

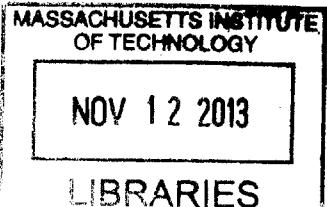
**Hierarchical controller for highly dynamic
locomotion utilizing pattern modulation and
impedance control: implementation on the MIT
Cheetah robot**

by

Jongwoo Lee

B.S., Seoul National University (2011)

ARCHIVES



LIBRARIES

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

© Massachusetts Institute of Technology 2013. All rights reserved.

Author
[Signature]

Department of Mechanical Engineering

August 23, 2013

Certified by

Sangbae Kim

Esther & Harold E. Edgerton Assistant Professor of Mechanical Engineering

[Signature] Thesis Supervisor

Accepted by

David E. Hardt

Ralph E. & Eloise F. Cross Professor of Mechanical Engineering
Chairman, Committee on Graduate Students

**Hierarchical controller for highly dynamic locomotion
utilizing pattern modulation and impedance control:
implementation on the MIT Cheetah robot**

by

Jongwoo Lee

Submitted to the Department of Mechanical Engineering
on August 23, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

This thesis presents a hierarchical control algorithm for quadrupedal locomotion. We address three challenges in developing a controller for high-speed running: locomotion stability, control of ground reaction force, and coordination of four limbs. To tackle these challenges, the proposed algorithm employs three strategies. Leg impedance control provides programmable virtual compliance of each leg which achieve self-stability in locomotion. The four legs exert forces to the ground using equilibrium-point hypothesis. A gait pattern modulator imposes a desired footfall sequence. The control algorithm is verified in a dynamic simulator constructed using MATLAB and then in the subsequent experiments on the MIT Cheetah robot. The experiments on the MIT Cheetah robot demonstrates high speed trot running reaching up to the speed of 6 m/s on a treadmill. This speed corresponds to a Froude number ($Fr = 7.34$), which is comparatively higher than other existing quadrupedal robots.

Thesis Supervisor: Sangbae Kim

Title: Esther & Harold E. Edgerton Assistant Professor of Mechanical Engineering

Acknowledgments

I would like to begin by thanking my advisor, Prof. Sangbae Kim for his advice, patience, and understanding. I thank the members of the MIT Biomimetic Robotics Laboratory for their help. I am especially pleased to have met Dr.Dong Jin Hyun in this lab. Without his support and encouragement, I would not have achieved anything. I also give many thanks to Sangok Seok for his advice and efforts to operate the MIT Cheetah robot.

It has been a pleasure to meet all of my friends in Boston, especially the members of MIT KGSA. It was memorable to hang out with Albert Wang and Meng Yee (Michael) Chuah. Also, I give thanks to Yoobin (Anna) Seo for correcting grammatical errors.

Additionally, I appreciate Samsung Foundation of Culture for their unconditional financial support for the last two years.

Finally, and most importantly, I would like to give thanks to my family; Kyuheon Lee, my father, Namhee Jung, my mother, and Jongmin Lee, my Wangjongmin. Without their love and support, I would not have finished my study.

The project is supported by the Defense Advanced Research Projects Agency (DARPA) Maximum Mobility and Manipulation (M3) program.

Contents

1	Introduction	19
1.1	Motivation and objectives	19
1.2	Current state of research	20
1.3	Our approaches and accomplishments	24
1.4	Structure of this thesis	27
2	Modeling of the MIT Cheetah robot	29
2.1	Hybrid dynamic systems	31
2.1.1	Hybrid dynamic systems in legged systems	31
2.1.2	hybrid dynamic systems realized in MATLAB	32
2.2	Constrained equations of motion	34
2.2.1	Multi-link rigid body equations of motion	34
2.2.2	Numerical error inhibition	35
2.3	Friction model	36
2.4	Impact map	40
3	Quadrupedal Locomotive Control Framework	45
3.1	Gait pattern modulator with proprioceptive sensory feedback	46
3.1.1	Phase signals	47
3.1.2	Synchronization by proprioceptive sensory feedback	48
3.1.3	Imposing gait patterns	49
3.1.4	Phase signal generation according to TD elapsed time and phase differences	49

