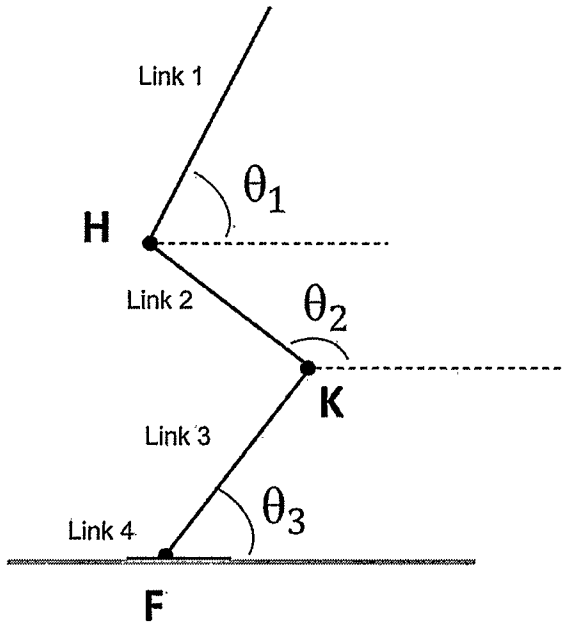

Figure 3.1: The model of the articulated hopper

where $P_{\mathrm{g}} = (X_{\mathrm{g}}, Y_{\mathrm{g}})^{\mathrm{T}}$ and $P_{\mathrm{f}} = (X_{\mathrm{f}}, Y_{\mathrm{f}})^{\mathrm{T}}$ are the positions of the CoM and the foot, respectively, $f(\cdot)$ is a function determined by kinematics, and $\Theta = (\theta_1, \theta_2, \theta_3)^{\mathrm{T}}$. Differentiating (3.9) once gives

$$\dot{P}_{\mathrm{g}} = \dot{P}_{\mathrm{f}} + \frac{\partial f}{\partial \Theta}\dot{\Theta}, \tag{3.10}$$

and twice gives

$$\ddot{P}_{\mathrm{g}} = \ddot{P}_{\mathrm{f}} + \frac{d}{dt}\left(\frac{\partial f}{\partial \Theta}\right)\dot{\Theta} + \left(\frac{\partial f}{\partial \Theta}\right)\ddot{\Theta}. \tag{3.11}$$

In the stance phase, gravity and GRFs act on the robot. In contrast, during the flight phase gravity provides the only external force. When completing the flight phase, the robot foot collides with the ground with a certain velocity. This collision causes an abrupt velocity jump at each joint. The flight phase is to be discussed first, then a solution to the velocity jump due to the foot/ground collision follows,

and finally the stance phase is explained.

3.2.1 The flight phase

During the flight phase, the robot has 5 DOFs. The generalized coordinate vector, $q^{\mathrm{f}} = (P_{\mathrm{g}}^{\mathrm{T}}, \Theta^{\mathrm{T}})^{\mathrm{T}}$, describes the robot's posture in the air.

Equations of motion

Using the Euler-Lagrange method leads to the equations of motion (EoM) during the flight phase (see A.1)

$$D^{\mathrm{f}}(q^{\mathrm{f}})\ddot{q}^{\mathrm{f}} + H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}})\dot{q}^{\mathrm{f}} + G^{\mathrm{f}}(q^{\mathrm{f}}) = B^{\mathrm{f}}\tau^{\mathrm{f}}, \tag{3.12}$$

where the superscript "f" specifies the flight phase, $D^{\mathrm{f}}(q^{\mathrm{f}}) \in \Re^{5\times5}$ is the inertia matrix, $H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}}) \in \Re^{5\times5}$ contains the centrifugal and Coriolis terms , and $G^{\mathrm{f}}(q^{\mathrm{f}}) \in \Re^{5}$ is the gravitational torque vector. The vector $\tau^{\mathrm{f}} = (\tau_1^{\mathrm{f}}, \tau_2^{\mathrm{f}}, \tau_3^{\mathrm{f}})^{\mathrm{T}}$ contains the torques of the hip, the knee, and the ankle. The term $B^{\mathrm{f}} \in \Re^{5\times3}$ is the constant coefficient matrix of τ^{f} and can be determined by using the virtual work principle (based on the selected generalized coordinates). For the sake of clarity, the superscript "f" disappears in the later part of this subsection.

One can partition the generalized coordinates into two parts: $q = (q_1^{\mathrm{T}}, q_2^{\mathrm{T}})^{\mathrm{T}}$, where $q_1 = P_{\mathrm{g}}$, and $q_2 = \Theta$. Partitioning the inertia matrix $D(q)$ produces

$$D(q) = \begin{bmatrix} D_1 & 0_{2\times3} \\ 0_{3\times2} & D_2(q_2) \end{bmatrix}, \tag{3.13}$$

where $D_1 = M_{\mathrm{t}} I_{2\times2}$, and $D_2(q_2) \in \Re^{3\times3}$ contains the moment of inertia related to the rotation of the robot's links. Similarly, the partitioned matrices $H(q, \dot{q})$ and B

become

$$H(q,\dot{q}) = \begin{bmatrix} 0_{2\times 2} & 0_{2\times 3} \\ 0_{3\times 2} & H_2(q_2,\dot{q}_2) \end{bmatrix},$$

and

$$B = \begin{bmatrix} 0_{2\times 3} \\ B_2 \end{bmatrix}.$$

The gravity torque vector is $G(q) = (0, M_t g, 0, 0, 0)^T$, with g being the gravity acceleration. The last three rows of (3.12) are:

$$D_2(q_2)\ddot{q}_2 + H_2(q_2,\dot{q}_2)\dot{q}_2 = B_2\tau. \tag{3.14}$$

Simplifying the first two rows of the EoM (3.12) yields

$$\ddot{X}_g = 0, \tag{3.15}$$

$$\ddot{Y}_g = -g, \tag{3.16}$$

which defines the acceleration of the robot's CoM and results in a parabolic trajectory. Clearly, during the flight phase, the forward velocity of the robot's CoM is a constant, denoted by V_x hereafter.

Denote the time instants at take-off and touchdown by subscripts "to" and "td", respectively, the step length by L_s, and the landing height by h_s. Assuming that the foot positions at take-off and touchdown are $P_{f,to} = (0,0)^T$ and $P_{f,td} = (L_s, h_s)^T$, respectively, and Θ_{to} and Θ_{td} have been picked in advance, equation (3.9) produces $P_{g,to}$ and $P_{g,td}$.

During the flight phase, the horizontal velocity $\dot{X}_{g,to} = \dot{X}_{g,td} = V_x$. The flight lasts for the duration

$$T^f = \frac{X_{g,td} - X_{g,to}}{V_x}. \tag{3.17}$$

The initial and final vertical velocity of the flight phase are

$$\dot{Y}_{\mathrm{g,to}} = \frac{Y_{\mathrm{g,td}} - Y_{\mathrm{g,to}} + \frac{1}{2}g\left(T^{\mathrm{f}}\right)^2}{T^{\mathrm{f}}}, \tag{3.18}$$

and

$$\dot{Y}_{\mathrm{g,td}} = \dot{Y}_{\mathrm{g,to}} - gT^{\mathrm{f}}. \tag{3.19}$$

Angular momentum about the CoM

In the flight phase gravity acts as the only external force at the CoM. Thus the angular momentum of the robot about the CoM is conserved. Namely,

$$\Gamma_{\mathrm{g,to}}(\Theta_{\mathrm{to}}, \dot{\Theta}_{\mathrm{to}}) = \Gamma_{\mathrm{g,td}}(\Theta_{\mathrm{td}}, \dot{\Theta}_{\mathrm{td}}), \tag{3.20}$$

where Γ_{g} stands for the angular momentum about the CoM. Denote the vector from the CoM of the i^{th} massive link to the CoM of the robot by p_{i}, and its first-order time derivative by $\dot{p}_{\mathrm{i}}$, Γ_{g} can be expressed as [145]:

$$\Gamma_{\mathrm{g}} = \sum_{\mathrm{i}=1}^{3} \left((p_{\mathrm{i}} \times M_{\mathrm{i}}\dot{p}_{\mathrm{i}})_{\vec{k}} + I_{\mathrm{i}}\dot{\theta}_{\mathrm{i}} \right), \tag{3.21}$$

where the subscript "$\vec{k}$" indicates the third component of the 3D vector $p_{\mathrm{i}} \times M_{\mathrm{i}}\dot{p}_{\mathrm{i}}$. Equation (3.21) can be simplified as

$$\Gamma_{\mathrm{g}} = s_1\dot{\theta}_1 + s_2\dot{\theta}_2 + s_3\dot{\theta}_3, \tag{3.22}$$

and $s_{\mathrm{i}} = \sum_{j=3}^{5} d_{\mathrm{i}+2,\mathrm{j}}$ for $\mathrm{i} = 1, 2, 3$, with each $d_{\mathrm{i,j}}$ being an element of $D(q)$ indexed by i, j. Therefore, the joint angles determine each s_{i}.

Energy analysis

Let Φ_{k} and Φ_{p} denote the kinetic and potential energy of the robot's CoM, respectively. It can be written that

$$\Phi_{\mathrm{k}} = \frac{1}{2}\dot{q}^{\mathrm{T}}D(q)\dot{q}. \tag{3.23}$$

The total energy of the robot, denoted by Φ_t, can be expressed as

$$\begin{aligned}
\Phi_t &= \Phi_k + \Phi_p, \\
&= \frac{1}{2}\dot{q}^T D(q)\dot{q} + \Phi_p, \\
&= \frac{1}{2}\dot{q}_1^T D_1 \dot{q}_1 + \frac{1}{2}\dot{q}_2^T D_2(q_2)\dot{q}_2 + \Phi_p, \\
&= \left(\frac{1}{2}\dot{q}_1^T D_1 \dot{q}_1 + \Phi_p\right) + \frac{1}{2}\dot{q}_2^T D_2(q_2)\dot{q}_2, \qquad (3.24) \\
&= \left(\frac{1}{2}M_t \dot{X}_g^2 + \frac{1}{2}M_t \dot{Y}_g^2 + \Phi_p\right) + \frac{1}{2}\dot{q}_2^T D_2(q_2)\dot{q}_2. \qquad (3.25)
\end{aligned}$$

Equation (3.24) implies that the total energy of the robot may be decomposed into three parts: the translational kinetic energy ($\frac{1}{2}\dot{q}_1^T D_1 \dot{q}_1$), the potential energy (Φ_p), and the rotational kinetic energy of the links ($\frac{1}{2}\dot{q}_2^T D_2(q_2)\dot{q}_2$).

Let the rotational kinetic energy of the links be denoted by Φ_{rot}. It can be inferred that

$$\begin{aligned}
\frac{d}{dt}(\Phi_p) &= [G(q)]^T \dot{q}, \\
&= M_t g \dot{Y}_g, \qquad (3.26)
\end{aligned}$$

and

$$\begin{aligned}
\frac{d}{dt}(\Phi_{rot}) &= \dot{q}_2^T D_2(q_2)\ddot{q}_2 + \frac{1}{2}\dot{q}_2^T \dot{D}_2(q_2)\dot{q}_2, \\
&= \dot{q}_2^T \left[D_2(q_2)\ddot{q}_2 + \frac{1}{2}\dot{D}_2(q_2)\dot{q}_2\right], \\
&= \dot{q}_2^T \left[B_2\tau - H_2(q_2, \dot{q}_2)\dot{q}_2 + \frac{1}{2}\dot{D}_2(q_2)\dot{q}_2\right], \\
&= \dot{q}_2^T B_2 \tau. \qquad (3.27)
\end{aligned}$$

Note that H_2 can be formulated such that $\dot{D}_2 - 2H_2$ is skew-symmetric, and hence $\dot{q}_2^T \left(\frac{1}{2}\dot{D}_2 - H_2\right)\dot{q}_2 = 0$ (see [137]).

Differentiating both sides of (3.25) yields

$$\frac{d}{dt}(\Phi_{\mathrm{t}}) = \dot{q}_2^{\mathrm{T}} B_2 \tau = \frac{d}{dt}(\Phi_{\mathrm{rot}}). \tag{3.28}$$

Equation (3.28) means that the sum of the translational kinetic energy and the potential energy is a constant, since the change rate of this sum is 0. In general, τ is not 0, and thus $\frac{d}{dt}(\Phi_{\mathrm{rot}}) \neq 0$. Therefore, the rotational kinetic energy is not conserved.

By integrating both sides of (3.27) over the whole flight phase and taking absolute values, it can be obtained that

$$|\Phi_{\mathrm{rot,td}} - \Phi_{\mathrm{rot,to}}| = \left| \int_{\mathrm{t_{to}}}^{\mathrm{t_{td}}} \dot{q}_2^{\mathrm{T}} B_2 \tau dt \right|, \tag{3.29}$$

$$\leq \int_{\mathrm{t_{to}}}^{\mathrm{t_{td}}} \sum_{\mathrm{j}=1}^{3} |\dot{q}_{2,\mathrm{j}}| \left| (B_2\tau)_{\mathrm{j}} \right| dt. \tag{3.30}$$

There are numerous possibilities for the robot to hop from one place to another. However, for an *energy-efficient* hopping gait, the rotational kinetic energy at the take-off, *i.e.* $\Phi_{\mathrm{rot,to}}$, must be small, within all physical constraints. Further, the right hand side (RHS) of (3.30) is essentially the energy supplied by the controller of the robot. The assumption (A1) says that on even ground, the energy consumed in the flight phase is small, and hence $|\Phi_{\mathrm{rot,td}} - \Phi_{\mathrm{rot,to}}|$ is small. This relationship may be formulated as

$$\Phi_{\mathrm{rot,to}} \approx \Phi_{\mathrm{rot,td}}.$$

Unfortunately, this treatment can hardly be extended to more complicated situations, such as hopping or running up stairs. When the robot hops up stairs, the energy consumed by the robot may be large, and thus the difference between $\Phi_{\mathrm{rot,to}}$ and

$\Phi_{\text{rot,td}}$ is large. To be more flexible, the assumption (A1) is formulated as a static optimization procedure, with an objective function proposed as

$$\mathcal{O}_1 = \gamma\Phi_{\text{rot,to}} + (1-\gamma)\left(\Phi_{\text{rot,to}} - \Phi_{\text{rot,td}}\right)^2, \tag{3.31}$$

where the constant $\gamma \in [0,1]$ is a weighting factor. Note that $\mathcal{O}_1$ is a function of $\dot{\Theta}_{\text{to}}$ and $\dot{\Theta}_{\text{td}}$, provided that Θ_{to} and Θ_{td} have been chosen in advance. Minimization of the two terms on the RHS of (3.31) forces $\Phi_{\text{rot,to}}$ and $|\Phi_{\text{rot,td}} - \Phi_{\text{rot,to}}|$ as small as possible.

Since energies $\Phi_{\text{rot,to}}$ and $\Phi_{\text{rot,td}}$ are positive, it can be deduced that

$$\Phi_{\text{rot,td}} - \Phi_{\text{rot,to}} \leq \left|\Phi_{\text{rot,td}} - \Phi_{\text{rot,to}}\right|, \tag{3.32}$$

and

$$\Phi_{\text{rot,td}} \leq \Phi_{\text{rot,to}} + \left|\Phi_{\text{rot,td}} - \Phi_{\text{rot,to}}\right|. \tag{3.33}$$

Thus if $\Phi_{\text{rot,to}}$ is small so is $\Phi_{\text{rot,td}}$. This leads to a second, more concise, objective function

$$\mathcal{O}_2 = \gamma\Phi_{\text{rot,to}} + (1-\gamma)\Phi_{\text{rot,td}}. \tag{3.34}$$

Minimization of $\mathcal{O}_2$ forces $\Phi_{\text{rot,to}}$ and $\Phi_{\text{rot,td}}$ to be small.

In practice, (3.34) outperforms (3.31) in two aspects. It results in more efficient gaits, and also, γ can be chosen in a larger range.

Boundary joint velocities

The boundary joint angles are picked manually, but the boundary joint velocities have to be determined. The problem can be solved by using a static optimization procedure, which is stated as:

Solving for $\dot{\Theta}_{to}$ and $\dot{\Theta}_{td}$, such that either one of the objective functions, (3.31) and (3.34), is minimized. The equality constraints could be

$$\begin{aligned} \dot{P}_{g,to} &= [V_x, \dot{Y}_{g,to}]^T, \\ \dot{P}_{g,td} &= [V_x, \dot{Y}_{g,td}]^T, \\ \Gamma_{g,to} &= \Gamma_{g,td}. \end{aligned}$$

Some inequality constraints, such as the angular velocity bounds at all joints, can also be applied. For example,

$$\omega_{i,min} \leq \omega_i \leq \omega_{i,max}, \tag{3.35}$$

where ω_i for $i = 1, 2, 3$ is the angular velocity at the i^{th} joint, $\omega_{i,min}$ and $\omega_{i,max}$ are the lower and upper bounds of the corresponding joint velocity.

With this static optimization procedure, $\dot{\Theta}_{to}$ and $\dot{\Theta}_{td}$ can easily be solved. Note that $\dot{\Theta}_{to}$ will be used as the initial value of flight phase, and $\dot{\Theta}_{td}$ is discarded. The true value of $\dot{\Theta}_{td}$ is searched by dynamic optimization, rather than the static optimization procedure.

Not all $\dot{\Theta}_{to}$, produced by the static optimization procedure, can be used as the initial velocity values of the dynamic optimization. This is not only determined by the flight phase, but also by the subsequent stance phase. The stance phase is also required to be efficient within constraints. A suitable initial velocity vector leading to a satisfactory hopping cycle depends on appropriate choices of γ, Θ_{to}, and Θ_{td}. With Θ_{to} and Θ_{td} picked in advance, there exists an optimal value of γ, denoted by γ_*, which results in an optimal flight phase in the sense of the cost function, (3.3). However, this γ_* may not be a good choice for the the stance phase. A more suitable γ can be found around γ_* by trial-and-error.

When $\gamma = 0$, (3.31) and (3.34), degrade to $\mathcal{O}_1 = (\Omega_{\text{rot,td}} - \Omega_{\text{rot,to}})^2$, and $\mathcal{O}_2 = \Omega_{\text{rot,td}}$, respectively. When $\gamma = 1$, both objective functions become $\Omega_{\text{rot,to}}$. As γ changes from 0 to 1, the resultant initial velocity vectors, respectively corresponding to the two objective functions, move along two different trajectories, and they intersect when $\gamma = 1$. This may be regarded as the geometrical connection between the two objective functions. When $\gamma \approx 1$, if one of the two objective functions produces satisfactory hopping cycle, it is highly possible that the other objective function can also generate a good result.

Foot velocity regulation

According to [145], energy loss due to the collision between the foot and the ground is proportional to the foot velocity immediately before the collision. Thus, if the foot velocity were 0, the robot would lose no energy when it collides with the ground. However, this setting may result in a dramatic increase of the control effort in the following stance phase. In this work, the foot velocity at the moment of touchdown is regulated to be

$$V_{\text{f,td}} = \begin{bmatrix} k_{\text{x}} & 0 \\ 0 & k_{\text{y}} \end{bmatrix} V_{\text{g,td}}, \tag{3.36}$$

where the constants $k_{\text{x}} \in [0, 1]$ and $k_{\text{y}} \in [0, 1]$ are recommended, trading off efficiency and control effort. A small k_{x} implies acceleration, and a large positive k_{x} slows down the robot. When k_{x} and k_{y} are both large, the energy loss due to the foot/ground collision is large, but the total energy consumption during a hopping cycle may be small.

3.2.2 Collision between the foot and the ground

The collision of the support foot with the ground is assumed to be instantaneous and inelastic. The impulsive force applied by the ground to the robot causes an abrupt jump of joint velocities, while joint angles remain unchanged. Equation (3.9) can be rearranged as

$$P_{\mathrm{f}} = f_1(q^{\mathrm{f}}) := P_{\mathrm{g}} - f(\Theta).$$

Letting the time instants immediately before and after the collision be denoted by superscripts "−" and "+", respectively, and defining the Jacobian

$$J = \frac{\partial f_1}{\partial q^{\mathrm{f}}}, \tag{3.37}$$

allows to write [34],

$$\dot{q}^{+} = \left[I_{5\times 5} - \left(D^{\mathrm{f}}\right)^{-1} J^{\mathrm{T}} \left(J \left(D^{\mathrm{f}}\right)^{-1} J^{\mathrm{T}} \right)^{-1} J \right] \dot{q}^{-}, \tag{3.38}$$

where $D^{\mathrm{f}} = D^{\mathrm{f}}(q)$ and $J = J(q)$ are evaluated at the touchdown. That is, $q = q_{\mathrm{td}}$ is inserted. Similarly, $\dot{q}^{-} = \dot{q}_{\mathrm{td}}$.

3.2.3 The stance phase

Adjustment of the forward velocity can only be achieved in the stance phase. Also, the stance phase prepares the necessary initial states for the next flight phase, and balances the whole body. Consequently, most control effort occurs in the stance phase.

Equations of motion

During the stance phase, the hopper has 3 DOFs, assuming that the foot is firmly in contact with the ground. Choose the convenient generalized coordinate vector

$q^s = \Theta$, which results in the dynamic model of the hopper in stance phase (refer to A.6):

$$D^s(q^s)\ddot{q}^s + H^s(q^s, \dot{q}^s)\dot{q}^s + G^s(q^s) = B^s\tau^s, \tag{3.39}$$

where $D^s \in \Re^{3\times3}$ is the inertia matrix, $H^s \in \Re^{3\times3}$ contains centrifugal and Coriolis terms, and $G^s \in \Re^3$ is the gravitational torque vector. The matrix $B^s \in \Re^{3\times3}$ is the coefficient matrix and depends on the choice of the generalized coordinates.

Time duration of the stance phase

The average forward speed in the stance phase is assumed to be equal to the horizontal speed in the flight phase ($\bar{V}_g^{st} = V_x$). Time duration of the stance phase becomes

$$T^s = \frac{X_{g,to}^{i+1} - X_{g,td}^{i}}{V_x}, \tag{3.40}$$

where the superscript "i" means the i^{th} hopping cycle, and $X_{g,to}^{i+1} - X_{g,td}^{i}$ is the horizontal distance that the robot's CoM travels in the stance phase.

Boundary joint velocities

The stance phase is a typical TPBVP. The joint angles at the two boundary points are chosen manually. The boundary joint velocities can be determined by

$$\begin{aligned} \dot{\Theta}_{initial}^{i,st} &= \dot{\Theta}_{td}^{i,+}, \\ \dot{\Theta}_{final}^{i,st} &= \dot{\Theta}_{to}^{i+1}, \end{aligned}$$

where the superscripts "i" and "+" have the same meaning as before, and the subscripts "initial" and "final" indicate the two boundary points.

The normal support force

The GRFs include the normal support force of the ground and the tangential friction force between the foot and the ground. The normal support force of the ground pushes vertically, formulated as

$$F_{\mathrm{y}} = M_{\mathrm{t}}(g + \ddot{Y}^{\mathrm{s}}_{\mathrm{g}}) \geq 0,$$

and hence

$$\ddot{Y}^{\mathrm{s}}_{\mathrm{g}} \geq -g. \tag{3.41}$$

Assuming the foot/ground collision occurs instantaneously, during the collision the resultant normal GRF may be extremely large. To prevent the robot from damage, impact-absorbing designs, such as passive compliance devices (*e.g.* rubbers and springs in [50, 27]) or active force control schemes, could be useful for online gait adaptation. In this off-line gait generator, an upper bound of the normal GRF is considered,

$$F_{\mathrm{y}} \leq F_{\mathrm{y,max}}. \tag{3.42}$$

At the end of the stance phase, the normal GRF reduces in amplitude, and vanishes at the take-off of the next flight phase. This implies that $F^{\mathrm{s}}_{\mathrm{y,final}} = 0$, and

$$\ddot{Y}^{\mathrm{s}}_{\mathrm{g,final}} = -g. \tag{3.43}$$

The tangential friction force

To ensure that the robot stands on the ground firmly, the horizontal inertial force of the robot must be less than the static friction force between the foot and the ground. That is

$$|\ddot{X}^{\mathrm{s}}_{\mathrm{g}}| \leq \frac{\mu F_{\mathrm{y}}}{M_{\mathrm{t}}}. \tag{3.44}$$

Clearly, once the ground, and hence the static friction coefficient μ, is determined, the horizontal acceleration in the stance phase must be bounded.

The ZMP

Assuming the foot does not rotate during the stance phase, the ZMP should be located within the foot range [147]. Following the arguments by Popovic *et al.* [111], we derive the ZMP for the planar hopper as

$$X_{\text{zmp}} = \frac{\sum_{\text{i}=1}^{3} M_{\text{i}} \left[X_{\text{g,i}}^{\text{s}} \left(\ddot{Y}_{\text{g,i}}^{\text{s}} + g \right) - Y_{\text{g,i}}^{\text{s}} \ddot{X}_{\text{g,i}}^{\text{s}} \right] + \sum_{\text{i}=1}^{3} I_{\text{i}} \ddot{\theta}_{\text{i}}}{M_{\text{t}} \left(\ddot{Y}_{\text{g}}^{\text{s}} + g \right)}. \tag{3.45}$$

Suppose front and rear lengths of the foot (separated by the ankle) are l_{f1} and l_{f2} respectively. Then

$$-l_{\text{f2}} \leq X_{\text{zmp}} \leq l_{\text{f1}}. \tag{3.46}$$

Starting and stopping

In the *starting phase* the robot moves from an upright static position, defined by $(\Theta^{\text{T}}, \dot{\Theta}^{\text{T}})^{\text{T}} = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, 0, 0, 0)^{\text{T}}$, into the desired initial state of the flight phase. The *stoppping phase* begins immediately after a foot/ground collision and ends when the robot reaches an upright static position. The same optimization procedure that works for the stance phase also works for the starting and stopping phases, with minor modifications of the boundary conditions.

Differing from the ordinary stance phase, the starting phase begins with

$$\begin{aligned} \ddot{X}_{\text{g,initial}}^{\text{start}} &= 0, \\ \ddot{Y}_{\text{g,initial}}^{\text{start}} &= 0, \\ X_{\text{zmp,initial}}^{\text{start}} &= 0, \end{aligned}$$

where the superscript "start" specifies the starting phase. Similarly, we can also formulate the constraints at the final time instant of the stopping phase as

$$\begin{aligned} \ddot{X}^{\text{stop}}_{\text{g,final}} &= 0, \\ \ddot{Y}^{\text{stop}}_{\text{g,final}} &= 0, \\ X^{\text{stop}}_{\text{zmp,final}} &= 0, \end{aligned}$$

where the superscript "stop" indicates the stopping phase.

When the robot stands upright, the horizontal and vertical accelerations of the CoM are zero, and the ZMP must be zero. These constraints respectively imply that the tangential friction force between the foot and the ground disappears, the normal support force of the ground is equal to the total weight of the robot body, and the projection of the CoM on the ground coincide with the ankle in our simplified hopper model.

3.3 Hopping optimization

The flight and stance hopping phases can be optimized separately, or together as a whole hopping cycle. The direct single shooting method reviewed in Section 3.1 is implemented by the Matlab optimization routine *fmincon()*.

3.3.1 Simple rules for choosing the boundary joint angles

The boundary joint angles of the flight phase, *i.e.* Θ_{to} and Θ_{td}, have to be picked manually. Although humans choose the initial posture easily, the underlying mechanisms are not completely revealed. For legged robots, a systematic method to pick

the appropriate initial posture does not exist. Specifically, for the hopper under study, humans' favorite initial posture in running can be tried first. Modifications can then be made according to the resultant ZMP trajectory in the stance phase. Some intuitive rules may be helpful for choosing appropriate initial values:

1. If the calculated ZMP stays inside the foot range at all times during the stance phase then Θ_{to} and Θ_{td} remain unchanged.

2. If the calculated ZMP is behind the heel in the beginning of the stance phase then decrease some components of Θ_{td}, moving the CoM of the robot forward.

3. If the calculated ZMP is in front of the toe in the beginning of the stance phase then increase some components of Θ_{td}, moving the robot's CoM backward.

4. If the calculated ZMP is behind the heel in the end of the stance phase then decrease some components of Θ_{to}.

5. If the calculated ZMP is in front of the toe in the end of the stance phase then increase some components of Θ_{to}.

6. If the calculated ZMP is always outside of the foot range, no matter how Θ_{to} and Θ_{td} are selected. In this case, smaller V_x, h_s, or L_s should be commanded.

3.3.2 Single-phase optimization

Single phase optimization proceeds on the two hopping phases separately. Each phase has a unique dynamic model, and also differs in constraints. However, the implementation method of the dynamic models, the definitions of the decision vectors, and the cost functions are almost identical for the two hopping phases.

Model implementation

Simulation of the second-order dynamic models (3.12) and (3.39) is straightforward. In principle, the direct single shooting method needs to find an appropriate initial decision vector U_0. Since inappropriate initialilization may cause the term $H(q,\dot{q})\dot{q}$ to grow too large for numerical integration, in the Simulink model a saturation block limits the output of this term.

The decision vector

The decision vector is defined as

$$U = \left[\tau(1)^{\mathrm{T}}, \tau(2)^{\mathrm{T}}, \cdots, \tau(N)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{3.47}$$

where $\tau(k) = [\tau_1(k), \tau_2(k), \tau_3(k)]^{\mathrm{T}}$, and k indicates the kth discretized time interval. U_0, the initial estimate of U, is taken a zero vector with same dimension as U. The torque limits of the joints form the bounds of the decision vector which have been expressed as (3.6)

The cost function

The two hopping phases have the same cost function, which is defined as

$$C_1 = \frac{1}{2}U^{\mathrm{T}}U\Delta t, \tag{3.48}$$

where Δt is the time interval with an assumption that the phase duration, T^{f} or T^{s}, is discretized equally. The number of time intervals, N, may be different in the two phases. Clearly, this cost function implies the *least control effort*. Note that the discretized Lagrangian is $\mathcal{L}(U(k)) = \frac{1}{2}U^{\mathrm{T}}U$.

Constraints in the flight phase

In the flight phase, the boundary values and (3.36) are the equality constraints. The physical ranges of the joints constitute the linear inequality constraints. Besides (3.35), the following constraints must be satisfied:

$$\theta^{\mathrm{f}}_{\mathrm{i,min}} \leq \theta^{\mathrm{f}}_{\mathrm{i}} \leq \theta^{\mathrm{f}}_{\mathrm{i,max}},$$

$$\alpha^{\mathrm{f}}_{\mathrm{min}} \leq \alpha^{\mathrm{f}} \leq \alpha^{\mathrm{f}}_{\mathrm{max}},$$

where $\alpha = \theta_2 - \theta_3$ is the relative angle between the upper leg and the lower leg. All the inequality constraints form the set $\zeta^{\mathrm{f}}(x)$.

Constraints in the stance phase

In the stance phase, the boundary values and (3.43) are the equality constraints. The linear inequality constraints could be written in the same way as those in the flight phase, with the superscript "f" being replaced by "s". The bound values in the stance phase may not be equal to their counterparts in the flight phase, but in our implementations, the corresponding bounds are designed to be identical.

Constraints defined by (3.41), (3.42), (3.44), and (3.46) are the nonlinear inequality constraints. All linear and nonlinear inequality constraints form the set of $\zeta^{\mathrm{s}}(x)$.

3.3.3 Complete-cycle optimization

Complete-cycle optimization synthesizes the two hopping phases together. The cost function, which evaluates the control effort of the whole hopping cycle, is defined as

$$\mathcal{C}_2 = \frac{1}{2}\left[\sum_{\mathrm{i}=0}^{N^{\mathrm{f}}-1} \left(U^{\mathrm{f}}\right)^{\mathrm{T}} \left(U^{\mathrm{f}}\right) \Delta t^{\mathrm{f}} + \sum_{\mathrm{i}=\mathrm{N}^{\mathrm{f}}}^{N^{\mathrm{f}}+N^{\mathrm{s}}-1} \left(U^{\mathrm{s}}\right)^{\mathrm{T}} \left(U^{\mathrm{s}}\right) \Delta t^{\mathrm{s}}\right], \tag{3.49}$$

where U^{f} and U^{s} are the decision vectors, N^{f} and N^{s} are the numbers of time intervals, Δt^{f} and Δt^{s}, of the discretized flight phase and stance phase, respectively.

The constraints of the whole cycle are essentially the same as the two separate hopping phases, but they have to be written only in one constraint function.

Correspondingly, the decision vector is $U = [(U^{\mathrm{f}})^{\mathrm{T}}, (U^{\mathrm{s}})^{\mathrm{T}}]^{\mathrm{T}}$.

3.3.4 A five step procedure for optimizing the hopping cycle

The following five step procedure generates an energy-efficient hopping cycle:

1. Pick Θ_{to} and Θ_{td}.
2. Search for the initial joint velocities $\dot{\Theta}_{\mathrm{to}}$, given L_{s}, h_{s}, V_x, and γ.
3. Search for the optimal flight phase.
4. Search for the optimal stance phase.
5. Optimize the generated flight phase and stance phase together as a complete cycle.

Step 3 and step 4 belong to *single-phase optimization*, and step 5 is a *complete-cycle optimization*. Step 3 is relatively straightforward but step 4 contains nonlinear constraints. The nonlinear constraints make the system sensitive to the initial estimate of the decision vector U_0, and hence the solution may converge at a local minimum. To address this issue, step 4 occurs at least twice. For the first run, applying only linear constraints results in quick search for a feasible solution. This solution then becomes the initial value for the second run, in which the nonlinear constraints are applied. Step 5 is usually optional. Note that an energy-efficient flight phase may

actually result in the stance phase consuming a large amount of energy. Step 5 not only reduces the energy cost further but balances the two hopping phases. If the resultant gait is still not satisfactory, one can generate a different gait by picking different Θ_{to}, Θ_{td}, γ and re-running the optimization procedure. If necessary, smaller V_{x}, L_{s}, or h_{s} can be tried. A one-legged hopper becomes unbalanced quite easily, so the range of suitable parameters is rather small.

3.3.5 A performance index

In the flight phase, the robot does work

$$W^{\text{f}} = \sum_{\text{j}=1}^{5} \sum_{\text{i}=0}^{N^{\text{f}}-1} \int_{t_{\text{to}}+i\Delta t^{\text{f}}}^{t_{\text{to}}+(i+1)\Delta t^{\text{f}}} \left| \dot{q}_{\text{j}}^{\text{f}} \left(B^{\text{f}} \tau^{\text{f}} \right)_{\text{j}} \right| dt, \tag{3.50}$$

where the subscript "j" indicates the j^{th} components of $\dot{q}^{\text{f}}$, "i" indicates the i^{th} time interval, N^{f} is the total number of time intervals, and Δt^{f} is the time interval. Note that each joint torque $\tau_{\text{j}}^{\text{f}}$ is a piecewise constant signal.

In the stance phase, the robot does work

$$W^{\text{s}} = \sum_{\text{j}=1}^{3} \sum_{\text{i}=0}^{N^{\text{s}}-1} \int_{t_{\text{td}}+i\Delta t^{\text{s}}}^{t_{\text{td}}+(i+1)\Delta t^{\text{s}}} \left| \dot{q}_{\text{j}}^{\text{s}} \left(B^{\text{s}} \tau^{\text{s}} \right)_{\text{j}} \right| dt, \tag{3.51}$$

and all symbols can be interpreted similar as those in the flight phase. The total work in a whole hopping cycle is

$$W = W^{\text{f}} + W^{\text{s}}. \tag{3.52}$$

To compare the energy efficiency of the robot, a performance index, *the cost of transport*, is defined as [120, 26]

$$\epsilon = \frac{W}{M_{\text{t}} g L}, \tag{3.53}$$

Table 3.1: Typical costs of running

Species	Values	Data sources
Humans	0.280	[120]
Wild turkeys	0.632	[120]
Monopod I	0.70	[3]
Monopod II	0.22	[3]
HRP-2L	3.57	[97]

where L is the distance traveled by the robot's CoM. For periodic hopping gaits, $L = \sqrt{L_s^2 + h_s^2}$.

A smaller cost of transport implies higher energy efficiency. For comparison, the costs of running for humans, wild turkeys, Monopod I, Monopod II, and the HRP-2L are listed in Table 3.1. In [97], the cost of transport of the HRP-2L was not explicitly given. The value is estimated by using the provided specifications. Note that efficiency of the Monopod II is higher than that of humans! The springs indeed help to reduce energy consumption.

3.4 Simulation results and discussions

In this section, algorithms generate typical hopping gaits using the two objective functions (3.31) and (3.34). The resulting cost of transport serves a performance comparison. The robot first hops on even ground, then hops up stairs. All simulations utilize some common parameters (Table 3.2). The step length L_s, the landing height h_s, and the average forward speed V_x are provided as inputs.

Table 3.2: The constants of the hopper

Parameters	Values	Units	Parameters	Values	Units
M_1	8	kg	I_1	0.4	kg-m^2
M_2	2	kg	I_2	0.02	kg-m^2
M_3	0.8	kg	I_3	0.01	kg-m^2
l_1	0.6	m	r_1	0.4	-
l_2	0.35	m	r_2	0.6	-
l_3	0.4	m	r_3	0.6	-
l_{f1}	0.15	m	l_{f2}	0.05	m
g	9.81	m/s^2	-	-	-
$\theta_{1,\min}$	45	deg	$\theta_{1,\max}$	90	deg
$\theta_{2,\min}$	75	deg	$\theta_{2,\max}$	225	deg
$\theta_{3,\min}$	30	deg	$\theta_{3,\max}$	120	deg
$\dot{\theta}_{1,\min}$	-150	deg/s	$\dot{\theta}_{1,\max}$	150	deg/s
$\dot{\theta}_{2,\min}$	-500	deg/s	$\dot{\theta}_{2,\max}$	500	deg/s
$\dot{\theta}_{3,\min}$	-450	deg/s	$\dot{\theta}_{3,\max}$	450	deg/s
$\alpha_{\min}$	0	deg	$\alpha_{\max}$	150	deg
$\tau_{1,\min}$	-40	N-m	$\tau_{1,\max}$	40	N-m
$\tau_{2,\min}$	-50	N-m	$\tau_{2,\max}$	50	N-m
$\tau_{3,\min}$	-30	N-m	$\tau_{3,\max}$	30	N-m

Table 3.3: The parameters for hopping on even ground

Parameters	Values	Units
L_s	0.25	m
h_s	0	m
V_x	0.5	m/s
Θ_{to}	$(78, 110, 75)^T$	deg
Θ_{td}	$(73, 115, 95)^T$	deg
μ	0.6	-
k_x	0.5	-
k_y	0.5	-
$F_{y,max}$	$2M_tg$	N

3.4.1 Hopping on even ground

In this simulation, the robot hops on even ground using both objective functions (3.31) and (3.34). Both cases use the same parameters (Table 3.3) such that the results can be fairly compared. The gaits are synthesized using the first 4 steps of the 5-step procedure.

$\dot{\Theta}_{to}$ vs. γ

From (3.31) and (3.34), it can be observed that the initial joint velocities are affected by choice of γ. In Fig. 3.2, solid curves show the initial velocities produced by use of (3.34), and the dashed curves by use of (3.31). Using (3.31) makes the range of $\dot{\Theta}_{to}$ small. Consequently, if a particular γ does not result in a decent hopping gait, the neighboring values of γ cannot be used either. A wider range of $\dot{\Theta}_{to}$ can be obtained by use of (3.34), facilitating generation of more gaits. Since the mass and inertia of the upper body are much larger than those of other links, the upper body contributes large portions of the total angular momentum and the total kinetic energy. Thus,

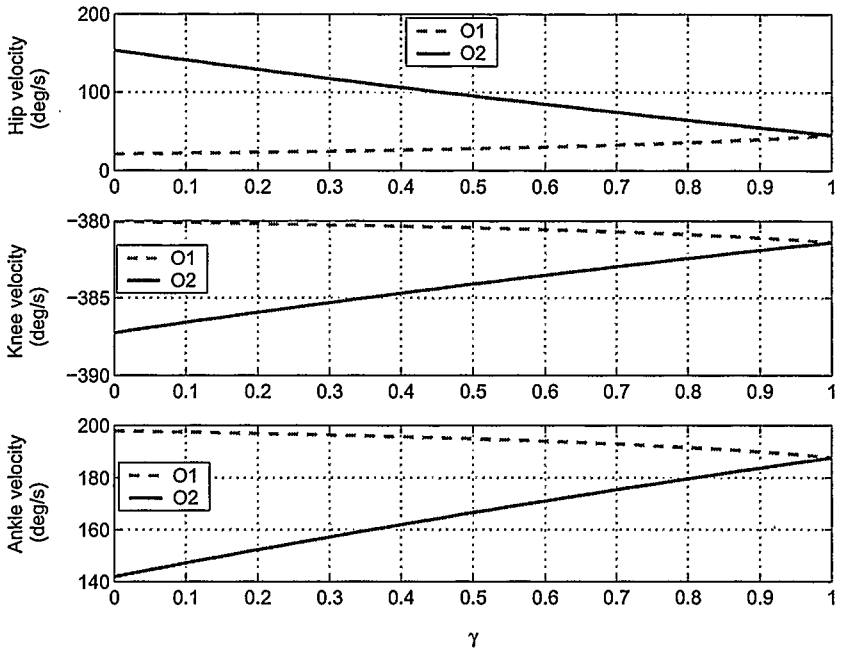

Figure 3.2: Initial velocities vs. γ

small $\dot{\theta}_{1,\text{to}}$ is preferred.

In each graph of Fig. 3.2, the corresponding velocity curves intersect at $\gamma = 1$, as discussed in Section 3.2.

$\Omega_{\text{rot,to}}$ and $\Omega_{\text{rot,td}}$ vs. γ

In Fig. 3.3, the three panes show how $\Omega_{\text{rot,to}}$, $\Omega_{\text{rot,td}}$ with γ, and $\Omega_{\text{rot,to}} - \Omega_{\text{rot,td}}$ change with γ. The solid curves and dashed curves represent use of (3.34) and (3.31), respectively. They are very small, and the corresponding curves connect at $\gamma = 1$, as expected.

Control effort vs. γ

The control effort of a complete hopping cycle is defined as

$$\mathcal{C}_1^{\text{cc}} = \mathcal{C}_1^{\text{f}} + \mathcal{C}_1^{\text{s}},$$

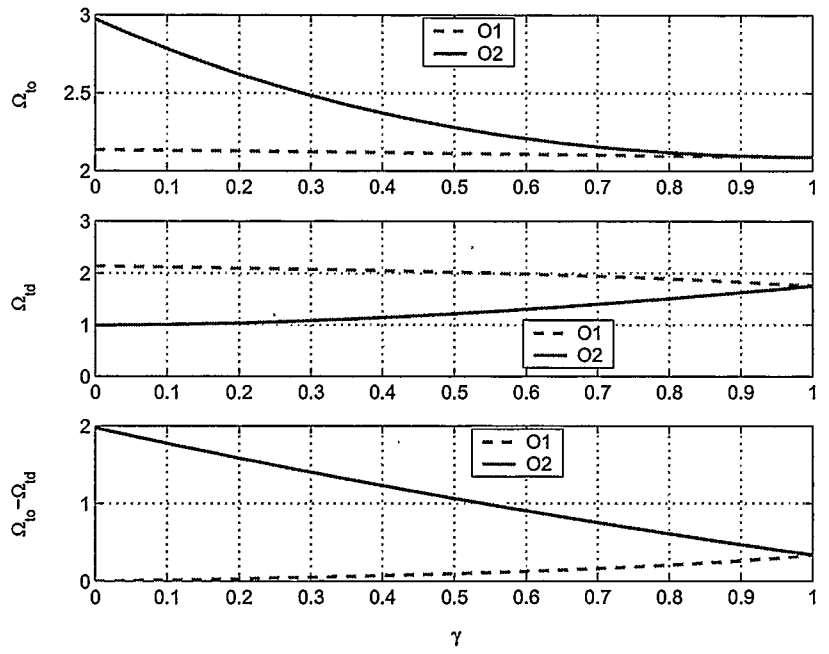

Figure 3.3: Boundary rotational kinietic energy vs. γ

where the subscript "cc" indicates the complete cycle. In fact, $\mathcal{C}_1^{cc}$ is the same as (3.49). This new notation, rather than C_2 in (3.49), indicate that step 5 (in the 5-step procedure) is not used. In Fig. 3.4, the solid curves correspond to the objective function (3.34), and the dashed curves to (3.31). Note that in the top pane, for $\gamma < 0.6$, use of (3.31) produces hopping gaits scuffing the ground in the beginning of the flight phase, and hence they are discarded. In contrast, when (3.34) is used, all $\gamma \in [0, 1]$ can be used to generate energy-efficient hopping gaits. For the parameters used in the simulations, when $\gamma \in [0, 0.4]$, the synthesized flight phases contain better take-off, with the foot fully cleared. From the top pane (Fig. 3.4) it can be observed that $\gamma_* \approx 0.46$.