3.2	Leg trajectory generator	52
3.2.1	Design of the trajectory in the swing phase	53
3.2.2	Design of the trajectory in the stance phase	56
3.2.3	Scaling and translation of the designed trajectory	59
3.3	Low level individual leg controller	59
4	Simulation result	63
4.1	Trot running with constant speed	64
4.2	Trot running with acceleration and gait transition to gallop	66
5	Experiments on the MIT Cheetah robot	71
5.1	Experimental setup	71
5.2	Control parameters for the experiment	73
5.3	Experimental results	75
5.3.1	Estimated locomotion speed	75
5.3.2	Measured leg trajectories	76
5.3.3	Estimation of generated leg-force based on applied currents	78
5.3.4	Gait transition	79
6	Discussion	81
6.1	Estimated pitch variation	81
6.2	Fr and COT	83
6.3	Evidences for the MIT Cheetah running	84
6.4	Estimated force exerted on foot end-effectors	85
6.5	Force analysis on the front legs	86
6.6	Contribution of virtual stiffness and virtual damping to leg compliance	88
6.7	Force analysis on the back legs	89
7	Concluding Remarks and Future Research Topics	91
A	Model Parameters	95
B	Back spine coordination	97

C Leg compliance	99
D InverseKinematics	101
E Supplementary Video	103

List of Figures

1-1	MIT Cheetah robot in the experiment, running with trot gait pattern on the treadmill while constrained on its sagittal plane.	27
2-1	The planar-rigid model of the MIT Cheetah robot. A planar model is constructed since the robot is constrained in its sagittal plane for the current experimental setup. Leg indices are also listed. FR: front and right side. FL: front and left side. BR: back and right side. BL: back and left side.	29
2-2	Typical schematic diagram for finite state machine	31
2-3	Finite state machine to model dynamics of legged mechanism	32
3-1	Schematic overall structure of the proposed hierarchical locomotive control framework. Each block represents each step and corresponding control parameters. In this schematic diagram, the <i>reference leg</i> is front right leg, FR.	46
3-2	Sawtooth phase signals generated by phase modulator for trot gait and its corresponding foot fall pattern diagram. The phase difference is represented in the diagram.	51
3-3	Sawtooth phase signals generated by phase modulator for gallop gait and its corresponding foot fall pattern diagram. The phase difference is represented in the diagram	51

3-4	The reference desired foot end trajectory with control parameters. 12 control points of Bézier curve are shown which define the swing phase trajectory(black solid line). Multiple overlapped points are denoted as ‘ $\times n$ ’. Stance phase trajectory as sinusoidal wave is also described in the figure(red solid line).	52
3-5	separation of 12 control points of Bézier curve for swing phase trajectory into vertical direction and horizontal direction	54
3-6	Stance trajectory design with equilibrium-point hypothesis: Along with vertical axis, the tracking error between the actual foot-end position and the designed foot-end trajectory is converted into the vertical force incorporated with the leg compliance. The time-integral of the force during the stance period is the vertical impulse.	58
3-7	Front leg with the impedance control: virtual leg compliance realization with stiffness and damping for radial/angular directions	60
3-8	Block diagram for leg control; implementation of the impedance control to create virtual leg compliance.	61
4-1	Animation interface constructed using MATLAB. 3-D animation of planar simulator, ground reaction forces, velocity, pitch variation, and leg phase(stance: 1 / swing: 0) are shown in the animation.	63
4-2	Leg trajectory designed for stable trot running. Both front and back leg trajectories are presented with respect to local shoulder/hip coordinates.	64
4-3	Trot running at 3.4 m/s. Velocity (top) and pitch of the front body (middle) are plotted over time. Height variation over distance traveled is also plotted (bottom). Pitch variation and height variation are damped out as the robot approaches to the steady state running.	65