The middle pane (Fig. 3.4) shows the energy consumed by the robot in a complete hopping cycle, varying with γ. Note that γ_* does not yield the least-energy gait. The

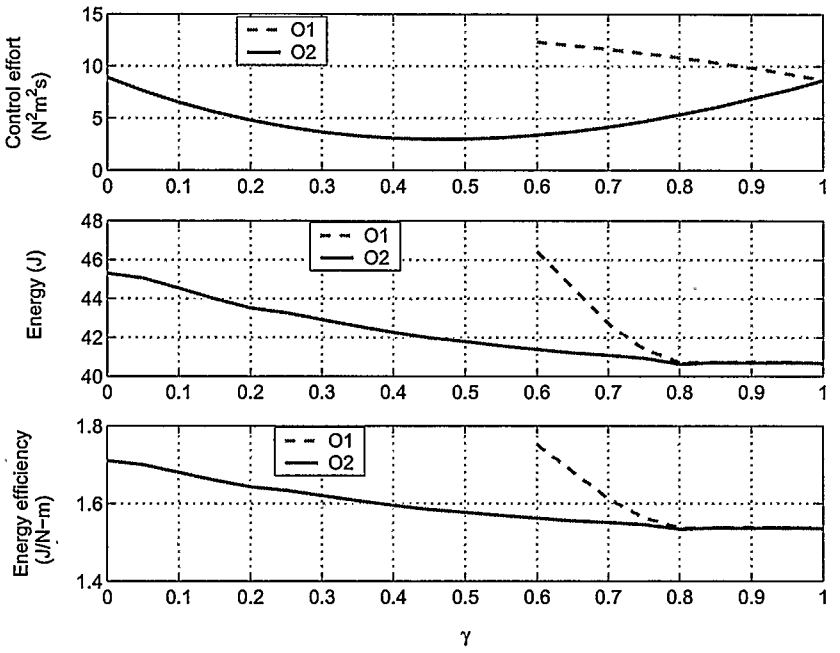

Figure 3.4: Performance vs. γ

efficiency index, *i.e.* the cost of transport ϵ, is presented in the bottom pane. Since the denominator in (3.53) is constant, ϵ and the consumed energy varies with γ in the same way. The efficiency index is around 1.6. Assuming that the transmission efficiency is 50% as in [26], the simulated hopper seems more efficient than the biped HRP-2L, but less than humans and the ARL Monopods.

Synthesized hopping gaits

When $\gamma = 0.8$, both (3.31) and (3.34) can generate efficient hopping cycles (Fig. 3.4). Figs. 3.5 and 3.6 show the two generated gaits, each with 3 periodic hopping cycles. Figs. 3.7, 3.8, and 3.9 compare the joint angles, velocities and torques, corresponding to the use of the two objective functions. Solid curves display the results from using (3.34), and dashed curves from (3.31). It is interesting to observe

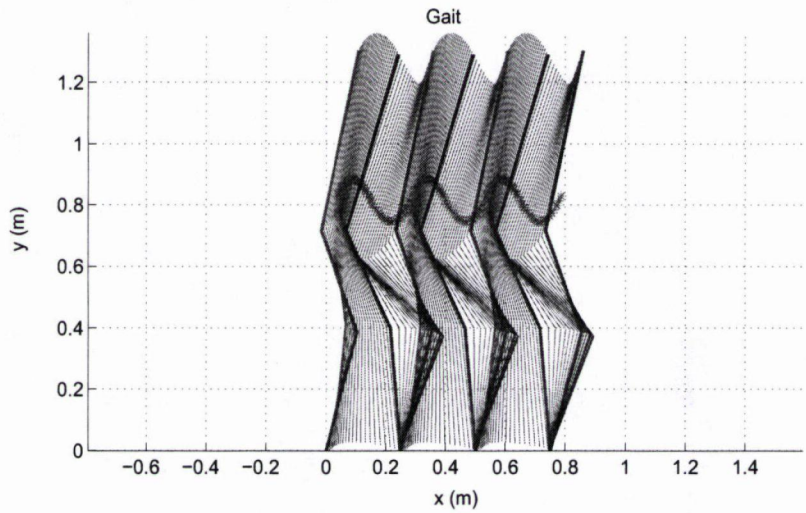

Figure 3.5: Hopping on even ground generated by use of (3.31)

that choice of the objective function mainly affects θ_1 and $\dot{\theta}_1$, while θ_2, $\dot{\theta}_2$, θ_3, and $\dot{\theta}_3$ remain nearly unchanged.

Correspondingly, the ZMP, the horizontal and vertical accelerations, and the normal GRF of the generated gaits are shown in Figs. 3.10 and 3.11, respectively. The neutral position of the ZMP, where $X_{\text{zmp}} = 0$, coincides with the ankle position in the inertial coordinate. Note also that the ZMP is not defined in the flight phase. Clearly, (3.34) yields a better ZMP trajectory.

Numerous simulations show that for the hopper model under study, (3.34) outperforms (3.31) in a wider range of γ, less control effort, and a better ZMP trajectory.

3.4.2 Hopping up stairs

As with hopping on even ground, this experiment uses $\gamma = 0.8$. Again, (3.31) and (3.34) generate the two periodic gaits for comparison. The simulations use the same parameters (Table 3.4). This time, all 5 steps of the 5-step procedure are applied.

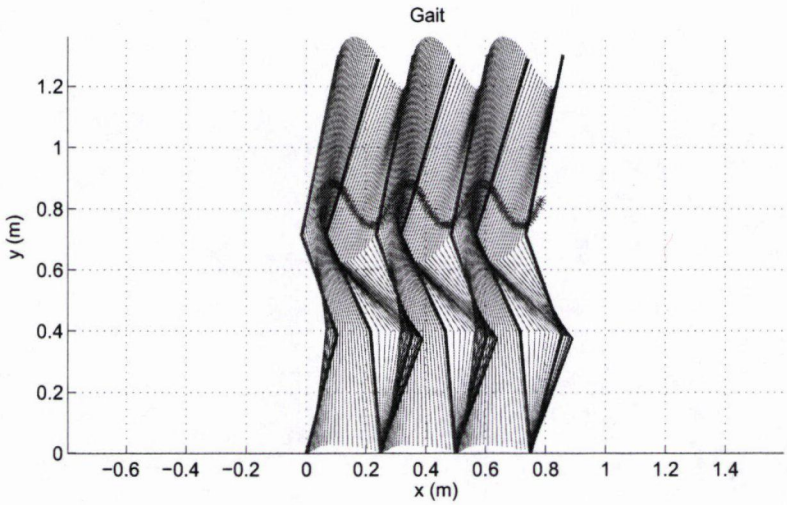

Figure 3.6: Hopping on even ground generated by use of (3.34)

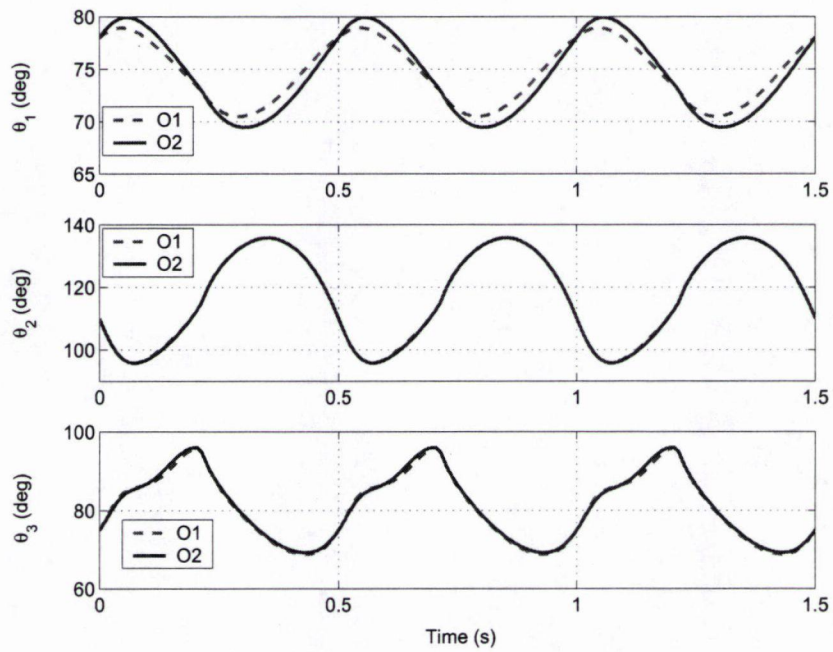

Figure 3.7: Comparison of joint angles

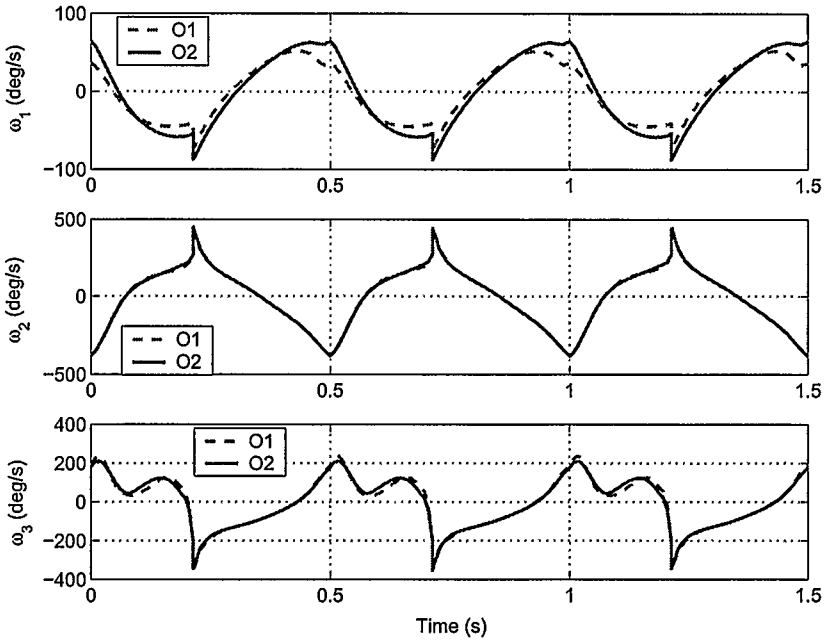

Figure 3.8: Comparison of joint velocities

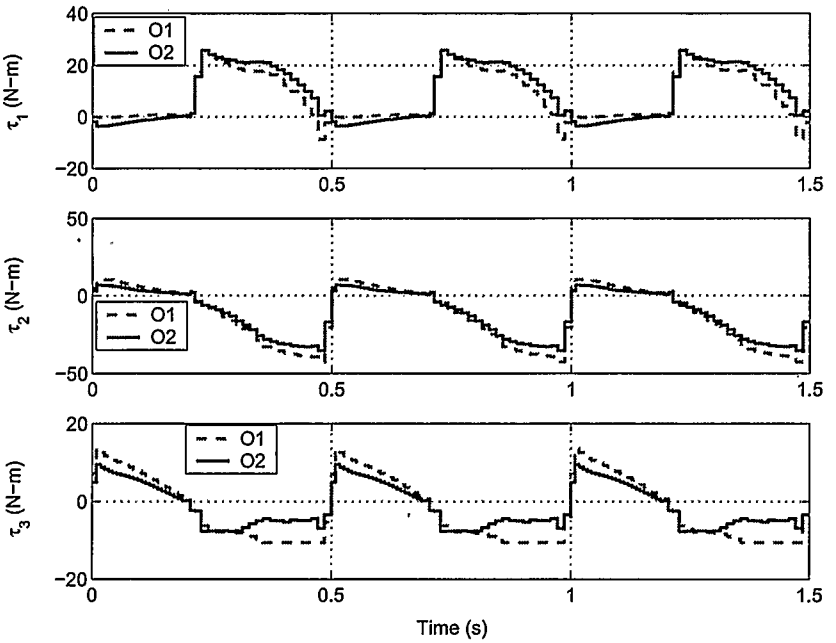

Figure 3.9: Comparison of joint torques

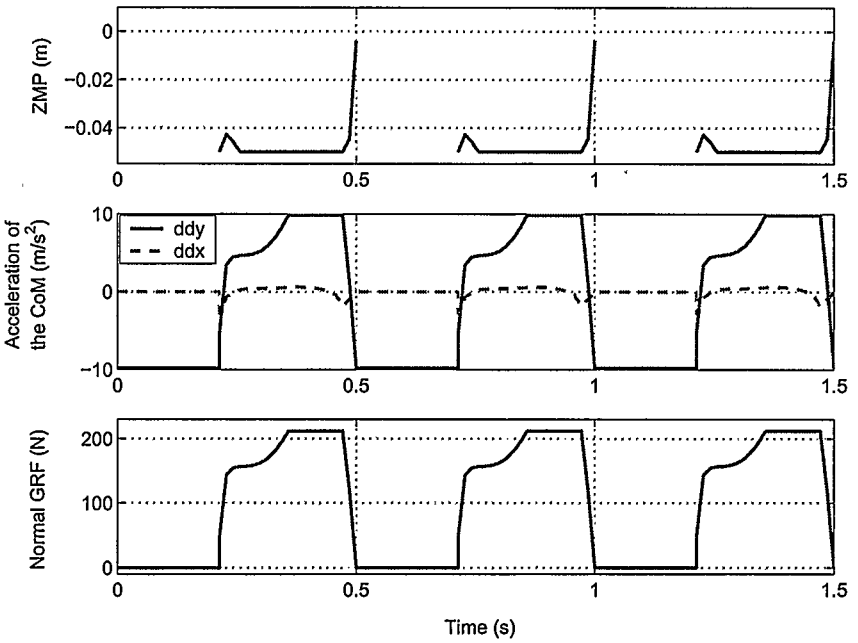

Figure 3.10: The ZMP, acceleration components of the CoM, and the normal GRF resulted from (3.31)

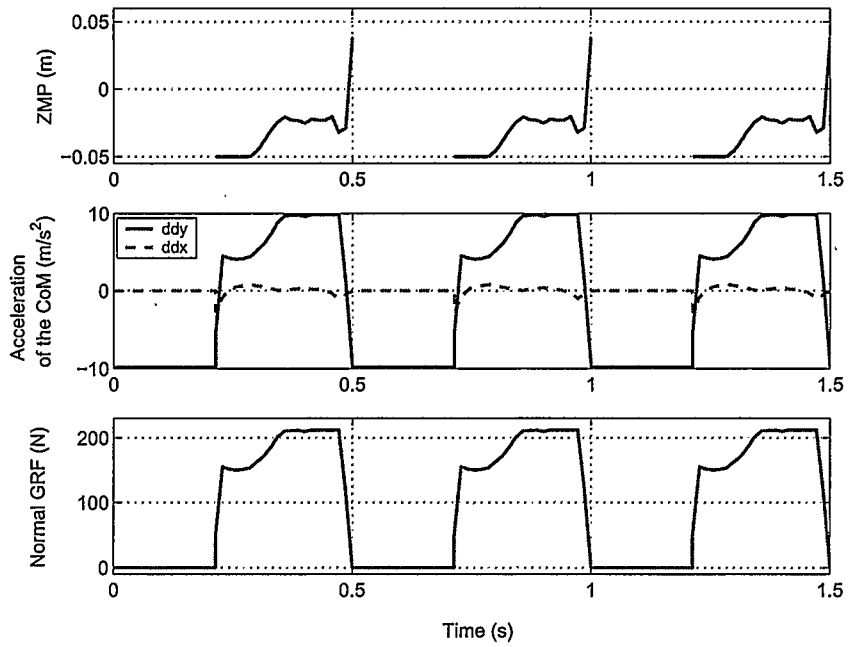

Figure 3.11: The ZMP, acceleration components of the CoM, and the normal GRF resulted from (3.34)

Table 3.4: The parameters for hopping up stairs

Parameters	Values	Units
L_s	0.25	m
h_s	0.08	m
V_x	0.5	m/s
Θ_{to}	$(73, 115, 70)^T$	deg
Θ_{td}	$(65, 133, 85)^T$	deg
μ	0.6	-
k_x	0.5	-
k_y	0.8	-
$F_{y,max}$	$2M_tg$	N

Figs. 3.12 and 3.13 present the gaits, corresponding to use of the two objective functions, respectively. Figs. 3.14, 3.15, and 3.16 show the comparisons of the resultant joint angles, velocities, and torques, respectively.

When (3.31) is used, the cost of transport is about $1.69JN^{-1}m^{-1}$. The flight phase and the stance phase consume energy, respectively of $7.34J$ and $37.45J$. The ratio of the consumed energy in the flight phase to that in the complete cycle is about 16.3%. As a comparison, when (3.34) is used, the cost of transport is about $1.58JN^{-1}m^{-1}$, the energy consumed in the two hopping phases are $5.28J$ for the flight phase, and $36.50J$ for the stance phase. The stance phase consumes about 13% of the total energy. Therefore, for this particular experiment, (3.34) performs better than (3.31).

In both cases, the ZMP trajectories remain within the foot range, and the normal GRFs are less than 2 times of the robot's weight.

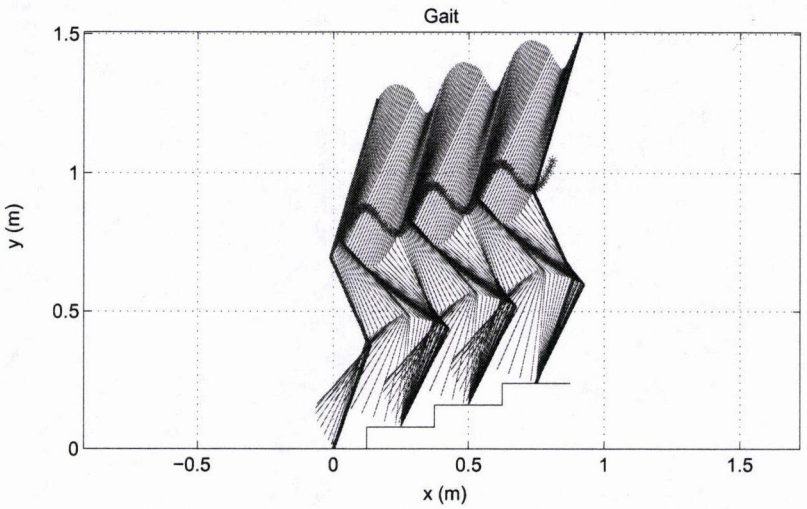

Figure 3.12: Hopping up stairs by using (3.31)

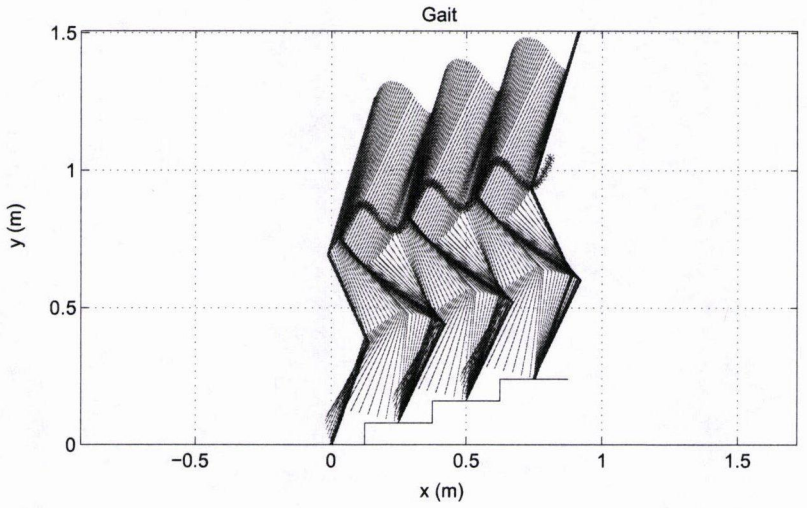

Figure 3.13: Hopping up stairs by using (3.34)

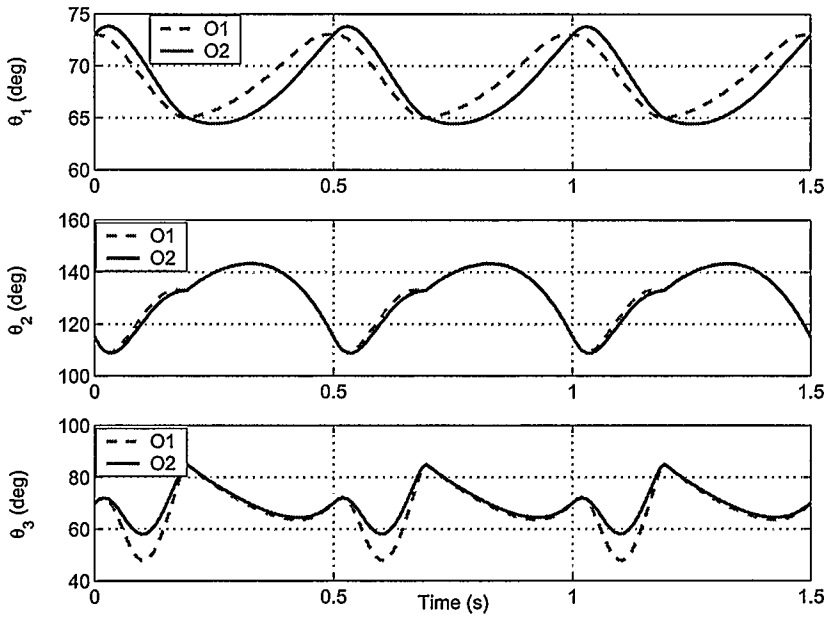

Figure 3.14: Comparison of joint angles when hopping up stairs

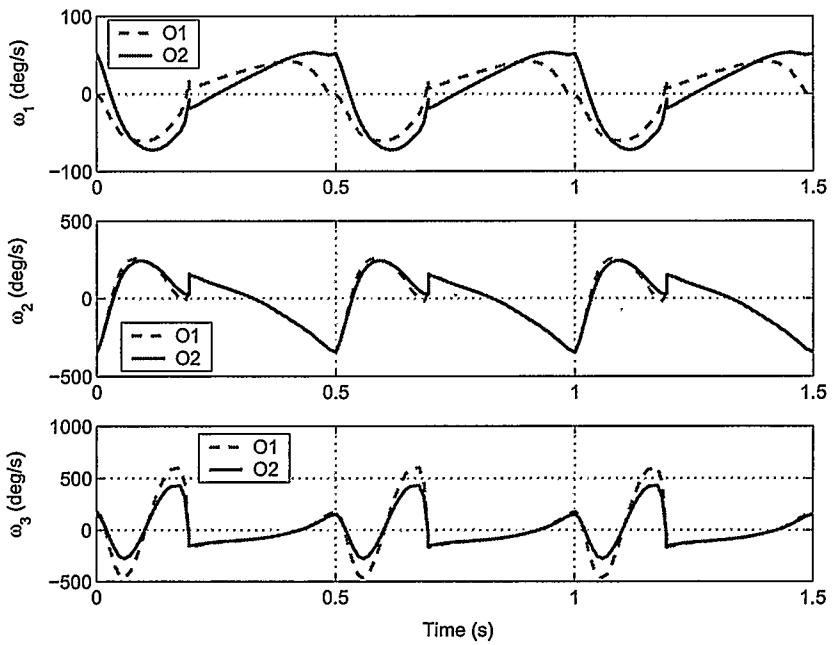

Figure 3.15: Comparison of joint velocities when hopping up stairs

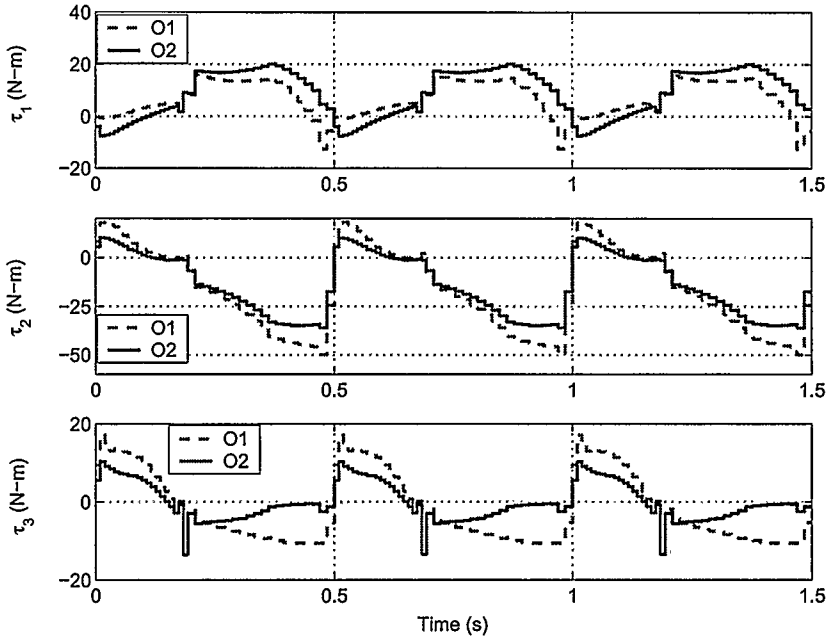

Figure 3.16: Comparison of joint torques when hopping up stairs

3.4.3 A complete hopping sequence

Fig. 3.17 presents a complete hopping sequence, containing a starting phase, an acceleration phase, two periodic cycles, a deceleration phase, and a stopping phase. The corresponding joint angles, velocities, and torques appear in Figs. 3.18, 3.19, and 3.20. All joint torques are within the limits listed in Table 3.2. The ZMP, acceleration of the CoM, and the normal GRF are presented in Fig. 3.21. The ZMP at the beginning of the sequence is 0, and at the end of the sequence, it returns to 0. The acceleration components of the robot's CoM also start from 0, and stop at 0.

3.4.4 Some remarks

The following remarks may help to understand synthesis of running-like gaits:

Remark 1) Note that the joint angles are defined with respect to the horizontal

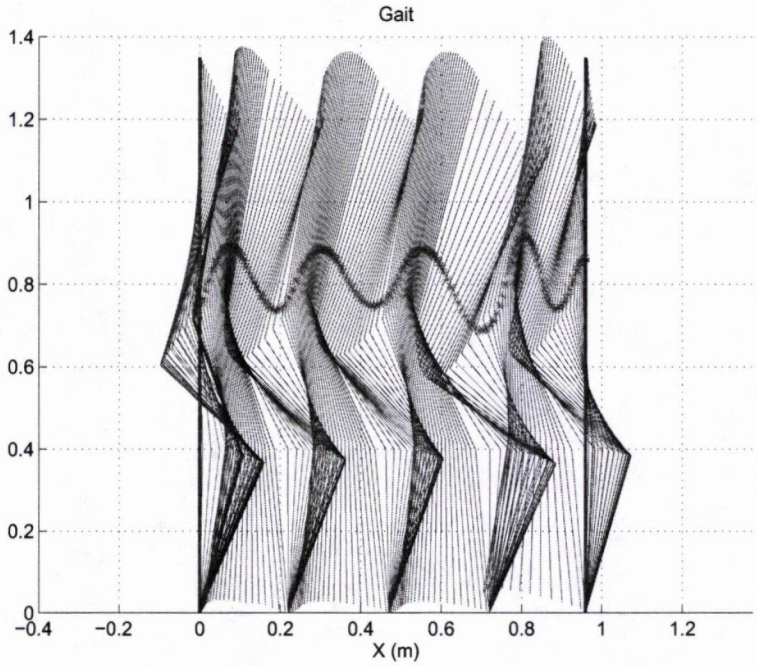

Figure 3.17: The complete hopping sequence

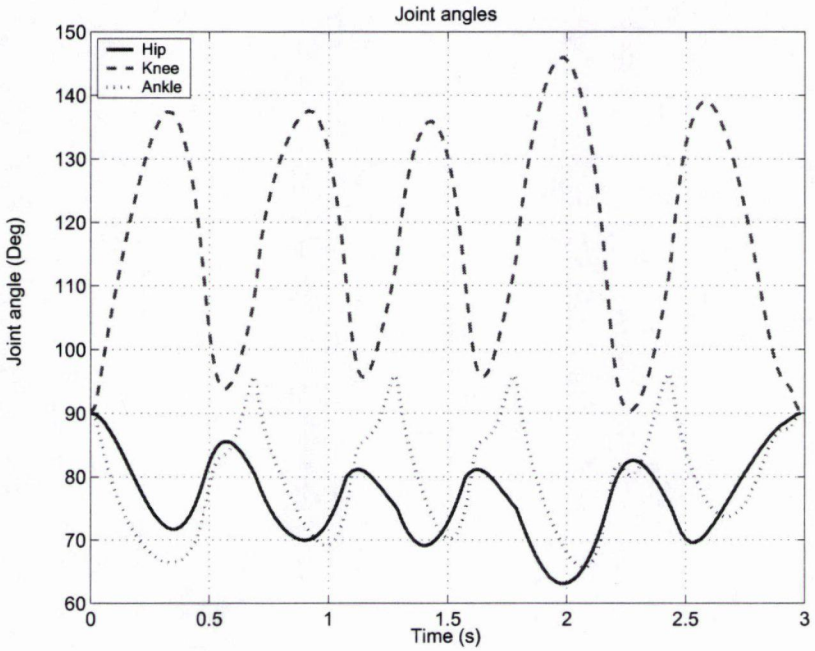

Figure 3.18: The joint angles of the complete sequence

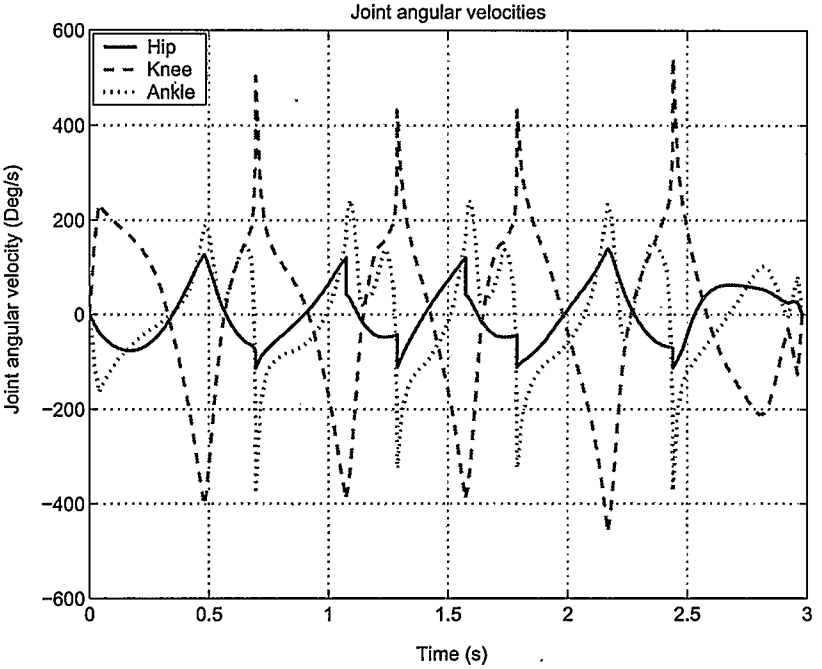

Figure 3.19: The joint velocities of the complete sequence

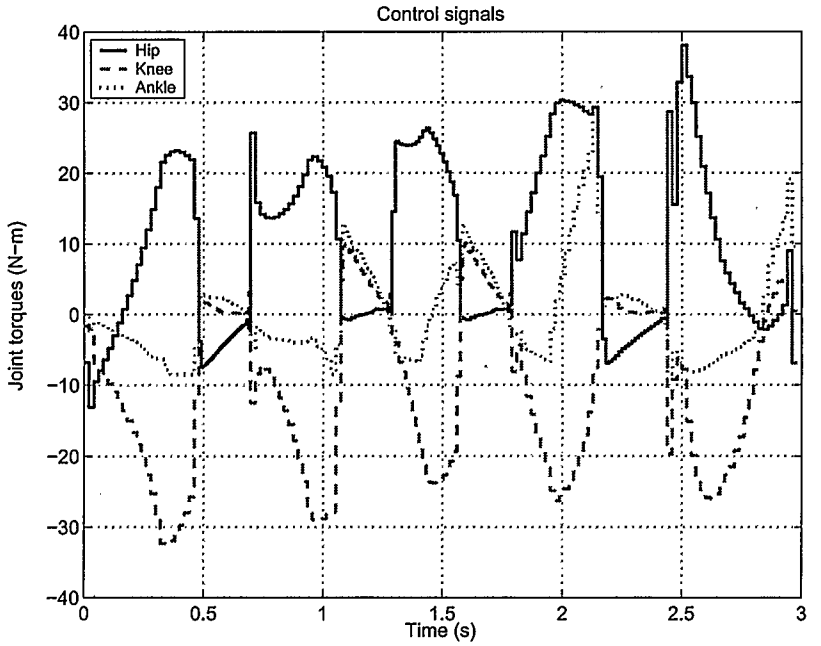

Figure 3.20: The joint torques of the complete sequence

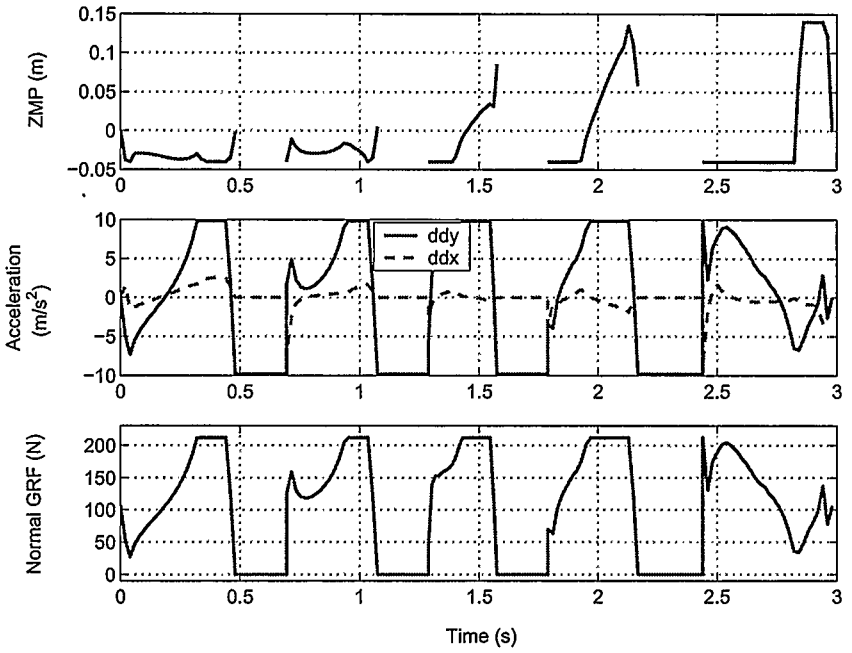

Figure 3.21: The ZMP, acceleration of the robot's robot, and the normal GRF of the complete hopping sequence

level, rather than the relative angles between two neighboring links. This definition greatly simplifies the model of the robot, and hence speeds up the computation. Also, it is easier to design constraints for the dynamic gait optimization such that the overturning in the flight phase can be avoided. However, in reality, it is very hard to measure the absolute angles accurately. One possible solution to this issue is to install the posture sensors, such as gyroscopes or inclinometers, on the upper body of the robot. In general, the mass and inertia of the upper body are much larger than those of other links. As we have seen, during the whole hopping cycle, the change range of the upper body is small, compared to those of the other links. Therefore, the orientation of the upper body can be used as a baseline of the posture, and other absolute angles used in our model can easily be recovered by using simple

linear transformations. Another possible solution to this problem is to design online observers to estimate the absolute joint angles, making use of the relative angle measurements. [76] may provide more useful details on this topic. This problem is left as a future research project.

Remark 2) To improve the energy efficiency, most weight, as well as the inertia, of the robot should be lumped at the upper body. It is recommend that the ratio of the weight of the upper body to that of the rest parts, denoted by λ, be greater than 2.5. Larger λ implies smaller cost of transport, and hence higher efficiency. This fact has been revealed for traditional passive walkers in [47].

Remark 3) It may be conservative to use the ZMP as constraints for synthesis of running-like motion (see [151], [71]). Without these strict constraints, it is easier to find the solutions. Despite the controversy on the ZMP stability criterion in running synthesis and control, use of the ZMP constraints indeed makes the generated joint trajectories, especially the joint accelerations, more reasonable. Moreover, the ZMP criterion can be treated as a weak gait stability indicator. When the ZMP is located outside of the support polygon, the robot may not fall, but the gait is less stable than when the ZMP is within the support polygon. Therefore, one goal of running synthesis is to force the ZMP as close as possible to the center of the support range.

3.5 Summary

This chapter presented an original method for producing an energy-efficient gait for a single-leg, three-joint hopper. The algorithm accepts three desired parameters: the hopping distance, the average forward velocity, and the landing (stair) height.

Predetermined joint angles at the two ends of the flight phase constrain the solution. A static optimization procedure, which assumes small energy consumption during the flight phase, produces the initial joint velocities. Dynamic optimization, with constraints, first synthesizes gaits for the flight phase and the stance phase separately. Constraints include physical joint limitations, ground reaction forces, and the ZMP trajectory. An additional procedure can optimize the energy over the entire cycle. In simulation, the robot hops on level ground and up stairs. A novel method for producing the initial state vector allows us to produce these different solutions by simply choosing appropriate parameters in the initialization stage. These techniques provide an appropriate foundation for achieving efficient bipedal running gaits in the future.

Chapter 4

Running Synthesis for a Biped with Rigid Articulated Legs

In this chapter, the techniques applied to the one-legged hopping synthesis, explained in the previous chapter, are to be extended to the bipedal running synthesis. Formulations for the two cases are quite similar, and hence, some of the contents in this chapter overlap with those presented in Chapter 3. To be self-contained, all overlapped contents are to be presented again with less details. Since the two legs of the biped play different roles, a switching matrix is to be used. Also, the clearance of the swing leg during the stance phase has to be treated. Section 4.1 explains the modeling of the biped. Section 4.2 implements the dynamic optimization of the bipedal running gaits. The simulation results are demonstrated in Section 4.3, followed by the concluding comments in Section 4.4.

4.1 Modeling of the biped

The planar biped studied here consists of 7 links and 6 frictionless pin joints: 2 hips, 2 knees, and 2 ankles. The masses, inertia, and heights of the two feet are assumed to be very small and hence negligible. The other 5 links have length and masses l_i and M_{i} ($\mathrm{i} = 1, \cdots, 5$), as specified in Fig. 4.1. The total mass is $M_{\mathrm{t}} = \sum_{i=1}^{5} M_{\mathrm{i}}$. Each moment of inertia I_{i} is taken about the CoM of the i^{th} massive link. Each r_{i} is a ratio of the CoM location of the massive link to the link lengths. Fig. 4.1

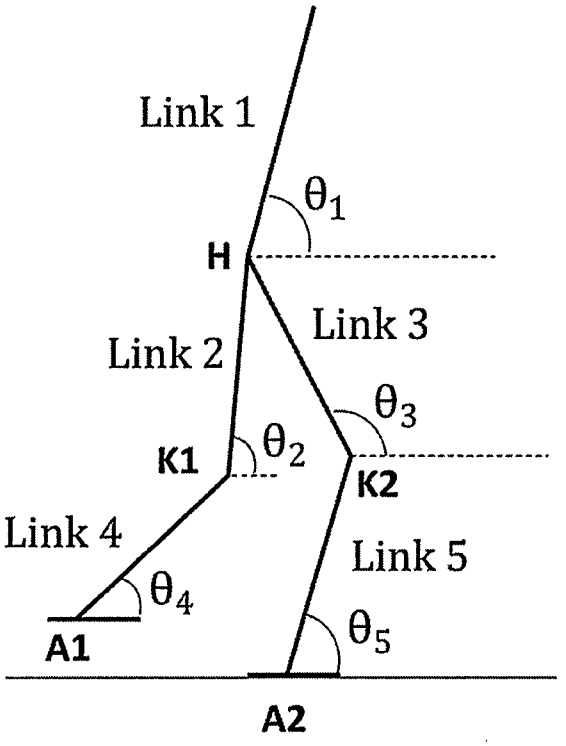

Figure 4.1: The model of the planar biped

illustrates the biped model.

Two assumptions are made as follows:

1. During the flight phase, the feet are parallel with the ground;

2. During the stance phase, the support foot grips the ground firmly, without bounce, slipping, or rotation.

The CoM and the two ankles, labeled by A_1 and A_2 in Fig. 4.1, are related by

$$P_g = P_{a1} + f_1(\Theta), \tag{4.1}$$

$$= P_{a2} + f_2(\Theta), \tag{4.2}$$

where $P_g = (X_g, Y_g)^T$, $P_{a1} = (X_{a1}, Y_{a1})^T$, and $P_{a2} = (X_{a2}, Y_{a2})^T$ are the positions of the CoM and the two ankles, respectively, $f_1(\cdot)$ and $f_2(\cdot)$ are two functions determined by kinematics, and Θ is defined to be $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^T$. Differentiating (4.1) and (4.2) respectively once gives

$$\dot{P}_g = \dot{P}_{a1} + \frac{\partial f_1}{\partial \Theta}\dot{\Theta}, \tag{4.3}$$

$$= \dot{P}_{a2} + \frac{\partial f_2}{\partial \Theta}\dot{\Theta}, \tag{4.4}$$

and twice gives

$$\ddot{P}_g = \ddot{P}_{a1} + \frac{d}{dt}\left(\frac{\partial f_1}{\partial \Theta}\right)\dot{\Theta} + \frac{\partial f_1}{\partial \Theta}\ddot{\Theta}, \tag{4.5}$$

$$= \ddot{P}_{a2} + \frac{d}{dt}\left(\frac{\partial f_2}{\partial \Theta}\right)\dot{\Theta} + \frac{\partial f_2}{\partial \Theta}\ddot{\Theta}. \tag{4.6}$$

In the stance phase, gravity and GRFs act on the robot. By contrast, during the flight phase gravity is the only external force. When completing the flight phase, the robot foot collides with the ground with a certain velocity. This collision causes an abrupt velocity jump at each joint. The flight phase will be explained first, followed by the velocity jump due to the collision between the support foot and the ground , and finally the support phase is discussed.

4.1.1 The flight phase

During the flight phase, the robot has 7 DOFs. The generalized coordinate vector, $q^f = (P_g^T, \Theta^T)^T$, describes the robot's posture in the air.

Equations of motion

The equations of motion (EoMs) during the flight phase can be derived as (see B.1)

$$D^f(q^f)\ddot{q}^f + H^f(q^f, \dot{q}^f)\dot{q}^f + G^f(q^f) = B^f\tau^f, \tag{4.7}$$

where the superscript "f" stands for the flight phase, $D^{\mathrm{f}}(q^{\mathrm{f}}) \in \Re^{7\times 7}$ is the inertia matrix, $H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}}) \in \Re^{7\times 7}$ contains the centrifugal and Coriolis terms , and $G^{\mathrm{f}}(q^{\mathrm{f}}) \in \Re^{7}$ is the gravitational torque vector. The vector $\tau^{\mathrm{f}} = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6)^{\mathrm{T}}$ contains the torques of the hips, the knees, and the ankles. The term $B^{\mathrm{f}} \in \Re^{7\times 6}$ is the constant coefficient matrix of τ^{f} and can be determined by using the virtual work principle, dependent on the chosen generalized coordinates. The superscript "f" will be dropped in the later part of this subsection where no ambiguity is introduced.

The generalized coordinates can be partitioned into two parts: $q = (q_1^{\mathrm{T}}, q_2^{\mathrm{T}})^{\mathrm{T}}$, where $q_1 = P_{\mathrm{g}}$, and $q_2 = \Theta$. Partitioning the inertia matrix $D(q)$ produces

$$D(q) = \begin{bmatrix} D_1 & 0_{2\times 5} \\ 0_{5\times 2} & D_2(q_2) \end{bmatrix}, \tag{4.8}$$

where $D_1 = M_{\mathrm{t}} I_{2\times 2}$, and $D_2(q_2) \in \Re^{5\times 5}$ contains the moment of inertia related to the rotation of the robot's links, $I_{n\times n}$ represents an $n \times n$ identity matrix. Accordingly, the matrices $H(q, \dot{q})$ and B can respectively be partitioned as

$$H(q, \dot{q}) = \begin{bmatrix} 0_{2\times 2} & 0_{2\times 5} \\ 0_{5\times 2} & H_2(q_2, \dot{q}_2) \end{bmatrix}, \tag{4.9}$$

and

$$B = \begin{bmatrix} 0_{2\times 6} \\ B_2 \end{bmatrix}, \tag{4.10}$$

where $H_2(q_2, \dot{q}_2) \in \Re^{5\times 5}$ and $B_2 \in \Re^{5\times 6}$ are sub-matrices of $H(q, \dot{q})$ and B, respectively. The gravity torque vector is $G(q) = (0, M_{\mathrm{t}} g, 0, 0, 0, 0, 0)^{\mathrm{T}}$, with g being the gravity acceleration. The last 5 rows of (4.7) are:

$$D_2(q_2)\ddot{q}_2 + H_2(q_2, \dot{q}_2)\dot{q}_2 = B_2\tau. \tag{4.11}$$

Simplifying the first two rows of the EoMs (4.7) yields

$$\ddot{X}_{\mathrm{g}} = 0, \tag{4.12}$$

$$\ddot{Y}_{\mathrm{g}} = -g, \tag{4.13}$$

which defines the acceleration of the robot's CoM and results in a parabolic trajectory.

Denote the time instants at take-off and touchdown by subscripts "to" and "td", the stride length and the landing height by L_{s} and h_{s}, respectively. Assuming that the position of the right ankle at the right foot take-off is $P_{\mathrm{a1,to}} = (0,0)^{\mathrm{T}}$, then the position of the left ankle at the left leg touchdown is $P_{\mathrm{a2,td}} = (L_{\mathrm{s}}, h_{\mathrm{s}})^{\mathrm{T}}$. Suppose that Θ_{to} and Θ_{td} have been picked in advance, $P_{\mathrm{g,to}}$ and $P_{\mathrm{g,td}}$ can be found by (4.1) and (4.2), respectively.

In this work, the robot is commanded to run in a constant average forward speed, V_{x}. Hence, the horizontal velocity $\dot{X}_{\mathrm{g,to}} = \dot{X}_{\mathrm{g,td}} = V_{\mathrm{x}}$. The flight lasts for the duration

$$T^{\mathrm{f}} = \frac{X_{\mathrm{g,td}} - X_{\mathrm{g,to}}}{V_{\mathrm{x}}}. \tag{4.14}$$

The initial and final vertical velocity of the flight phase are

$$\dot{Y}_{\mathrm{g,to}} = \frac{Y_{\mathrm{g,td}} - Y_{\mathrm{g,to}} + \frac{1}{2} g \left(T^{\mathrm{f}}\right)^2}{T^{\mathrm{f}}}, \tag{4.15}$$

and

$$\dot{Y}_{\mathrm{g,td}} = \dot{Y}_{\mathrm{g,to}} - g T^{\mathrm{f}}. \tag{4.16}$$

Angular momentum about the CoM

In the flight phase gravity acts as the only external force at the CoM, implying that the angular momentum of the robot about the CoM is conserved. Therefore,

$$\Gamma_{\mathrm{g,to}}(\Theta_{\mathrm{to}}, \dot{\Theta}_{\mathrm{to}}) = \Gamma_{\mathrm{g,td}}(\Theta_{\mathrm{td}}, \dot{\Theta}_{\mathrm{td}}), \tag{4.17}$$

where Γ_{g} stands for the angular momentum about the CoM, and it can be expressed as

$$\Gamma_{\mathrm{g}} = \sum_{\mathrm{i}=1}^{5} s_{\mathrm{i}} \dot{\theta}_{\mathrm{i}}, \tag{4.18}$$

and $s_{\mathrm{i}} = \sum_{\mathrm{j}=3}^{7} d_{\mathrm{i}+2,\mathrm{j}}$ for $\mathrm{i} = 1, \cdots, 5$, with each $d_{i,j}$ being an element of $D(q)$ indexed by i, j.

Energy analysis

As what has been analyzed in Chapter 3, the total energy of the robot can be decomposed into three parts: translational kinetic energy, rotational kinetic energy, and potential energy. The total translational kinetic energy plus potential energy is conserved in the flight phase. The rotational kinetic energy of the robot in the flight phase is

$$\Phi_{\mathrm{rot}}(\Theta, \dot{\Theta}) = \frac{1}{2} \dot{q}_2^{\mathrm{T}} D_2(q_2) \dot{q}_2, \tag{4.19}$$

which depends on the joint torques. Making use of the skew-symmetric property of $\dot{D}_2 - 2H_2$ (see [137]), it can be shown that $\frac{d(\Phi_{\mathrm{rot}})}{dt} = \dot{q}_2^{\mathrm{T}} B_2 \tau$. During the flight phase, τ is usually not 0, and thus $\frac{d(\Phi_{\mathrm{rot}})}{dt} \neq 0$. Thus, the rotational kinetic energy is not conserved.

By the assumption (A1), the flight phase consumes much less energy that the stance phase does. By the same arguments presented in Chapter 3, this understand-

ing leads to $\Phi_{\text{rot,to}} \approx \Phi_{\text{rot,td}}$ (*i.e.* $\Phi_{\text{rot,to}} - \Phi_{\text{rot,td}} \approx 0$), and an objective function

$$\mathcal{O}_1 = \gamma\Phi_{\text{rot,to}} + (1-\gamma)\left(\Phi_{\text{rot,to}} - \Phi_{\text{rot,td}}\right)^2, \tag{4.20}$$

where the constant $\gamma \in [0,1]$ is a weighting factor. Note that $\mathcal{O}_1$ is a function of $\dot{\Theta}_{\text{to}}$ and $\dot{\Theta}_{\text{td}}$, provided that Θ_{to} and Θ_{td} have been chosen in advance. Minimization of the first term on the RHS of (4.20) gives rise to a minimal take-off rotational kinetic energy that guarantees the required running motion defined by the three commanded parameters L_{s}, h_{s}, and V_{x}. Minimization of the second term on the RHS of (4.20) is inspired by the assumption (A1). In real running motion, $|\Phi_{\text{rot,to}} - \Phi_{\text{rot,td}}|$ is usually a small positive quantity. Since $\Phi_{\text{rot,td}} \leq \Phi_{\text{rot,to}} + |\Phi_{\text{rot,td}} - \Phi_{\text{rot,to}}|$, in the case that $\Phi_{\text{rot,to}}$ has been minimized, the minimization of $(\Phi_{\text{rot,to}} - \Phi_{\text{rot,td}})^2$ means that $\Phi_{\text{rot,td}}$ is also minimized (see Chapter 3 for detailed derivation). Thus, a second, more concise, objective function arises

$$\mathcal{O}_2 = \gamma\Phi_{\text{rot,to}} + (1-\gamma)\Phi_{\text{rot,td}}. \tag{4.21}$$

Similar to the hopping synthesis reported in the previous chapter, (4.21) outperforms (4.20) in two aspects. It results in more efficient gaits, and also, γ can be chosen in a larger range. In later sections of this chapter, only (4.21) is applied.

Boundary joint velocities

The boundary joint angles are picked manually, but the boundary joint velocities have to be determined. The problem can be stated as:

Solve for $\dot{\Theta}_{\text{to}}$ and $\dot{\Theta}_{\text{td}}$, such that the objective function (4.21) is minimized. The

equality constraints could be

$$\begin{aligned} \dot{P}_{\mathrm{g,to}} &= [V_{\mathrm{x}}, \dot{Y}_{\mathrm{g,to}}]^{\mathrm{T}}, \\ \dot{P}_{\mathrm{g,td}} &= [V_{\mathrm{x}}, \dot{Y}_{\mathrm{g,td}}]^{\mathrm{T}}, \\ \Gamma_{\mathrm{g,to}} &= \Gamma_{\mathrm{g,td}}. \end{aligned}$$

Inequality constraints limit the motion. The angular velocities of the joints are bounded by

$$\omega_{\mathrm{i,min}} \leq \omega_{\mathrm{i}} \leq \omega_{\mathrm{i,max}}, \tag{4.22}$$

where ω_{i} for $\mathrm{i} = 1, \cdots, 5$ is the angular velocity at the i^{th} joint, $\omega_{\mathrm{i,min}}$ and $\omega_{\mathrm{i,max}}$ are the lower and upper bounds of the corresponding joint velocity.

A static optimization procedure, with constraints, then finds $\dot{\Theta}_{\mathrm{to}}$ and $\dot{\Theta}_{\mathrm{td}}$. $\dot{\Theta}_{\mathrm{to}}$ will be used as the initial value of flight phase, and $\dot{\Theta}_{\mathrm{td}}$ is discarded. The true value of $\dot{\Theta}_{\mathrm{td}}$ is searched by dynamic optimization. This treatment makes the entire algorithm more flexible.

Foot velocity regulation

Energy loss due to the collision between the foot and the ground is proportional to the front foot velocity immediately before the collision [145]. Although a front foot velocity of 0 at touch-down could be achieved, this would require a lot of extra control effort in current flight phase and the following stance phase. The front foot velocity immediately before the touchdown can be chosen such that

$$V_{\mathrm{a2,td}} = \begin{bmatrix} k_{\mathrm{x}} & 0 \\ 0 & k_{\mathrm{y}} \end{bmatrix} V_{\mathrm{g,td}}, \tag{4.23}$$

where the constants $k_x \in [0, 1]$ and $k_y \in [0, 1]$ are recommended to trade off collision energy loss and control effort. Choosing k_x smaller increases forward velocity and *vice versa*. Although large k_x and k_y may result in significant energy loss at touch-down, it may be worth the decreased control effort overall.

4.1.2 Collision between the foot and the ground

By assumption, the foot collides with the ground instantaneously and inelastically at touch-down. The joint angles change continuously but the joint velocities jump discontinuously. Equation (4.2) can be rearranged into

$$P_{a2} = f(q^f) := P_g - f_2(\Theta).$$

A Jacobian matrix J is defined as

$$J = \frac{\partial f}{\partial q^f}. \tag{4.24}$$

Using superscripts "−" and "+" for states immediately before and after collision, respectively, the collision model in [34] predicts discrete velocity changes

$$\dot{q}^+ = \left(I_{7\times 7} - \left(D^f\right)^{-1} J^T \left(J \left(D^f\right)^{-1} J^T \right)^{-1} J \right) \dot{q}^-, \tag{4.25}$$

where $D^f = D^f(q)$ and $J = J(q)$ are evaluated at the touchdown when $q = q_{td}$ and $\dot{q}^- = \dot{q}_{td}$. The velocity of the CoM immediately after the collision, $\dot{P}^+_{g,td}$, follows directly from (4.4).

4.1.3 The stance phase

Assuming the foot remains firmly in contact with the ground during the stance phase, the robot moves with five DOFs. The superscript "s" indicates the variables used in the stance phase.

The stance dynamics

Defining the stance-phase generalized coordinates as $q^{\mathrm{s}} = \Theta$ results in the dynamic model (B.6):

$$D^{\mathrm{s}}(q^{\mathrm{s}})\ddot{q}^{\mathrm{s}} + H^{\mathrm{s}}(q^{\mathrm{s}}, \dot{q}^{\mathrm{s}})\dot{q}^{\mathrm{s}} + G^{\mathrm{s}}(q^{\mathrm{s}}) = B^{\mathrm{s}}\tau^{\mathrm{s}}, \tag{4.26}$$

where $D^{\mathrm{s}} \in \Re^{5\times5}$ is the inertia matrix, $H^{\mathrm{s}} \in \Re^{5\times5}$ contains centrifugal and Coriolis terms, and $G^{\mathrm{s}} \in \Re^{5}$ is the gravitational torque vector. The matrix $B^{\mathrm{s}} \in \Re^{5\times6}$ is the coefficient matrix and depends on the choice of the generalized coordinates.

Constraints

In the stance phase, the time duration, boundary joint velocities, GRFs, and the ZMP have to be determined or bounded. The GRFs include the normal support force of the ground, and the tangential friction force between the support foot and the ground. Assuming the robot moves at commanded velocity V_{x} the stance phase lasts for time

$$T^{\mathrm{s}} = \frac{X^{\mathrm{i+1}}_{\mathrm{g,to}} - X^{\mathrm{i}}_{\mathrm{g,td}}}{V_{\mathrm{x}}}, \tag{4.27}$$

where the superscript "i" means the i^{th} half running cycle, and $X^{\mathrm{i+1}}_{\mathrm{g,to}} - X^{\mathrm{i}}_{\mathrm{g,td}}$ is the horizontal distance traveled by the robot's CoM in the stance phase.

Since the boundary states are known in advance, the stance phase is a typical TPBVP. The boundary angles are chosen manually, the joint velocities are solved by the static optimization procedure, and the joint velocities just before touch-down are searched by the dynamic optimization of the flight phase. Thus, in the stance phase, the initial velocities follow from (4.25) and the final velocities are the velocities at the next take-off.

The support force from the ground is

$$F_{\mathrm{y}} = M_{\mathrm{t}}(g + \ddot{Y}^{\mathrm{s}}_{\mathrm{g}}) \geq 0, \tag{4.28}$$

and must point upward, implying

$$\ddot{Y}^{\mathrm{s}}_{\mathrm{g}} \geq -g. \tag{4.29}$$

Assuming the touch-down collision is instantaneous implies the resultant normal touch-down GRF may be extremely large. To prevent the robot from damage, the maximum vertical GRF must be limited. Namely,

$$F_{\mathrm{y}} \leq F_{\mathrm{y,max}}. \tag{4.30}$$

At the end of the stance phase, the normal GRF vanishes, implying $F^{\mathrm{s}}_{\mathrm{y,final}} = 0$, where the subscript "final" indicates the end of the stance phase. This can further be expressed as

$$\ddot{Y}^{\mathrm{s}}_{\mathrm{g,final}} = -g. \tag{4.31}$$

Assuming the robot does not slip, the horizontal inertia force of the robot is less than the static friction at the foot,

$$|\ddot{X}^{\mathrm{s}}_{\mathrm{g}}| \leq \frac{\mu F_{\mathrm{y}}}{M_{\mathrm{t}}}, \tag{4.32}$$

where μ is the friction coefficient between the support foot and the ground.

The ZMP

Assuming the support foot does not rotate during the stance phase, the ZMP should be located within the range of the support foot. The ZMP for the planar biped

follows from standard arguments [111]

$$X_{\text{zmp}} = \frac{\sum_{\text{i}=1}^{5} M_{\text{i}} \left[X_{\text{g,i}}^{\text{s}} \left(\ddot{Y}_{\text{g,i}}^{\text{s}} + g \right) - Y_{\text{g,i}}^{\text{s}} \ddot{X}_{\text{g,i}}^{\text{s}} \right] + \sum_{\text{i}=1}^{5} I_{\text{i}} \ddot{\theta}_{\text{i}}}{M_{\text{t}} \left(\ddot{Y}_{\text{g}}^{\text{s}} + g \right)}, \tag{4.33}$$

with the restriction that

$$-l_{\text{f2}} \leq X_{\text{zmp}} \leq l_{\text{f1}}, \tag{4.34}$$

where l_{f1} and l_{f2} are the front and rear lengths of the support foot, as measured from the ankle pivot point.

4.2 Optimization of bipedal running

The running gaits can be synthesized by using dynamic optimization. The EoMs (4.7) and (4.26) are highly nonlinear, requiring numerical methods. The Matlab function, $fmincon(\cdot)$, is chosen again as the optimization solver.

The two running phases can be optimized separately, or together as a half running cycle. A five step optimization procedure is used in this work.

1. Pick Θ_{to} and Θ_{td}.
2. Search for the initial joint velocities of the flight phase $\dot{\Theta}_{\text{to}}$, given L_{s}, h_{s}, V_x, and γ.
3. Optimize the flight phase.
4. Optimize the stance phase.
5. Re-optimize the generated flight phase and stance phase together as a half running cycle.

4.2.1 Single-phase optimization

Single phase optimization proceeds on the two running phases separately. In general, a nonlinear optimization solver requires a decision vector, boundary values of the decision vector, a cost function and some constraints (see [10] for more details). The decision vector, denoted by U, contains the discretized joint torques τ_{i} for $\mathrm{i} = 1, 2, \cdots, 6$. The modeling method, the decision vector, and the cost function are almost identical for the two phases. However, the constraints in different phases may be very different.

Implementation

The computer model of second-order dynamic models (4.7) and (4.26) consists of Matlab Simulink blocks. As a matter of practical implementation, a saturation block limits the output of term $H(q, \dot{q})\dot{q}$ to ensure the integrator does not fail. The definition of the decision vector is

$$U = \left[\tau(1)^{\mathrm{T}}, \tau(2)^{\mathrm{T}}, \cdots, \tau(k)^{\mathrm{T}}, \cdots, \tau(N)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{4.35}$$

where N is the number of discretized time intervals, k indicates the k^{th} discretized time interval, and $\tau(k) = [\tau_1(k), \tau_2(k), \tau_3(k), \tau_4(k), \tau_5(k), \tau_6(k)]^{\mathrm{T}}$. The initial estimate of U is U_0, a zero vector with the same dimension as U. The torque limits of the joints form the bounds of the decision vector,

$$\tau_{\mathrm{i,min}} \leq \tau_{\mathrm{i}} \leq \tau_{\mathrm{i,max}}. \tag{4.36}$$

Correspondingly, the decision vector is bounded by U_{min} and U_{max} which are appropriate formations of $\tau_{\mathrm{i,min}}$ and $\tau_{\mathrm{i,max}}$. The cost function in each running phase has

the same form

$$C = \frac{1}{2} U^{\mathrm{T}} U \Delta t, \tag{4.37}$$

where Δt is the time interval (assuming the phase duration, T^{f} or T^{s}, is equally discretized), and it may be different in the two phases.

Constraints in the flight phase

In the flight phase, the boundary joint angles, the initial joint velocities, and (4.23) are the equality constraints. The linear inequality constraints include all physical joint limits. Besides (4.22), the following constraints must be satisfied:

$$\Theta^{\mathrm{f}}_{\mathrm{i,min}} \leq \Theta^{\mathrm{f}}_{\mathrm{i}} \leq \Theta^{\mathrm{f}}_{\mathrm{i,max}},$$

$$\alpha^{\mathrm{f}}_{\mathrm{min}} \leq \alpha^{\mathrm{f}} \leq \alpha^{\mathrm{f}}_{\mathrm{max}},$$

where α is the relative angle of a knee.

Constraints in the stance phase

In the stance phase, the boundary joint angles, the boundary joint velocities, and (4.31) are the equality constraints. The linear inequality constraints could be written in the same way as those in the flight phase, with the superscript "f" being replaced by "s". In general, the values of the bounds in the stance phase may not be equal to their counterparts in the flight phase but in the implementation they are identical.

Constraints defined by (4.29),(4.30), (4.32), and (4.34) impose the most important nonlinear inequality constraints.

During the stance phase, the swing foot may scuff the ground, if its lowest height is not controlled. This leads to

$$Y_{\mathrm{a,sw}} \geq Y_{\mathrm{a,sw,min}}, \tag{4.38}$$

where the subscript "sw" indicates the swing leg. This is also considered as a non-linear inequality constraint.

4.2.2 Optimization of an entire half running cycle

The half-running-cycle optimization synthesizes the two running phases together. The cost function evaluating the control effort of the half cycle is

$$C_1 = \frac{1}{2}\left[\sum_{i=1}^{N^f}\left(U^f\right)^T\left(U^f\right)\Delta t^f + \sum_{i=N^f+1}^{N^f+N^s}\left(U^s\right)^T\left(U^s\right)\Delta t^s\right], \tag{4.39}$$

where N is the number of discretized time intervals. The flight and stance constraints remain the same, but are now included as one constraint function. The decision vector for the half-running-cycle optimization is $U = [(U^f)^T, (U^s)^T]^T$.

4.2.3 The second half cycle

The second half running cycle is simply a duplication of the first half cycle, with the leg label switched. This can be thought as change of the state values of the two legs. Namely,

$$\begin{aligned}\Theta_{to}^{i+1} &= S\Theta_{to}^{i},\\ \dot{\Theta}_{to}^{i+1} &= S\dot{\Theta}_{to}^{i},\end{aligned}$$

where the superscript "i" specifies the i^{th} half running cycle, S is the switching matrix, and

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4.40}$$

4.2.4 A performance index

In the flight phase, the robot does work

$$W^{\mathrm{f}} = \sum_{\mathrm{i}=0}^{N^{\mathrm{f}}-1} \int_{t_{\mathrm{to}}+i\Delta t^{\mathrm{f}}}^{t_{\mathrm{to}}+(i+1)\Delta t^{\mathrm{f}}} \sum_{\mathrm{j}=1}^{5} \left|\dot{q}^{\mathrm{f}}_{2,\mathrm{j}}\right| \left|\left(B^{\mathrm{f}}_{2}\tau^{\mathrm{f}}\right)_{\mathrm{j}}\right| dt. \tag{4.41}$$

where the subscript "j" indicates the angular velocity of the j^{th} joint, and "i" indicates the i^{th} discretized time interval. Note that each joint torque $\tau^{\mathrm{f}}_{\mathrm{j}}$ is a piecewise constant signal. The work in the stance phase is similar, but with "f" replaced by "s". The total work in a half running cycle is

$$W = W^{\mathrm{f}} + W^{\mathrm{s}}. \tag{4.42}$$

To compare the energy efficiency of the robot, the cost of transport is chosen again as the performance index

$$\epsilon = \frac{W}{M_t g L}, \tag{4.43}$$

where L is the distance traveled by the robot's CoM. A smaller cost of transport implies higher energy efficiency.

Table 4.1: The constants in the simulations

Constants	Values	Units	Constants	Values	Units
M_1	8	kg	I_1	0.4	kg-m^2
$M_{2/3}$	1	kg	$I_{2/3}$	0.02	kg-m^2
$M_{4/5}$	0.8	kg	$I_{4/5}$	0.01	kg-m^2
l_1	0.6	m	r_1	0.4	-
$l_{2/3}$	0.35	m	$r_{2/3}$	0.6	-
$l_{4/5}$	0.4	m	$r_{4/5}$	0.6	-
l_{f1}	0.15	m	l_{f2}	0.05	m
g	9.81	m/s^2	$Y_{\mathrm{a,sw,min}}$	0.05	m
$\theta_{1,\mathrm{min}}$	45	deg	$\theta_{1,\mathrm{max}}$	90	deg
$\theta_{2/3,\mathrm{min}}$	60	deg	$\theta_{2/3,\mathrm{max}}$	225	deg
$\theta_{4/5,\mathrm{min}}$	30	deg	$\theta_{4/5,\mathrm{max}}$	120	deg
α_{min}	0	deg	α_{max}	150	deg
$\dot{\theta}_{1,\mathrm{min}}$	-4	rad/s	$\dot{\theta}_{1,\mathrm{max}}$	4	rad/s
$\dot{\theta}_{2/3,\mathrm{min}}$	-9	rad/s	$\dot{\theta}_{2/3,\mathrm{max}}$	9	rad/s
$\dot{\theta}_{4/5,\mathrm{min}}$	-6	rad/s	$\dot{\theta}_{4/5,\mathrm{max}}$	6	rad/s
$\tau_{1/2,\mathrm{min}}$	-80	N-m	$\tau_{1/2,\mathrm{max}}$	80	N-m
$\tau_{3/4,\mathrm{min}}$	-40	N-m	$\tau_{3/4,\mathrm{max}}$	40	N-m
$\tau_{5/6,\mathrm{min}}$	-40	N-m	$\tau_{5/6,\mathrm{max}}$	40	N-m
μ	0.6	–	$F_{\mathrm{y,max}}$	$2M_{\mathrm{t}}g$	N

4.3 Simulation results and discussions

The simulated robot runs on even ground and up stairs. The simulation produces a complete running cycle (*i.e.* two half cycles) for each gait. The constants listed in Table 4.1 apply in all simulations. The robot achieves a desired step length L_{s}, landing height h_{s}, and average forward speed V_{x}.

4.3.1 Running on even ground

This simulation uses the parameters listed in Table 4.2 and generates a complete running cycle (see Fig. 4.2) using the 5-step optimization procedure. In the stance

Table 4.2: The parameters for running on even ground

Parameters	Values	Units
L_s	0.6	m
h_s	0	m
V_x	1.6	m/s
Θ_{to}	$(80, 100, 140, 70, 60)^T$	deg
Θ_{td}	$(80, 100, 115, 45, 100)^T$	deg
γ	0.3	-
k_x	0.5	-
k_y	0.5	-

phase, the trajectory of the CoM behaves like an elastic inverted pendulum, consistent with the SLIP model. At mid-stance the CoM arrives at its minimum height. The joint angles are continuous (Fig. 4.3) whereas the joint velocities jump due to the collision between the foot and the ground (Fig. 4.4). Immediately after the touchdown, some of the joint velocities, *e.g.* $\dot{\theta}_4$ and $\dot{\theta}_5$, may violate the allowed velocity range ($\pm 8 rad/s$, or $\pm 459 deg/s$) due to the overly-simplified collision model. In reality, the collisions should exhibit some elasticity resulting in smaller (within bound) velocity jumps. All joint torques fall within the corresponding ranges (Fig. 4.5).

The neutral position of the ZMP, where $X_{zmp} = 0$, coincides with the ankle positions in the inertial coordinate (Fig. 4.6 top). Note that the ZMP is not defined in the flight phase. Since the ZMP lies within the support foot, the robot will not tip over. The vertical acceleration of the CoM grows from $6.05 m/s^2$ to g, and then decreases to $-g$ (Fig. 4.6 middle). Thus, the normal GRF increases from $184N$ to $F_{y,max}$, then decreases dramatically before finally vanishing at the subsequent take-off (Fig. 4.6 bottom). Interestingly, the normal GRF appears in an M-shape

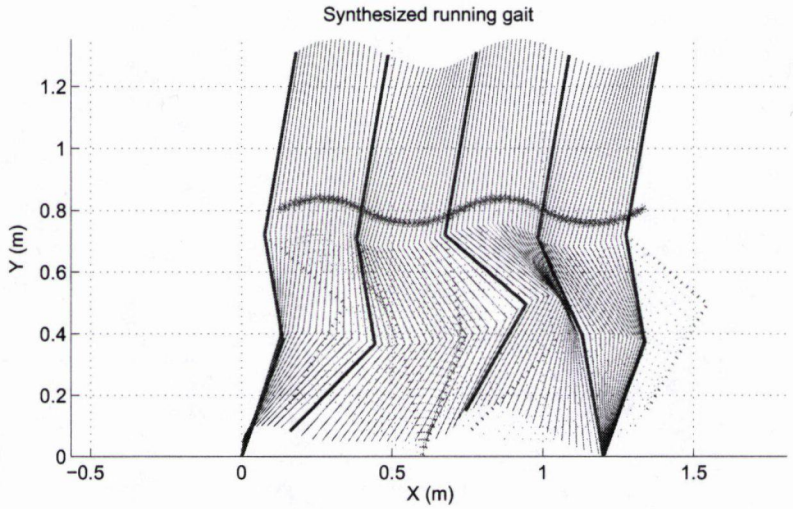

Figure 4.2: A running cycle on even ground. The solid lines indicate right leg and upper body. The dotted lines indicate left leg. The curve formed by "*" indicates CoM trajectory.

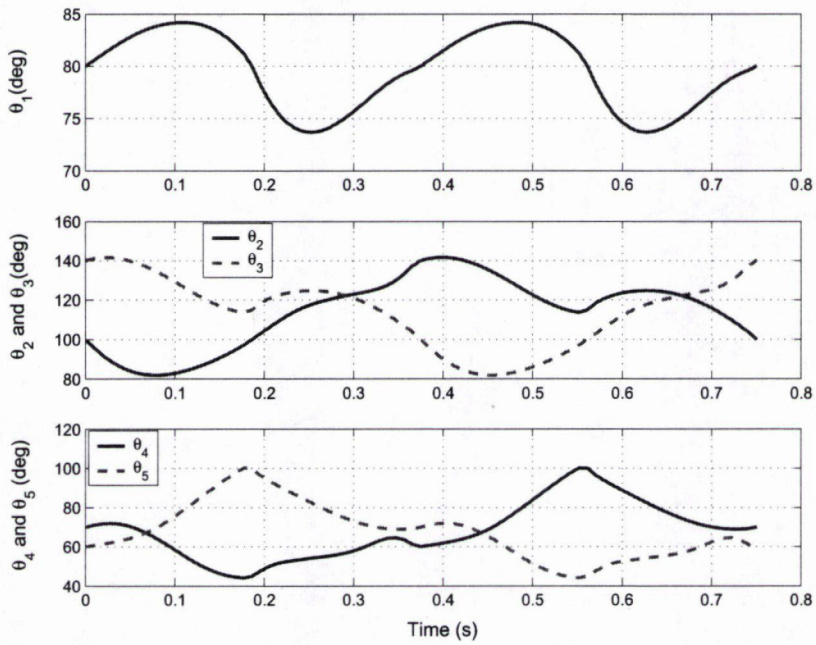

Figure 4.3: Joint angles for running on even ground

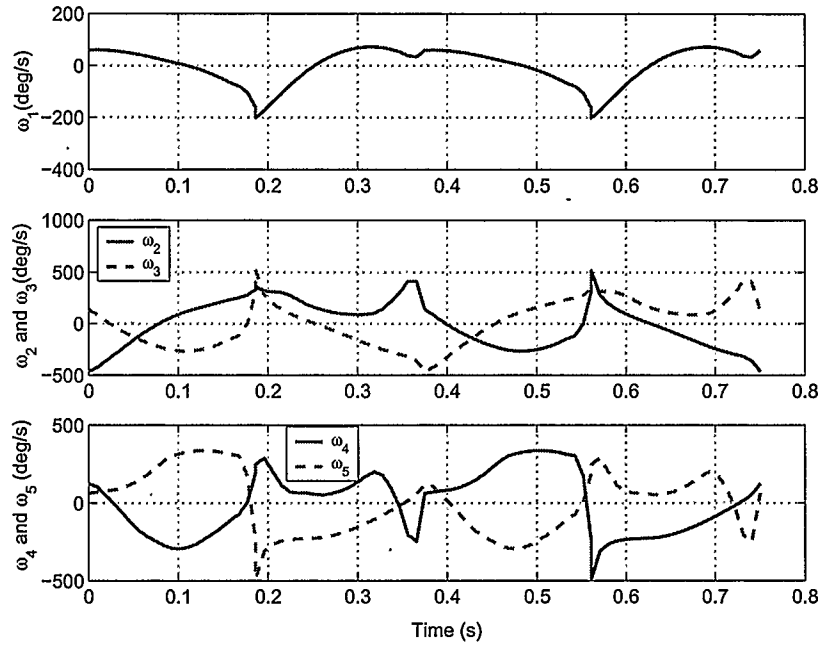

Figure 4.4: Joint velocities for running on even ground

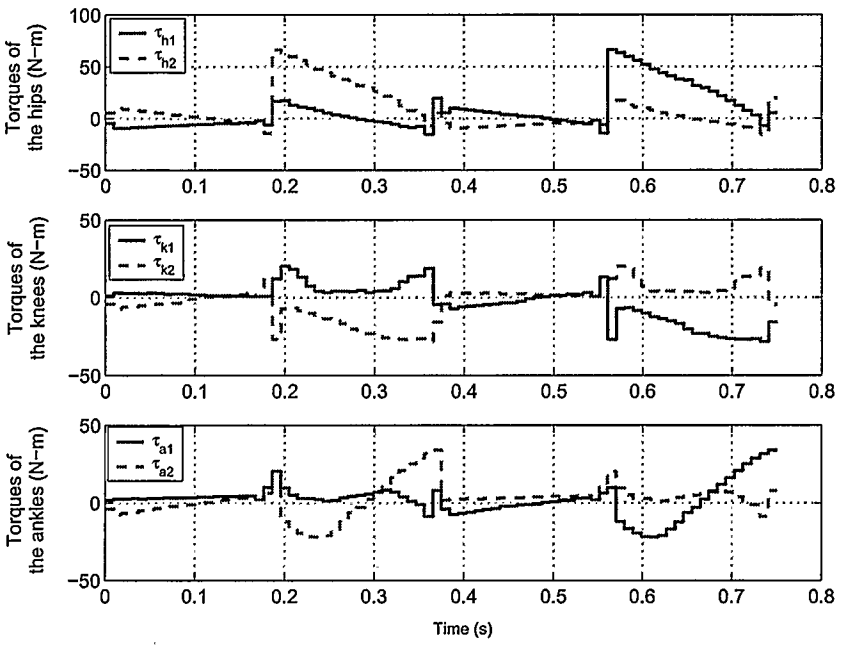

Figure 4.5: Control signals for running on even ground

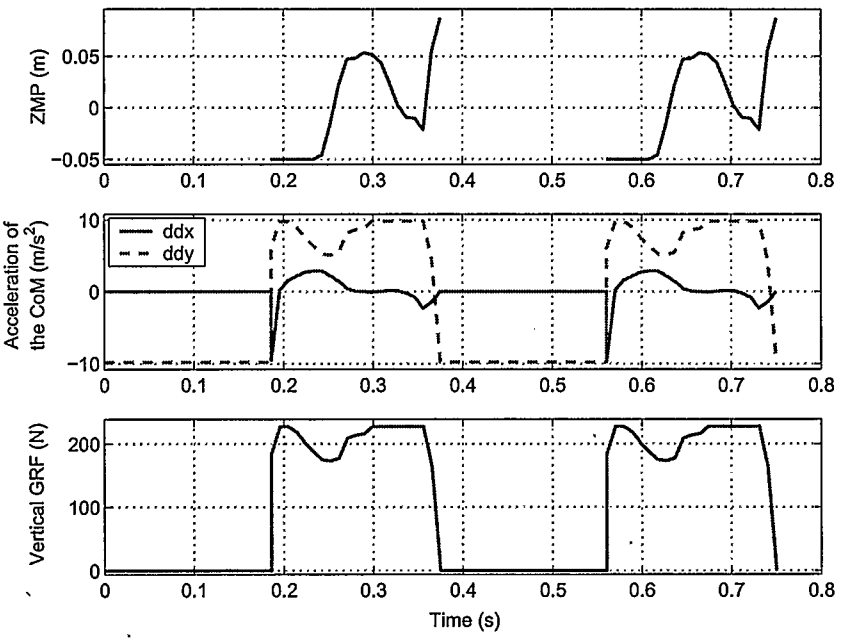

Figure 4.6: The ZMP trajectory (top), acceleration components of the CoM (middle), and the vertical GRF (bottom) for running on even ground. For the middle graph: horizontal component, solid line, vertical component, dashed line.

which is rare in human running. Due to (4.32), the horizontal acceleration of the CoM becomes zero at the end of the stance phase (Fig. 4.6 middle). Therefore, the horizontal acceleration of the CoM remains continuous over the whole running cycle.

In each half running cycle, the energy consumed in the flight phase and the stance phase are 11 Joules and 61.4 Joules, respectively. The flight phase consumes about 15.2% of the total energy of the half running cycle. The cost of transport of the simulated robot is $1.06JN^{-1}m^{-1}$. Assuming that the transmission efficiency of the robot is about 50%, the true cost of transport is estimated to be around $2.2JN^{-1}m^{-1}$. The cost appears to be higher than humans and the spring-loaded Monopods I and II, but possibly lower than HRP-2L (see Table 3.1).

4.3.2 Running up stairs

Using the parameters listed in Table 4.3 results in the robot running up stairs (Fig. 4.7). The joint angles (Fig. 4.8), the joint velocities (Fig. 4.9), the joint torques (Fig. 4.10) differ in a reasonable manner from the even ground results. The ZMP trajectories, the horizontal and vertical accelerations of the CoM, and the normal GRF appear in Fig. 4.11

The energy consumed in the flight phase and the stance phase are 8.1 Joules and 68.8 Joules, respectively, and the flight phase consumes about 10.5% of the total energy of the half running cycle. The calculated cost of transport of the robot is about $1.7JN^{-1}m^{-1}$, which shows that running up stairs is not as efficient as on even ground, even though the average forward speed is greatly reduced, from $1.6m/s$ to $0.8m/s$. If the transmission efficiency of the robot is about 50%, the actual cost of running up stairs is about $3.4JN^{-1}m^{-1}$. It may be a little more efficient than the

Table 4.3: The parameters for running up stairs

Parameters	Values	Units
L_{s}	0.4	m
h_{s}	0.08	m/s
V_{x}	0.8	m/s
Θ_{to}	$(80, 100, 135, 70, 60)^{\mathrm{T}}$	deg
Θ_{td}	$(75, 105, 115, 45, 95)^{\mathrm{T}}$	deg
γ	0.3	-
k_{x}	0.5	-
k_{y}	0.8	-

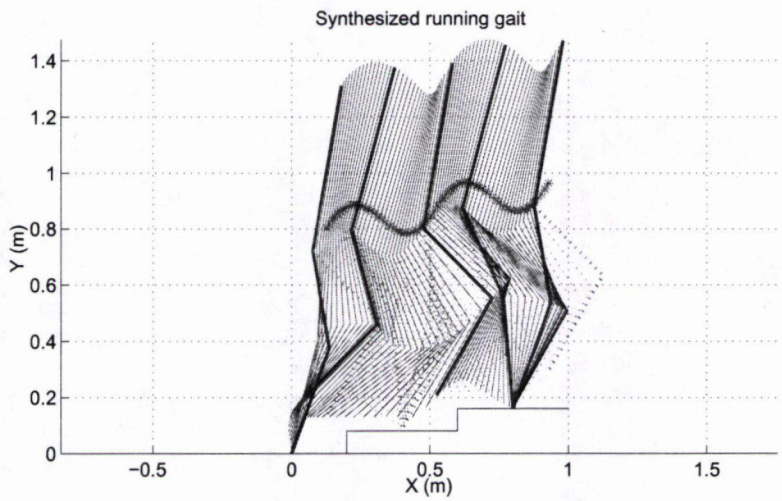

Figure 4.7: A gait cycle for running up stairs. The solid lines indicate right leg and upper body. The dotted lines indicate left leg. The curve formed with "*" indicates CoM trajectory.

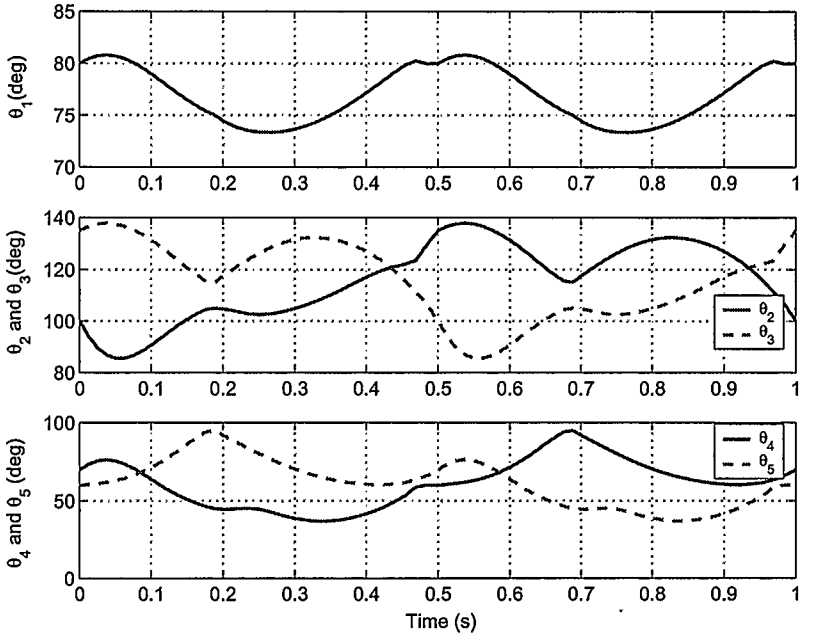

Figure 4.8: Joint angles for running up stairs

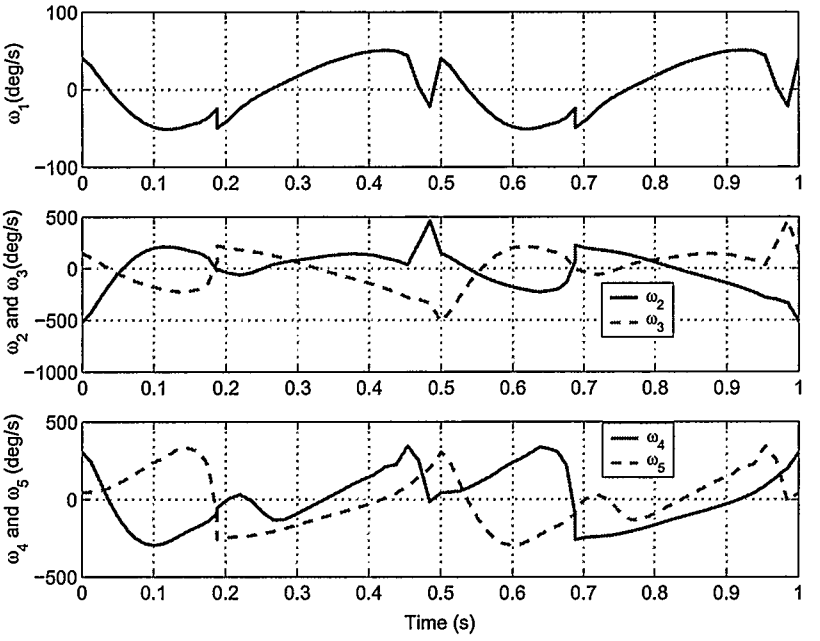

Figure 4.9: Joint velocities for running up stairs

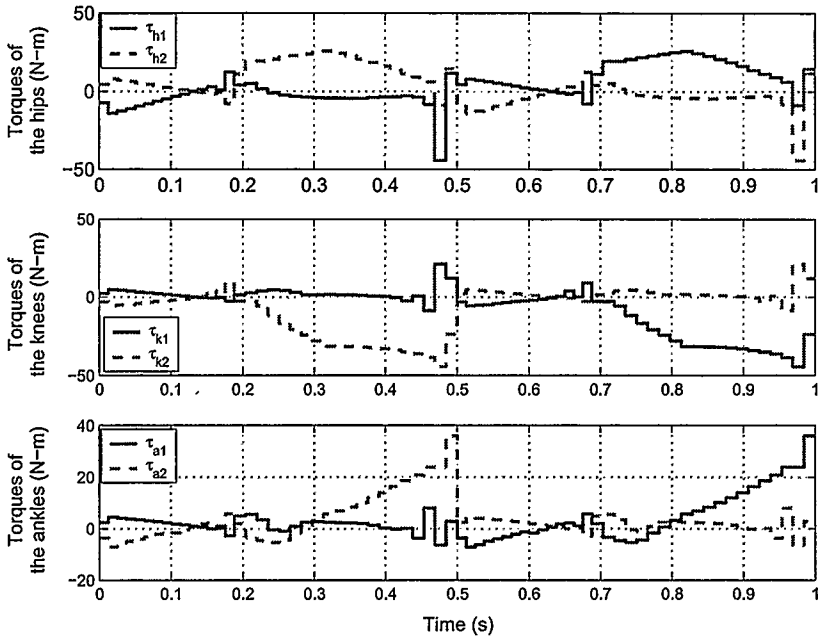

Figure 4.10: Control signals for running up stairs

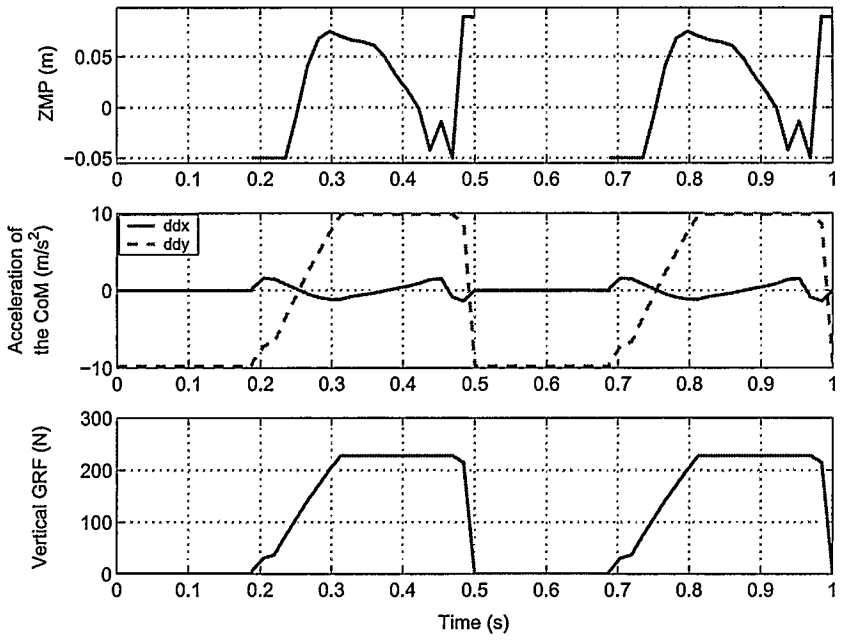

Figure 4.11: The ZMP trajectory, acceleration components of the CoM, and the vertical GRF for running up stairs

HRP-2L's running on even ground.

4.3.3 Some remarks

Simulations of bipedal running also confirm the fundamental assumption (A1) presented in Chapter 3. The flight phases consume relatively little energy compared to the stance phases. This applies running on even ground and up stairs. The maximum normal GRF, $F_{\mathrm{y,max}}$, is set to be twice the robot's weight. Although it is a reasonable value for humans and animals, this value may be too large for a real running robot. A physical robot may require active force control or passive compliant feet in order to partly damp the large vertical GRF. The static friction coefficient $\mu = 0.6$ represents a typical real-world value. However, if smaller $F_{\mathrm{y,max}}$ and μ have to be used, the robot could reduce the average forward speed V_{x}, the landing height h_{s}, or the stride length L_s to achieve safe running.

Compared with [97], the robot model under study is more realistic. The generated running gaits are more efficient, in the sense of cost of transport. Compared with [34], the proposed method provides a systematic approach to choice of the initial joint velocities in the running phases, by using the fundamental assumption (A1). The rigid biped clearly does not achieve the energy-efficiency of designs that incorporate compliant elements, *e.g.* the Monopods I and II. However, this approach lays a useful mathematical framework for future research into bipedal running, including compliant designs.

The two legs of the biped play different roles. The foot that touches the ground is not the foot that leaves the ground. This results in two expressions of the CoM with respect to the two ankles, *i.e.* (4.1) and (4.2). During the stance phase, to prevent

the swing leg from scuffing the ground, the lowest ankle height has to be limited. Moreover, to switch the role of the two legs, a switching matrix is used. These particular characteristics do not appear in one-legged hopping. It is interesting to observe that solutions to the bipedal running synthesis are easier to find than those to the one-legged hopping.

4.4 Summary

In this chapter, the techniques, presented in Chapter 3, were extended to synthesize bipedal running gaits. The assumption (A1) was employed again as the cornerstone of bipedal running synthesis. Different from hopping synthesis, a switching matrix helps to switch the role of the two legs. When the swing leg recovers from hind to front during the stance phase, the lowest ankle height is forced to be higher than a threshold in order to avoid scuffing with the ground. Other designs are similar as those for hopping synthesis. The computational time of bipedal running synthesis is usually shorter than that of one-legged hopping, revealing that the bipedal robots are essentially more stable than the one-legged hopper. Simulation results show that the generated running gaits, both on even ground and up stairs, are visually appealing. The energy efficiency in the two cases is acceptable.

Chapter 5

Hybrid Finite-Time Control Designs for the One-Legged Hopper

The hopping robot is a hybrid system, including two continuous hopping phases, and two discrete events. A touchdown triggers a transition from the flight phase to the stance phase, and a take-off symbolizes the end of a stance phase and the start of a flight phase. The two discrete events determine the initial conditions of the subsequent motion phase. In general, a two-level control architecture is needed. The top level is a supervisory controller, typically implemented as a finite-state machine (FSM), *e.g.* the control designs in Raibert's hoppers [118]. The low-level controllers stabilize the continuous motion phases. A good control subsystem must at least satisfy the following two conditions:

1. The control subsystem itself is stable in some sense.

2. The robot, with the control algorithm applied, must be free from damage or falling. In other words, the controlled gait must be stable.

In accordance with different definitions of gait stability, the existing control algorithms for robots with one or two rigid articulated legs roughly fall in two categories. The ZMP-based controllers employ the ZMP stability criterion such that the resultant ZMP trajectory is within the support polygon [148], or more strictly, within the safety region which is defined as the middle part of the support polygon [73].

Typical robots based on ZMP concepts include the Honda's Asimo [50], the Sony's Qrio [95], and others [100, 94]. The other type of controllers are based on orbital stability [54, 40]. Rather than tracking the joint trajectories, these controllers pursue to track the phase orbits [151, 92]. Since no time information is contained on the phase portrait, these controllers are also called event-based [151, 92].