4-4	Measured ground reaction forces (top) and estimated ground reaction forces at each leg (bottom) between time $t = 12$ s to $t = 14$ s. Vertical ground reaction forces (solid lines) and horizontal ground reaction forces (dotted lines) are plotted together. Solid-dotted horizontal line indicates the weight, mg , of the robot, where $m = 29.443kg$ in the simulation. Input joint torques are coordinate transformed to estimate ground reaction forces.	66
4-5	Snapshots of trot running at 3.4 m/s in the simulation	66
4-6	Leg trajectory designed for gallop running. Both front and back leg trajectories are presented with respect to local shoulder/hip coordinates.	68
4-7	The state of all four legs from $t = 6$ s to $t = 9$ s. Gait transition happened at $t = 8$ s.	69
4-8	Trot acceleration and gait transition to gallop running. Velocity (top) and pitch of the front body (middle) are plotted over time. Height variation over distance traveled is also plotted (bottom). From $t = 7$ s to $t = 8$ s, the robot accelerates and at $t = 8$ s, transits its gait from trot to gallop.	70
4-9	Measured ground reaction forces between time $t = 6$ s to $t = 9$ s. Gait transition happened at $t = 8$ s. Solid lines are vertical forces, and dotted lines are horizontal forces.	70
4-10	Snapshots of gait transition from trot to gallop by varying gait pattern parameters $\Delta\vec{S}$ in linear over time	70
5-1	Experimental setup: the MIT Cheetah robot runs on the speed-controlled treadmill. Minimal friction sliding guides constrain the robot on its sagittal plane.	72
5-2	Gait trajectories implemented on the experiment with leg workspaces	73
5-3	Snapshot for the trot running gait of the MIT Cheetah robot with each footfall diagram: the existence of the aerial phase is observed between each paired legs' step in trotting gait	74

5-4	Estimated forward velocity (top) of the MIT Cheetah robot: ± 0.15 m/s bounded difference was observed between the estimated speed and the treadmill speed measured by a rotary encoder (CUI Inc. AMT 102-V) directly attached to the DC motor main axis. Froude number (middle, blue solid line) and cost of transport(middle, red solid line) of the robot are plotted together.	75
5-5	Graph set of generated desired trajectories by the leg trajectory generator and measured actual gait trajectories during trotting gait from 1 to 6 m/s for a) front right leg, b) front left leg, c) back right leg, d) back left leg. In each leg graph, <i>top left</i> : desired trajectories, <i>top right</i> : actual trajectories with respect to speed, <i>bottom left</i> : comparison between desired/actual trajectories, <i>bottom right</i> : comparison between actual trajectories at low/high speeds, respectively	76
5-6	Commanded currents applied to each EM motor for shoulder/hip and knee joints at leg $i \in \{FL, FR, BR, BL\}$: The motor driver supplies the commanded current to the EM motor which proportionally generate torque by the torque constant K_τ	78
5-7	Sawtooth phase signals generated by phase modulator for gait transition from trot to gallop. The phase signal converges from one to another, linearly with time. Gait transition through linear convergence from the trot parameters to the gallop parameters	79
5-8	Snapshots of gait transition from trot to gallop recorded by a high speed video captured at 500 fps depicting 150 ms	80
6-1	Actual measured trajectories of four legs with respect to each should/hip joint, where all the trajectories for four legs are overlapped with respect to each shoulder/hip joint. The data allow us to approximate the height/pitch variation of the robot during trot experiment.	82

6-2	Estimated shoulder height trajectories: convex shape of shoulder height trajectory as a typical representation of running is observed after 2.2 m/s.	84
6-3	Estimated duty factor D_{est} for the front right leg: measured kinematic data for virtual leg is used to estimate duration of the stance/swing phases.	85
6-4	Estimation of generated forces in Cartesian coordinate at foot-end of front legs. Stance phase(high vertical force region) and swing phase(low vertical force region) can be approximated from the graph. Increase in peak force value is observed with decrease in duty factor.	86
6-5	Estimation of generated forces in Cartesian coordinate at foot-end of back legs. Increase in peak force value is observed with decrease in duty factor	87
6-6	Desired/actual vertical virtual leg lengths L_v	87
6-7	Virtual stiffness and damping contribution to create vertical force at the foot-end	88
C-1	Measured virtual leg compliance in the polar coordinate (r, θ) : a) longitudinal stiffness b) angular stiffness	99
D-1	Inverse kinematics of three-segment leg. The proximal and the distal segment are parallel due to pantographic leg design.	101