The biggest challenge to monopods or bipeds comes from the interaction between the robots and the environment. Due to the unstructured environment, the robot has to adjust its gaits frequently in real time, and the gaits are rarely periodic as what the event-based controllers assume. Varying with the environment, the GRFs change step by step. In case that the GRFs are greatly larger than the permitted limits, the robot gets damaged. An active force control mechanism must be embedded in the control algorithm, such that the excessive GRFs can be suppressed and damage to the robot can be avoided.

An intuitive approach to suppress the excessive GRFs is to crouch the body, as humans do. Since the model of the environment is in general unknown, the *impedance control*, originally proposed by Hogan to regulate the contact force of robotic manipulators against the environment [51], may help to address this issue. In applications where a robotic manipulator has to maintain firm contact with the environment, the robot in the presence of contact forces is usually modeled as a mass-spring-damper system [150, 68, 82, 139], a mass-damper system [80, 123], or a mass-spring system [25, 81]. Also, it is convenient to formulate such a problem in the task space, based on the pioneering work done by Khatib [69]. Arimoto has shown that this problem can also be formulated in joint space without a force model [8]. In these works, a position-tracking control module achieves asymptotic stability,

and a PI (proportional-integral) control module, with the excessive forces being the inputs, limits the contact forces. Experiments have verified the effectiveness of the impedance control [146, 123].

Force-regulation control has been used in Honda's Asimo, but the technical details have not been disclosed [50]. Park and Chung proposed a hybrid control algorithm to achieve bipedal walking [103]. The impedance control method was applied to the swing leg to fulfill soft landing. The impedance parameters varied with the walking phases. The computed torque control was used for the support leg. The same algorithm was then tested in different biped platforms [102, 105, 73, 74]. Lim *et al.* implemented another impedance control algorithm [78] where the outputs of the controller are joint angles, rather than the joint torques as in [103]. In these work, the contact forces were modeled as mass-spring-damper systems in task space. Silva and Tenreiro Machado proposed another force control algorithm in which the environment was modeled as a spring-damper system. The desired vertical GRF was assumed to be the robot weight plus PD (proportional-derivative) compensation terms which regulate the hip height. Katić *et al.* employed a PI regulator to reject the excessive GRFs. Unfortunately, strict stability analysis has not been presented in these reports.

In reality, apart from the variable environment, system uncertainties, unmodeled dynamics, external disturbances, and imperfect control actions may cause the robot to lose its balance, or to fall. For the ZMP-based robots, this implies that the actual ZMP location is not within the support polygon. An online ZMP compensation module is needed to drag the ZMP back inside the safety region. Park and Chung suggested changing the height of the hip to achieve this goal, without sacrificing the

robot's forward speed [104]. Okumura *et al.* proposed altering the joint acceleration to compensate the ZMP deviation [99]. This method changes the forward speed. Prahlad *et al.* tried to adjust the torque at the ankle to sustain the desired ZMP trajectory [113]. This problem has also been formulated in the task space. A "Jacobian compensation" method, aiming to modify the pre-planned joint trajectories, has been reported [135, 154].

The forward speed can only be controlled in the stance phase. In the stance phase, the final states (joint angles and joint velocities) should converge to the pre-planned values before the phase end. The traditional asymptotic convergence does not satisfy this requirement. Moreover, the flight phase should also be stabilized before the phase end, in order to land at the right time and at the right foothold. Thus, the velocity jumps at the joints would not differ too much from the predicted values. This affects greatly the balance of the subsequent stance phase. To guarantee finite-time convergence, a finite-time controller is needed.

The concepts of finite-time control and finite-time stability have existed for a long time [30], but the modern meanings of these concepts may be attributed to Haimo [44]. Bhat and Bernstein extended Haimo's work and gave more complete definitions of the concepts [13]. The completeness of the theory makes the finite-time control design more attractive than other fast sliding mode control algorithms, *e.g.* reported in [162, 107]. The finite-time controllers proposed by Haimo [44] and other terminal sliding mode controllers have been examined extensively for robotic manipulators (see [31, 159, 158, 79] and the references therein). In these algorithms, besides the controllers, the sliding surfaces also contain the finite-time functions. This may result in complex solutions.

In this chapter, a novel control algorithm for the one-legged hopper is presented. The sliding surfaces do not contain any finite-time functions. Instead, a 3^{rd} order polynomial is embedded in each of the sliding surfaces, following [106]. A reaching interval[1] is not needed, and thus the system is more robust than traditional sliding mode control algorithms [155]. Haimo's finite-time functions are used such that the system trajectories converge to the desired references in finite-time, if system uncertainties and external disturbances are absent. With system uncertainties and external disturbances present, the system trajectories are *uniformly bounded* (Definition 9.3, page 36 in [157]). The finite-time stability is briefly reviewed in Section 5.1. To reject the excessive GRFs while keeping tracking accuracy, a force-suppression module is implemented. The excessive GRFs are modeled as a mass-damper system in joint space, and stability analysis is provided. An original online ZMP compensation module is also proposed. To maintain the ZMP within the safety region, the torso pitch angle is modified in real time. Details of the control algorithm are explained in Section 5.2. Simulation results are presented in Section 5.3. Concluding remarks follow in Section 5.4.

5.1 Finite-time stability

Consider a continuous system

$$\dot{x}(t) = f(x(t)), \quad x(t_0) = x_0, \tag{5.1}$$

[1]Also well known as "reaching phase" in sliding mode control theory. In this thesis, the word "phase" indicates a particular time period of a certain gait.

where $x(t) \in \mathcal{X} \subseteq \Re^n$ for $t \in \mathcal{T}_{x_0}$ is system state vector, $\mathcal{T}_{x_0}$ is the maximal interval of existence of a solution $x(t)$ to (5.1), $\mathcal{X}$ is an open set, $0 \in \mathcal{X}$, $f(0) = 0$. Assume that (5.1) has a unique solution in forward time for all initial conditions except probably the origin.

Definition 1 (Definition 4.7, page 254 in [43], Definition 2.2 in [13]). For the system (5.1), the equilibrium $x_e(t) = 0$ of (5.1) is *(locally) finite-time stable*, if there exists an open neighborhood $\mathcal{N} \subseteq \mathcal{X}$ of the origin and a function $T_s : \mathcal{N}\backslash\{0\} \mapsto (0, \infty)$, called the *settling-time function*, such that the following statements hold:

i) *Finite-time convergence.* For every $x_0 \in \mathcal{N}\backslash\{0\}$, $x(t) \in \mathcal{N}\backslash\{0\}$, for all $t \in [0, T_s(x_0))$, and $\lim_{t \to T_s(x_0)} x(t) = 0$;

ii) *Lyapunov stability.* For every $\epsilon > 0$ there exists a $\delta > 0$ such that for every $x_0 \in \mathcal{N}\backslash\{0\}$ and $\|x_0\| \leq \delta$, $\|x(t)\| \leq \epsilon$ for $t \in [0, T_s(x_0))$.

The equilibrium $x_e(t) = 0$ of (5.1) is *globally finite-time stable*, if it is finite-time stable with $\mathcal{N} = \mathcal{X} = \Re^n$.

If the equilibrium $x_e = 0$ is finite-time stable, then it is also asymptotically stable. Therefore, finite-time stability is stronger than asymptotic stability. The settling time function $T_s(x_0)$ varies with the initial state x_0.

Specifically, if $x_0 = 0$, $x(t) = x_e(t) = 0$ for all $t \geq 0$. By defining $T_s(0) := 0$, $T_s(x_0)$ is continuous on $\mathcal{N}$.

Theorem 5.1.1. *(Theorem 4.17, page 257 in [43], Theorem 4.2 in [13]) For system (5.1), assume that there exists a continuously differentiable function $V : \mathcal{X} \mapsto \Re_{\geq 0}$, real numbers $k > 0$ and $\gamma \in (0, 1)$, and a neighborhood $\mathcal{N} \subseteq \mathcal{X}$ of the origin, such*

that

$$V(0) = 0, \tag{5.2}$$

$$V(x) > 0, \quad x \in \mathcal{N}\backslash\{0\}, \tag{5.3}$$

$$\dot{V}(x) \leq -k(V(x))^{\gamma}, \quad x \in \mathcal{N}\backslash\{0\}. \tag{5.4}$$

Then the equilibrium $x_e = 0$ is finite-time stable. Moreover, there exists an open neighborhood $\mathcal{N}$ of the origin, and a settling-time function $T_{s,1} : \mathcal{N} \mapsto \Re_{\geq 0}$ such that

$$T_{s,1}(x_0) \leq \frac{1}{k(1-\gamma)}(V(x_0))^{1-\gamma}, \quad x_0 \in \mathcal{N}, \tag{5.5}$$

and $T_{s,1}(x_0)$ is continuous on $\mathcal{N}$.

If $\mathcal{N} = \Re^n$, the equilibrium $x_e = 0$ is globally finite-time stable.

Example: (Example 4.12, page 255 in [43], Example 2.1 in [13]) Following Haimo [44], denote $\mathrm{sig}^{\gamma}(\mathrm{x}) = |\mathrm{x}|^{\gamma}\,\mathrm{sgn}(\mathrm{x})$ where $\mathrm{sgn}(\cdot)$ is the signum function and $\gamma \in (0, 1)$. A nonlinear dynamic system is described by

$$\dot{x}(t) = -k\mathrm{sig}^{\gamma}(\mathrm{x}), \quad \mathrm{x}(0) = \mathrm{x}_0, \quad \mathrm{t} \geq 0,$$

with k being positive. It is easy to check that this system is globally finite-time stable, by using $V(x) = x^2$, and the settling-time function is

$$T_s(x_0) = \frac{1}{k(1-\gamma)}\,|x_0|^{1-\gamma}.$$

Theorem 5.1.2. *(An extension of Theorem 5.1.1) For system (5.1), assume that there exists a continuously differentiable function $V : \mathcal{X} \mapsto \Re_{\geq 0}$, real numbers $k_1 > 0$,*

$k_2 > 0$, and $\gamma \in (0,1)$, and a neighborhood $\mathcal{N} \subseteq \mathcal{X}$ of the origin, such that

$$V(0) = 0, \tag{5.6}$$

$$V(x) > 0, \quad x \in \mathcal{N}\backslash\{0\}, \tag{5.7}$$

$$\dot{V}(x) \leq -k_1 V(x) - k_2 (V(x))^{\gamma}, \quad x \in \mathcal{N}\backslash\{0\}. \tag{5.8}$$

Then the equilibrium $x_e = 0$ is finite-time stable. Moreover, there exists an open neighborhood $\mathcal{N}$ of the origin, and a settling-time function $T_{s,2} : \mathcal{N} \mapsto \Re_{\geq 0}$ such that

$$T_{s,2}(x_0) \leq \frac{1}{k_1(1-\gamma)} \ln\left(\frac{k_1 (V(x_0))^{1-\gamma} + k_2}{k_2}\right), \quad x_0 \in \mathcal{N}, \tag{5.9}$$

and $T_{s,2}(x_0)$ is continuous on $\mathcal{N}$.

If $\mathcal{N} = \Re^n$, the equilibrium $x_e = 0$ is globally finite-time stable.

Proof: (5.6), (5.7), and (5.8) indicate that the equilibrium $x_e = 0$ is Lyapunov stable. Also, direct integration of (5.8) yields (5.9), implying finite-time convergence. Therefore, the equilibrium is finite-time stable. Correspondingly, the other results follow.

□

Theorem 5.1.3. *Consider (5.5), and let $k = k_2$. The following inequality always holds:*

$$T_{s,2}(x_0) \leq T_{s,1}(x_0). \tag{5.10}$$

Proof: Let

$$m = \frac{T_{s,2}}{T_{s,1}},$$

and hence

$$m = \frac{\ln(p+1)}{p},$$

where $p = \frac{k_1(V(x_0))^{1-\gamma}}{k_2}$. It is obvious that $m \leq 1$, and only when $p = 0$, the equality in (5.10) holds.

□

Theorem 5.1.3 states that adding a negative proportional term to the right-hand side of (5.4) can speed up the convergence. Especially, increasing p leads to faster convergence.

5.2 The control algorithm

This section explains the details of the control algorithm. To proceed clearly, fundamental mathematical preliminaries are presented first.

5.2.1 Mathematical preliminaries

Here, three important inequalities are introduced. The first one is actually an extension of the classical Hölder inequality.

Lemma 5.2.1. *(Theorem 3, Page 52, in [89]) Let* $a_{\mathrm{i,j}}$ $(\mathrm{i} = 1, \cdots, \mathrm{n}; j = 1, \cdots, m)$ *be positive numbers, and let* $k_1, k_2, \cdots, k_{\mathrm{m}}$ *be positive numbers such that*

$$\frac{1}{k_1} + \frac{1}{k_2} + \cdots + \frac{1}{k_{\mathrm{m}}} \geq 1.$$

Then

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} a_{\mathrm{i},1} \cdots a_{\mathrm{i,m}} \leq \left(\sum_{\mathrm{i}=1}^{\mathrm{n}} a_{\mathrm{i},1}^{k_1} \right)^{\frac{1}{k_1}} \cdots \left(\sum_{\mathrm{i}=1}^{\mathrm{n}} a_{\mathrm{i,m}}^{k_{\mathrm{m}}} \right)^{\frac{1}{k_{\mathrm{m}}}}. \tag{5.11}$$

Lemma 5.2.2. *Let* a_{i} $(\mathrm{i} = 1, \cdots, \mathrm{n})$ *be positive numbers. For* $1 < \mathrm{p} < 2$, *the following inequality holds:*

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} a_{\mathrm{i}}^{\mathrm{p}} \geq \left(\sum_{\mathrm{i}=1}^{\mathrm{n}} a_{\mathrm{i}}^2 \right)^{\frac{\mathrm{p}}{2}}. \tag{5.12}$$

Proof: Let $m = 2$, $a_{i,1} = a_{i,2} = a_i$, $k_1 = 2$, and $1 < k_2 = p < 2$. Clearly, $\frac{1}{2} + \frac{1}{p} \geq 1$. By applying Lemma 5.2.1, it can be deduced that

$$\sum_{i=1}^{n} a_i^2 \leq \left(\sum_{i=1}^{n} a_i^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^{n} a_i^p \right)^{\frac{1}{p}},$$

which leads to

$$\left(\sum_{i=1}^{n} a_i^2 \right)^{\frac{1}{2}} \leq \left(\sum_{i=1}^{n} a_i^p \right)^{\frac{1}{p}},$$

and hence (5.12) is satisfied.

□

Lemma 5.2.3. *Let* $s = (s_1, s_2, \cdots, s_n)^T$ *and* $\gamma \in (0, 1)$. *By extending Haimo's notation [44], define a vector function:*

$$sig^{\gamma}(s) = \begin{bmatrix} |s_1|^{\gamma} \operatorname{sgn}(s_1) \\ |s_2|^{\gamma} \operatorname{sgn}(s_2) \\ \cdots \\ |s_n|^{\gamma} \operatorname{sgn}(s_n) \end{bmatrix},$$

and a diagonal matrix $K = \operatorname{diag}(k_1, k_2, \cdots, k_n)$, *where* $k_i > 0$. *Then*

$$s^T K sig^{\gamma}(s) \geq k_{min} \left(\sum_{i=1}^{n} s_i^2 \right)^{\frac{1+\gamma}{2}}, \tag{5.13}$$

where $k_{min} = \min(k_1, k_2, \cdots, k_n)$.

Proof: Note that

$$s_i |s_i|^{\gamma} \operatorname{sgn}(s_i) = |s_i|^{1+\gamma}.$$

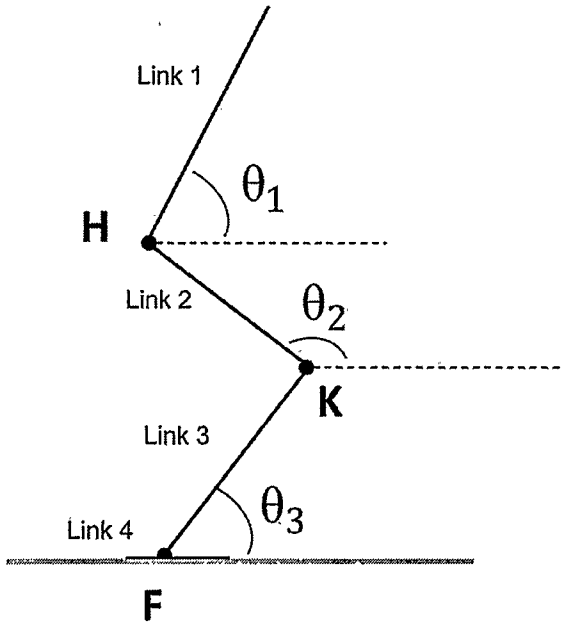

Figure 5.1: The model of the articulated hopper

Thus,

$$\begin{aligned} s^{\mathrm{T}} K \mathrm{sig}^{\gamma}(\mathrm{s}) &= \sum_{\mathrm{i}=1}^{\mathrm{n}} k_{\mathrm{i}} \left|s_{\mathrm{i}}\right|^{1+\gamma}, \\ &\geq k_{\min} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left|s_{\mathrm{i}}\right|^{1+\gamma}. \end{aligned} \tag{5.14}$$

By using Lemma 5.2.2, inequality (5.14) produces (5.13).

□

5.2.2 The hopping model

Fig. 5.1 presents the planar articulated hopper under study. It is a duplication of Fig. 3.1. The physical meanings of the parameters remain identical to those in Chapter 3.

5.2.3 The flight phase

During the flight phase, the robot has 5 degrees of freedom (DOFs). The generalized coordinate vector, $q^{\mathrm{f}} = (X_{\mathrm{g}}, Y_{\mathrm{g}}, \theta_1, \theta_2, \theta_3)^{\mathrm{T}}$, describes the robot's posture in the air.

The practical model of the flight phase

Without considering the parameter uncertainties and external disturbances, the hopper in the flight phase can be modeled as:

$$D^{\mathrm{f}}(q^{\mathrm{f}})\ddot{q}^{\mathrm{f}} + H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}})\dot{q}^{\mathrm{f}} + G^{\mathrm{f}}(q^{\mathrm{f}}) = B^{\mathrm{f}}\tau^{\mathrm{f}}, \tag{5.15}$$

where $D^{\mathrm{f}}(q^{\mathrm{f}}) \in \Re^{5\times5}$ is the positive-definite, symmetric inertia matrix, $H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}}) \in \Re^{5\times5}$ contains the Centrifugal and Coriolis terms , and $G^{\mathrm{f}}(q^{\mathrm{f}}) \in \Re^5$ is the gravitational torque vector. The vector $\tau^{\mathrm{f}} = (\tau_1^{\mathrm{f}}, \tau_2^{\mathrm{f}}, \tau_3^{\mathrm{f}})^{\mathrm{T}}$ contains the torques of the hip, the knee, and the ankle. The constant coefficient matrix $B^{\mathrm{f}} \in \Re^{5\times3}$ can easily be determined by using the virtual work principle. For the sake of clarity, the superscript "f" in the remaining of this subsection disappears if no ambiguity is introduced.

By denoting $q_1 = (X_{\mathrm{g}}, Y_{\mathrm{g}})^{\mathrm{T}}$, and $q_2 = (\theta_1, \theta_2, \theta_3)^{\mathrm{T}}$, equation (5.15) can be decomposed into [42]:

$$D_1\ddot{q}_1 + G_1 = 0, \tag{5.16}$$

$$D_2(q_2)\ddot{q}_2 + H_2(q_2, \dot{q}_2)\dot{q}_2 = B_2\tau, \tag{5.17}$$

where $D_1 = \mathrm{diag}(\mathrm{M_t}, \mathrm{M_t})$, $G_1 = (0, M_t g)$, with g being the gravity acceleration. $D_2 \in \Re^{3\times3}$ and $H_2 \in \Re^{3\times3}$ are right-bottom submatrices of D and H, respectively. $B_2 \in \Re^{3\times3}$ is a bottom submatrix of B.

Equation (5.16) means that the hopper's CoM cannot be manipulated in the flight phase. The control actions can only affect the joint angles which are described

by (5.17). Considering the parameter uncertainties, viscous friction, and the external disturbances, the model (5.17) is extended as:

$$D_2(q_2)\ddot{q}_2 + H_2(q_2, \dot{q}_2)\dot{q}_2 = B_2\tau - \tau_{\mathrm{f}} + \tau_{\mathrm{dis}}, \tag{5.18}$$

The matrices $D_2(q_2)$ and $H_2(q_2, \dot{q}_2)$ possess the following important properties [137]:

1. $D_2(q_2)$ is bounded by

$$d_{\min} I_{\mathrm{n\times n}} \leq D_2(q_2) \leq d_{\max} I_{\mathrm{n\times n}}, \tag{5.19}$$

   where $d_{\min}$ and $d_{\max}$ are the minimum and maximum eigenvalues of $D_2(q_2)$, respectively.

2. $H_2(q_2, \dot{q}_2)$ can be chosen such that $\dot{D}_2(q_2) - 2H_2(q_2, \dot{q}_2)$ is skew-symmetric. Mathematically,

$$x^{\mathrm{T}}(\dot{D}_2 - 2H_2)x = 0, \tag{5.20}$$

   where $x \in \Re^3$ is an arbitrary vector.

3. The left-hand side (LHS) of (5.18) can be linearly expressed as

$$D_2(q_2)\ddot{q}_2 + H_2(q_2, \dot{q}_2)\dot{q}_2 = \mathcal{Y}(q_2, \dot{q}_2, \ddot{q}_2)P, \tag{5.21}$$

   where $\mathcal{Y}(q_2, \dot{q}_2, \ddot{q}_2)$ is the regressor, and P contains system parameters.

The following assumptions are made:

1. The external disturbance torque τ_{dis} is bounded by $\|\tau_{\mathrm{dis}}\| \leq \eta_{\mathrm{d}}$.

2. The viscous friction torque can be modeled as [108]

$$\tau_{\mathrm{f}} = K_{\mathrm{f}}\dot{q}_2, \tag{5.22}$$

where $K_{\mathrm{f}} = \mathrm{diag}(\mathrm{k}_{\mathrm{f},1}, \mathrm{k}_{\mathrm{f},2}, \mathrm{k}_{\mathrm{f},3})$ with each element being positive, and denote $k_{\mathrm{f,max}} = \max(\mathrm{k}_{\mathrm{f},1}, \mathrm{k}_{\mathrm{f},2}, \mathrm{k}_{\mathrm{f},3})$.

3. Denote the nominal parameter of the system by $\hat{P}$. The system uncertainty $\mathcal{Y}(P-\hat{P}) = \mathcal{Y}\tilde{P}$ is assumed to be bounded by $\left\|\mathcal{Y}\tilde{P}\right\| \leq m_0+m_1\left\|q_2\right\|+m_2\left\|\dot{q}_2\right\|^2$ [31, 159], where $m_{\mathrm{i}} > 0$.

4. The overall effect of system uncertainties, viscous friction, and external disturbances are bounded, *i.e.*,

$$\left\|\mathcal{Y}\tilde{P}\right\| + \left\|K_{\mathrm{f}}\dot{q}_2\right\| + \left\|\tau_{\mathrm{dis}}\right\| \leq \eta_{\mathrm{t}}(q_2, \dot{q}_2),$$

where $\eta_{\mathrm{t}}(q_2, \dot{q}_2) = m_0 + m_1\left\|q_2\right\| + m_2\left\|\dot{q}_2\right\|^2 + k_{\mathrm{f,max}}\left\|\dot{q}_2\right\| + \eta_{\mathrm{d}}$.

Control design for the flight phase

Define the tracking error as

$$\tilde{q}_2 = q_2 - q_{2,\mathrm{d}}, \tag{5.23}$$

where $q_{2,\mathrm{d}}$ consists of the desired joint angles.

Define a sliding surface [106]

$$\begin{aligned} s &= \dot{\tilde{q}}_2 + \Lambda\tilde{q}_2 - \dot{v}(t) - \Lambda v(t), && (5.24)\\ &= \dot{w}(t) + \Lambda w(t), && (5.25)\\ &= \dot{q}_2 - \dot{q}_{2,\mathrm{r}}, && (5.26) \end{aligned}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$, $\lambda_\mathrm{i} > 0$ for $\mathrm{i} = 1, 2, 3$, $\lambda_{\min} = \min(\lambda_1, \lambda_2, \lambda_3)$, $v(t)$ is the *desired tracking error*, and

$$\begin{aligned} \dot{q}_{2,\mathrm{r}} &= \dot{q}_{2,\mathrm{d}} - \Lambda\tilde{q}_2 + \dot{v}(t) + \Lambda v(t), \\ w(t) &= \tilde{q}_2 - v(t). \end{aligned} \tag{5.27}$$

The desired tracking error, $v(t)$, is defined as [106]

$$v(t) = \begin{cases} p_0 + p_1 t + p_2 t^2 + p_3 t^3, & t \in [0, T_\mathrm{c}^\mathrm{f}]; \\ 0, & t \in (T_\mathrm{c}^\mathrm{f}, \infty), \end{cases} \tag{5.28}$$

where T_c^f is the convergence time to be picked manually. The coefficients in (5.28) are chosen to be

$$\begin{aligned} p_0 &= \tilde{q}_2(0), \\ p_1 &= \dot{\tilde{q}}_2(0), \\ p_2 &= -\frac{3}{(T_\mathrm{c}^\mathrm{f})^2}\tilde{q}_2(0) - \frac{2}{T_\mathrm{c}^\mathrm{f}}\dot{\tilde{q}}_2(0), \\ p_3 &= \frac{2}{(T_\mathrm{c}^\mathrm{f})^3}\tilde{q}_2(0) + \frac{1}{(T_\mathrm{c}^\mathrm{f})^2}\dot{\tilde{q}}_2(0). \end{aligned}$$

Clearly, $v(0) = \tilde{q}_2(0)$, $\dot{v}(0) = \dot{\tilde{q}}_2(0)$. Hence, $w(0) = 0$ and $s(0) = 0$.

Differentiating both sides of (5.27) gives

$$\ddot{q}_{2,\mathrm{r}} = \ddot{q}_{2,\mathrm{d}} - \Lambda\dot{\tilde{q}}_2 + \ddot{v}(t) + \Lambda\dot{v}(t). \tag{5.29}$$

Rearranging (5.26) yields

$$\dot{q}_2 = s + \dot{q}_{2,\mathrm{r}}. \tag{5.30}$$

Substituting (5.30) into (5.18) produces

$$D_2\dot{s} + D_2\ddot{q}_{2,\mathrm{r}} + H_2 s + H_2\dot{q}_{2,\mathrm{r}} = B_2\tau - K_\mathrm{f}\dot{q}_2 - \tau_\mathrm{dis},$$

which leads to

$$D_2\dot{s} + H_2 s = B_2\tau - K_{\mathrm{f}}\dot{q}_2 - \tau_{\mathrm{dis}} - D_2\ddot{q}_{2,\mathrm{r}} - H_2\dot{q}_{2,\mathrm{r}}. \tag{5.31}$$

Theorem 5.2.4. *For the system (5.18), if the following controller is applied:*

$$\begin{aligned} \tau &= B_2^{-1}\left(\hat{D}_2\ddot{q}_{2,\mathrm{r}} + \hat{H}_2\dot{q}_{2,\mathrm{r}} - K_1 s - K_2\mathrm{sig}^{\gamma}(\mathrm{s})\right), \\ &= B_2^{-1}\left(\mathcal{Y}(q_2, \dot{q}_2, \dot{q}_{2,\mathrm{r}}, \ddot{q}_{2,\mathrm{r}})\hat{P} - K_1 s - K_2\mathrm{sig}^{\gamma}(\mathrm{s})\right), \end{aligned} \tag{5.32}$$

where $K_1 = \mathrm{diag}(\mathrm{k}_{1,1}, \mathrm{k}_{1,2}, \mathrm{k}_{1,3})$, $K_2 = \mathrm{diag}(\mathrm{k}_{2,1}, \mathrm{k}_{2,2}, \mathrm{k}_{2,3})$, $k_{\mathrm{i,j}} > 0$ *for* $\mathrm{i} = 1, 2; \mathrm{j} = 1, 2, 3$, *then the system trajectories are uniformly bounded for* $t \geq 0$, *and the bound of the sliding surface is* $\|s\| \leq \|s\|_*$, *where* $\|s\|_*$ *is the unique solution of the equation*

$$k_{1,\min}\|s\| + k_{2,\min}\|s\|^{\gamma} - \eta_{\mathrm{c}} = 0, \tag{5.33}$$

with $\gamma \in (0, 1)$, $k_{1,\min} = \min(\mathrm{k}_{1,1}, \mathrm{k}_{1,2}, \mathrm{k}_{1,3})$, $k_{2,\min} = \min(\mathrm{k}_{2,1}, \mathrm{k}_{2,2}, \mathrm{k}_{2,3})$, *and*

$$\eta_{\mathrm{c}} = \sup_{q_2, \dot{q}_2 \in \Re^3} \left(\eta_{\mathrm{t}}(q_2, \dot{q}_2)\right). \tag{5.34}$$

Moreover, the tracking errors are bounded by

$$|v_{\mathrm{i}}| - \frac{\|s\|_*}{\lambda_{\min}} \leq |\tilde{q}_{2,\mathrm{i}}| \leq |v_{\mathrm{i}}| + \frac{\|s\|_*}{\lambda_{\min}}, \tag{5.35}$$

where $\tilde{q}_{2,\mathrm{i}}$ *and* v_{i} *are the* i^{th} *elements of* $\tilde{q}_2$ *and* v, *respectively.*

Proof: Substituting (5.32) into (5.31) generates

$$\begin{aligned} D_2\dot{s} + H_2 s &= -\tilde{D}_2\ddot{q}_{2,\mathrm{r}} - \tilde{H}_2\dot{q}_{2,\mathrm{r}} - K_1 s - K_2\mathrm{sig}^{\gamma}(\mathrm{s}) - \mathrm{K_f}\dot{\mathrm{q}} + \tau_{\mathrm{dis}}, \\ &= -\mathcal{Y}(q_2, \dot{q}_2, \dot{q}_{2,\mathrm{r}}, \ddot{q}_{2,\mathrm{r}})\tilde{P} - K_1 s - K_2\mathrm{sig}^{\gamma}(\mathrm{s}) - \mathrm{K_f}\dot{\mathrm{q}} + \tau_{\mathrm{dis}}, \end{aligned} \tag{5.36}$$

where $\hat{(\cdot)}$, $\tilde{(\cdot)}$ are the nominal and uncertain parts of $(\cdot)$, and $\tilde{(\cdot)} = (\cdot) - \hat{(\cdot)}$.

Define a Lyapunov function

$$V = \frac{1}{2} s^{\mathrm{T}} D_2 s, \tag{5.37}$$

and its time derivative is

$$\dot{V} = s^{\mathrm{T}} D_2 \dot{s} + \frac{1}{2} s^{\mathrm{T}} \dot{D}_2 s. \tag{5.38}$$

Due to (5.20), equation (5.38) implies

$$\begin{aligned}
\dot{V} &= s^{\mathrm{T}}(D_2\dot{s} + H_2 s), \\
&= s^{\mathrm{T}}\left(-\mathcal{Y}\tilde{P} - K_{\mathrm{f}}\dot{q}_2 + \tau_{\mathrm{dis}}\right) - s^{\mathrm{T}} K_1 s - s^{\mathrm{T}} K_2 \mathrm{sig}^{\gamma}(\mathrm{s}), \\
&\leq \|s\|\left(\left\|\mathcal{Y}\tilde{P}\right\| + k_{\mathrm{f,max}}\|\dot{q}_2\| + \|\tau_{\mathrm{dis}}\|\right) - k_{1,\min}\|s\|^2 - k_{2,\min}\|s\|^{1+\gamma}, \\
&\leq \|s\|\,\eta_{\mathrm{t}}(q_2, \dot{q}_2) - k_{1,\min}\|s\|^2 - k_{2,\min}\|s\|^{1+\gamma}.
\end{aligned} \tag{5.39}$$

At this moment, let us ignore the system uncertainties, the viscous friction, and the external disturbances. This means $\eta_{\mathrm{t}}(q_2, \dot{q}_2) = 0$. Then inequality (5.39) becomes

$$\begin{aligned}
\dot{V} &\leq -k_{1,\min}\|s\|^2 - k_{2,\min}\|s\|^{1+\gamma}, \\
&\leq -c_1 V - c_2 V^{\rho} \leq 0,
\end{aligned} \tag{5.40}$$

where $c_1 = 2k_{1,\min}$, $c_2 = 2^{\rho}k_{2,\min}$, and $\rho = \frac{1+\gamma}{2} \in (\frac{1}{2}, 1)$. Clearly, the system is finite-time stable, and according to Theorem 5.1.2, the settling time is

$$T_{\mathrm{s}} = \frac{1}{c_1(1-\rho)} \ln\left(\frac{c_1 (V(0))^{1-\rho} + c_2}{c_2}\right), \tag{5.41}$$

where $V(0) = \frac{1}{2}(s(0))^{\mathrm{T}} D_2(q_2(0))s(0) = 0$. Thus by (5.41)

$$T_{\mathrm{s}} = 0, \tag{5.42}$$

and

$$\begin{aligned} V &\equiv 0, \\ s &\equiv 0. \end{aligned} \tag{5.43}$$

Thus, equation (5.25) becomes

$$s = \dot{w} + \Lambda w \equiv 0,$$

which implies that

$$w(t) = e^{-\Lambda t} w(0). \tag{5.44}$$

Since $w(0) = \tilde{q}_2(0) - v(0) = 0$, the relations $w(t) \equiv 0$ and $\tilde{q}_2 \equiv v$ hold for $t \geq 0$. Therefore, $\tilde{q}_2$ converges to 0 at T_c^f by definition of v, and $\tilde{q}_2 = 0$ for $t \geq T_c^f$.

If system uncertainties, viscous friction, and external disturbances are taken into account, $\eta_t(q_2, \dot{q}_2) \neq 0$. Now, the design task is to choose the appropriate K_1, K_2, and Λ, such that $\eta_t(q_2, \dot{q}_2)$ is bounded by a constant η_c. At this moment, assume that the inequality $\|\eta_t(q_2, \dot{q}_2)\| \leq \eta_c$ holds true for $t \geq 0$. This condition needs to be verified *a posteriori*.

When $0 < \|\eta_t(q, \dot{q})\| \leq \eta_c$, inequality (5.39) becomes

$$\begin{aligned} \dot{V} &\leq \|s\| \eta_c - k_{1,\min} \|s\|^2 - k_{2,\min} \|s\|^{1+\gamma}, \\ &\leq -\|s\| \left(k_{1,\min} \|s\| + k_{2,\min} \|s\|^{\gamma} - \eta_c \right). \end{aligned} \tag{5.45}$$

Define a function

$$f(\|s\|) = k_{1,\min} \|s\| + k_{2,\min} \|s\|^{\gamma} - \eta_c,$$

which has following properties:

1. It is monotonically increasing with $\|s\|$;

2. $f(\|s\| = 0) = -\eta_c < 0$;

3. $f\left(\|s\| = \frac{\eta_c}{k_{1,\min}}\right) = k_{2,\min}\|s\|^\gamma > 0$;

4. $f\left(\|s\| = \left(\frac{\eta_c}{k_{2,\min}}\right)^{\frac{1}{\gamma}}\right) = k_{1,\min}\|s\| > 0$.

Therefore, the equation $f(\|s\|) = 0$ has a unique solution $\|s\|_*$, and

$$0 < \|s\|_* < \min\left(\frac{\eta_c}{k_{1,\min}}, \left(\frac{\eta_c}{k_{2,\min}}\right)^{\frac{1}{\gamma}}\right). \tag{5.46}$$

When $\|s\| > \|s\|_*$, $f(\|s\|) > 0$, implying $\dot{V} < 0$. Clearly,

$$\{s \mid \|s\| \leq \|s\|_*\} \tag{5.47}$$

is an invariant set. Outside this set, $\dot{V} < 0$. By (5.46), $\|s\|_*$ can be reduced by increasing $k_{1,\min}$ or $k_{2,\min}$, and hence the invariant set (5.47) can be as small as desired.

Since $s(0) = 0$ is inside the invariant set (5.47), $s(t)$ for $t \geq 0$ is always entrapped inside the invariant set. Therefore, the system trajectories are uniformly bounded for $t \geq 0$.

Due to (5.25), it can be inferred that

$$|w_i| \leq \frac{|s_i|}{\lambda_i} \leq \frac{\|s\|_*}{\lambda_i} \leq \frac{\|s\|_*}{\lambda_{\min}}, \tag{5.48}$$

following arguments by Slotine and Li in [134].

Furthermore, since

$$|\tilde{q}_i| - |v_i| \leq |\tilde{q}_i - v_i| = |w_i| \leq \frac{\|s\|_*}{\lambda_{\min}},$$

the relationship (5.35) follows.

□

In practice, $|w_i|$ can be very small by properly picking $k_{1,\min}$, $k_{2,\min}$, or $\lambda_{\min}$. The tracking error $\tilde{q}_{2,i}$ can follow the desired error trajectory v_i with high accuracy. Also, due to $\|\dot{w}\| \leq \|s\|_* + \|\Lambda w\|$, $\dot{w}$ is upper bounded. It can easily be concluded that $\dot{q}_2 = \dot{w} + \dot{v} + \dot{q}_{2,d}$ is also upper bounded, and thus $\eta_t(q, \dot{q})$ is indeed upper bounded.

In above discussions, $k_{1,\min}$, $k_{2,\min}$, and $\lambda_{\min}$ can be arbitrarily large. In reality, large $k_{1,\min}$, $k_{2,\min}$, or $\lambda_{\min}$ may cause dramatic oscillations of the system states. If $\dot{q}_2$ becomes too large, the assumption that $\eta_t(q_2, \dot{q}_2)$ is upper bounded may not hold true. In cases that $\eta_t > k_{1,\min}\|s\| + k_{2,\min}\|s\|^\gamma$, $\dot{V}$ becomes positive. Since the control signals cannot be too large, due to physical saturations, $\dot{V} > 0$ sustains, resulting in failure of the algorithm. Therefore, trade-offs must be made when the gain matrices are chosen.

5.2.4 The stance phase

In the stance phase, the number of the DOFs of the hopper is three. The vector of generalized coordinates is chosen as $q^s = (\theta_1, \theta_2, \theta_3)^T$.

Practical model of the stance phase

The practical model of the stance phase may be expressed as

$$D^s(q^s)\ddot{q}^s + H^s(q^s, \dot{q}^s)\dot{q}^s + G^s(q^s) = B^s\tau^s - \tau_f^s + J_g^T(q^s)\tilde{F} + \tau_{dis}^s, \tag{5.49}$$

where $D^s(q^s) \in \Re^{3\times3}$, $H^s(q^s, \dot{q}^s) \in \Re^{3\times3}$, and $G^s(q^s) \in \Re^3$ are the symmetric, positive definite inertia matrix, the matrix containing the Centrifugal and Coriolis terms, and the gravitational vector, respectively, $\tau \in \Re^3$ is the vector of joint torques, $B \in \Re^{3\times3}$

is the constant coefficient matrix of τ, $\tau_{\mathrm{f}}^{\mathrm{s}}$ represents the torque caused by the viscous friction forces, $J_{\mathrm{g}}(q^{\mathrm{s}}) \in \Re^{2\times 3}$ is the Jacobian mapping from the hopper's CoM to the ankle, $\tilde{F}$ is the vector of excessive external forces equivalently applied to the hopper's CoM, and $\tau_{\mathrm{dis}}^{\mathrm{s}}$ represents other external disturbances. In the remaining of this subsection, the superscript "s" will be dropped, if no confusion is introduced.

In an *ideal* stance phase, the support foot were expected to maintain firm touch with the ground, and it could be regarded as the base of a robotic arm. Thus, the GRFs could be treated as internal forces, and would not appear in the equations of motion. However, this is never true in practice. Due to unstructured ground conditions, unmodeled dynamics, uncertain parameters, unpredictable external disturbances and imperfect control, the robot may lose its posture balance. The actual GRFs often go outside of their permitted ranges. Denote the vector of the GRFs by F. The lower and upper bounds of F are $F_{\min}$ and $F_{\max}$, respectively. Suppose that F can be measured in real time with satisfactory accuracy. $\tilde{F}$ can be formulated as:

$$\tilde{F} = \begin{cases} 0, & F_{\min} \leq F \leq F_{\max}; \\ F - F_{\max}, & F > F_{\max}; \\ F - F_{\min}, & F < F_{\min}. \end{cases}$$

Note that F essentially represents all calculated external forces acting on the robot's CoM and the true external forces are not required to be known. $\tilde{F}$, expressed in the task space, can be mapped into the joint space as a torque vector τ_{F} via the Jacobian J_{g}, complying with

$$\tau_{\mathrm{F}} = J_{\mathrm{g}}^{\mathrm{T}}(q)\tilde{F}.$$

With the superscript "s" dropped, the stance model (5.49) can be re-written as

$$\begin{aligned}\hat{D}\ddot{q}+\hat{H}\dot{q}+\hat{G} &= B\tau-\tau_{\mathrm{f}}+\tau_{\mathrm{F}}+\tau_{\mathrm{dis}}-\tilde{D}\ddot{q}-\tilde{H}\dot{q}-\tilde{G},\\ &= B\tau-\tau_{\mathrm{f}}+\tau_{\mathrm{F}}+\tau_{\mathrm{dis}}-\mathcal{Y}(q,\dot{q},\ddot{q})\tilde{P}, \qquad (5.50)\end{aligned}$$

where $\hat{(\cdot)}$ and $\tilde{(\cdot)}$ are the nominal and uncertain parts of $(\cdot)$, and $\hat{(\cdot)}+\tilde{(\cdot)}=(\cdot)$.

Control design for the stance phase

The tracking error is defined as

$$\tilde{q}=q-q_{\mathrm{d}}. \qquad (5.51)$$

The desired error trajectory $v(t)$ is defined in a similar way as (5.28), with the convergence time being $T_{\mathrm{c}}^{\mathrm{s}}$.

Define a sliding surface [106]

$$\begin{aligned}s(t) &= \dot{\tilde{q}}+\Lambda\tilde{q}-\dot{v}(t)-\Lambda v(t), \qquad (5.52)\\ &= \dot{w}(t)+\Lambda w(t), \qquad (5.53)\\ &= \dot{q}-\dot{q}_{\mathrm{r}}, \qquad (5.54)\end{aligned}$$

where

$$\begin{aligned}w(t) &= \tilde{q}-v(t), \qquad (5.55)\\ \dot{q}_{\mathrm{r}} &= \dot{q}_{\mathrm{d}}-\Lambda\tilde{q}+\dot{v}+\Lambda v. \qquad (5.56)\end{aligned}$$

Note that the tracking error $\tilde{q}$ is expected to evolve along $v(t)$, and $w(t)$ is the absolute tracking error with the bias $v(t)$ removed. In other words, $w(t)$ is expected to evolve along zero. Differentiating both sides of (5.56) leads to

$$\ddot{q}_{\mathrm{r}}=\ddot{q}_{\mathrm{d}}-\Lambda\dot{\tilde{q}}+\ddot{v}+\Lambda\dot{v}. \qquad (5.57)$$

Clearly, $w(0) = 0$ and $s(0) = 0$.

Denote the estimate of τ_{F} by $\hat{\tau}_{\mathrm{F}}$, and

$$\hat{\tau}_{\mathrm{F}} = \hat{J}^{\mathrm{T}}(q)\tilde{F}, \tag{5.58}$$

where $\hat{J}_{\mathrm{g}}(q)$ is the estimate of $J_{\mathrm{g}}(q)$.

Theorem 5.2.5. *For the system (5.50), if*

1. *the following controller is applied:*

$$\tau = \tau_{\mathrm{p}} + \tau_{\mathrm{F}}, \tag{5.59}$$

with

$$\tau_{\mathrm{p}} = \underbrace{B^{-1}\left(\hat{D}\ddot{q}_{\mathrm{r}} + \hat{H}\dot{q} + \hat{G} - \hat{D}K_1 s - \hat{D}K_2 \mathrm{sig}^{\gamma}(\mathrm{s})\right)}_{position-tracking\ module}, \tag{5.60}$$

$$\tau_{\mathrm{F}} = \underbrace{-B^{-1}\left((I_{3\times 3} + \hat{D}K_\alpha)\hat{\tau}_{\mathrm{F}} + \hat{D}K_\beta \int_0^{\mathrm{t}} \hat{\tau}_{\mathrm{F}} dt\right)}_{force-suppression\ module}, \tag{5.61}$$

where $\gamma \in (0,1)$, $K_1 = \mathrm{diag}(\mathrm{k}_{1,1}, \mathrm{k}_{1,2}, \mathrm{k}_{1,3})$, $K_2 = \mathrm{diag}(\mathrm{k}_{2,1}, \mathrm{k}_{2,2}, \mathrm{k}_{2,3})$, $k_{\mathrm{i,j}} > 0$ *for* $\mathrm{i} = 1, 2; \mathrm{j} = 1, 2, 3$, $K_\alpha \in \Re^{3\times 3}$ *and* $K_\beta \in \Re^{3\times 3}$ *are diagonal, positive definite matrices;*

2. *to suppress the external forces, each joint is desired to behave like a mass-damper system, complying with*

$$\hat{\tau}_{\mathrm{F}} = h_0\ddot{w} + h_1\dot{w}, \tag{5.62}$$

where $h_0 \in \Re^{3\times 3}$ *and* $h_1 \in \Re^{3\times 3}$ *are diagonal, positive definite gain matrices, representing the desired moment of inertia and the desired damping ratio, and they can be picked arbitrarily to satisfy the design specifications;*

then

1. *the system trajectories are uniformly bounded for* $t \geq 0$, *and the bound of the sliding surface is* $\|s\| \leq \|s\|_*$, *where* $\|s\|_*$ *is the unique solution of the equation*

$$k'_{1,\min} \|s\| + k_{2,\min} \|s\|^\gamma - \eta_c = 0, \tag{5.63}$$

with $k'_{1,\min}$ *is the minimum element of the matrix* $K'_1 = K_1 + K_\alpha h_1$, $k_{2,\min} = \min(k_{2,1}, k_{2,2}, k_{2,3})$, $\eta_c = \sup\left(\hat{D}^{-1}\left(\tilde{\tau}_F - \tau_f + \tau_{dis} - \mathcal{Y}(q,\dot{q},\ddot{q})\tilde{P}\right)\right)$, *and* $\tilde{\tau}_F = \tau_F - \hat{\tau}_F$.

2. *the tracking errors are bounded by*

$$|v_i| - \frac{\|s\|_*}{\lambda_{\min}} \leq |\tilde{q}_i| \leq |v_i| + \frac{\|s\|_*}{\lambda_{\min}}, \tag{5.64}$$

where $\tilde{q}_i$ *and* v_i *are the* i^{th} *elements of* $\tilde{q}$ *and* v, *respectively.*

Proof: Substituting (5.59) into (5.50) results in

$$\begin{aligned} \dot{s} &= \underbrace{\hat{D}^{-1}\left(\tilde{\tau}_F - \tau_f + \tau_{dis} - \mathcal{Y}(q,\dot{q},\ddot{q})\tilde{P}\right)}_{\eta_t} \\ &\quad -K_1 s - K_2 \mathrm{sig}^\gamma(s) - K_\alpha \hat{\tau}_F - K_\beta \int_0^t \hat{\tau}_F dt, \\ &= \eta_t - K_1 s - K_2 \mathrm{sig}^\gamma(s) - K_\alpha \hat{\tau}_F - K_\beta \int_0^t \hat{\tau}_F dt. \end{aligned} \tag{5.65}$$

Due to (5.62),

$$\begin{aligned} K_\alpha \hat{\tau}_F + K_\beta \int_0^t \hat{\tau}_F dt &= K_\alpha h_0 \ddot{w} + (K_\alpha h_1 + K_\beta h_0)\dot{w} + K_\beta h_1 w, \\ &:= K_\alpha h_0 \dot{s} + K_\beta h_1 \Lambda^{-1} s, \end{aligned} \tag{5.66}$$

The purpose of (5.66) is to translate a function of $\ddot{w}(t)$, $\dot{w}(t)$, and $w(t)$ into a function of $\dot{s}(t)$ and $s(t)$, for convenience of stability analysis. Equation (5.66) holds true if

and only if the following condition satisfies:

$$K_\alpha h_0 \Lambda + K_\beta h_1 \Lambda^{-1} = K_\alpha h_1 + K_\beta h_0. \tag{5.67}$$

In (5.67), all matrices are diagonal with same dimension. After simple algebraic operations, it can be deduced that

$$K_\beta = K_\alpha \Lambda = \Lambda K_\alpha, \tag{5.68}$$

which is independent of h_0 and h_1. Inserting (5.68) into (5.66) gives

$$K_\alpha \hat{\tau}_{\mathrm{F}} + K_\beta \int_0^t \hat{\tau}_{\mathrm{F}} dt = K_\alpha h_0 \dot{s} + K_\alpha h_1 s. \tag{5.69}$$

Substituting (5.69) into (5.65) produces

$$\begin{aligned} \left(I_{3\times 3} + K_\alpha h_0\right) \dot{s} &= \eta_{\mathrm{t}} - \left(K_1 + K_\alpha h_1\right) s - K_2 \mathrm{sig}^\gamma(\mathrm{s}), \\ &= \eta_{\mathrm{t}} - K_1' s - K_2 \mathrm{sig}^\gamma(\mathrm{s}). \end{aligned} \tag{5.70}$$

Define a Lyapunov function

$$V(s) = \frac{1}{2} s^{\mathrm{T}} \left(I_{3\times 3} + K_\alpha h_0\right) s. \tag{5.71}$$

Time-derivative of (5.71) is

$$\begin{aligned} \dot{V}(s) &= s^{\mathrm{T}} \left(I_{3\times 3} + K_\alpha h_0\right) \dot{s}, \\ &= s^{\mathrm{T}} \left(\eta_{\mathrm{t}} - K_1' s - K_2 \mathrm{sig}^\gamma(\mathrm{s})\right). \end{aligned} \tag{5.72}$$

Temporarily, assume η_{t} is upper bounded and $\|\eta_{\mathrm{t}}\| \leq \eta_{\mathrm{c}}$, $k'_{1,\min}$ is the minimum element of K_1', then

$$\dot{V} \leq - \|s\| \left(k'_{1,\min} \|s\| + k_{2,\min} \|s\|^\gamma - \eta_{\mathrm{c}}\right). \tag{5.73}$$

Note that (5.73) has the same form as (5.45), and thus same arguments can be applied. Roughly speaking, by appropriately choosing the gain matrices, K_2, K_2, and K_α, inequality (5.73) is ensured for $\|s\| \geq \|s\|_*$, where $\|s\|_*$ is the unique solution of the equation

$$k'_{1,\min} \|s\| + k_{2,\min} \|s\|^\gamma - \eta_c = 0.$$

Therefore, $\{s | \|s\| \leq \|s\|_*\}$ is an invariant set. In theory, once the sliding surface s enters this invariant set, it does not escape. Since $s(0) = 0$, the sliding surface is entrapped inside the invariant set from $t = 0$. Also, the bound of the invariant set $\|s\|_*$ can be arbitrarily small, by choosing large $k'_{1,\min}$ or $k_{2,\min}$.

By (5.53), $w(t)$ is bounded, and

$$|w_i| \leq \frac{\|s\|_*}{\lambda_{\min}}. \tag{5.74}$$

By (5.55), $\tilde{q}$ is bounded by $\|\tilde{q}\| \leq \|w\| + \|v\|$ and hence $\|q\| \leq \|\tilde{q}\| + \|q_d\|$. By (5.56), $\dot{q}_r$ is bounded by $\|\dot{q}_r\| \leq \|\dot{q}_d\| + \|\dot{v}\| + \|\Lambda w\|$. By (5.54), $\dot{q}$ is bounded by $\|\dot{q}\| \leq \|s\| + \|\dot{q}_r\|$. Therefore, the system trajectories are uniformly bounded for $t \geq 0$. Since q and $\dot{q}$ are bounded, $\left\|\mathcal{Y}(q, \dot{q}, \ddot{q})\tilde{P}\right\| \leq m_0^s + m_1^s \|q\| + m_2^s \|\dot{q}\|^2$ is bounded, leading to that η_t is bounded, provided that τ_f is modeled as (5.22) and τ_{dis}^s is bounded. Let $\eta_c = \sup\left(\hat{D}^{-1}\left(\tilde{\tau}_F - \tau_f + \tau_{dis} - \mathcal{Y}(q, \dot{q}, \ddot{q})\tilde{P}\right)\right)$, $\|\eta_t\| \leq \|\eta_c\|$ is ensured.

From (5.74) and (5.55), (5.64) follows.

□

Note that the diagonal elements of $\hat{D}$ are all positive constants, and K_1 and K_2 are diagonal matrices. Consequently, in (5.60), $\hat{D}K_1$ and $\hat{D}K_2$ can be simplified as two diagonal positive definite matrices, respectively. Likewise, in (5.61), $\hat{D}K_\alpha$ and

$\hat{D}K_\beta$ can also be simplified as two diagonal positive definite matrices. Therefore, (5.60) and (5.61) can be written as

$$\tau_{\mathrm{p}} = B^{-1}\left(\hat{D}\ddot{q}_{\mathrm{r}} + \hat{H}\dot{q} - K_{\mathrm{a}}s - K_{\mathrm{b}}\mathrm{sig}^{\gamma}(\mathrm{s})\right), \tag{5.75}$$

$$\tau_{\mathrm{F}} = -B^{-1}\left((I_{3\times 3} + K_\alpha^{'})\hat{\tau}_{\mathrm{F}} + K_\beta^{'}\int_0^{\mathrm{t}}\hat{\tau}_{\mathrm{F}}dt\right), \tag{5.76}$$

where $K_{\mathrm{a}} \in \Re^{3\times 3}$, $K_{\mathrm{b}} \in \Re^{3\times 3}$, $K_\alpha^{'} \in \Re^{3\times 3}$, and $K_\beta^{'} \in \Re^{3\times 3}$ are diagonal, positive definite matrices, and $K_\beta^{'} = K_\alpha^{'}\Lambda = \Lambda K_\alpha^{'}$.

The tracking accuracy can be increased by using large K_1 and K_2. More excessive external forces can be suppressed by using large K_α and Λ. However, large gain matrices may result in system chattering and instability. Trade-offs must be made when choosing these values.

Online ZMP compensation

Re-writing (3.45), the ZMP location is

$$X_{\mathrm{zmp}} = \frac{\sum_{\mathrm{i}=1}^{3} M_{\mathrm{i}}\left[X_{\mathrm{g,i}}^{\mathrm{s}}\left(\ddot{Y}_{\mathrm{g,i}}^{\mathrm{s}} + g\right) - Y_{\mathrm{g,i}}^{\mathrm{s}}\ddot{X}_{\mathrm{g,i}}^{\mathrm{s}}\right] + \sum_{\mathrm{i}=1}^{3} I_{\mathrm{i}}\ddot{\theta}_{\mathrm{i}}}{M_{\mathrm{t}}\left(\ddot{Y}_{\mathrm{g}}^{\mathrm{s}} + g\right)}. \tag{5.77}$$

Equation (5.77) can be expanded into (see Appendix A.3.2):

$$X_{\mathrm{zmp}} = \frac{\sum_{\mathrm{i}=1}^{3} a_{\mathrm{i}}\ddot{\theta}_{\mathrm{i}} + \sum_{\mathrm{i}=1}^{3} b_{\mathrm{i}}\dot{\theta}_{\mathrm{i}}^2 + M_{\mathrm{t}}gX_{\mathrm{g}}}{M_{\mathrm{t}}\left(\sum_{\mathrm{i}=1}^{3} c_{\mathrm{i}}\ddot{\theta}_{\mathrm{i}} + \sum_{\mathrm{i}=1}^{3} d_{\mathrm{i}}\dot{\theta}_{\mathrm{i}}^2 + g\right)}, \tag{5.78}$$

where a_{i}, b_{i}, c_{i} and d_{i} are trigonometric functions of $q^{\mathrm{s}} = (\theta_1, \theta_2, \theta_3)^{\mathrm{T}}$.

The variational ranges of $\ddot{\theta}_{\mathrm{i}}$ and $\dot{\theta}_{\mathrm{i}}$ are usually very large, compared to the ranges of θ_{i}. Thus, a_{i}, b_{i}, c_{i}, and b_{i} are assumed nearly constant within a small sampling interval. Equation (5.78) can be re-organized as

$$\sum_{\mathrm{i}=1}^{3}\left(a_{\mathrm{i}} - X_{\mathrm{zmp}}M_{\mathrm{t}}c_{\mathrm{i}}\right)\ddot{\theta}_{\mathrm{i}} + \sum_{\mathrm{i}=1}^{3}\left(b_{\mathrm{i}} - X_{\mathrm{zmp}}M_{\mathrm{t}}d_{\mathrm{i}}\right)\dot{\theta}_{\mathrm{i}}^2 + M_{\mathrm{t}}g\left(X_{\mathrm{g}} - X_{\mathrm{zmp}}\right) = 0. \tag{5.79}$$

If the actual ZMP is located outside of the safety region, the robot is likely to fall. The boundary of the ZMP safety region is denoted by $X_{\text{zmp,b}}$, and the joint angles when the ZMP is at the boundary of the safety region are denoted by $\theta_{\text{i,b}}$ for $\text{i} = 1, 2, 3$, then

$$\sum_{\text{i=1}}^{3} \left(a_{\text{i}} - X_{\text{zmp,b}} M_{\text{t}} c_{\text{i}}\right) \ddot{\theta}_{\text{i,b}} + \sum_{\text{i=1}}^{3} \left(b_{\text{i}} - X_{\text{zmp,b}} M_{\text{t}} d_{\text{i}}\right) \dot{\theta}_{\text{i,b}}^2 + M_{\text{t}} g \left(X_{\text{g}} - X_{\text{zmp,b}}\right) = 0. \tag{5.80}$$

Let $\tilde{X}_{\text{zmp}} = X_{\text{zmp,b}} - X_{\text{zmp}}$ and $\tilde{\theta}_{\text{i}} = \theta_{\text{i,b}} - \theta_{\text{i}}$. Subtracting (5.79) from (5.80) yields

$$\sum_{\text{i=1}}^{3} \left(a_{\text{i}} - X_{\text{zmp,b}} M_{\text{t}} c_{\text{i}}\right) \ddot{\tilde{\theta}}_{\text{i}} + \sum_{\text{i=1}}^{3} \left(b_{\text{i}} - X_{\text{zmp,b}} M_{\text{t}} d_{\text{i}}\right) \dot{\tilde{\theta}}_{\text{i}} \left(\dot{\tilde{\theta}}_{\text{i}} + 2\dot{\theta}_{\text{i}}\right) = \tilde{X}_{\text{zmp}} F_{\text{y,b}}, \tag{5.81}$$

where $F_{\text{y,b}}$ is the vertical GRF when the ZMP is at the boundary of the safety region, and

$$F_{\text{y,b}} = M_{\text{t}} \left(\sum_{\text{i=1}}^{3} c_{\text{i}} \ddot{\theta}_{\text{i,b}} + \sum_{\text{i=1}}^{3} d_{\text{i}} \dot{\theta}_{\text{i,b}}^2 + g\right) = M_{\text{t}} \left(\ddot{Y}_{\text{g,b}}^{\text{s}} + g\right), \tag{5.82}$$

with $\ddot{Y}_{\text{g,b}}^{\text{s}}$ being the vertical acceleration of the hopper's CoM when the ZMP is at the boundary of the safety region. The actual value of $F_{\text{y,b}}$ is unknown and hard to predict since $\ddot{\theta}_{\text{i,b}}$ and $\dot{\theta}_{\text{i,b}}$ are unknown. In this work, the measured value F_{y} is used to replace $F_{\text{y,b}}$, based on an additional assumption that $F_{\text{y,b}} \approx F_{\text{y}}$. Thus,

$$\sum_{\text{i=1}}^{3} \left(a_{\text{i}} - X_{\text{zmp,b}} M_{\text{t}} c_{\text{i}}\right) \ddot{\tilde{\theta}}_{\text{i}} + \sum_{\text{i=1}}^{3} \left(b_{\text{i}} - X_{\text{zmp,b}} M_{\text{t}} d_{\text{i}}\right) \dot{\tilde{\theta}}_{\text{i}} \left(\dot{\tilde{\theta}}_{\text{i}} + 2\dot{\theta}_{\text{i}}\right) = \tilde{X}_{\text{zmp}} F_{\text{y}}, \tag{5.83}$$

To maintain the ZMP within the safety region, the pre-planned joint trajectories can be modified in real time according to (5.83). Note that (5.83) is a scalar equation with three unknowns ($\tilde{\theta}_{\text{i}}$ for $\text{i} = 1, 2, 3$) and there are infinite solutions to online ZMP compensation. In this work, the pitch angle of the upper body, *i.e.* θ_1, is chosen to be adjusted online. Thus, (5.83) can be simplified as

$$\left(a_1 - X_{\text{zmp,b}} M_{\text{t}} c_1\right) \ddot{\tilde{\theta}}_1 + \left(b_1 - X_{\text{zmp,b}} M_{\text{t}} d_1\right) \dot{\tilde{\theta}}_1 \left(\dot{\tilde{\theta}}_1 + 2\dot{\theta}_1\right) = \tilde{X}_{\text{zmp}} F_{\text{y}},$$

which can be changed into

$$\ddot{\tilde{\theta}}_1 = \frac{\tilde{X}_{\text{zmp}} F_{\text{y}}}{a_1 - X_{\text{zmp,b}} M_{\text{t}} c_1} - \frac{b_1 - X_{\text{zmp,b}} M_{\text{t}} d_1}{a_1 - X_{\text{zmp,b}} M_{\text{t}} c_1} \left(2\dot{\theta}_1 + \dot{\tilde{\theta}}_1\right) \dot{\tilde{\theta}}_1. \tag{5.84}$$

By assumption, X_{zmp} and F_{y} can be measured with satisfactory accuracy. Since θ_{i} for $\text{i} = 1, 2, 3$ can be measured accurately in real time, a_1, b_1, c_1, and d_1 can be computed online. Integrating twice both sides of (5.84), with the initial values of $\tilde{\theta}_1$ and $\dot{\tilde{\theta}}_1$ being set as zero, gives the compensation angle $\tilde{\theta}_1$.

The stability of the closed-loop control subsystem can be analyzed by using the same arguments for the stance phase, with q_{d}^{s} being modified online.

5.3 Simulations

Simulations have been carried out to verify the effectiveness of the control laws (5.32), (5.59), and the online ZMP compensator (5.84). The desired joint trajectories, *i.e.* q_{d}^{f}, q_{d}^{s} and their 1^{st} and 2^{nd} order derivatives, have been generated by using dynamic optimization reported in Chapter 3 (also in [42]).

5.3.1 Simulation strategies

First of all, it is worth pointing out that accurate position tracking for legged robots is not necessary. Instead, landing at the right time at the right foothold is more significant. More importantly, safe operation is always the primary concern. In the simulations, large convergence times T_{c}^{f} and T_{c}^{s} are chosen to achieve smooth joint trajectories. In general, T_{c} is chosen in $[0.85T_{\text{d}}, 0.95T_{\text{d}}]$ with T_{d} being the phase duration. If T_{c} is small, the actual joint trajectories converge to the desired references quickly, often resulting in large joint accelerations during the transient time. The

resultant gait can hardly be elegant. In the stance phase, quick convergence may cause the robot to lose its balance, since the ZMP is very sensitive to the joint accelerations, as suggested by (5.78). Moreover, the GRFs may become very large, due to the inter-dependency between the GRFs and the joint accelerations, as described by (5.82). As a result, the robot is likely to be damaged. Whenever the robot is at risk of a damage due to large GRFs or at risk of a fall due to loss of balance, a temporary retrieval motion has to be taken, sacrificing the accuracy of position tracking.

For rigid legged robots, the large GRFs may affect the measurement accuracy. In general, the joint angles can be measured with high accuracy. However, joint velocities may contain strong noise. Also, the real system may deviate from the nominal model due to parameter uncertainties. Therefore, in the simulations to be presented, the initial joint angles are randomly perturbed by $\pm 2\%$. The initial joint velocites and all system parameters are randomly perturbed by $\pm 5\%$, respectively. To simulate the viscous friction force τ_{f} and the external disturbances τ_{dis}, all viscous friction coefficients of the joints are set to be $0.01 \pm 5\% Nms$, and all joint torques are perturbed by white noise with zero mean and limited variance ($\sigma^2 \leq 2.0$).

Particularly, to simulate the external forces in the stance phase, two external forces, in horizontal and vertical directions, respectively, are applied to the robot's CoM. It is expected to see that they are suppressed by using the force-suppression module. For both the flight phase and the stance phase, γ is chosen to be 0.8.

5.3.2 The flight phase

In the flight phase, the control parameters are listed in Table 5.1. Figs. 5.2 and 5.3 show the joint angles and joint velocities. The solid curves represent the references,

Table 5.1: Parameters of the controller in the flight phase

Parameters	Values
Λ^{f}	$\mathrm{diag}(30, 30, 30)$
K_1^{f}	$\mathrm{diag}(25, 25, 25)$
K_2^{f}	$\mathrm{diag}(50, 50, 50)$

and the dashed curves are the actual joint trajectories. At the end of the flight phase, the actual joint trajectories converge to their references with small errors (Fig. 5.4). The three generalized coordinates are represented by solid, dotted, and dashed curves. These discrepancies, even small, may cause the ZMP at the beginning of the next stance phase out of the safety region. Fig. 5.5 shows the joint torques which appear noisy due to the noise intentionally added to the system. The joint torques are within their physical limits. As a comparison, Fig. 5.6 presents the joint torques when the traditional sliding mode $s = \dot{\tilde{q}} + \Lambda\tilde{q}$ is used. In this trial, no friction force and external disturbances are applied. Clearly, the controller proposed in this chapter performs better. Especially, in the beginning of the flight phase, the control signals do not jump dramatically to compensate for the deviation of the initial state. The proposed controller stabilizes the system from the very beginning, due to the absence of the reaching surface.

5.3.3 The stance phase

In the stance phase, the controller contains three modules: the position-tracking module, the force-suppression module, and the ZMP compensator. To verify the effectiveness of these control modules, three control schemes are tested:

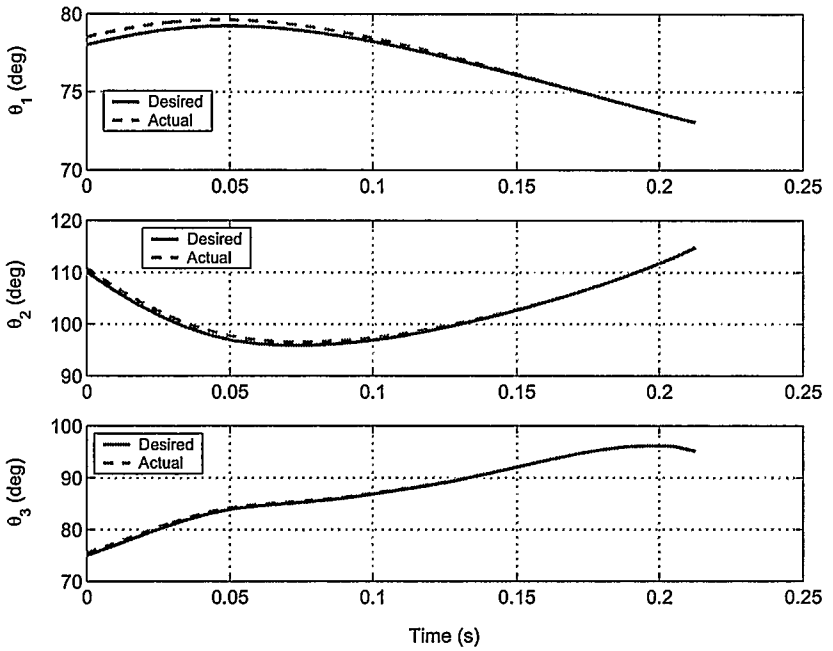

Figure 5.2: Joint angles in the flight phase

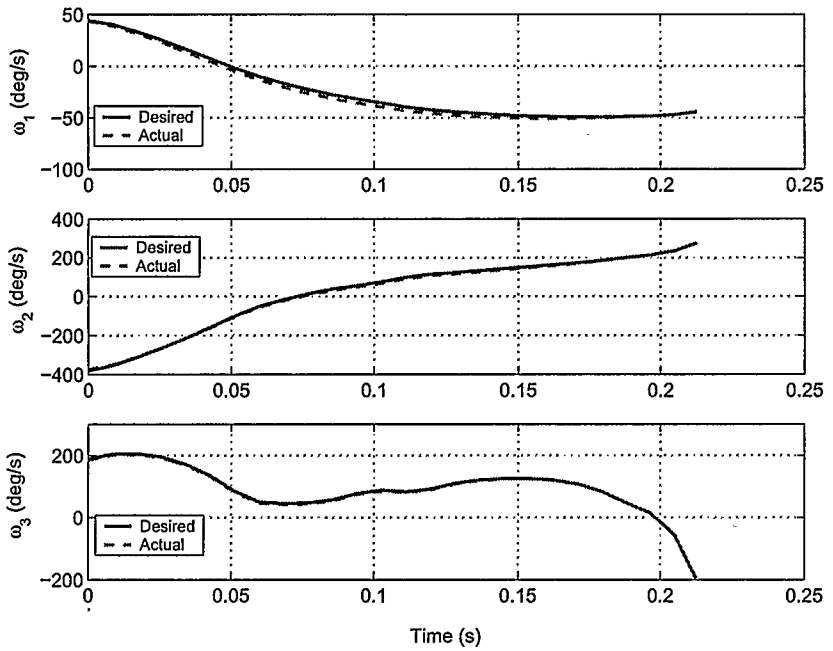

Figure 5.3: Joint velocities in the flight phase

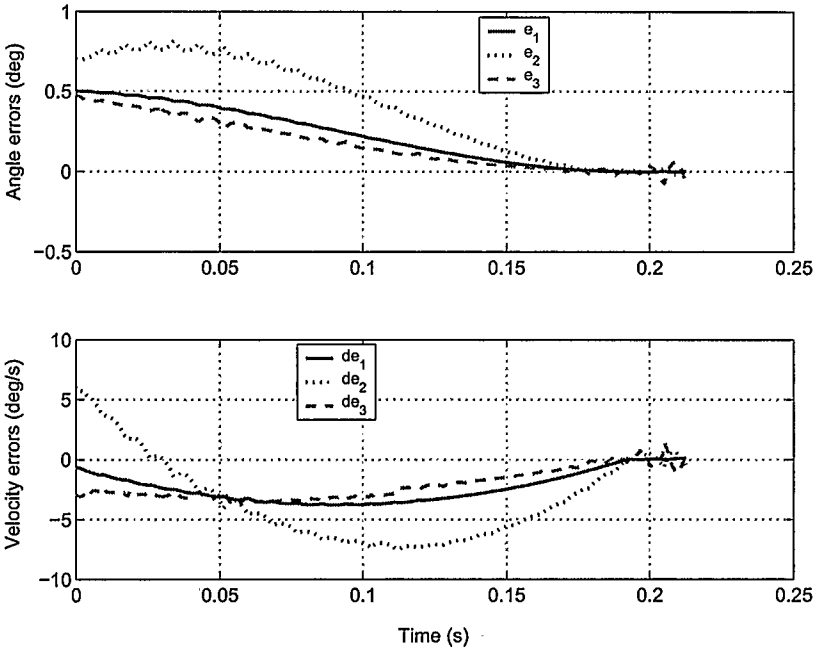

Figure 5.4: Joint errors in the flight phase

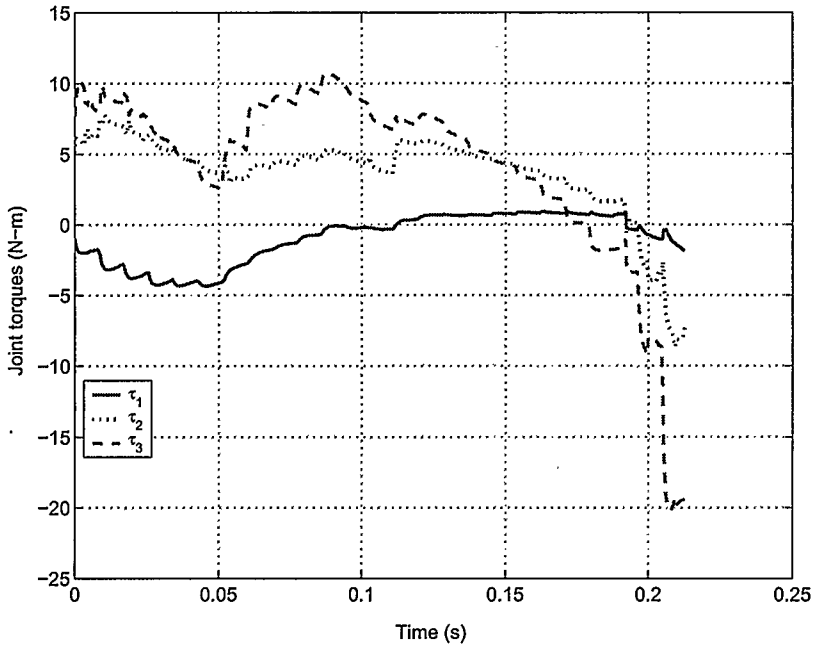

Figure 5.5: Joint torques in the flight phase

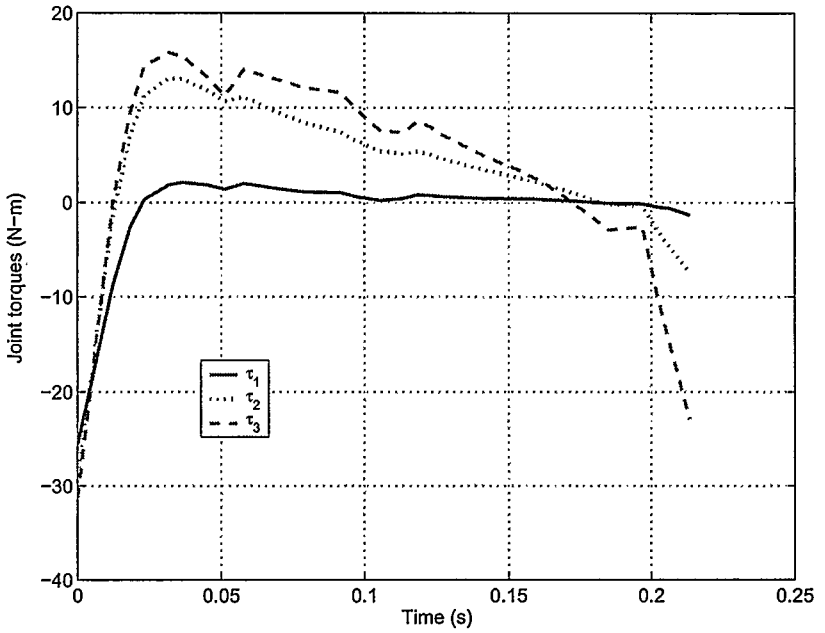

Figure 5.6: Joint torques in the flight phase when a traditional sliding mode is used

1. Scheme-1: only the position-tracking module is turned on;

2. Scheme-2: the position-tracking and the force-suppression modules are switched on;

3. Scheme-3: all of the three modules are applied simultaneously.

The parameters of the proposed controller are listed in Table 5.2.

The two external forces, with magnitude of $65N$ and $75N$, respectively, are acting at $0.0573s$ after the touchdown, and last for $0.01s$.

The results corresponding to the three control schemes are compared in the Figs. 5.7 - 5.14. The columns in Figs. 5.7 - 5.9 show θ_1, $\dot{\theta}_1$, θ_2, $\dot{\theta}_2$, θ_3, and $\dot{\theta}_3$, respectively. Each row in the figures correspond to the Scheme- 1, 2, and 3. Solid curves stand for the references, and dashed curves for the actual joint angles or actual joint velocities.

Table 5.2: Parameters of the controller in the stance phase

Parameters	Values
Λ^s	diag(35, 35, 35)
K_1^s	diag(44, 44, 44)
K_2^s	diag(50, 50, 50)
K_α	diag(10, 10, 10)
K_β	diag(350, 350, 350)

When Scheme-3 is applied, the actual θ_1 and $\dot{\theta}_1$ do not converge to the pre-planned references. The tracking accuracy is sacrificed such that the ZMP trajectory remains within the range of the support foot. Tracking errors are shown in Figs. 5.10 and 5.11. Note that when Scheme-3 is employed, the angle error and velocity error are actually $\theta_1^s + \tilde{\theta}_1^s - \theta_{1,d}^s$ and $\dot{\theta}_1^s + \dot{\tilde{\theta}}_1^s - \dot{\theta}_{1,d}^s$, respectively. After the convergence time T_c^s, all tracking errors are small. For the three control schemes, all final angle errors are within ± 0.0015 radians (± 0.086 degrees) and all final velocity errors are within ± 0.06 radians per second (± 3.44 degrees per second). The actual joint trajectories appear smooth, in the presence of parameter uncertainties, viscous friction, and external disturbances.

Fig. 5.12 demonstrates the joint torques. The solid, dashed, and dotted lines correspond to the Scheme- 1, 2, and 3, respectively. The torques are infected by noise added intentionally to the system. When the force-suppression module is active, immediately after the external forces are applied, the joint torques change abruptly to reject the external forces. The excessive vertical GRF is reduced from more than $240N$ to about $213.9N$ (the bottom graph of Fig. 5.14). As a contrast, the maximum horizontal GRF (in the top graph of Fig. 5.14) does not drop much, since it is less

than μF_y for $\mu = 0.6$. Clearly, the force-suppression module works.

Fig. 5.13 illustrates the ZMP trajectories for the three tests. The safety region is set to be identical to the foot range. In other words, if $-0.05m \leq X_{\text{zmp}} \leq 0.15m$ satisfy, the hopping gait is stable. When Scheme-1 is employed, the ZMP is pushed backward by the external forces (see Fig. 5.14) to the hind of the safety region during $0.05s \leq t \leq 0.135s$ (top graph of Fig. 5.13). Use of Scheme-2 relieves the risk of damage, but the risk of falling remains, since the ZMP trajectory violates the stability criterion twice (middle graph of Fig. 5.13). Scheme-3 helps to maintain the ZMP trajectory within the safety region, while the GRF profiles are within permitted values.

In simulations, the location of ZMP is calculated by using (5.78), which is sensitive to the joint accelerations. Consequently, the ZMP trajectory often goes outside of the safety region, even though the online ZMP compensator is switched on. Actually, for fast gaits, applicability of the ZMP stability criterion is controversial [71, 151, 153, 92]. In this work, the ZMP trajectory is treated as a weak stability indicator. A ZMP that is located outside of the safety region does not necessarily make a fall, but a ZMP that is closer to the center of the safety region has higher stability. The online ZMP compensator helps to improve the running stability, by sacrificing the tracking accuracy.

5.4 Summary

In this chapter, a novel hopping control algorithm was presented. In the flight phase, a position-tracking controller was proposed based on the finite-time stability

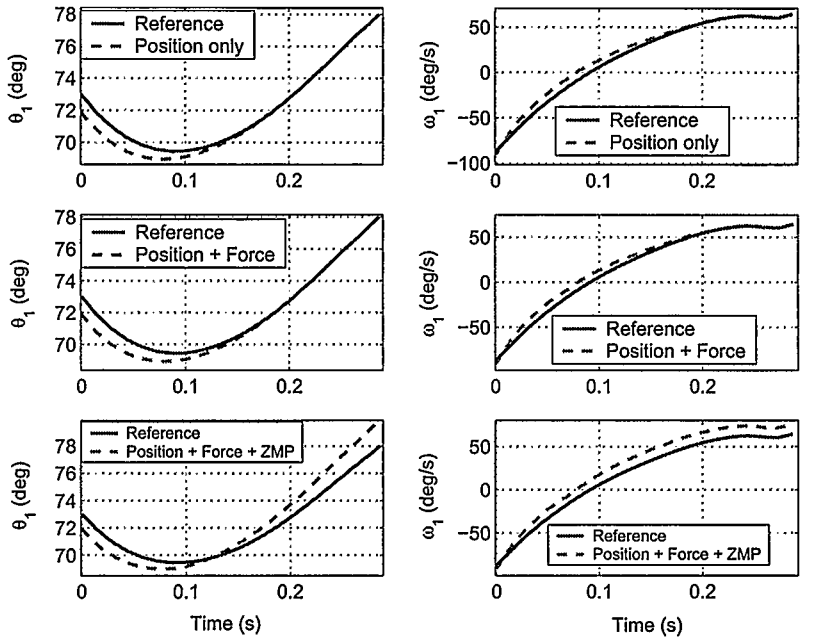

Figure 5.7: Comparison of the hip trajectories

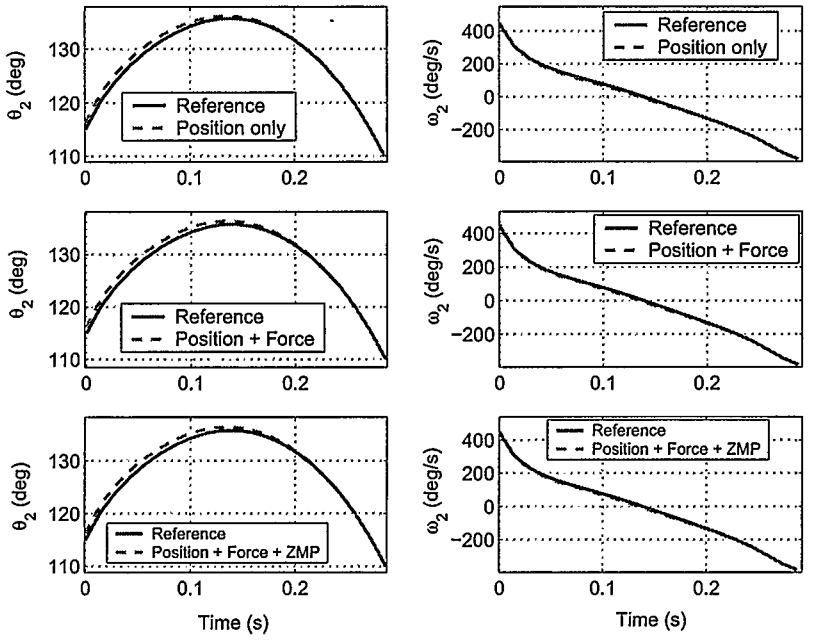

Figure 5.8: Comparison of the knee trajectories

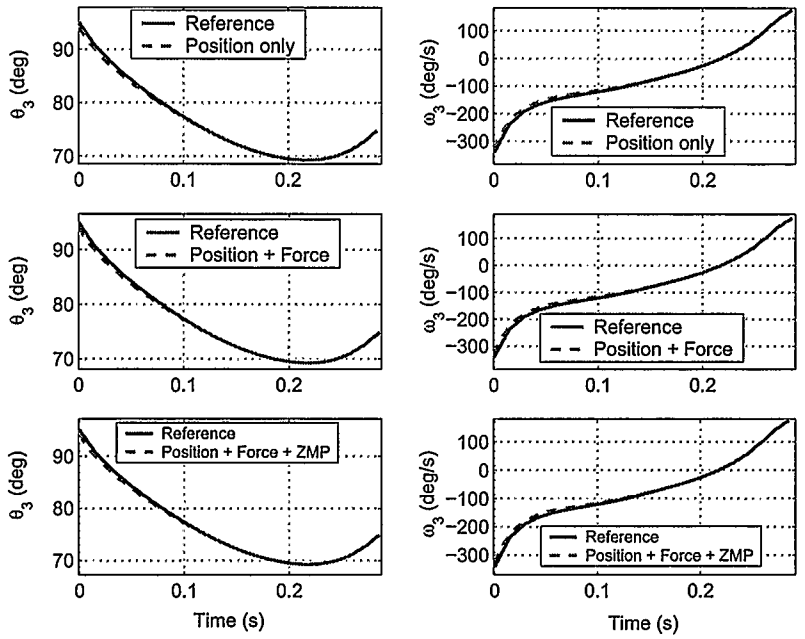

Figure 5.9: Comparison of the ankle trajectories

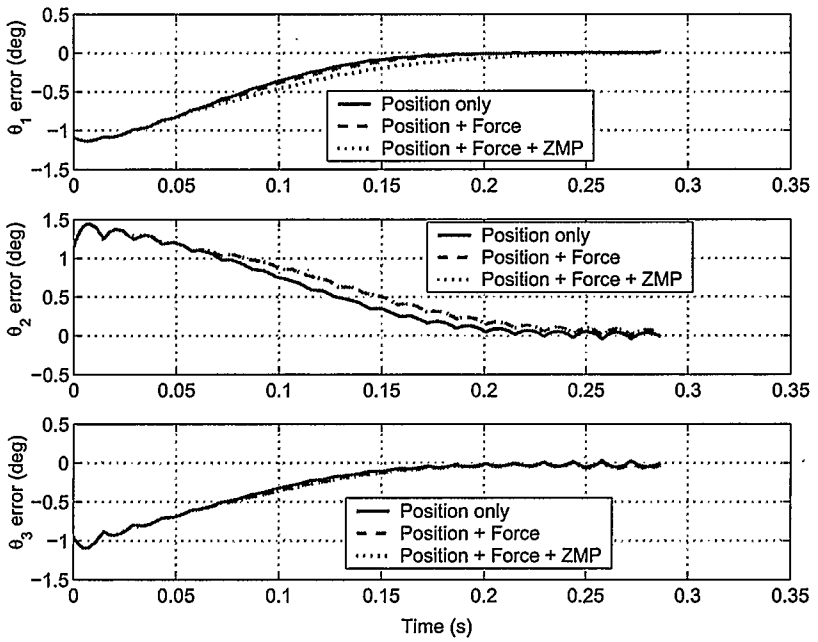

Figure 5.10: Comparison of the angle errors

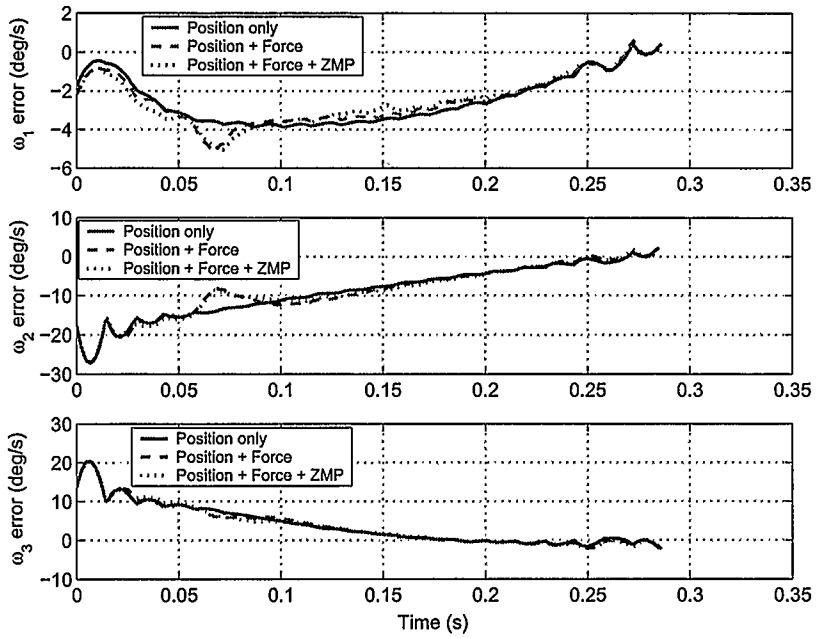

Figure 5.11: Comparison of the velocity errors

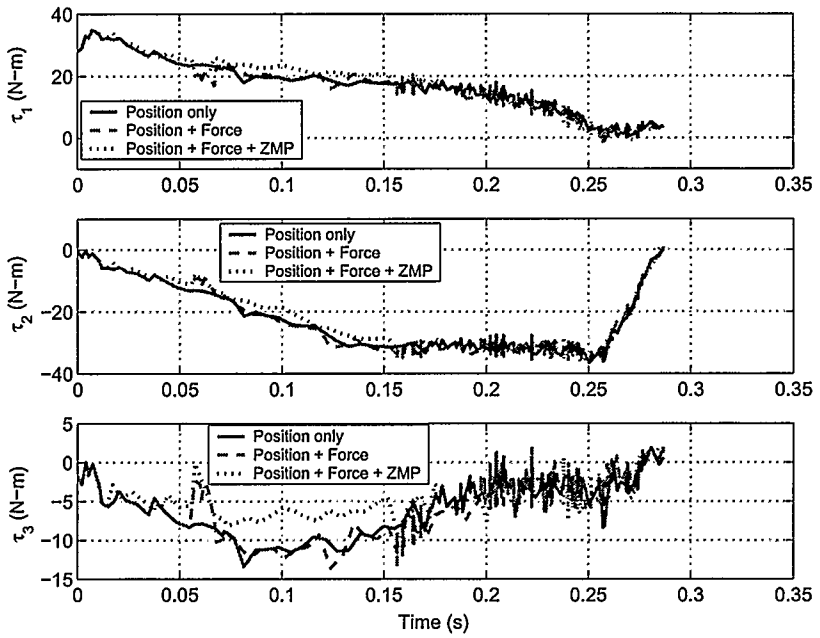

Figure 5.12: Comparison of the joint torques

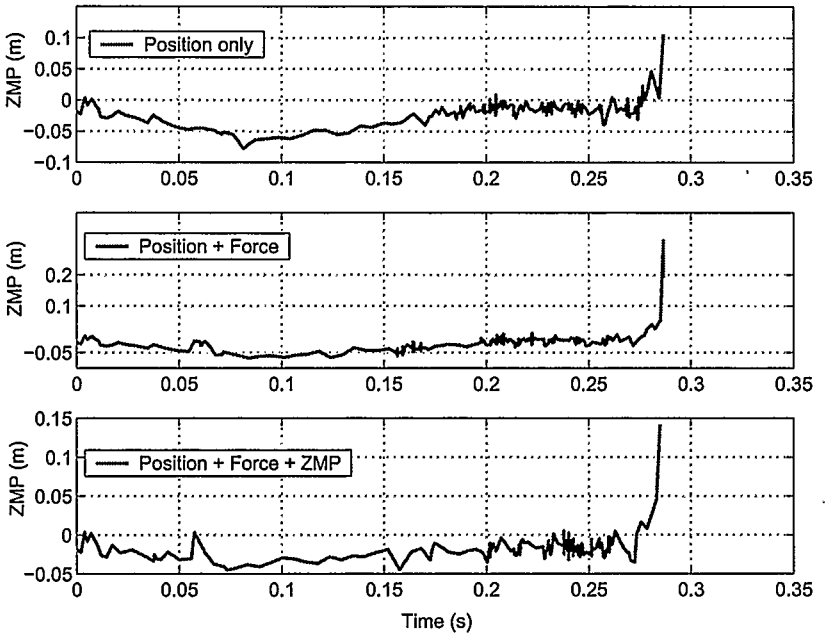

Figure 5.13: Comparison of the ZMP trajectories

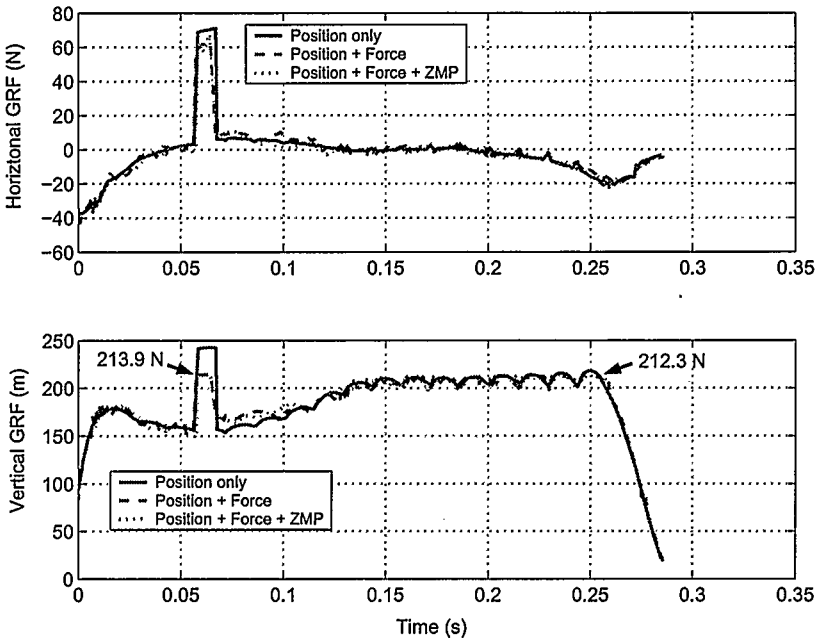

Figure 5.14: Comparison of the GRF trajectories

theory. The sliding surface is specially designed and no reaching surface is needed. The desired tracking errors are modeled as a set of 3^{rd} order polynomials and are embedded into the sliding surface. By appropriately choosing the control gains, the sliding surface can be maintained within a small neighborhood of zero. Thus, the actual tracking errors converge to the vicinity of the desired tracking errors. Since the convergence time can be arbitrarily chosen, the resultant joint trajectories can be very smooth, and the resultant flight phase appears elegant. In spite of the parameter uncertainties, the viscous friction in the actuators, and the unpredictable external disturbances, the system trajectories converge to the desired values in finite time with high accuracy.

In the stance phase, the controller is composed of three modules, the position-tracking module, the force-suppression module, and the online ZMP compensator. The position-tracking module is designed in the same way as that in the flight phase. PI terms, with the excessive external forces being the inputs, are adopted as the force-suppression module. The control gains can be chosen such that the tracking errors, including the position errors and the excessive external forces, converge to the vicinity of zero along the desired errors. Moreover, the online ZMP compensator aims to modify the desired hip angle in real time. If the actual ZMP tends to go outside of the safety region, this module is switched on to force the ZMP to move closer to the center of the safety region. Thus, the running stability can be improved.

Simulation experiments showed that the proposed control algorithm works satisfactorily for hopping of an articulated leg. One of the future work directions is to extend the algorithm to bipedal running control.

Chapter 6

Hybrid Finite-Time Control Designs for Bipedal Running

Compared with the one-legged hoppers, the bipedal robots are more versatile since they can perform different gaits. Therefore, the bipedal robots have the potential to work in practical applications, and study of bipedal robots is in essence more significant than study of one-legged hoppers. In this chapter, the control algorithm presented in Chapter 5 is extended to bipedal running. The desired joint trajectories are obtained by the offline running gait generator discussed in Chapter 4.

The hopper shown in Chapters 3 and 5 is a fully-actuated system with as many control inputs as system DOFs [17]. In contrast, the running biped under study is an over-actuated system in which the number of actuators is more than the number of system DOFs [77], without considering the two DOFs of the CoM trajectory in the flight phase (since they cannot be manipulated by changing the joint torques). This difference implies that the control algorithm for the one-legged hopper cannot directly be applied to bipedal running if no modifications are made.

Fortunately, due to similarity between one-legged hopping and bipedal running (see Chapter 2), the control designs in this chapter are similar to those in Chapter 5. Section 6.1 presents the control algorithm for bipedal running. Section 6.3 demonstrates the simulation results. Some concluding remarks are given in 6.4.

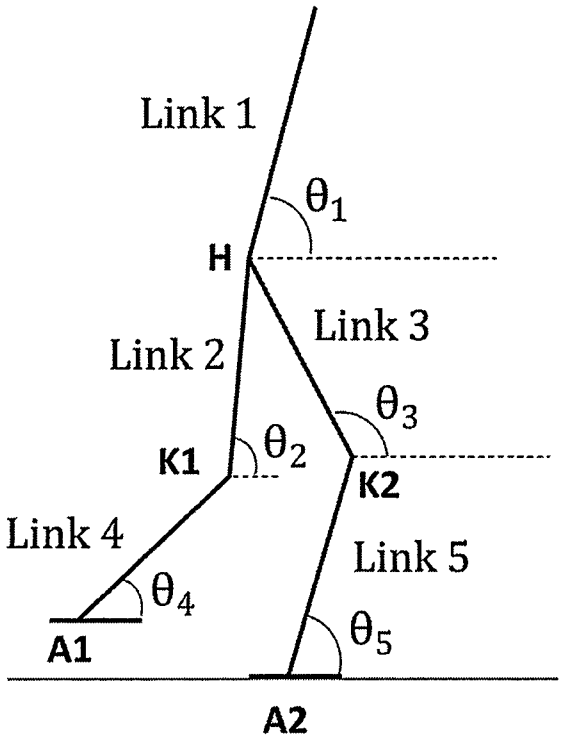

Figure 6.1: The model of the planar biped

6.1 The control algorithm

The control algorithm for bipedal running is briefly discussed in the following subsections. Stability analysis is ignored since it can be carried out in the same way as that in Chapter 5.

6.1.1 The bipedal model

The biped model is re-drawn in Fig. 6.1. The meanings of all parameters are identical to those in Section 4.1.

6.1.2 The flight phase

In the flight phase, the biped has 7 DOFs, and the vector of generalized coordinates can be chosen as $q^{\mathrm{f}} = (X_{\mathrm{g}}, Y_{\mathrm{g}}, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^{\mathrm{T}}$.

The practical model of the flight phase

At this moment, suppose that no parameter uncertainties and external disturbances are considered. By denoting $q_1^{\mathrm{f}} = (X_{\mathrm{g}}, Y_{\mathrm{g}})^{\mathrm{T}}$, and $q_2^{\mathrm{f}} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^{\mathrm{T}}$, the flight phase can be modeled as

$$D_1^{\mathrm{f}} \ddot{q}_1^{\mathrm{f}} + G_1^{\mathrm{f}} = 0, \tag{6.1}$$

$$D_2^{\mathrm{f}}(q_2^{\mathrm{f}}) \ddot{q}_2^{\mathrm{f}} + H_2^{\mathrm{f}}(q_2^{\mathrm{f}}, \dot{q}_2^{\mathrm{f}}) \dot{q}_2^{\mathrm{f}} = B_2 \tau, \tag{6.2}$$

where $D_1^{\mathrm{f}} = \mathrm{diag}(\mathrm{Mt}, \mathrm{Mt})$, $G_1^{\mathrm{f}} = (0, M_{\mathrm{t}} g)$, $D_2^{\mathrm{f}} \in \Re^{5\times5}$ is the symmetric, positive definite inertia matrix, $D_2^{\mathrm{f}} \in \Re^{5\times5}$ is the matrix containing the Centrifugal and Coriolis terms, $\tau = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6)^{\mathrm{T}}$, and $B_2 \in \Re^{5\times6}$ is a constant coefficient matrix of τ. In the remaining of this subsection, the superscript "f" will be dropped if no confusion is introduced.

Equation (6.1) implies that the CoM of the biped cannot be manipulated in the flight phase. The control actions can only influence the joint angles, governed by (6.2). In (6.2), the number of generalized coordinates is 5, and the number of control inputs is 6. Therefore, equation (6.2) describes an over-actuated system.

Considering the parameter uncertainties, viscous frictions, and the external disturbances, the model (6.2) is extended as:

$$D_2(q_2)\ddot{q}_2 + H_2(q_2, \dot{q}_2)\dot{q}_2 = B_2\tau - \tau_{\mathrm{f}} + \tau_{\mathrm{dis}}, \tag{6.3}$$

where τ_{f} contains the torques caused by viscous friction forces, τ_{dis} represents the torques due to external disturbances. D_2 and H_2 are composed of nominal and uncertain parts, respectively. D_2 is bounded, H_2 is chosen such that $\dot{D}_2 - 2H_2$ is skew-symmetric, and the LHS of (6.3) is linear in parameters, *i.e.*,

$$D_2\ddot{q}_2 + H_2\dot{q}_2 = \mathcal{Y}(q_2, \dot{q}_2, \ddot{q}_2)P, \tag{6.4}$$

with $\mathcal{Y}(q_2, \dot{q}_2, \ddot{q}_2)$ and P being the regressor and parameter vector.

The following assumptions are made:

1. The external disturbance torque τ_{dis} is bounded by $\|\tau_{\mathrm{dis}}\| \leq \eta_{\mathrm{d}}$.

2. The viscous friction torque can be modeled as [108]

$$\tau_{\mathrm{f}} = K_{\mathrm{f}}\dot{q}_2, \tag{6.5}$$

   where $K_{\mathrm{f}} = \mathrm{diag}(\mathrm{k_{f,1}}, \mathrm{k_{f,2}}, \cdots, \mathrm{k_{f,5}})$ with $k_{\mathrm{f,i}} > 0$ for $\mathrm{i} = 1, 2, \cdots, 5$, and denote $k_{\mathrm{f,max}} = \max(\mathrm{k_{f,1}}, \mathrm{k_{f,2}}, \cdots, \mathrm{k_{f,5}})$.

3. Suppose the nominal parameter is $\hat{P}$. The system uncertainty $\mathcal{Y}(P - \hat{P}) = \mathcal{Y}\tilde{P}$ is assumed to be bounded by $\left\|\mathcal{Y}\tilde{P}\right\| \leq m_0 + m_1 \|q_2\| + m_2 \|\dot{q}_2\|^2$ [31, 159], where $m_{\mathrm{i}} > 0$ for $\mathrm{i} = 0, 1, 2$.

4. The complex effect of system uncertainties, viscous frictions, and external disturbances is bounded, *i.e.*,

$$\left\|\mathcal{Y}\tilde{P}\right\| + \|K_{\mathrm{f}}\dot{q}_2\| + \|\tau_{\mathrm{dis}}\| \leq \eta_{\mathrm{t}}(q_2, \dot{q}_2),$$

   where $\eta_{\mathrm{t}}(q_2, \dot{q}_2) = m_0 + m_1 \|q_2\| + m_2 \|\dot{q}_2\|^2 + k_{\mathrm{f,max}} \|\dot{q}_2\| + \eta_{\mathrm{d}}$.

Control design for the flight phase

Define the tracking error

$$\tilde{q}_2 = q_2 - q_{2,\mathrm{d}}, \tag{6.6}$$

where $q_{2,\mathrm{d}}$ consists of the desired joint angles.

Define a sliding surface

$$\begin{aligned} s &= \dot{\tilde{q}}_2 + \Lambda\tilde{q}_2 - \dot{v}(t) - \Lambda v(t), && (6.7)\\ &= \dot{w}(t) + \Lambda w(t), && (6.8)\\ &= \dot{q}_2 - \dot{q}_{2,\mathrm{r}}, && (6.9) \end{aligned}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_5)$, $\lambda_\mathrm{i} > 0$ for $\mathrm{i} = 1, 2, \cdots, 5$, $\lambda_{\min} = \min(\lambda_1, \lambda_2, \cdots, \lambda_5)$, $v(t)$ is the desired tracking error, and

$$\begin{aligned} \dot{q}_{2,\mathrm{r}} &= \dot{q}_{2,\mathrm{d}} - \Lambda\tilde{q}_2 + \dot{v}(t) + \Lambda v(t), && (6.10)\\ w(t) &= \tilde{q}_2 - v(t). \end{aligned}$$

The desired error trajectory $v(t)$ is defined as (5.28). Note that $v(0) = \tilde{q}_2(0)$, $\dot{v}(0) = \dot{\tilde{q}}_2(0)$, $w(0) = 0$ and $s(0) = 0$.

Differentiating both sides of (6.10) gives

$$\ddot{q}_{2,\mathrm{r}} = \ddot{q}_{2,\mathrm{d}} - \Lambda\dot{\tilde{q}}_2 + \ddot{v}(t) + \Lambda\dot{v}(t). \tag{6.11}$$

Theorem 6.1.1. *For the system (6.3), if the following controller is applied:*

$$\begin{aligned} \tau &= B_2^{\#}\left(\hat{D}_2\ddot{q}_{2,\mathrm{r}} + \hat{H}_2\dot{q}_{2,\mathrm{r}} - K_1 s - K_2\mathrm{sig}^{\gamma}(\mathrm{s})\right),\\ &= B_2^{\#}\left(\mathcal{Y}(q_2, \dot{q}_2, \dot{q}_{2,\mathrm{r}}, \ddot{q}_{2,\mathrm{r}})\hat{P} - K_1 s - K_2\mathrm{sig}^{\gamma}(\mathrm{s})\right), && (6.12) \end{aligned}$$

where $B_2^{\#} = B_2^{\mathrm{T}}(B_2B_2^{\mathrm{T}})^{-1} \in \Re^{6\times 5}$ *is the Moore-Penrose pseudo-inverse of* B_2 *[53],* $K_1 = \mathrm{diag}(\mathrm{k}_{1,1}, \mathrm{k}_{1,2}, \cdots, \mathrm{k}_{1,5})$, $K_2 = \mathrm{diag}(\mathrm{k}_{2,1}, \mathrm{k}_{2,2}, \cdots, \mathrm{k}_{2,5})$, $k_{\mathrm{i,j}} > 0$ *for* $\mathrm{i} = 1, 2; \mathrm{j} =$

$1, 2, \cdots, 5$, $\hat{D}_2$ *and* $\hat{H}_2$ *are the nominal parts of* D_2 *and* H_2, *respectively, then the system trajectories are uniformly bounded for* $t \geq 0$, *and the bound of the sliding surface is* $\|s\| \leq \|s\|_*$, *where* $\|s\|_*$ *is the unique solution of the equation*

$$k_{1,\min} \|s\| + k_{2,\min} \|s\|^{\gamma} - \eta_c = 0, \tag{6.13}$$

with $\gamma \in (0,1)$, $k_{1,\min} = \min(k_{1,1}, k_{1,2}, \cdots, k_{1,5})$, $k_{2,\min} = \min(k_{2,1}, k_{2,2}, \cdots, k_{2,5})$, *and*

$$\eta_c = \sup_{q_2, \dot{q}_2 \in \Re^5} (\eta_t(q_2, \dot{q}_2)). \tag{6.14}$$

Moreover, the tracking errors are bounded by

$$|v_i| - \frac{\|s\|_*}{\lambda_{\min}} \leq |\tilde{q}_{2,i}| \leq |v_i| + \frac{\|s\|_*}{\lambda_{\min}}, \tag{6.15}$$

where $\tilde{q}_{2,i}$ *and* v_i *are the* i^{th} *elements of* $\tilde{q}_2$ *and* v, *respectively.*

Proof: Multiplying both sides of (6.12) by B_2 gives:

$$B_2 \tau = \mathcal{Y}(q_2, \dot{q}_2, \dot{q}_{2,r}, \ddot{q}_{2,r}) \hat{P} - K_1 s - K_2 \mathrm{sig}^{\gamma}(s). \tag{6.16}$$

Mimicking the proof of Theorem 5.2.4, the conclusions can easily be drawn.

□

6.1.3 The stance phase

In the stance phase, the robot possesses 5 DOFs. The vector of generalized coordinates is chosen as $q^s = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^T$.

The practical model of the stance phase

The practical model of the stance phase may be expressed as

$$D^s(q^s)\ddot{q}^s + H^s(q^s, \dot{q}^s)\dot{q}^s + G^s(q^s) = B\tau - \tau_f^s + J_g^T(q^s)\tilde{F} + \tau_{dis}^s, \tag{6.17}$$

where $D^{\mathrm{s}}(q^{\mathrm{s}}) \in \Re^{5\times5}$, $H^{\mathrm{s}}(q^{\mathrm{s}}, \dot{q}^{\mathrm{s}}) \in \Re^{5\times5}$, and $G^{\mathrm{s}}(q^{\mathrm{s}}) \in \Re^{5}$ are the symmetric, positive definite inertia matrix, the matrix containing the Centrifugal and Coriolis terms, and the gravitational vector, respectively, $\tau \in \Re^{6}$ is the vector of joint torques, $B \in \Re^{5\times6}$ is the constant coefficient matrix of τ, $\tau_{\mathrm{f}}^{\mathrm{s}}$ represents the torque caused by the viscous friction forces, $J_{\mathrm{g}}(q^{\mathrm{s}}) \in \Re^{2\times5}$ is the Jacobian mapping from the robot's CoM to the ankle of the support leg, $\tilde{F}$ is the vector of excessive external forces equivalently applied to the robot's CoM, and $\tau_{\mathrm{dis}}^{\mathrm{s}}$ represents other external disturbances. For clarity, the superscript "s" disappears in the remaining of this subsection, unless ambiguity is introduced.

With the same arguments as those in Chapter 5, the excessive external forces can be mapped to the joint space via the Jacobian J_{g}:

$$\tau_{\mathrm{F}} = J_{\mathrm{g}}^{\mathrm{T}}(q)\tilde{F}.$$

Without the superscript "s", the stance model (6.17) can be re-written as

$$\begin{aligned} \hat{D}\ddot{q} + \hat{H}\dot{q} + \hat{G} &= B\tau - \tau_{\mathrm{f}} + \tau_{\mathrm{F}} + \tau_{\mathrm{dis}} - \tilde{D}\ddot{q} - \tilde{H}\dot{q} - \tilde{G}, \\ &= B\tau - \tau_{\mathrm{f}} + \tau_{\mathrm{F}} + \tau_{\mathrm{dis}} - \mathcal{Y}(q, \dot{q}, \ddot{q})\tilde{P}, \end{aligned} \tag{6.18}$$

where $\hat{(\cdot)}$ and $\tilde{(\cdot)}$ are the nominal and uncertain parts of $(\cdot)$, and $\hat{(\cdot)} + \tilde{(\cdot)} = (\cdot)$.

Control design for the stance phase

The tracking error is defined as

$$\tilde{q} = q - q_{\mathrm{d}}. \tag{6.19}$$

The desired error trajectory $v(t)$ is defined in a similar way as (5.28), with the convergence time being $T_{\mathrm{c}}^{\mathrm{s}}$.

Define a sliding surface

$$\begin{aligned} s(t) &= \dot{\tilde{q}} + \Lambda\tilde{q} - \dot{v}(t) - \Lambda v(t), && (6.20)\\ &= \dot{w}(t) + \Lambda w(t), && (6.21)\\ &= \dot{q} - \dot{q}_{\mathrm{r}}, && (6.22) \end{aligned}$$

where

$$\begin{aligned} w(t) &= \tilde{q} - v(t), && (6.23)\\ \dot{q}_{\mathrm{r}} &= \dot{q}_{\mathrm{d}} - \Lambda\tilde{q} + \dot{v} + \Lambda v. && (6.24) \end{aligned}$$

The absolute tracking error $w(t)$, with the bias $v(t)$ removed, is expected to evolve along zero. Differentiating both sides of (6.24) leads to

$$\ddot{q}_{\mathrm{r}} = \ddot{q}_{\mathrm{d}} - \Lambda\dot{\tilde{q}} + \ddot{v} + \Lambda\dot{v}. \tag{6.25}$$

Clearly, $w(0) = 0$ and $s(0) = 0$.

Denote the estimate of τ_{F} by $\hat{\tau}_{\mathrm{F}}$, and

$$\hat{\tau}_{\mathrm{F}} = \hat{J}_{\mathrm{g}}^{\mathrm{T}}(q)\tilde{F}, \tag{6.26}$$

where $\hat{J}_{\mathrm{g}}(q)$ is the estimate of $J_{\mathrm{g}}(q)$.

Theorem 6.1.2. *For the system (6.18), if*

1. *the following controller is applied:*

$$\tau = \tau_{\mathrm{p}} + \tau_{\mathrm{F}}, \tag{6.27}$$

with

$$\tau_{\mathrm{p}} = \underbrace{B^{\#}\left(\hat{D}\ddot{q}_{\mathrm{r}} + \hat{H}\dot{q} + \hat{D} - \hat{D}K_1 s - \hat{D}K_2 \mathrm{sig}^{\gamma}(\mathrm{s})\right)}_{position-tracking\ module}, \tag{6.28}$$

$$\tau_{\mathrm{F}} = \underbrace{-B^{\#}\left((I_{5\times5} + \hat{D}K_{\alpha})\hat{\tau}_{\mathrm{F}} + \hat{D}K_{\beta}\int_0^{\mathrm{t}} \hat{\tau}_{\mathrm{F}} dt\right)}_{force-suppression\ module}, \tag{6.29}$$

where $B^{\#} = B^{\mathrm{T}}(BB^{\mathrm{T}})^{-1} \in \Re^{6\times5}$ *is the Moore-Penrose pseudo-inverse of* B, $K_1 = \mathrm{diag}(\mathrm{k}_{1,1}, \mathrm{k}_{1,2}, \cdots, \mathrm{k}_{1,5})$, $K_2 = \mathrm{diag}(\mathrm{k}_{2,1}, \mathrm{k}_{2,2}, \cdots, \mathrm{k}_{2,5})$, $k_{\mathrm{i,j}} > 0$ *for* $\mathrm{i} = 1, 2; \mathrm{j} = 1, 2, \cdots, 5$, $K_{\alpha} \in \Re^{5\times5}$ *and* $K_{\beta} \in \Re^{5\times5}$ *are diagonal, positive definite matrices;*

2. *to suppress the external forces, each joint is desired to behave like a damper, complying with*

$$\hat{\tau}_{\mathrm{F}} = h_0\ddot{w} + h_1\dot{w}, \tag{6.30}$$

where $h_0 \in \Re^{5\times5}$ *and* $h_1 \in \Re^{5\times5}$ *are diagonal, positive definite gain matrices, representing the desired moment of inertia and the desired damping ratio.* h_0 *and* h_1 *can be picked arbitrarily to satisfy the design specifications;*

then

1. *the system trajectories are uniformly bounded for* $t \geq 0$, *and the bound of the sliding surface is* $\|s\| \leq \|s\|_*$, *where* $\|s\|_*$ *is the unique solution of the equation*

$$k'_{1,\min}\|s\| + k_{2,\min}\|s\|^{\gamma} - \eta_{\mathrm{c}} = 0, \tag{6.31}$$

with $\gamma \in (0, 1)$, $k'_{1,\min}$ *is the minimum element of the matrix* $K'_1 = K_1 + K_{\alpha}h_1$, $k_{2,\min} = \min(\mathrm{k}_{2,1}, \mathrm{k}_{2,2}, \cdots, \mathrm{k}_{2,5})$, $\eta_{\mathrm{c}} = \sup\left\|\hat{D}^{-1}\left(\tilde{\tau}_{\mathrm{F}} - \tau_{\mathrm{f}} + \tau_{\mathrm{dis}} - \mathcal{Y}(q, \dot{q}, \ddot{q})\tilde{P}\right)\right\|$, *and* $\tilde{\tau}_{\mathrm{F}} = \tau_{\mathrm{F}} - \hat{\tau}_{\mathrm{F}}$.

2. the tracking errors are bounded by

$$|v_{\rm i}| - \frac{\|s\|_*}{\lambda_{\min}} \leq |\tilde{q}_{\rm i}| \leq |v_{\rm i}| + \frac{\|s\|_*}{\lambda_{\min}}, \tag{6.32}$$

where $\tilde{q}_{\rm i}$ and $v_{\rm i}$ are the i^{th} elements of $\tilde{q}$ and v, respectively.

Proof: Making use of the equality $BB^{\#} = I_{5\times 5}$, and mimicking the proof of Theorem 5.2.5, the conclusions of Theorem 6.1.2 follow.

□.

Online ZMP compensation

Re-writing (4.33), the ZMP location is

$$X_{\rm zmp} = \frac{\sum_{\rm i=1}^{5} M_{\rm i}\left[X^{\rm s}_{\rm g,i}\left(\ddot{Y}^{\rm s}_{\rm g,i}+g\right) - Y^{\rm s}_{\rm g,i}\ddot{X}^{\rm s}_{\rm g,i}\right] + \sum_{\rm i=1}^{5} I_{\rm i}\ddot{\theta}_{\rm i}}{M_{\rm t}\left(\ddot{Y}^{\rm s}_{\rm g}+g\right)}, \tag{6.33}$$

Equation (6.33) can be expanded into (see Appendix B.3.2)

$$X_{\rm zmp} = \frac{\sum_{\rm i=1}^{5} a_{\rm i}\ddot{\theta}_{\rm i} + \sum_{\rm i=1}^{5} b_{\rm i}\dot{\theta}_{\rm i}^2 + M_{\rm t}gX_{\rm g}}{M_{\rm t}\left(\sum_{\rm i=1}^{5} c_{\rm i}\ddot{\theta}_{\rm i} + \sum_{\rm i=1}^{5} d_{\rm i}\dot{\theta}_{\rm i}^2 + g\right)}, \tag{6.34}$$

where $a_{\rm i}$, $b_{\rm i}$, $c_{\rm i}$ and $d_{\rm i}$ are trigonometric functions of $q^{\rm s} = (\theta_1, \theta_2, \cdots, \theta_5)^{\rm T}$.

The boundary of the ZMP safety region is denoted by $X_{\rm zmp,b}$, and $\tilde{X}_{\rm zmp} = X_{\rm zmp,b} - X_{\rm zmp}$. The actual ZMP location can be changed by adjusting the torso pitch angle in real time. Denote the desired torso pitch angle when the ZMP is at the boundary of the ZMP safety region by $\theta_{1,\rm b}$, and $\tilde{\theta}_1 = \theta_{1,\rm b} - \theta_1$. With simple derivations similar to those in Chapter 5, it can be obtained that

$$\ddot{\tilde{\theta}}_1 = \frac{\tilde{X}_{\rm zmp}F_{\rm y}}{a_1 - X_{\rm zmp,b}M_{\rm t}c_1} - \frac{b_1 - X_{\rm zmp,b}M_{\rm t}d_1}{a_1 - X_{\rm zmp,b}M_{\rm t}c_1}\left(2\dot{\theta}_1 + \dot{\tilde{\theta}}_1\right)\dot{\tilde{\theta}}_1, \tag{6.35}$$

where $F_{\rm y}$ is the vertical GRF, as before.

Integrating twice both sides of (6.35), with the initial values of $\tilde{\theta}_1$ and $\dot{\tilde{\theta}}_1$ being set as zero, gives the compensation angle $\tilde{\theta}_1$. Note that the mass and inertia of the hind leg are small, and hence little placement change of the hind leg does not apparently influence the ZMP location. Adjustment of the torso pitch angle offers a better solution to biped stabilization.

Stability analysis of the closed-loop system can be conducted in the same way as that of the one-legged hopper.

6.2 Remarks on the control algorithm

The following remarks may help to understand the design idea of the control algorithm:

1. The basic idea of the control algorithm is essentially the same as that for the hopper. Both of them are based on the finite-time control theory, and Haimo's finite-time function plays important roles.

2. Since the biped is over-actuated, B_2 and B, the coefficient matrices of τ in different running phases, are not square. Thus, their corresponding Moore-Penrose pseudo-inverse matrices have to be used in the control designs.

3. For more complicated tasks, subtask optimizations can further be employed. This treatment has been widely adopted in redundant manipulators [69, 52, 164, 70, 142, 98].

6.3 Simulation results

Simulations have been done to verify the efficacy of the control laws (6.12), (6.27), and the online ZMP compensator (6.35). The desired joint trajectories are generated by using dynamic optimization reported in Chapter 4.

Same as Chapter 5, all system parameters are randomly perturbed by ±5% to mimic the parameter uncertainties. The initial joint angles and initial joint velocities are randomly perturbed by ±2% and ±5%, respectively, to account for the measurememt inaccuracy. Viscous friction coefficients take $0.01 \pm 5\% Nms$. All joints are infected by white noise with zero mean and variance of 2.0 (*i.e.* $\sigma^2 = 2.0$). For all running phases, $\gamma = 0.8$.

6.3.1 The flight phase

In the flight phase, the control parameters are listed in Table 6.1. Figs. 6.2 and 6.4 show θ_1, $\dot{\theta}_1$, θ_2, and $\dot{\theta}_2$, respectively, with the conventions that the solid curves stand for the references, and the dashed curves represent the actual angles and the actual angular velocities (see the legends in the graphs). The corresponding tracking errors are shown in Figs. 6.3 and 6.5. The evolutions of other angles and angular velocities come out similar, and hence are not presented here. At the end of the flight phase the angle errors are within ±0.004 radians (or ±0.23 degrees), and the velocity errors are within ±0.25 radians per second (or 14.4 degrees per second). For practical tasks, the tracking performance is satisfactory. In Fig. 6.6, from top to bottom, the torques, of the hips, the knees, and the ankles, are shown. The swing leg and the support leg are represented by the solid and dashed curves, respectively.

Table 6.1: Parameters of the controller in the flight phase

Parameters	Values
Λ^{f}	diag(15, 15, 15, 15, 15)
K_1^{f}	diag(30, 30, 30, 30, 30)
K_2^{f}	diag(150, 150, 150, 150, 150)

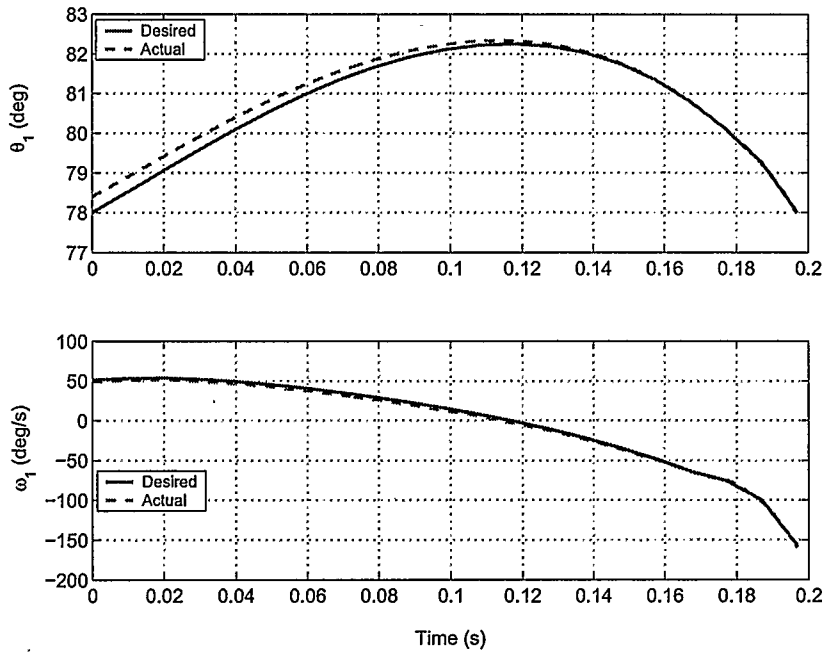

Figure 6.2: θ_1 and $\dot{\theta}_1$ in the flight phase

Clearly, they are infected by the noise added artificially to the system.

6.3.2 The stance phase

In the stance phase, at $0.0866s$ after the touchdown, two external forces with amplitude of $65N$ and $85N$, in horizontal and vertical directions, respectively, are applied to the biped's CoM, and they last for $0.01s$. Three simulation strategies, termed Scheme-1, Scheme-2, and Scheme-3, respectively, are tried. They are explained as

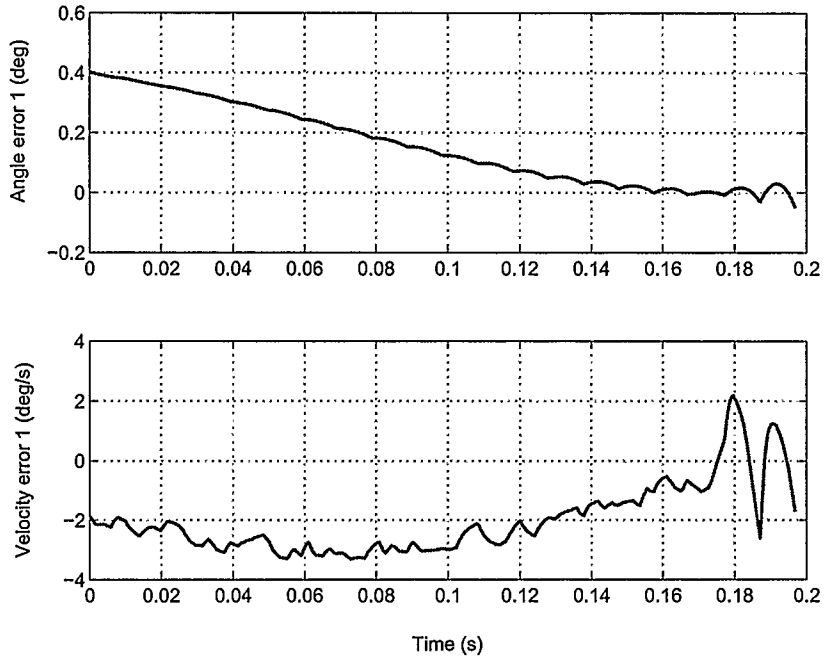

Figure 6.3: Tracking errors $\tilde{\theta}_1$ and $\dot{\tilde{\theta}}_1$ in the flight phase

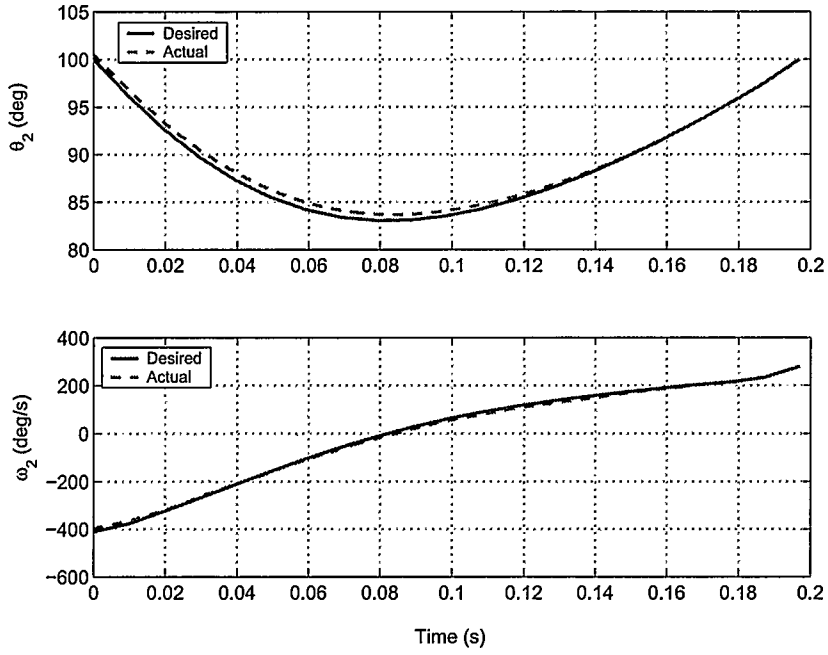

Figure 6.4: θ_2 and $\dot{\theta}_2$ in the flight phase

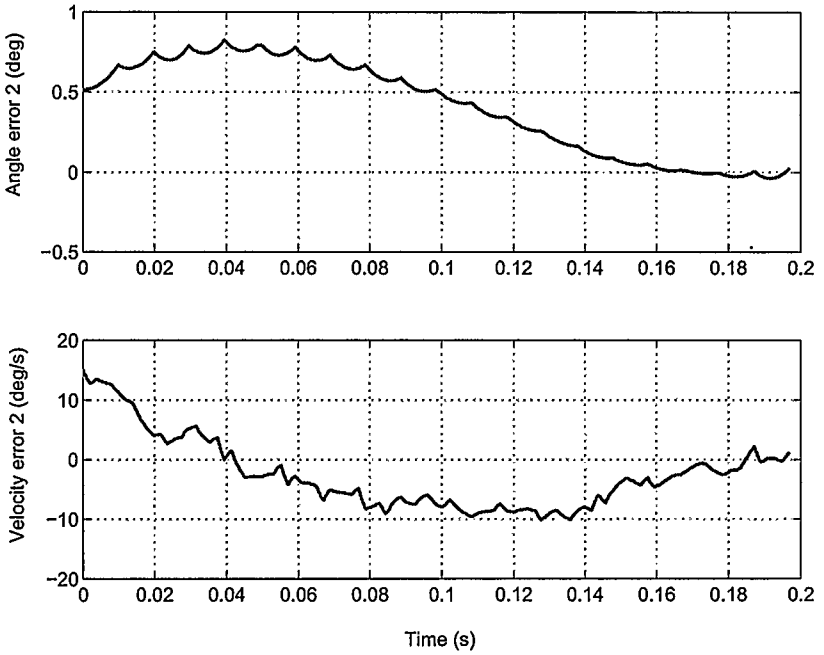

Figure 6.5: Tracking errors $\tilde{\theta}_2$ and $\dot{\tilde{\theta}}_2$ in the flight phase

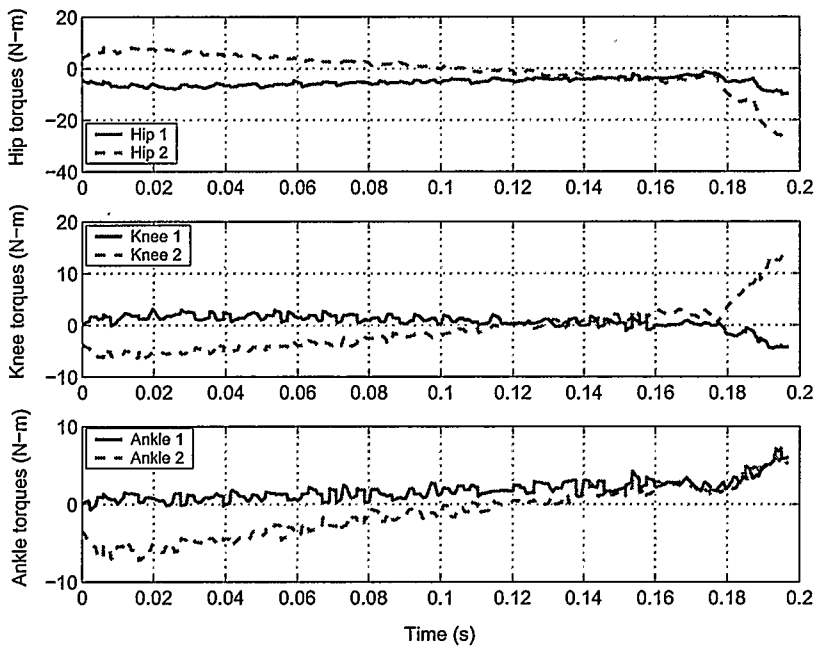

Figure 6.6: Joint torques in the flight phase

follows:

1. Scheme-1: only position-tracking module is applied.

2. Scheme-2: the position-tracking module and the force-suppression module are turned on.

3. Scheme-3: all of the three control modules are used.

The parameters of the controller are listed in Table 6.2. Figs. 6.7 - 6.9 show the trajectories of θ_1 and $\dot{\theta}_1$, corresponding to the three strategies, respectively. When Scheme-3 is active, the tracking error of θ_1 is actually $\theta_1^s + \tilde{\theta}_1^s - \theta_{1,d}^s$ and the error of $\dot{\theta}_1$ is defined as $\dot{\theta}_1^s + \dot{\tilde{\theta}}_1^s - \dot{\theta}_{1,d}^s$. The actual signals (dashed curves) converge to the references (solid curves) with small tracking errors after the convergence time T_c^s (Fig. 6.10).

Scheme- 2 and 3 deteriorate slightly the tracking accuracy of the support leg (Figs. 6.15 - 6.18), but do not degrade the tracking performance of the swing leg, as shown in Figs. 6.11 - 6.14. For all of the three strategies, the maximal final errors of the angles and the angular velocities are 0.006 radians (or 0.35 degrees) and 0.4 radians per second (or 22.9 degrees per second), respectively. θ_4, $\dot{\theta}_4$, θ_5, $\dot{\theta}_5$, and their tracking errors are not presented, for conciseness of the thesis.

The GRFs and ZMP trajectories are shown in Figs. 6.19 and 6.20. The safety region is set to be within $[-0.05m, 0.15m]$, same as the foot range. When Scheme-1 is tested, large vertical GRF arises three times (bottom graph of Fig. 6.19). The ZMP trajectory goes outside of the safety region twice (top graph of Fig. 6.20). If Scheme-2 is employed, the large vertical GRF values are greatly reduced to $235.9N$,

Table 6.2: Parameters of the controller in the stance phase

Parameters	Values
Λ^s	diag(15, 15, 15, 15, 15)
K_1^s	diag(30, 30, 30, 30, 30)
K_2^s	diag(150, 150, 150, 150, 150)
K_α	diag(30, 30, 30, 30, 30)
K_β	diag(450, 450, 450, 450, 450)

$233.9N$, and $230.8N$, respectively. They are still greater than $2M_t g$, the permitted maximal vertical GRF value. To protect the robot, passive force-reducing elements, *e.g.* springs or rubber pads, can be installed in the feet [50, 27, 102]. Unfortunately, use of Scheme-2 cannot guarantee satisfaction of the ZMP trajectory (middle graph of Fig. 6.20). Scheme-3 produces satisfactory vertical GRF profile and ZMP trajectory simultaneously, with trivial deterioration of tracking accuracy. The torques of the hips, the knees, and the ankles are shown in sequence as Figs. 6.21 - 6.23. They are noisy, infected by the white noise intentionally injected to the system.

6.4 Summary

In this chapter, the control algorithm for the one-legged hopper, discussed in Chapter 5, was extended to bipedal running with two modifications. Vectors and gain matrices possess higher dimensions. Moore-Penrose pseudo-inverse of the coefficient matrix of τ was used to resolve the joint torques.

The basic structure of the control algorithm for the hopper was inherited. The flight controller focuses only on position tracking. The stance controller consists of three modules: a position-tracking module, a force-suppression module, and an

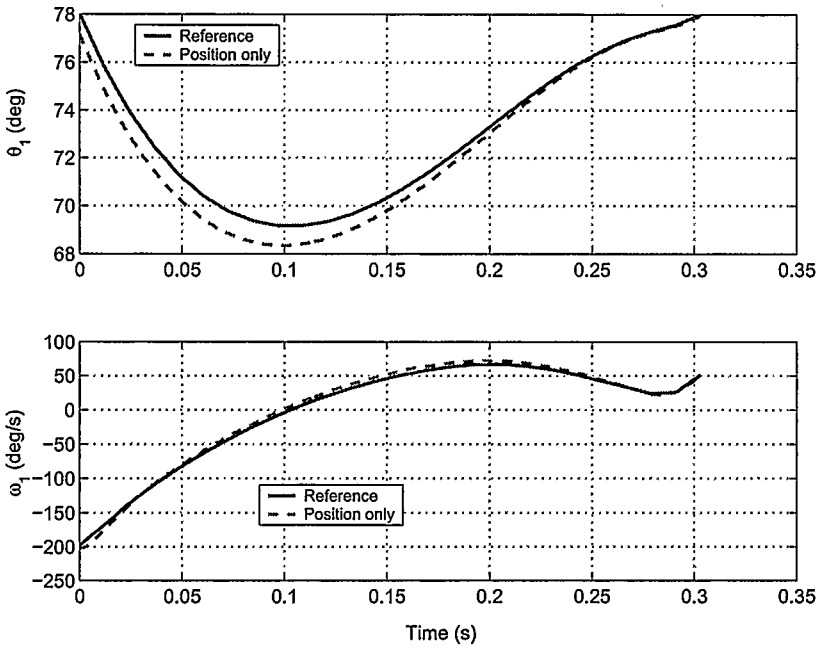

Figure 6.7: θ_1 and $\dot{\theta}_1$ in the stance phase by using Scheme-1

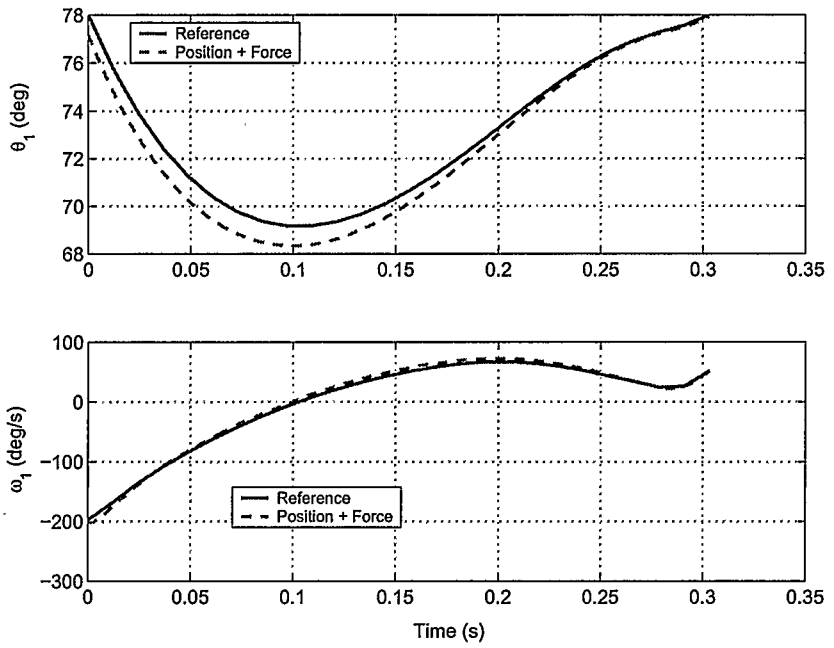

Figure 6.8: θ_1 and $\dot{\theta}_1$ in the stance phase by using Scheme-2

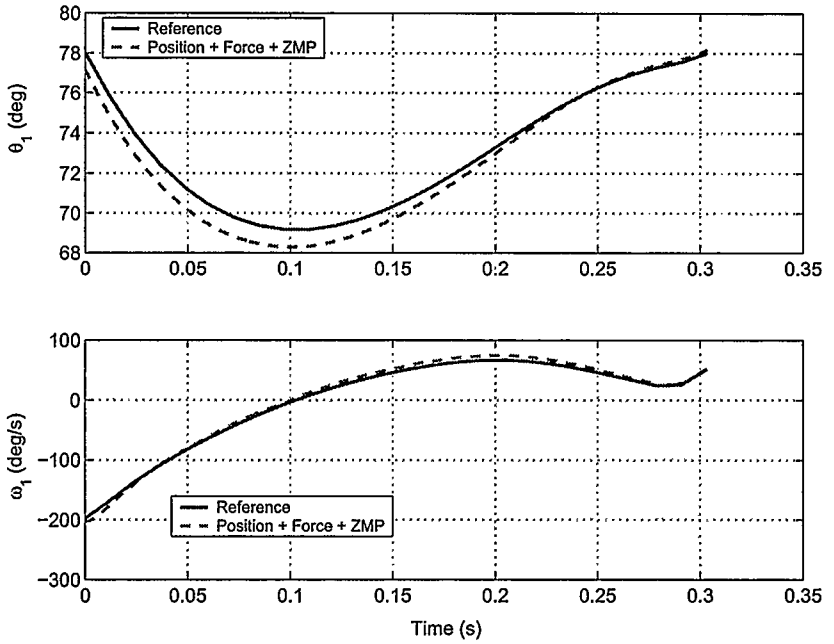

Figure 6.9: θ_1 and $\dot{\theta}_1$ in the stance phase by using Scheme-3

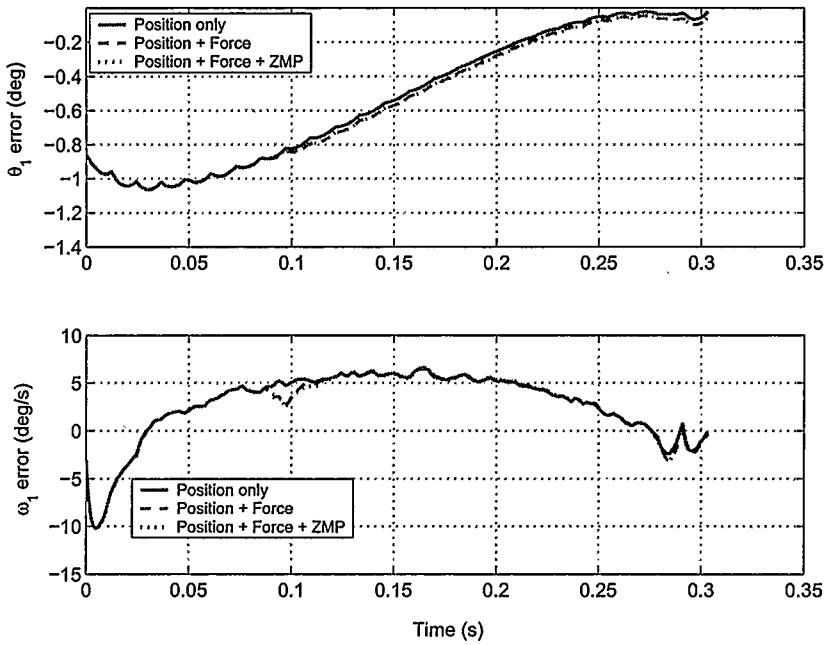

Figure 6.10: Tracking errors of θ_1 and $\dot{\theta}_1$ in the stance phase

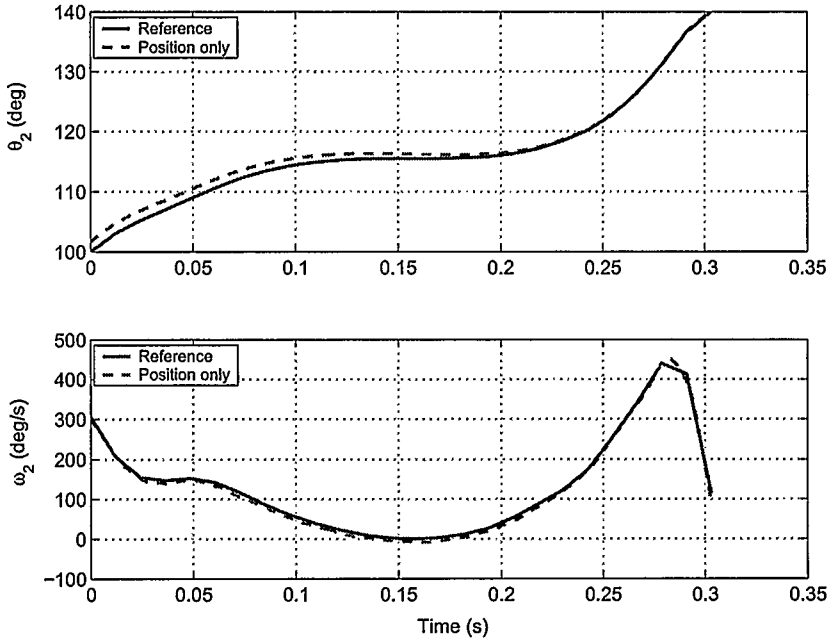

Figure 6.11: θ_2 and $\dot{\theta}_2$ in the stance phase by using Scheme-1

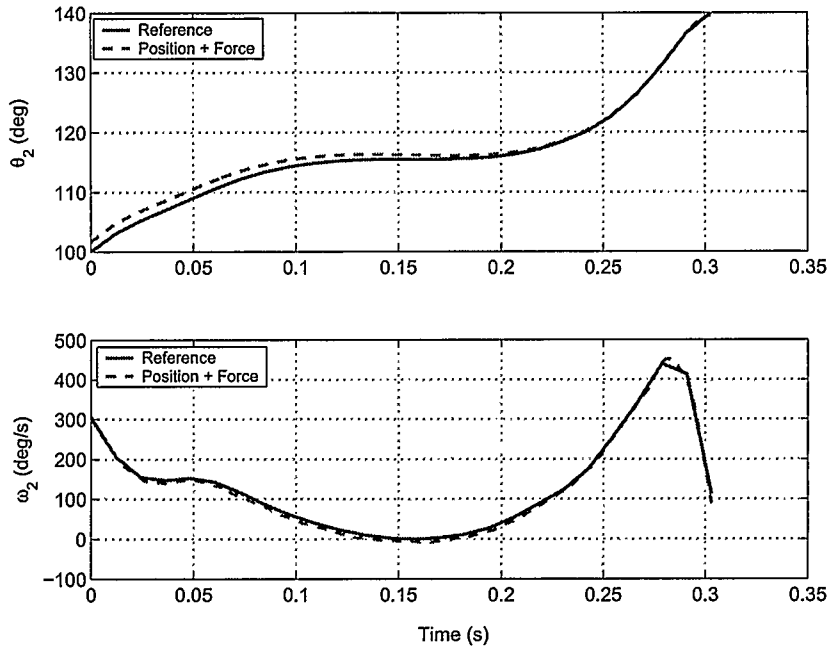

Figure 6.12: θ_2 and $\dot{\theta}_2$ in the stance phase by using Scheme-2

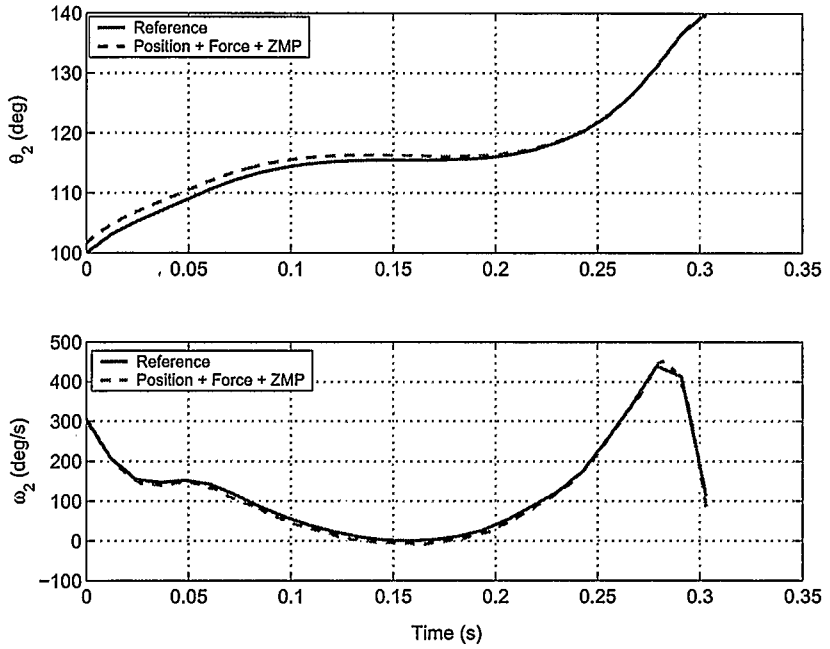

Figure 6.13: θ_2 and $\dot{\theta}_2$ in the stance phase by using Scheme-3

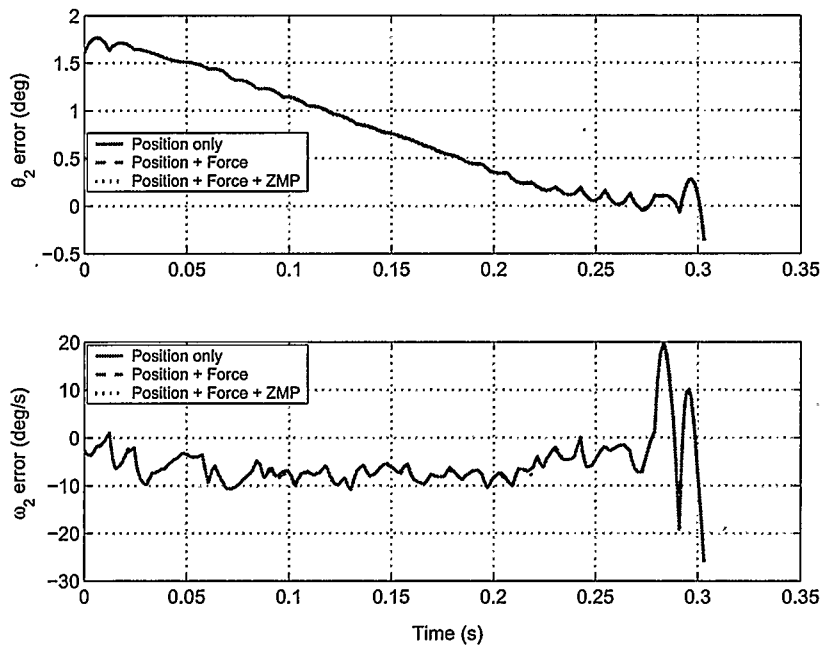

Figure 6.14: Tracking errors of θ_2 and $\dot{\theta}_2$ in the stance phase

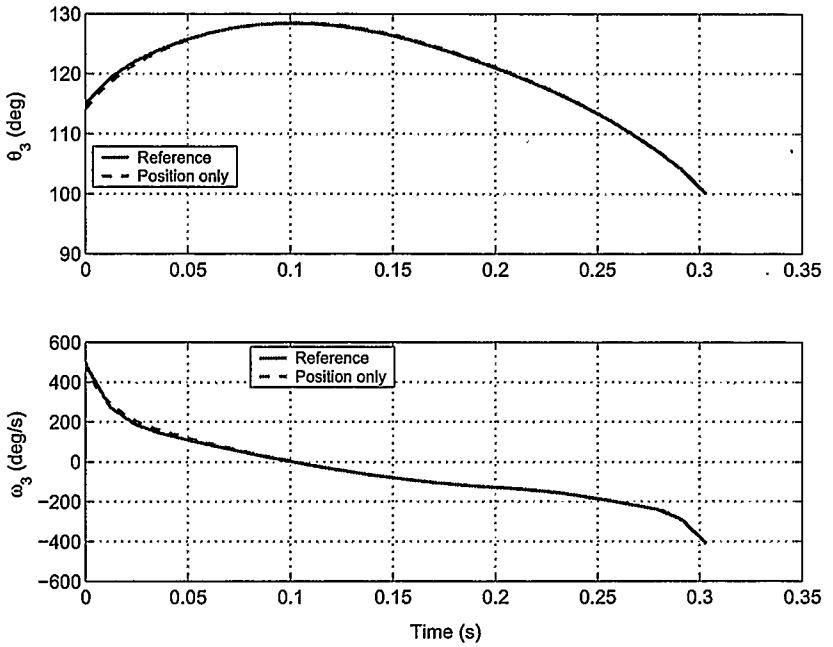

Figure 6.15: θ_3 and $\dot{\theta}_3$ in the stance phase by using Scheme-1

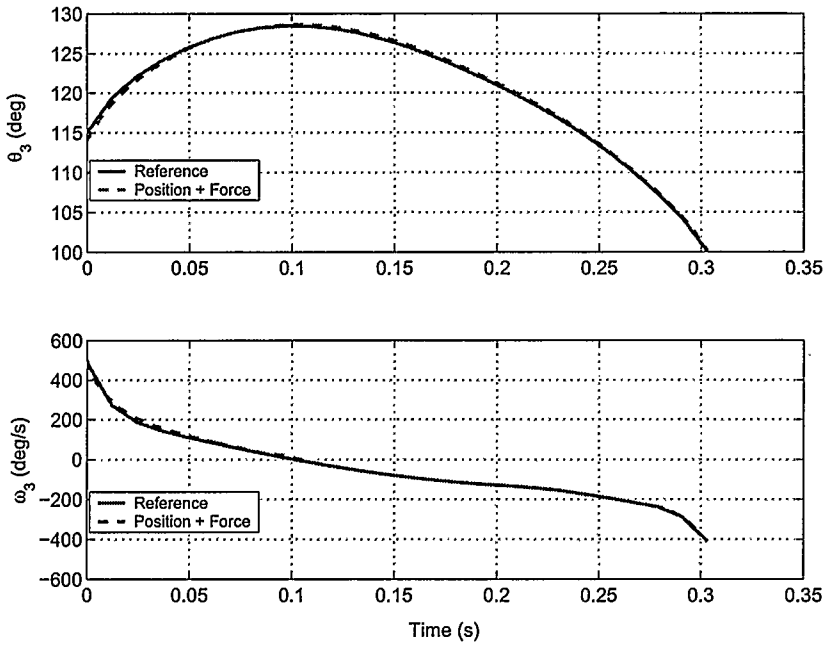

Figure 6.16: θ_3 and $\dot{\theta}_3$ in the stance phase by using Scheme-2

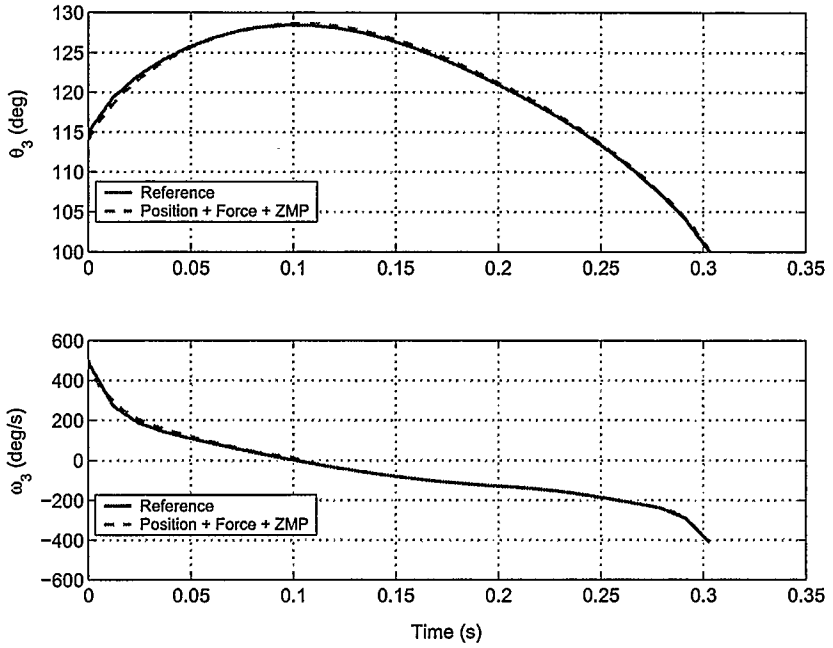

Figure 6.17: θ_3 and $\dot{\theta}_3$ in the stance phase by using Scheme-3

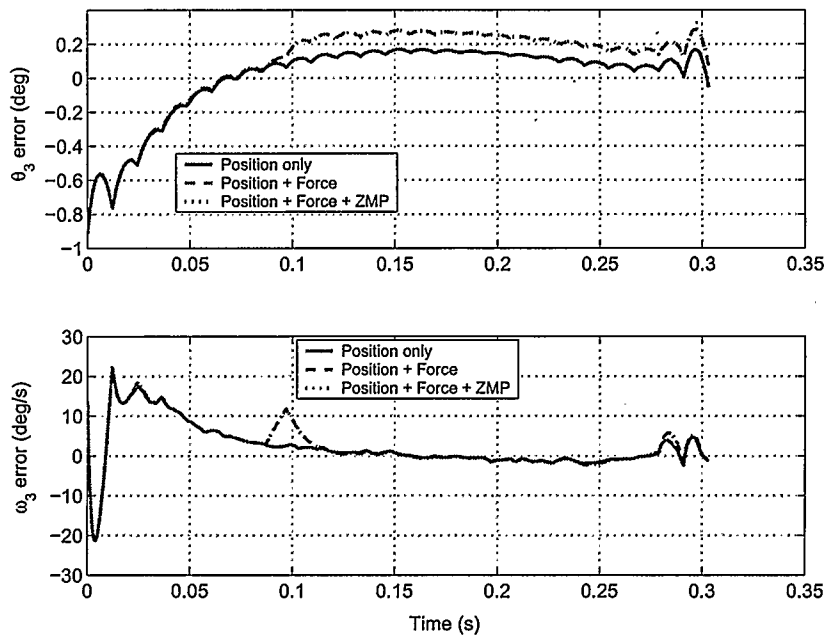

Figure 6.18: Tracking errors of θ_3 and $\dot{\theta}_3$ in the stance phase

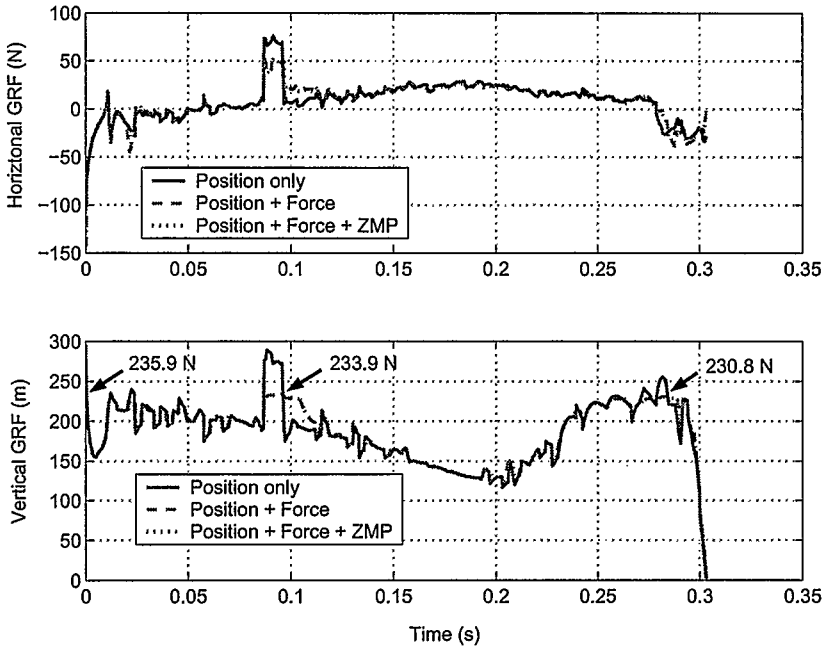

Figure 6.19: GRFs in the stance phase

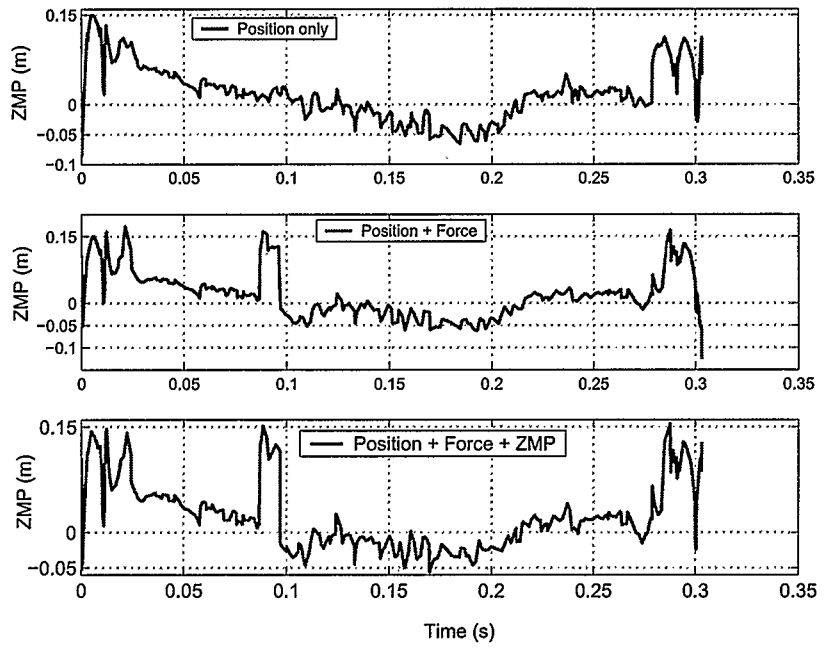

Figure 6.20: ZMP trajectories in the stance phase

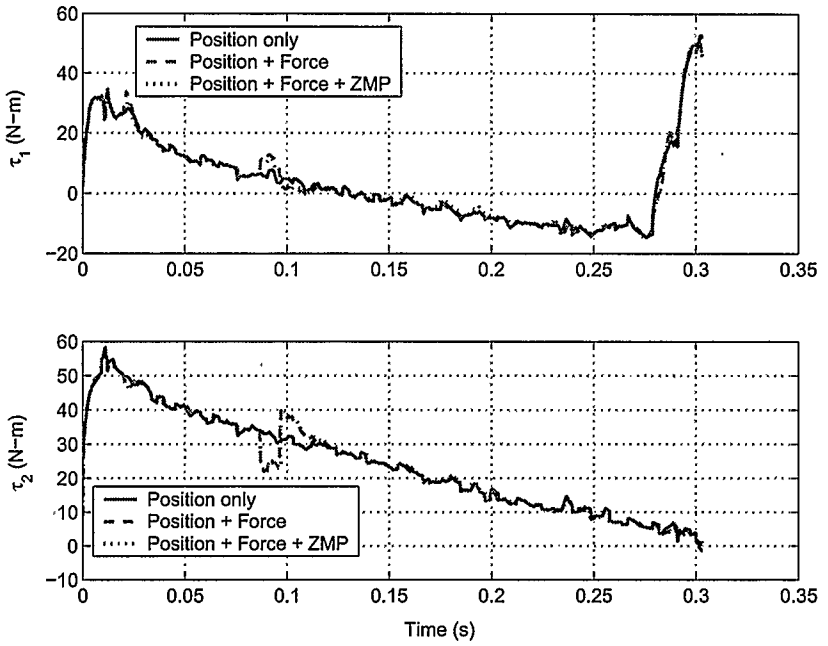

Figure 6.21: Torques of the hips in the stance phase

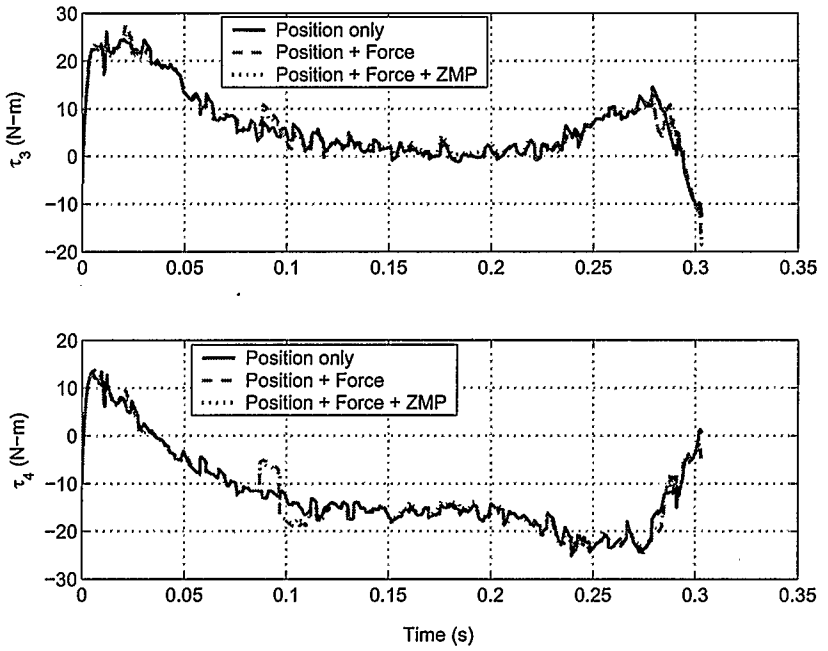

Figure 6.22: Torques of the knees in the stance phase

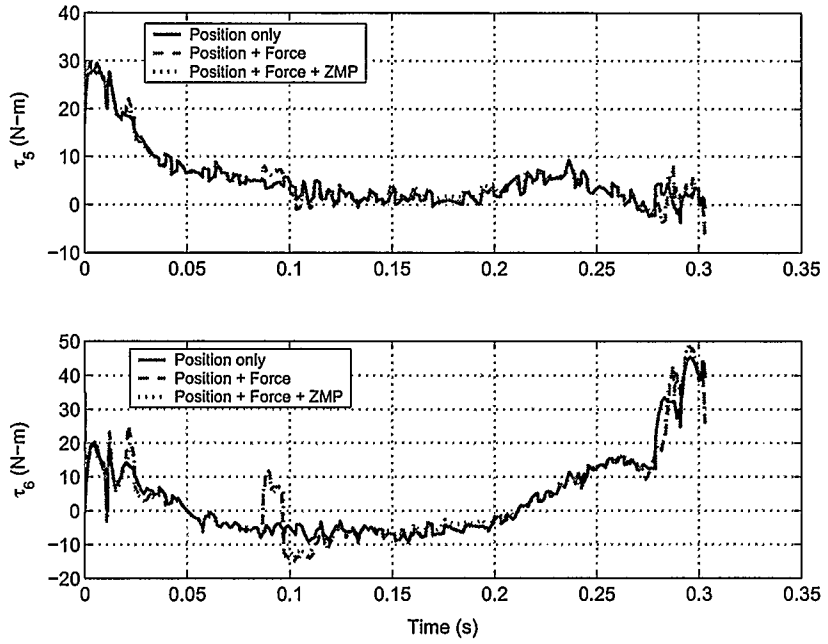

Figure 6.23: Torques of the ankles in the stance phase

online ZMP compensator. Their effectiveness was verified by simulations.

Chapter 7

Conclusions and Future Work Directions

This thesis concentrates on running synthesis and running control designs for robots with one and two articulated legs. Although success of SLIP hoppers and bipeds with compliant legs suggests promising possibilities in constructing running robots, no compliant elements have been considered in the robot models studied in this thesis. Without use of compliant elements, the robots suffer very large impulsive impacts from the ground. This challenge distinguishes the rigid running robots from the compliant running robots and the walking machines. It is believed that solutions to this problem offer new understanding of bouncing gaits, which may facilitate development of more versatile one- and two-legged robots that can adjust gaits in real time and are ready for practical tasks.

Concluding remarks are given in Section 7.1. Future research directions are proposed in Section 7.2.

7.1 Concluding remarks

Two sub-projects have been discussed in previous chapters. Chapter 3 and Chapter 5 deal with hopping synthesis and hopping control designs, respectively. The techniques are then extended to synthesize and stabilize bipedal running in Chapter 4 and Chapter 6 with minor modifications.

Directly inspired by new observations in biological sciences and biomechanical

analysis, it is assumed that the energy consumed by the robot in the flight phase is small when the robot hops or runs on flat even ground. This assumption is the cornerstone of this thesis. By using energy analysis, the assumption has been formulated as a simple static optimization problem, whose solutions provide the initial joint velocities of the flight phase with the boundary joint angles given in advance. The running gait can then be generated by dynamic optimization.

The forward speed of the robot cannot be altered during the flight phase. One goal of the stance phase is to prepare the correct initial states for the subsequent flight phase such that the desired stride length, landing height, and forward speed can be reached within error tolerances. The stance phase is also responsible for interaction between the robot and the ground. The robot may be damaged due to large GRFs. Meanwhile, large GRFs introduce large tipping moment about the support foot, and consequently, the robot loses its balance and falls down. To address the above two issues, a novel controller has been proposed. The controller is composed of three modules: a finite-time position-tracking module, a force-suppression module, and an online ZMP compensator. The finite-time position-tracking module helps to prepare desired initial conditions for the subsequent flight phase. The force-suppression module rejects most of the excessive GRFs. The online ZMP compensator pushes the actual ZMP location closer to the center of the safety region, such that the stance stability can be improved.

To improve landing accuracy, a finite-time position-tracking controller is also employed in the flight phase.

Clearly, the finite-time control algorithm forms the basis of the whole control system. In this work, 3^{rd} order polynomials have been adopted as the desired tracking

errors. The coefficients of the polynomials are chosen in such a manner that the reaching surface is removed and the sliding surface always contains the initial states of the system. Thus, the control system is highly robust to parameter uncertainties and external disturbances. Simulations verify the effectiveness of the control algorithm.

Compared to synthesis and control designs for bipedal running, hopping synthesis and hopping control designs have been explained in more details. This does not mean that hoppers are more important than bipeds. There is no doubt that bipeds are more versatile and more suitable for practical use than hoppers. As a good stepping stone, the one-legged hopping model provides sharp insight into the problem, and thus greatly simplifies the formulations and discussions. Starting directly from a robot with complicated mechanical structure may result in daunting equations (see *e.g.* [88]).

7.2 Future research directions

Running with rigid articulated legs has been shown difficult by recent experimental practices [92, 74]. Numerous questions are to be answered. Explorations of the following ideas are direct extensions of the work reported in this thesis:

1. In this thesis, the feet are assumed to be parallel with the ground for safe landing. This may not be necessary. When a human is running, the swing leg folds compactly in the beginning of the stance phase, resulting in easier recovery motion of the swing leg and less energy cost of the gait. This observation may be utilized to synthesize more efficient and more elegant bouncing gaits for legged robots. Unfortunately, this treatment complicates the stance phase. A

heel-strike happens first, a rotation of the support foot about the heel then occurs, followed by a second collision of the sole with the ground. The simple collision model employed in this thesis may not be plausible anymore.

2. In this thesis, all motions take place in the sagittal plane and only planar robot models have been studied. More realistic 3D robot models may help to understand better the principles of legged locomotion systems. The effects of the yaw moments caused by the swing leg should be examined.

3. Optimization of the flight phase can be very quick with the help of the assumption (A1). Unfortunately, optimization of the stance phase is time-consuming. It is desirable to generate the running gaits in real time. A more efficient algorithm for synthesis of the stance phase is needed.

4. Elastic elements should be added to the legged robots to improve energy efficiency. Recent observations in biomechanical analysis and legged robotics suggest that compliant bipeds can also perform multiple gaits, including walking and running [37, 130, 61, 60]. Combinations of these hardware designs with the fundamental assumption (A1) proposed in this thesis may lead to more efficient gaits, since collision loss can be greatly reduced due to use of compliant elements. Considering 3D gaits, exciting results are expected.

5. According to above arguments, a series of robot prototypes can be built. Comparisons between the rigid robots and the robots with one or two compliant articulated legs may be interesting. The simulation predictions need to be verified by experiments.

6. Event-based finite-time control algorithm can also be tried, following the works done by Westervelt [151] and Morris [92]. The actual phase trajectories are expected to converge to the desired phase trajectories in finite time. The orbital stability criterion, rather than the ZMP stability criterion, has to be embedded into the control designs.

Locomotion with one and two legs is an exciting subject which is still in infancy. The possible extensions listed above are directly inspired by the work reported in this thesis. New ideas sparkle whenever new observations are made, and research activities never stop.

Bibliography

[1] M. E. Abdallah and K. J. Waldron. A physical model and control strategy for biped running. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3982–3988, Roma, Italy, April 2007.

[2] M. Ahmadi and M. Buehler. Stable control of a simulated one-legged running robot with hip and leg compliance. *IEEE Transactions on Robotics and Automation*, 13(1):96–103, 1997.

[3] M. Ahmadi and M. Buehler. The arl monopod ii running robot: control and energetics. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1689–1694, Detroit, MI, USA, May 1999.

[4] R. McN. Alexander. Three uses for springs in legged locomotion. *International Journal of Robotics Research*, 9(2):53–61, 1990.

[5] R. McN. Alexander. A model of bipedal locomotion on compliant legs. *Transactions of the Royal Society B: Biological Sciences*, 338(1284):189–198, 1992.

[6] R. Altendorfer, D. E. Koditschek, and P. Holmes. Stability analysis of legged locomotion models by symmetry-factored return maps. *International Journal of Robotics Research*, 23(10-11):979–999, 2004.

[7] T. Arakawa and T. Fukuda. Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of ga, ep layers. In *IEEE International Conference on Robotics and Automation*, pages 211–216, Albuquerque, NM, USA, April 1997.

[8] S. Arimoto. *Control Theory of Non-linear Mechanical Systems: A Passivity-based and Circuit-Theoretic Approach.* Oxford University Press, New York, USA, 1996.

[9] C. Azevedo, P. Poignet, and B. Espiau. Moving horizon control for biped robots without reference trajectory. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2762–2767, Washington, DC, USA, May 2002.

[10] V. M. Becerra. Solving optimal control problems with state constraints using nonlinear programming and simulation tools. *IEEE Transactions on Education*, 47(3):377–384, 2004.

[11] M. D. Berkemeier and R. S. Fearing. Sliding and hopping gaits for the underactuated acrobot. *IEEE Transactions on Robotics and Automation*, 14(4):629–634, 1998.

[12] G. Bessonnet, S. Chesse, and P. Sardain. Optimal gait synthesis of a seven-link planar biped. *International Journal of Robotics Research*, 23(10/11):1059–1073, 2004.

[13] S. P. Bhat and B. S. Bernstein. Finite-time stability of countinuous autonomous systems. *SIAM Journal of Control and Optimization*, 38(3):751–766, 2000.

[14] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11/12):1217–1227, 1989.

[15] R. Blickhan and R. Full. Similarity in multilegged locomotion: bouncing like a monopode. *Journal of Comparative Physiology A*, 173(5):509–517, 1993.

[16] B. Brown and G. Zeglin. The bow leg hopping robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 781–786, Leuven, Belgium, May 1998.

[17] F. Bullo and R. M. Murray. Tracking for fully actuated mechanical systems: a geometric framework. *Journal of Intelligent and Robotic Systems*, 35(1):17–34, 1999.

[18] G. Cabodevila and G. Abba. Quasi optimal gait for a biped robot using genetic algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics - Computational Cybernetics and Simulations*, pages 3960–3965, Orlando, FL, USA, Oct. 1997.

[19] G. A. Cavagna, H. Thys, and A. Zamboni. The sources of external work in level walking and running. *Journal of Physiology*, 262(3):639–657, 1976.

[20] P. H. Channon, S. J. Hopkins, and D. H. Pham. Simulation and optimal walking motions for a bipedal robot. *Mathematical and Computer Modelling*, 14(1):463–467, 1990.

[21] P. H. Channon, S. J. Hopkins, and D. H. Pham. Derivation of optimal walking motions for a bipedal walking robot. *Robotica*, 10(2):165–172, 1992.

[22] N. Cherouvim and E. Papadopoulos. Single actuator control analysis of a planar 3dof hopping robot. In S. Thrun, G. Sukhatme, and S. Schaal, editors,

Experimental Robotics I, pages 145–152. The MIT Press, Cambridge, MA, USA, 2005.

[23] C. Chevallereau and Y. Aoustin. Optimal reference trajectories for walking and running of a biped robot. *Robotica*, 19(5):557–569, 2001.

[24] C. Chevallereau, E. R. Westervelt, and J. W. Grizzle. Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. *International Journal of Robotics Research*, 24(6):431–464, 2005.

[25] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993.

[26] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[27] S. H. Collins, M. Wisse, and A. Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *International Journal of Robotics Research*, 20(2):607–615, 2001.

[28] C. Canudas de Wit, L. Roussel, and A. Goswami. Periodic stabilization of a 1-dof hopping robot on nonlinear compliant surface. In *Proceedings of the IFAC Symposium on Robot Control*, pages 405–410, Nantes, France, Sept. 1997.

[29] J. M. Donelan, R. Kram, and A. D. Kuo. Mechanical and metabolic determinants of the preferred step width in human walking. *Proceedings of the Royal Society of London, Series B: Biological Science*, 268(1480):1985–1992, 2001.

[30] P. Dorato. An overview of finite-time stability. In L. Menini, L. Zaccarian, and C. T. Abdallah, editors, *Current Trends in Nonlinear Systems and Control: In Honor of Petar Kokotović and Turi Nicosia*, pages 185–194. Birkhäuser Boston, Cambridge, MA, USA, 2006.

[31] Y. Feng, X. Yu, and Z. Man. Non-singular terminal sliding mode control of rigid manipulators. *Automatica*, 38(12):2159–2167, 2002.

[32] C. Francois and C. Samson. A new approach to the control of the planar one-legged hopper. *International Journal of Robotics Research*, 17(11):1150–1166, 1998.

[33] C. Frohlich. Do springboard divers violate angular momentum conservation? *American Journal of Physics*, 47(7):583–592, 1979.

[34] Y. Fujimoto. Trajectory generation of biped running robot with minimum energy consumption. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3803–3808, New Orleans, LA, USA, April 2004.

[35] R. J. Full and D. E. Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 1999.

[36] E. Garcia, J. Estremera, and P. Gonzalez de Santos. A comparative study of stability margins for walking machines. *Robotica*, 20(6):595–605, 2002.

[37] H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behaviour explains

basic dynamics of walking and running. *Proceeding of the Royal Society of London, Series B: Biological Sciences*, 273(1603):2861–2867, 2006.

[38] R. M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek. A simply stabilized running model. *SIAM Review*, 47(3):519–549, 2005.

[39] A. Goswami. Postural stability of biped robots and the foot-rotation indicator (fri) point. *International Journal of Robotics Research*, 18(6):523–533, 1999.

[40] A. Goswami, B. Thuilot, and B. Espiau. A study of the passive gait of a compass-like biped robot: symmetry and chaos. *International Journal of Robotics Research*, 17(12):1282–1301, 1998.

[41] P. Gregorio, M. Ahmadi, and M. Buehler. Design, control and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 27(4):626–634, 1997.

[42] Q. Guo, C. J. B. Macnab, and J. K. Pieper. Hopping on even ground and up stairs with a single articulated leg. *Journal of Intelligent and Robotic Systems*, 53(4):331–358, 2008.

[43] W. M. Haddad and V. Chellaboina. *Nonlinear Dynamical System and Control, A Lyapunov-Based Approach.* Princeton University Press, Princeton, NJ, USA, 2008.

[44] V. T. Haimo. Finite-time controller. *SIAM Journal of Control and Optimization*, 24(4):760–770, 1986.

[45] R. Van Ham, B. Vanderborght, M. Van Damme, B. Verrelst, and D. Lefeber. Maccepa, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems*, 55(10):761–768, 2007.

[46] M. W. Hardt. *Multibody dynamical algorithms, numerical optimal control, with detailed studies in the control of jet engine compressors and biped walking.* PhD thesis, University of California, San Diego, CA, USA, 1999.

[47] J. Hass, J. M. Herrmann, and T. Geisel. Optimal mass distribution for passivity-based bipedal robots. *The International Journal of Robotics Research*, 25(11):1087–1098, 2006.

[48] C. Hayashi. *Nonlinear Oscillations in Physical Systems.* Princeton University Press, Princeton, NJ, USA, 1985.

[49] H. Hemami and C. L. Golliday. The inverted pendulum and biped stability. *Mathematical Biosciences*, 34(1/2):95–110, 1977.

[50] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *IEEE International Conference on Robotics and Automation*, pages 1321–1326, Leuven, Belgium, May 1998.

[51] N. Hogan. Impedance control: An approach to manipulation: Part i-theory, part ii-implementation, part iii-applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–24, 1985.

[52] J. M. Hollerbach and K. C. Suh. Redundancy resolution of manipulators

through torque optimization. *IEEE Journal of Robotics and Automation*, Ra-3(4):308–316, 1987.

[53] P. Hsu, J. Hauser, and S. Sastry. Dynamic control of redundant manipulators. *Journal of Robotic Systems*, 6(2):133–148, 1989.

[54] Y. Hürmüzlü and G. D. Moskowitz. Bipedal locomotion stabilized by impact and switching i and ii. *Dynamics and Stability of Systems*, 2(2):73–112, 1987.

[55] J. W. Hurst, J. E. Chestnutt, and A. A. Rizzi. Design and philosophy of the bimasc, a highly dynamic biped. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1863–1868, Roma, Italy, April 2007.

[56] J. W. Hurst and A. A. Rizzi. Physically variable compliance in running. In *Proceedings of the International Conference on Climbing and Walking Robots*, pages 123–133, Madrid, Spain, Sept. 2004.

[57] S.-H. Hyon and T. Emura. Energy-preserving control of a passive one-legged running robot. *Advanced Robotics*, 18(4):357–381, 2004.

[58] S.-H. Hyon and T. Emura. Running control of a planar biped robot based on energy -preserving strategy. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3791–3796, New Orleans, LA, USA, April 2004.

[59] S.-H. Hyon and T. Mita. Development of a biologically inspired hopping robot - "kenken". In *Proceedings of the IEEE International Conference on Robotics*

and Automation, pages 3984–3991, Washington DC, USA, May 2002.

[60] F. Iida, Y. Minekawa, J. Rummel, and A. Seyfarth. Toward a human-like biped robot with compliant legs. *Robotics and Autonomous Systems*, 57(2):139–144, 2009.

[61] F. Iida, J. Rummel, and A. Seyfarth. Bipedal walking and running with spring-like biarticular muscles. *Journal of Biomechanics*, 41(3):656–667, 2008.

[62] T. Ikeda, Y. Iwatani, K. Suse, and T. Mita. Analysis and design of running robots in touchdown phase. In *Proceedings of the IEEE International Conference on Control Applications*, pages 496–501, Kohala Coast-Island, Hawaii, USA, Aug. 1999.

[63] J. L. Johansson, D. M. Sherrill, P. O. Riley, P. Bonato, and H. Herr. A clinical comparison of variable damping and mechanically passive prosthetic knee devices. *American Journal of Physical Medicine and Rehabilitation*, 84(8):563–575, 2005.

[64] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hiukawa. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1644–1650, Las Vegas, NV, USA, Oct. 2003.

[65] S. Kajita, O. Matsumoto, and M. Saigo. Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *Proceedings of the IEEE In-*

ternational Conference on Robotics and Automation, pages 2299–2036, Seoul, Korea, May 2001.

[66] S. Kajita, T. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie. A running controller of humanoid biped hrp-2lr. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 616–622, Barcelona, Spain, April 2005.

[67] S. Kajita, T. Nagasaki, K. Yokoi, K. Kaneko, and K. Tanie. Running pattern generation for a humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2755–2761, Washington DC, USA, May 2002.

[68] B. Karan. Robust position-force control of robot manipulator in contact with linear dynamic environment. *Robotica*, 23(6):799–803, 2005.

[69] O. Khatib. A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987.

[70] C. A. Klein and L.-C. Chu. Comparison of extended jacobian and lagrange multiplier based methods for resolving kinematic redundancy. *Journal of Intelligent and Robotic Systems*, 19(1):39–54, 1997.

[71] K. Kondak and G. Hommel. Control of online computation of stable movement for biped robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 874–879, Las Vegas, NV, USA, Oct. 2003.

[72] A. D. Kuo. A simple model predicts the step length-speed relationship in human walking. *Journal of Biomechanical Engineering*, 123(3):264–269, 2001.

[73] O. Kwon and J. H. Park. Gait transitions for walking and running of biped robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1350–1355, Taipei, Taiwan, Sept. 2003.

[74] O. Kwon and J. H. Park. Asymmetric trajectory generation and impedance control for running of biped robots. *Autonomous Robots*, 26(1):47–78, 2009.

[75] W. Lee and M. H. Raibert. Control of hoof rolling in an articulated leg. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1386–1391, Sacramento, CA, USA, April 1991.

[76] A. Lesbastard, Y. Aoustin, and Y. Plestan. Observer-based control of a walking biped robot without orientation measurement. *Robotica*, 24(3):384–400, 2006.

[77] M. Lidberg. *Design of optimal processes for closed-loop chain SCARA-like robots.* PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 2004.

[78] H.-O. Lim, S. A. Setiawan, and A. Takanishi. Position-based impedance control of a biped humanoid robot. *Advanced Robotics*, 18(4):415–435, 2004.

[79] C.-K. Lin. Nonsingular terminal sliding mode control of robot manipulators using fuzzy wavelet networks. *IEEE Transactions on Fuzzy Systems*, 14(6):849–859, 2006.

[80] G. Liu and A. A. Goldenberg. Robust hybrid impedance control of robot manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 287–292, Sacramento, CA, USA, April 1991.

[81] A. Loria and R. Ortega. Force/position regulation for robot manipulators with unmeasurable velocities and uncertain gravity. *Automatica*, 32(6):939–943, 1996.

[82] V. Mallapragada, D. Erol, and N. Sarkar. A new method of force control for unknown environment. *International Journal of Advanced Robotic Systems*, 4(3):313–322, 2007.

[83] T. McGeer. Passive bipedal running. *Proceeding of the Royal Society of London, Series B: Biological Sciences*, 240(1297):107–134, 1990.

[84] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.

[85] R. B. McGhee and A. Frank. On the stability properties of quadruped creeping gaits. *Mathematical Bioscience*, 3(1/2):331–351, 1968.

[86] R. T. M'Clowskey and J. M. Burdick. Periodic motions of a hopping robot with vertical and forward motion. *International Journal of Robotics Research*, 12(3):197–218, 1993.

[87] T. A. McMahon and G. Cheng. The mechanics of running: how does stiffness couple with speed? *Journal of Biomechanics*, 23(Suppl 1):65–78, 1990.

[88] A. Meghdari and M. Aryanpour. Dynamic modeling and analysis of the human jumping process. *Journal of Intelligent and Robotic Systems*, 37(1):97–115, 2003.

[89] D. S. Mitrinović. *Analytic Inequalities.* Springer-Verlag, Berlin, Germany, 1970.

[90] M. Miyazaki, M. Sampei, M. Koga, and A. Takahashi. A control of underactuated hopping gait systems: acrobot example. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 4797–4802, Sydney, Australia, Dec. 2000.

[91] K. D. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schlöder. Open-loop stable running. *Robotica*, 23(1):21–33, 2005.

[92] B. Morris. *Stabilizing highly dynamic locomotion in planar bipedal robots with dimension reducing control.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 2008.

[93] B. Morris, E. R. Westervelt, C. Chevallereau, G. Buche, and J. W. Grizzle. Achieving bipedal running with rabbit: Six steps toward infinity. In *Fast Motions in Biomechanics and Robotics*, pages 277–297, Heidelberg, Germany, Sept. 2005.

[94] G. Muscato and G. Spampinato. Kinematical model and control architecture for a human inspired five dof robotic leg. *Mechatronics*, 17(1):45–63, 2007.

[95] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi. Integrated

motion control for walking, jumping and running on a small bipedal entertainment robot. In *Proceedings of the 2004 IEEE international conference on robotics and automation*, pages 3189–3194, New Orleans, LA, USA, April 2004.

[96] T. Nagasaki, S. Kajita, K. Kaneko, K. Yokoi, and K. Tanie. A running experiment of humanoid biped. In *Proceedings of 2004 IEEE/RSI International Conference on Intelligent Robots and Systems*, pages 135–141, Sendai, Japan, Sept. 2004.

[97] T. Nagasaki, S. Kajita, K. Yokoi, K. Kaneko, and K. Tanie. Running pattern generation and its evaluation using a realistic humanoid model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1336–1342, Taipei, Taiwan, Sept. 2003.

[98] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Comparative experiments on task space control with redundancy resolution. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3901–3908, Edmonton, Canada, Aug. 2005.

[99] Y. Okumura, T. Tawara, K. Endo, T. Furuta, and M. Shimizu. Realtime zmp compensation for biped walking robot using adaptive inertia force control. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 335–339, Las Vegas, NV, USA, Oct. 2003.

[100] J. Or and A. Takanishi. A biologically inspired cpg-zmp control system for the real-time balance of a single-legged belly dancing robot. In *Proceedings of*

2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 931–936, Sendai, Japan, Sept.-Oct. 2004.

[101] D. Owaki, K. Osuka, and A. Ishiguro. On the embodiment that enables passive dynamic bipedal running. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 341–346, Pasadena, CA, USA, May 2008.

[102] J. H. Park. Impedance control for biped robot locomotion. *IEEE Transactions on Robotics and Automation*, 17(6):870–882, 2001.

[103] J. H. Park and H. Chung. Hybrid control for biped robots using impedance control and computed-torque control. In *Proceedings of 1999 IEEE International Conference on Robotics and Automation*, pages 1365–1370, Detroit, MI, USA, May 1999.

[104] J. H. Park and H. Chung. Zmp compensation by on-line trajectory generation for biped robots. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2568–2573, Sendai, Japan, Sept.-Oct. 2004.

[105] J. H. Park and O. Kwon. Impedance control for running of biped robots. In *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 944–949, Kobe, Japan, July 2003.

[106] K.-B. Park and T. Tsuji. Terminal sliding mode control of second-order nonlinear uncertain systems. *International Journal of Robust and Nonlinear Control*, 9(11):769–780, 1999.

[107] V. Parra-Vega and G. Hirzinger. Finite-time tracking for robot manipulators with singularity-free continuous control: A passivity-based approach. In *Proceedings of the 39Ih IEEE Conference on Decision and Control*, pages 5085–5090, Sydney, Australia, Dec. 2000.

[108] V. Parra-Vega, A. Rodriguez-Angeles, and G. Hirzinger. Perfect position/force tracking of robots with dynamical terminal sliding mode control. *Journal of Robotic Systems*, 18(9):517–532, 2001.

[109] R. P. Pingrose. *Self-stabilizing running.* PhD thesis, MIT, Cambridge, MA, USA, 1997.

[110] M. Popovic, A. Englehart, and H. Herr. Angular momentum primitives for human walking: Biomechanics and control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1685–1691, Sendai, Japan, Sept. 2004.

[111] M. B. Popovic, A. Goswami, and H. Herr. Ground reference points in legged locomotion: definitions, biological trajectories and control implications. *International Journal of Robotics Research*, 24(12):1013–1032, 2005.

[112] I. Poulakakis and J. W. Grizzle. Monopedal running control: Slip embedding and virtual constraint controllers. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 323–330, San Diego, CA, USA, Oct. 2007.

[113] V. Prahlad, G. Dip, and M. H. Chia. Disturbance rejection by online zmp compensation. *Robotica*, 26(1):9–17, 2008.

[114] G. A. Pratt. Legged robots at mit: What's new since raibert. *IEEE Robotics and Automation Magzine*, 7(3):15–19, 2000.

[115] G. A. Pratt and M. M. Williamson. Series elastic actuators. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 399–406, Pittsburgh, PA, USA, Aug. 1995.

[116] J. Pratt and R. Tedrake. Velocity based stability margins for fast bipedal walking. In *Fast Motions in Biomechanics and Robotics*, pages 299–324, Heidelberg, Germany, Sept. 2005.

[117] H. Rad, P. Gregorio, and M. Buehler. Design, modeling and control of a hopping robot. In *Proceedings of the IEEE/RSJ Conference On Intelligent Systems and Robots*, pages 1778–1785, Yokohama, Japan, July 1993.

[118] M. H. Raibert. *Legged Robots That Balance.* The MIT Press, Cambridge, MA, USA, 1986.

[119] C. Ridderström. *Legged locomotion: Balance, control and tools - from equation to action.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2003.

[120] T. S. Roberts, R. Kram, P. G. Weyand, and R. Taylor. Energetics of bipedal running, i: metabolic cost of generating force. *The Journal of Experimental Biology*, 201(19):2745–2751, 1998.

[121] M. Rostami and G. Bessonnet. Sagittal gait of a biped robot during the single stance phase. part 2: optimal motion. *Robotica*, 19(3):241–253, 2001.

[122] L. Roussel, C. Canudas de Wit, and A. Goswami. Comparative study of method for energy-optimal gait eneration for biped robots. In *Proceedings of the International Conference on Informatics and Control*, pages 1205–1212, St. Petersburg, Russia, June 1997.

[123] J. Roy and L. L. Whitcomb. Adaptive force control of position/velocity controlled robots: Theory and experiment. *IEEE Transactions on Robotics and Automation*, 18(2):121–137, 2002.

[124] U. Saranli, W. J. Schwind, and D. E. Koditschek. Toward the control of a multi-jointed, monoped runner. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2676–2682, Leuven, Belgium, May 1998.

[125] P. Sardain and G. Bessonnet. Forces acting on a biped robot, center of pressure-zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(5):630–637, 2004.

[126] A. Sayyad, B. Seth, and P. Seshu. Single-legged hopping robotics research - a review. *Robotica*, 25(5):587–613, 2007.

[127] W. J. Schwind. *Spring loaded inverted pendulum running: a plant model.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1998.

[128] J. E. Seipel and P. Holmes. Running in three dimensions: analysis of a point-mass sprung-leg model. *International Journal of Robotics Research*, 24(8):657–674, 2005.

[129] A. Seyfarth, H. Geyer, M. Guenther, and R. Blickhan. A movement criterion for running. *Journal of Biomechanics*, 35(5):649–655, 2002.

[130] A. Seyfarth, R. Tausch, M. Stelzer, F. Iida, A. Karguth, and O. von Stryk. Towards bipedal running as a natural result of optimizing walking speed for passively compliant three-segmented legs. In *Proceedings of the 9th International Conference on Climbing and Walking Robots*, Brussels, Belgium, Sept. 2006.

[131] Z. Sheng and K. Yamafuji. Postural stability of a human riding a unicycle and its emulation by a robot. *IEEE Transactions on Robotics and Automation*, 13(5):709–720, 1997.

[132] T. Shimizu, S. Nakaura, and M. Sampei. The control of a bipedal running robot based on output zeroing considered rotation of the ankle joint. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5456–5461, San Diego, CA, USA, Dec. 2006.

[133] S. P. Singh. *Self-contained measurement of dynamic legged locomotion: Design for robot and field environments.* PhD thesis, Stanford University, Stanford, CA, USA, 2006.

[134] J. J. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

[135] M. Sobotka, D. Wollherr, and M. Buss. A jacobian method for online modification of precalculated gait trajectories. In *Proceedings of the 6th International*

Conference on Climbing and Walking Robots, pages 435–442, Catania, Italy, Sept 2003.

[136] S.-M. Song and K. J. Waldron. *Machines that Walk: the adaptive suspension vehicle.* The MIT Press, Cambridge, MA, USA, first edition, 1989.

[137] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control.* John Wiley & Sons, Hoboken, NJ, USA, 2006.

[138] M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75, 2006.

[139] S. Sung and Y. Youm. Landing motion control of articulated hopping robot. *International Journal of Advanced Robotic Systems*, 4(3):303–312, 2007.

[140] A. Takanishi, M. Ishida, Y. Yamazaki, and I Kato. The realization of dynamic walking by the biped robot wl-10rd. In *International Conference on Advanced Robotics*, pages 459–466, Tokyo, Japan, Sept. 1985.

[141] C. M. Thompson and M. H. Raibert. Passive dynamical running. In V. Hayward and O. Khatib, editors, *Experimental Robotics I*, pages 74–83. Springer-Verlag, NY, USA, 1989.

[142] F. E. Udwadia. A new perspective on the tracking control of nonlinear structural and mechanical systems. *Royal Society Proceedings: Mathematical, Physical and Engineering Sciences*, 459(2035):1783–1800, 2003.

[143] A. F. Vakakis and J. W. Burdick. Chaotic motions in the dynamics of a hopping robot. In *Proceedings of IEEE International Conference on Robotics*

and Automation, pages 1464–1469, Cincinnati, OH, USA, May 1990.

[144] R. Q. van der Linde. Passive bipedal walking with phasic muscle contraction. *Biological Cybernetics*, 81(3):227–237, 1999.

[145] J. Vermeulen. *Trajectory generation for planar hopping and walking robots: an objective parameter and angular momentum approach.* PhD thesis, Vrije Universiteit, Brussels, Belgium, 2004.

[146] M. K. Vokubratović and A. D. Rodić. Control of manipulation robots interacting with dynamic environment: Implementation and experiments. *IEEE Transactions on Industrial Electronics*, 42(4):358–366, 1995.

[147] M. Vukobratović and B. Borovać. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[148] M. Vukobratović, B. Borovać, D. Surla, and D. Stokić. *Biped Locomotion: dynamic stability, control and application.* Springer-Verlag, Berlin, 1990.

[149] M. Vukobratović, V. Potkonjak, and V. Matijević. Internal redundancy - the way to improve robot dynamics and control performances. *Journal of Intelligent and Robotic Systems*, 27(1/2):31–66, 2000.

[150] J. T. Wen and S. Murphy. Stability analysis of position and force control for robot arms. *IEEE Transactions on Automatic Control*, 36(3):365–370, 1991.

[151] E. Westervelt. *Toward a coherent framework for the control of planar biped locomotion.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 2003.

[152] P.-B. Wieber. On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, pages 53–59, Tsukuba, Japan, Oct. 2002.

[153] M. Wisse. *Essentials of dynamic walking - Analysis and design of two-legged robots.* PhD thesis, Delft University of Technology, Netherlands, 2004.

[154] D. Wollherr and M. Buss. Posture modification for biped humanoid robots based on jacobian method. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 124–129, Sendai, Japan, Sept.-Oct. 2004.

[155] R. Xu. *Optimal sliding mode control and stabilization of underactuated systems.* PhD thesis, Ohio State University, Columbus, OH, USA, 2007.

[156] V. Yen and M. L. Nagurka. Suboptimal trajectory planning of a five-link human locomotion model. In J. L. Stein, editor, *Biomechanics of Normal and Prosthetic Gait - ASME Winter Annual Meeting.*

[157] T. Yoshizawa. *Stability Theory by Liapunov's Second Method.* The Mathematical Society of Japan, Tokoyo, Japan, 1966.

[158] S. Yu, G. Guo, Z. Ma, and J. Du. Global fast terminal sliding mode control for robotic manipulators. *International Journal of Modelling, Identification and Control*, 1(1):72–79, 2006.

[159] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man. Continuous finite-time control for

robotic manipulators with terminal sliding mode. *Automatica*, 41(12):1957–1964, 2005.

[160] G. Zeglin. *Uniroo: A one legged dynamic hopping robot.* BS. Thesis, MIT, Cambridge, MA, USA, 1991.

[161] G. Zeglin and B. Brown. Control of a bow leg hopping robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 793–798, Leuven, Belgium, May 1998.

[162] D. Q. Zhang and S. K. Panda. Chattering-free and fast-response sliding mode controller. *IEE Proceedings of Control Theory and Applications*, 146(2):171–177, 1999.

[163] W. Zhang, G. Wang, T. Chambers, and W. E. Simon. Toward a folding-legged uniped that can learn to jump. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pages 4315–4319, Orlando, FL, USA, Oct. 1997.

[164] Z.-L. Zhou and C. C. Nguyen. Globally optimal trajectory planning for redundant manipulators using state space augmentation method. *Journal of Intelligent and Robotic Systems*, 19(1):105–117, 1997.

Appendix A

Model details of the hopper

This appendix presents the model details of the one-legged hopper discussed in Chapter 3. It may be useful for the readers who are interested in the complexity of legged locomotion problems.

A.1 System constants

Let $l_{g,i}$ be the length between the CoM and the lower end of the i^{th} link, and $r_i = l_{g,i}/l_i$. The system constants are listed as below:

$$\begin{aligned} c_1 &= \frac{l_1 M_1 r_1}{M_t}, \\ c_2 &= \frac{l_2(M_1 + M_2 r_2)}{M_t}, \\ c_3 &= \frac{l_3(M_1 + M_2 + M_3 r_3)}{M_t}. \end{aligned}$$

A.2 Model details of the flight phase

Constants for the flight phase are:

$$\begin{aligned} e_1 &= c_1 l_2 (M_3 + M_2(1 - r_2)), \\ e_2 &= c_1 l_3 M_3 (1 - r_3), \\ e_3 &= c_2 l_3 M_3 (1 - r_3). \end{aligned}$$

Rewriting (3.12), the model of the flight phase is

$$D^{\mathrm{f}}(q^{\mathrm{f}})\ddot{q}^{\mathrm{f}} + H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}})\dot{q}^{\mathrm{f}} + G^{\mathrm{f}}(q^{\mathrm{f}}) = B^{\mathrm{f}}\tau, \tag{A.1}$$

where $D^{\mathrm{f}}(q^{\mathrm{f}})$, $H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}})$, $G^{\mathrm{f}}(q^{\mathrm{f}})$, and B^{f} are

$$D^{\mathrm{f}}(q^{\mathrm{f}}) = \begin{bmatrix} M_{\mathrm{t}} & 0 & 0 & 0 & 0 \\ 0 & M_{\mathrm{t}} & 0 & 0 & 0 \\ 0 & 0 & d^{\mathrm{f}}_{3,3} & d^{\mathrm{f}}_{3,4} & d^{\mathrm{f}}_{3,5} \\ 0 & 0 & d^{\mathrm{f}}_{4,3} & d^{\mathrm{f}}_{4,4} & d^{\mathrm{f}}_{4,5} \\ 0 & 0 & d^{\mathrm{f}}_{5,3} & d^{\mathrm{f}}_{5,4} & d^{\mathrm{f}}_{5,5} \end{bmatrix}, \tag{A.2}$$

$$H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h^{\mathrm{f}}_{3,4} & h^{\mathrm{f}}_{3,5} \\ 0 & 0 & h^{\mathrm{f}}_{4,3} & 0 & h^{\mathrm{f}}_{4,5} \\ 0 & 0 & h^{\mathrm{f}}_{5,3} & h^{\mathrm{f}}_{5,4} & 0 \end{bmatrix}, \tag{A.3}$$

$$G^{\mathrm{f}}(q^{\mathrm{f}}) = (0, M_{\mathrm{t}}g, 0, 0, 0)^{\mathrm{T}}, \tag{A.4}$$

$$B^{\mathrm{f}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \tag{A.5}$$

with non-zero elements of $D^{\mathrm{f}}(q^{\mathrm{f}})$ being

$$\begin{aligned}
d^{\mathrm{f}}_{3,3} &= I_1 + \frac{l_1^2 r_1^2 M_1 (M_2 + M_3)}{M_{\mathrm{t}}}, \\
d^{\mathrm{f}}_{3,4} &= e_1 \cos(\theta_1 - \theta_2), \\
d^{\mathrm{f}}_{3,5} &= e_2 \cos(\theta_1 - \theta_3), \\
d^{\mathrm{f}}_{4,3} &= d^{\mathrm{f}}_{3,4}, \\
d^{\mathrm{f}}_{4,4} &= I_2 + \frac{l_2^2 (M_1 M_2 (1 - r_2)^2 + M_3 (M_1 + M_2 r_2^2))}{M_{\mathrm{t}}}, \\
d^{\mathrm{f}}_{4,5} &= e_3 \cos(\theta_2 - \theta_3), \\
d^{\mathrm{f}}_{5,3} &= d^{\mathrm{f}}_{3,5}, \\
d^{\mathrm{f}}_{5,4} &= d^{\mathrm{f}}_{4,5}, \\
d^{\mathrm{f}}_{5,5} &= I_3 + \frac{l_3^2 (M_1 + M_2) M_3 (1 - r_3)^2}{M_{\mathrm{t}}},
\end{aligned}$$

and non-zero elements of $H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}})$ being

$$\begin{aligned}
h^{\mathrm{f}}_{3,4} &= e_1 \sin(\theta_1 - \theta_2) \dot{\theta}_2, \\
h^{\mathrm{f}}_{4,3} &= -e_1 \sin(\theta_1 - \theta_2) \dot{\theta}_1, \\
h^{\mathrm{f}}_{3,5} &= e_2 \sin(\theta_1 - \theta_3) \dot{\theta}_3, \\
h^{\mathrm{f}}_{5,3} &= -e_2 \sin(\theta_1 - \theta_3) \dot{\theta}_1, \\
h^{\mathrm{f}}_{4,5} &= e_3 \sin(\theta_2 - \theta_3) \dot{\theta}_3, \\
h^{\mathrm{f}}_{5,4} &= -e_3 \sin(\theta_2 - \theta_3) \dot{\theta}_2.
\end{aligned}$$

A.3 Model details of the stance phase

Constants for the stance phase are:

$$\begin{aligned} t_1 &= c_1 M_{\mathrm{t}}, \\ t_2 &= c_2 M_{\mathrm{t}}, \\ t_3 &= c_3 M_{\mathrm{t}}, \\ t_4 &= t_1 l_2, \\ t_5 &= t_1 l_3, \\ t_6 &= t_2 l_3. \end{aligned}$$

A.3.1 The stance model

Rewriting (3.39), the model of the flight phase is

$$D^{\mathrm{s}}(q^{\mathrm{s}})\ddot{q}^{\mathrm{s}} + H^{\mathrm{s}}(q^{\mathrm{s}}, \dot{q}^{\mathrm{s}})\dot{q}^{\mathrm{s}} + G^{\mathrm{s}}(q^{\mathrm{s}}) = B^{\mathrm{s}}\tau^{\mathrm{s}}, \tag{A.6}$$

where $D^{s}(q^{s})$, $H^{s}(q^{s},\dot{q}^{s})$, $G^{s}(q^{s})$, and B^{s} are

$$D^{s}(q^{s}) = \begin{bmatrix} d^{s}_{1,1} & d^{s}_{1,2} & d^{s}_{1,3} \\ d^{s}_{2,1} & d^{s}_{2,2} & d^{s}_{2,3} \\ d^{s}_{3,1} & d^{s}_{3,2} & d^{s}_{3,3} \end{bmatrix}, \tag{A.7}$$

$$H^{s}(q^{s},\dot{q}^{s}) = \begin{bmatrix} 0 & h^{s}_{1,2} & h^{s}_{1,3} \\ h^{s}_{2,1} & 0 & h^{s}_{2,3} \\ h^{s}_{3,1} & h^{s}_{3,2} & 0 \end{bmatrix}, \tag{A.8}$$

$$G^{s}(q^{s}) = (g_1, g_2, g_3)^{\mathrm{T}}, \tag{A.9}$$

$$B^{s} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \tag{A.10}$$

with non-zero elements of $D^{s}(q^{s})$ being

$$\begin{aligned} d^{s}_{1,1} &= I_1 + l_1^2 r_1^2 M_1, \\ d^{s}_{1,2} &= t_4 \cos(\theta_1 - \theta_2), \\ d^{s}_{1,3} &= t_5 \cos(\theta_1 - \theta_3), \\ d^{s}_{2,1} &= d^{s}_{1,2}, \\ d^{s}_{2,2} &= I_2 + l_2^2 (M_1 + M_2 r_2^2), \\ d^{s}_{2,3} &= t_6 \cos(\theta_2 - \theta_3), \\ d^{s}_{3,1} &= d^{s}_{1,3}, \\ d^{s}_{3,2} &= d^{s}_{2,3}, \\ d^{s}_{3,3} &= I_3 + l_3^2 (M_1 + M_2 + M_3 r_3^2), \end{aligned}$$

non-zero elements of $H^s(q^s, \dot{q}^s)$ being

$$\begin{aligned}
h^s_{1,2} &= t_4 \sin(\theta_1 - \theta_2)\dot{\theta}_2, \\
h^s_{2,1} &= -t_4 \sin(\theta_1 - \theta_2)\dot{\theta}_1, \\
h^s_{1,3} &= t_5 \sin(\theta_1 - \theta_3)\dot{\theta}_3, \\
h^s_{3,1} &= -t_5 \sin(\theta_1 - \theta_3)\dot{\theta}_1, \\
h^s_{2,3} &= t_6 \sin(\theta_2 - \theta_3)\dot{\theta}_3, \\
h^s_{3,2} &= -t_6 \sin(\theta_2 - \theta_3)\dot{\theta}_2,
\end{aligned}$$

and elements of $G^s(q^s)$ being

$$\begin{aligned}
g_1 &= gt_1 \cos(\theta_1), \\
g_2 &= gt_2 \cos(\theta_2), \\
g_3 &= gt_3 \cos(\theta_3).
\end{aligned}$$

A.3.2 ZMP formulation

Some variables are defined as

$$
\begin{aligned}
a_1 &= c_1 \cos(\theta_1), \\
a_2 &= c_2 \cos(\theta_2), \\
a_3 &= c_3 \cos(\theta_3), \\
b_1 &= -c_1 \sin(\theta_1), \\
b_2 &= -c_2 \sin(\theta_2), \\
b_3 &= -c_3 \sin(\theta_3), \\
s_1 &= \sum_{\mathrm{i}=1}^{3} d^{\mathrm{s}}_{1,\mathrm{i}}, \\
s_2 &= \sum_{\mathrm{i}=1}^{3} d^{\mathrm{s}}_{2,\mathrm{i}}, \\
s_3 &= \sum_{\mathrm{i}=1}^{3} d^{\mathrm{s}}_{3,\mathrm{i}}, \\
s_4 &= -t_4 \sin(\theta_1 - \theta_2) - t_5 \sin(\theta_1 - \theta_3), \\
s_5 &= t_4 \sin(\theta_1 - \theta_2) - t_6 \sin(\theta_2 - \theta_3), \\
s_6 &= t_5 \sin(\theta_1 - \theta_3) + t_6 \sin(\theta_2 - \theta_3), \\
s_7 &= g M_{\mathrm{t}} \sum_{\mathrm{i}=1}^{3} a_{\mathrm{i}} = M_{\mathrm{t}} g X^{\mathrm{s}}_{\mathrm{g}}.
\end{aligned}
$$

By using the above variables, the ZMP location can be formulated as

$$
X_{\mathrm{zmp}} = \frac{\sum_{\mathrm{i}=1}^{3} s_{\mathrm{i}} \ddot{\theta}_{\mathrm{i}} + \sum_{\mathrm{i}=1}^{3} s_{\mathrm{i}+3} \dot{\theta}_{\mathrm{i}}^2 + s_7}{M_{\mathrm{t}} (\sum_{\mathrm{i}=1}^{3} a_{\mathrm{i}} \ddot{\theta}_{\mathrm{i}} + \sum_{\mathrm{i}=1}^{3} b_{\mathrm{i}} \dot{\theta}_{\mathrm{i}}^2 + g)}, \tag{A.11}
$$

$$
= \frac{\sum_{\mathrm{i}=1}^{3} s_{\mathrm{i}} \ddot{\theta}_{\mathrm{i}} + \sum_{\mathrm{i}=1}^{3} s_{\mathrm{i}+3} \dot{\theta}_{\mathrm{i}}^2 + M_{\mathrm{t}} g X^{\mathrm{s}}_{\mathrm{g}}}{M_{\mathrm{t}} (\ddot{Y}^{\mathrm{s}}_{\mathrm{g}} + g)}. \tag{A.12}
$$

Appendix B

Model of the biped

This appendix presents the model details of the biped discussed in Chapter 4.

B.1 System constants

Let $l_{g,i}$ be the length between the CoM and the lower end of the i^{th} link, and $r_i = l_{g,i}/l_i$. The system constants are listed as below:

$$\begin{aligned}
c_1 &= \frac{l_1 r_1 M_1}{M_t}, \\
c_2 &= \frac{l_2(M_1 + M_2(1+r_2) + M_3)}{M_t}, \\
c_3 &= -\frac{l_2(M_2(1-r_2) + M_3)}{M_t}, \\
c_4 &= \frac{l_3(M_1 + 2M_2 + M_3(1+r_3))}{M_t}, \\
c_5 &= -\frac{l_3 M_3(1-r_3)}{M_t}.
\end{aligned}$$

B.2 Model details of the flight phase

Constants for the flight phase are:

$$\begin{aligned} t_1 &= -c_3 M_\mathrm{t}, \\ t_2 &= -c_5 M_\mathrm{t}, \\ e_1 &= c_1 t_1, \\ e_2 &= c_1 t_2, \\ e_3 &= c_3 t_1, \\ e_4 &= c_2 t_2, \\ e_5 &= c_5 t_1, \\ e_6 &= c_3 t_2, \\ e_7 &= c_5 t_2, \end{aligned}$$

Rewriting (4.7), the model of the flight phase is

$$D^\mathrm{f}(q^\mathrm{f})\ddot{q}^\mathrm{f} + H^\mathrm{f}(q^\mathrm{f}, \dot{q}^\mathrm{f})\dot{q}^\mathrm{f} + G^\mathrm{f}(q^\mathrm{f}) = B^\mathrm{f}\tau^\mathrm{f}, \tag{B.1}$$

where $D^{\mathrm{f}}(q^{\mathrm{f}})$, $H^{\mathrm{f}}(q^{\mathrm{f}},\dot{q}^{\mathrm{f}})$, $G^{\mathrm{f}}(q^{\mathrm{f}})$, and B^{f} are

$$D^{\mathrm{f}}(q^{\mathrm{f}}) = \begin{bmatrix} M_{\mathrm{t}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M_{\mathrm{t}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d^{\mathrm{f}}_{3,3} & d^{\mathrm{f}}_{3,4} & d^{\mathrm{f}}_{3,5} & d^{\mathrm{f}}_{3,6} & d^{\mathrm{f}}_{3,7} \\ 0 & 0 & d^{\mathrm{f}}_{4,3} & d^{\mathrm{f}}_{4,4} & d^{\mathrm{f}}_{4,5} & d^{\mathrm{f}}_{4,6} & d^{\mathrm{f}}_{4,7} \\ 0 & 0 & d^{\mathrm{f}}_{5,3} & d^{\mathrm{f}}_{5,4} & d^{\mathrm{f}}_{5,5} & d^{\mathrm{f}}_{5,6} & d^{\mathrm{f}}_{5,7} \\ 0 & 0 & d^{\mathrm{f}}_{6,3} & d^{\mathrm{f}}_{6,4} & d^{\mathrm{f}}_{6,5} & d^{\mathrm{f}}_{6,6} & d^{\mathrm{f}}_{6,7} \\ 0 & 0 & d^{\mathrm{f}}_{7,3} & d^{\mathrm{f}}_{7,4} & d^{\mathrm{f}}_{7,5} & d^{\mathrm{f}}_{7,6} & d^{\mathrm{f}}_{7,7} \end{bmatrix}, \tag{B.2}$$

$$H^{\mathrm{f}}(q^{\mathrm{f}},\dot{q}^{\mathrm{f}}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h^{\mathrm{f}}_{3,4} & h^{\mathrm{f}}_{3,5} & h^{\mathrm{f}}_{3,6} & h^{\mathrm{f}}_{3,7} \\ 0 & 0 & h^{\mathrm{f}}_{4,3} & 0 & h^{\mathrm{f}}_{4,5} & h^{\mathrm{f}}_{4,6} & h^{\mathrm{f}}_{4,7} \\ 0 & 0 & h^{\mathrm{f}}_{5,3} & h^{\mathrm{f}}_{5,4} & 0 & h^{\mathrm{f}}_{5,6} & h^{\mathrm{f}}_{5,7} \\ 0 & 0 & h^{\mathrm{f}}_{6,3} & h^{\mathrm{f}}_{6,4} & h^{\mathrm{f}}_{6,5} & 0 & h^{\mathrm{f}}_{6,7} \\ 0 & 0 & h^{\mathrm{f}}_{7,3} & h^{\mathrm{f}}_{7,4} & h^{\mathrm{f}}_{7,5} & h^{\mathrm{f}}_{7,6} & 0 \end{bmatrix}, \tag{B.3}$$

$$G^{\mathrm{f}}(q^{\mathrm{f}}) = (0, M_{\mathrm{t}}g, 0, 0, 0, 0, 0)^{\mathrm{T}}, \tag{B.4}$$

$$B^{\mathrm{f}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \tag{B.5}$$

with non-zero elements of $D^{\mathrm{f}}(q^{\mathrm{f}})$ being

$$
\begin{aligned}
d^{\mathrm{f}}_{3,3} &= I_1 + 2c_1 r_1 l_1 (M_2 + M_3), \\
d^{\mathrm{f}}_{3,4} &= e_1 \cos(\theta_1 - \theta_2), \\
d^{\mathrm{f}}_{3,5} &= e_1 \cos(\theta_1 - \theta_3), \\
d^{\mathrm{f}}_{3,6} &= e_2 \cos(\theta_1 - \theta_4), \\
d^{\mathrm{f}}_{3,7} &= e_2 \cos(\theta_1 - \theta_5), \\
d^{\mathrm{f}}_{4,3} &= d^{\mathrm{f}}_{3,4}, \\
d^{\mathrm{f}}_{4,4} &= I_2 + \frac{l_2^2 (M_2(1-r_2)^2(M_1+M_2+M_3) + M_3(M_1 + M_2(1+r_2^2) + M_3))}{M_{\mathrm{t}}}, \\
d^{\mathrm{f}}_{4,5} &= e_3 \cos(\theta_2 - \theta_3), \\
d^{\mathrm{f}}_{4,6} &= e_4 \cos(\theta_2 - \theta_4), \\
d^{\mathrm{f}}_{4,7} &= e_5 \cos(\theta_2 - \theta_5), \\
d^{\mathrm{f}}_{5,3} &= d^{\mathrm{f}}_{3,5}, \\
d^{\mathrm{f}}_{5,4} &= d^{\mathrm{f}}_{4,5}, \\
d^{\mathrm{f}}_{5,5} &= d^{\mathrm{f}}_{4,4}, \\
d^{\mathrm{f}}_{5,6} &= e_6 \cos(\theta_3 - \theta_4), \\
d^{\mathrm{f}}_{5,7} &= e_4 \cos(\theta_3 - \theta_5), \\
d^{\mathrm{f}}_{6,3} &= d^{\mathrm{f}}_{3,6}, \\
d^{\mathrm{f}}_{6,4} &= d^{\mathrm{f}}_{4,6}, \\
d^{\mathrm{f}}_{6,5} &= d^{\mathrm{f}}_{5,6}, \\
d^{\mathrm{f}}_{6,6} &= I_3 + \frac{l_3^2 M_3 (1-r_3)^2 (M_1 + 2M_2 + M_3)}{M_{\mathrm{t}}}, \\
d^{\mathrm{f}}_{6,7} &= e_7 \cos(\theta_4 - \theta_5),
\end{aligned}
$$

$$
\begin{aligned}
d^{\mathrm{f}}_{7,3} &= d^{\mathrm{f}}_{3,7}, \\
d^{\mathrm{f}}_{7,4} &= d^{\mathrm{f}}_{4,7}, \\
d^{\mathrm{f}}_{7,5} &= d^{\mathrm{f}}_{5,7}, \\
d^{\mathrm{f}}_{7,6} &= d^{\mathrm{f}}_{6,7}, \\
d^{\mathrm{f}}_{7,7} &= d^{\mathrm{f}}_{6,6},
\end{aligned}
$$

and non-zero elements of $H^{\mathrm{f}}(q^{\mathrm{f}}, \dot{q}^{\mathrm{f}})$ being

$$
\begin{aligned}
h^{\mathrm{f}}_{3,4} &= e_1 \sin(\theta_1 - \theta_2)\dot{\theta}_2, \\
h^{\mathrm{f}}_{3,5} &= e_1 \sin(\theta_1 - \theta_3)\dot{\theta}_3, \\
h^{\mathrm{f}}_{3,6} &= e_2 \sin(\theta_1 - \theta_4)\dot{\theta}_4, \\
h^{\mathrm{f}}_{3,7} &= e_2 \sin(\theta_1 - \theta_5)\dot{\theta}_5, \\
h^{\mathrm{f}}_{4,3} &= -e_1 \sin(\theta_1 - \theta_2)\dot{\theta}_1, \\
h^{\mathrm{f}}_{4,5} &= e_3 \sin(\theta_2 - \theta_3)\dot{\theta}_3, \\
h^{\mathrm{f}}_{4,6} &= e_4 \sin(\theta_2 - \theta_4)\dot{\theta}_4, \\
h^{\mathrm{f}}_{4,7} &= e_5 \sin(\theta_2 - \theta_5)\dot{\theta}_5, \\
h^{\mathrm{f}}_{5,3} &= -e_1 \sin(\theta_1 - \theta_3)\dot{\theta}_1, \\
h^{\mathrm{f}}_{5,4} &= -e_3 \sin(\theta_2 - \theta_3)\dot{\theta}_2, \\
h^{\mathrm{f}}_{5,6} &= e_6 \sin(\theta_3 - \theta_4)\dot{\theta}_4, \\
h^{\mathrm{f}}_{5,7} &= e_4 \sin(\theta_3 - \theta_5)\dot{\theta}_5, \\
h^{\mathrm{f}}_{6,3} &= -e_2 \sin(\theta_1 - \theta_4)\dot{\theta}_1, \\
h^{\mathrm{f}}_{6,4} &= -e_4 \sin(\theta_2 - \theta_4)\dot{\theta}_2,
\end{aligned}
$$

$$
\begin{aligned}
h^{\mathrm{f}}_{6,5} &= -e_6 \sin(\theta_3 - \theta_4)\dot{\theta}_3, \\
h^{\mathrm{f}}_{6,7} &= e_7 \sin(\theta_4 - \theta_5)\dot{\theta}_5, \\
h^{\mathrm{f}}_{7,3} &= -e_2 \sin(\theta_1 - \theta_5)\dot{\theta}_1, \\
h^{\mathrm{f}}_{7,4} &= -e_5 \sin(\theta_2 - \theta_5)\dot{\theta}_2, \\
h^{\mathrm{f}}_{7,5} &= -e_4 \sin(\theta_3 - \theta_5)\dot{\theta}_3, \\
h^{\mathrm{f}}_{7,6} &= -e_7 \sin(\theta_4 - \theta_5)\dot{\theta}_4,
\end{aligned}
$$

B.3 Model details of the stance phase

Constants for the stance phase are:

$$
\begin{aligned}
t_1 &= c_1 M_{\mathrm{t}}, \\
t_2 &= c_2 M_{\mathrm{t}}, \\
t_3 &= c_3 M_{\mathrm{t}}, \\
t_4 &= c_4 M_{\mathrm{t}}, \\
t_5 &= c_5 M_{\mathrm{t}}, \\
e_1 &= t_1 l_2, \\
e_2 &= t_1 l_3, \\
e_3 &= t_3 l_2, \\
e_4 &= t_5 l_2, \\
e_5 &= t_3 l_3, \\
e_6 &= t_5 l_3, \\
e_7 &= t_2 l_3.
\end{aligned}
$$

B.3.1 The stance model

Define some variables as follows:

$$\begin{aligned}
n_1 &= e_1 \sin(\theta_1 - \theta_3), \\
n_2 &= e_2 \sin(\theta_1 - \theta_5), \\
n_3 &= e_3 \sin(\theta_2 - \theta_3), \\
n_4 &= e_4 \sin(\theta_2 - \theta_4), \\
n_5 &= e_5 \sin(\theta_2 - \theta_5), \\
n_6 &= e_4 \sin(\theta_3 - \theta_4), \\
n_7 &= e_7 \sin(\theta_3 - \theta_5), \\
n_8 &= e_6 \sin(\theta_4 - \theta_5).
\end{aligned}$$

Rewriting (4.26), the dynamic model of the stance phase is

$$D^s(q^s)\ddot{q}^s + H^s(q^s, \dot{q}^s)\dot{q}^s + G^s(q^s) = B^s\tau^s, \tag{B.6}$$

where $D^s(q^s)$, $H^s(q^s, \dot{q}^s)$, $G^s(q^s)$, and B^s are

$$D^s(q^s) = \begin{bmatrix} d^s_{1,1} & d^s_{1,2} & d^s_{1,3} & d^s_{1,4} & d^s_{1,5} \\ d^s_{2,1} & d^s_{2,2} & d^s_{2,3} & d^s_{2,4} & d^s_{2,5} \\ d^s_{3,1} & d^s_{3,2} & d^s_{3,3} & d^s_{3,4} & d^s_{3,5} \\ d^s_{4,1} & d^s_{4,2} & d^s_{4,3} & d^s_{4,4} & d^s_{4,5} \\ d^s_{5,1} & d^s_{5,2} & d^s_{5,3} & d^s_{5,4} & d^s_{5,5} \end{bmatrix}, \tag{B.7}$$

$$H^s(q^s, \dot{q}^s) = \begin{bmatrix} h^s_{1,1} & h^s_{1,2} & h^s_{1,3} & h^s_{1,4} & h^s_{1,5} \\ h^s_{2,1} & h^s_{2,2} & h^s_{2,3} & h^s_{2,4} & h^s_{2,5} \\ h^s_{3,1} & h^s_{3,2} & h^s_{3,3} & h^s_{3,4} & h^s_{3,5} \\ h^s_{4,1} & h^s_{4,2} & h^s_{4,3} & h^s_{4,4} & h^s_{4,5} \\ h^s_{5,1} & h^s_{5,2} & h^s_{5,3} & h^s_{5,4} & h^s_{5,5} \end{bmatrix}, \tag{B.8}$$

$$G^s(q^s) = (g_1, g_2, g_3, g_4, g_5)^{\mathrm{T}}, \tag{B.9}$$

$$B^s = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \tag{B.10}$$

with all elements of $D^s(q^s)$ being

$$
\begin{aligned}
d^s_{1,1} &= I_1 + r_1^2 l_1^2 M_1, \\
d^s_{1,2} &= 0, \\
d^s_{1,3} &= e_1 \cos(\theta_1 - \theta_3), \\
d^s_{1,4} &= 0, \\
d^s_{1,5} &= e_2 \cos(\theta_1 - \theta_5), \\
d^s_{2,1} &= d^s_{1,2}, \\
d^s_{2,2} &= I_2 + l_2^2(M_2(1 - r_2)^2 + M_3), \\
d^s_{2,3} &= e_3 \cos(\theta_2 - \theta_3), \\
d^s_{2,4} &= -e_4 \cos(\theta_2 - \theta_4), \\
d^s_{2,5} &= e_5 \cos(\theta_2 - \theta_5), \\
d^s_{3,1} &= d^s_{1,3}, \\
d^s_{3,2} &= d^s_{2,3}, \\
d^s_{3,3} &= I_2 + l_2^2(M_1 + M_2(1 + r_2^2) + M_3), \\
d^s_{3,4} &= e_4 \cos(\theta_3 - \theta_4), \\
d^s_{3,5} &= e_7 \cos(\theta_3 - \theta_5), \\
d^s_{4,1} &= d^s_{1,4}, \\
d^s_{4,2} &= d^s_{2,4}, \\
d^s_{4,3} &= d^s_{3,4}, \\
d^s_{4,4} &= I_3 + l_3^2 M_3(1 - r_3)^2, \\
d^s_{4,5} &= e_6 \cos(\theta_4 - \theta_5),
\end{aligned}
$$

$$
\begin{aligned}
d^{s}_{5,1} &= d^{s}_{1,5}, \\
d^{s}_{5,2} &= d^{s}_{2,5}, \\
d^{s}_{5,3} &= d^{s}_{3,5}, \\
d^{s}_{5,4} &= d^{s}_{4,5}, \\
d^{s}_{5,5} &= I_3 + l_3^2(M_1 + 2M_2 + M_3(1 + r_3^2)),
\end{aligned}
$$

all elements of $H^{s}(q^{s}, \dot{q}^{s})$ being

$$
\begin{aligned}
h^{s}_{1,1} &= 0, \\
h^{s}_{1,2} &= 0, \\
h^{s}_{1,3} &= n_1\dot{\theta}_3, \\
h^{s}_{1,4} &= 0, \\
h^{s}_{1,5} &= n_2\dot{\theta}_5, \\
h^{s}_{2,1} &= 0, \\
h^{s}_{2,2} &= 0, \\
h^{s}_{2,3} &= n_3\dot{\theta}_3, \\
h^{s}_{2,4} &= -n_4\dot{\theta}_4, \\
h^{s}_{2,5} &= n_5\dot{\theta}_5, \\
h^{s}_{3,1} &= -n_1\dot{\theta}_1, \\
h^{s}_{3,2} &= -n_3\dot{\theta}_2, \\
h^{s}_{3,3} &= 0, \\
h^{s}_{3,4} &= n_6\dot{\theta}_4, \\
h^{s}_{3,5} &= n_7\dot{\theta}_5,
\end{aligned}
$$

$$
\begin{aligned}
h^{s}_{4,1} &= 0, \\
h^{s}_{4,2} &= n_4\dot{\theta}_2, \\
h^{s}_{4,3} &= -n_6\dot{\theta}_3, \\
h^{s}_{4,4} &= 0, \\
h^{s}_{4,5} &= n_8\dot{\theta}_5, \\
h^{s}_{5,1} &= -n_2\dot{\theta}_1, \\
h^{s}_{5,2} &= -n_5\dot{\theta}_2, \\
h^{s}_{5,3} &= -n_7\dot{\theta}_3, \\
h^{s}_{5,4} &= -n_8\dot{\theta}_4, \\
h^{s}_{5,5} &= 0,
\end{aligned}
$$

and elements of $G^{s}(q^{s})$ being

$$
\begin{aligned}
g_1 &= gt_1\cos(\theta_1), \\
g_2 &= gt_3\cos(\theta_2), \\
g_3 &= gt_2\cos(\theta_3), \\
g_4 &= gt_5\cos(\theta_4), \\
g_5 &= gt_4\cos(\theta_5).
\end{aligned}
$$

B.3.2 ZMP formulation

Define some variables as

$$
\begin{aligned}
a_1 &= c_1 \cos(\theta_1), \\
a_2 &= c_3 \cos(\theta_2), \\
a_3 &= c_2 \cos(\theta_3), \\
a_4 &= c_5 \cos(\theta_4), \\
a_5 &= c_4 \cos(\theta_5), \\
b_1 &= -c_1 \sin(\theta_1), \\
b_2 &= -c_3 \sin(\theta_2), \\
b_3 &= -c_2 \sin(\theta_3), \\
b_4 &= -c_5 \sin(\theta_4), \\
b_5 &= -c_4 \sin(\theta_5),
\end{aligned}
$$

$$
\begin{aligned}
s_1 &= \sum_{1}^{5} d^{s}_{1,i}, \\
s_2 &= \sum_{1}^{5} d^{s}_{2,i}, \\
s_3 &= \sum_{1}^{5} d^{s}_{3,i}, \\
s_4 &= \sum_{1}^{5} d^{s}_{4,i}, \\
s_5 &= \sum_{1}^{5} d^{s}_{5,i}, \\
p_1 &= -n_1 - n_2, \\
p_2 &= -n_3 + n_4 - n_5, \\
p_3 &= n_1 + n_3 - n_6 - n_7, \\
p_4 &= -n_4 + n_6 - n_8, \\
p_5 &= n_2 + n_5 + n_7 + n_8, \\
q &= gM_t \sum_{i=1}^{5} a_i = gM_t X^{s}_{g}.
\end{aligned}
$$

By using the above variables, the ZMP location can be formulated as

$$
X_{zmp} = \frac{\sum_{i=1}^{5} s_i \ddot{\theta}_i + \sum_{i=1}^{5} p_i \dot{\theta}_i^2 + q}{M_t(\sum_{i=1}^{5} a_i \ddot{\theta}_i + \sum_{i=1}^{5} b_i \dot{\theta}_i^2 + g)}, \tag{B.11}
$$

$$
= \frac{\sum_{i=1}^{5} s_i \ddot{\theta}_i + \sum_{i=1}^{5} p_i \dot{\theta}_i^2 + M_t g X^{s}_{g}}{M_t(\ddot{Y}^{s}_{g} + g)}. \tag{B.12}
$$