List of Tables

2.1	Possible states of a leg after impact and corresponding necessary conditions	42
3.1	Nomenclature of the set of control parameters	47
3.2	12 control points for Bézier curve for designing swing phase trajectory.	55
4.1	Set of control parameters designed for the trot running simulation.	64
4.2	Set of control parameters designed for the gallop running simulation.	69
5.1	Set of control parameters designed for the experiment.	73
A.1	Model Parameters	95
E.1	Supplementary videos and urls are listed in the table	103

Chapter 1

Introduction

1.1 Motivation and objectives

Developing dynamic legged machines has been an active field of research in robotics to exploit potential advantages of the legged locomotion. It is widely believed that legged machines can outperform wheeled machines on rough, irregular and discontinuous terrains in terms of mobility, versatility and energy efficiency [1]. This belief is supported by locomotion skills that animals exhibit on natural terrains which are unreachable for wheeled vehicles [2]. These advantages have increased investigations of legged robots for a variety of tasks such as future transportation, disaster response operations, and military purposes. Despite many robots already proved their ability for stable walking, the performance of legged robots has not reached our expectation; few robots show both stable and agile performance in a rough terrain on which animals can traverse easily.

In this thesis, we mainly aim to develop a controller to accomplish a stable high-speed quadrupedal locomotion. To develop such a controller, we try to exploit insights from nature as well as the previous approaches performed by other roboticists. Therefore, a study on legged locomotion in general such as biological findings, simplified mathematical models for quadrupeds, and controllers developed for quadrupedal robots is conducted. In order to verify the controller, we develop a dynamic simulator using MATLAB, and mathematical derivation of the simulator is also presented. Fi-

nally, the controller is implemented on the MIT Cheetah robot, and the experimental results and analyses of the data are presented.

1.2 Current state of research

Biological findings

Quadrupedal animals have various kinds of gait patterns (walk, trot, pronk, bound, half-bound, rotary gallop, transverse gallop, etc.). Interestingly, Alexander revealed dynamic similarity between different sized quadrupeds, by defining a non-dimensional speed, Froude number Fr , which is strongly correlated with gait patterns [3]. The mechanism and reason of gait transition from one to another is still in question, but it was observed that quadrupeds use the same gait pattern when their corresponding Fr are the same.

The studies of quadrupeds discovered the relation between speed of locomotion, gait pattern and spatial/temporal limb coordination such as stride length and stride frequency. Dogs and horses tend to use different gait patterns over different speeds. It has been revealed that the way animals modulate their speed varies by gait pattern: dogs and horses modulate their stride frequency when they use trot, whereas stride length is modulated when they use gallop. In both gait patterns, the duration of leg swing is maintained constant for various speeds [4], [5], [6] and [7]. Metabolic energy efficiency is often suggested to explain the reason why dogs and horses change their gait from trot for mid-speed running to gallop for high-speed running [5], [7], while other researchers speculate that horses transit gait pattern to reduce peak forces for each limb [8].

Study of quadrupedal locomotion with reduced-order models

Researchers found reduced-order models that represent the dynamic behaviors of legged locomotion, which provided better understanding of their characteristics. The Spring-Loaded Inverted Pendulum (SLIP) model is the most well known model which

well predicts the center of mass behavior of running animals with symmetric gait pattern [9]. In this model, the center of mass behavior is represented as the motion of a point mass which is attached to an ideal spring. Farley et al. examined that the effective stiffness of the spring is independent from locomotion speed but determined by body mass [6].

Ruina et al. [10] developed a collisional model to study the mechanism of redirecting vertical body motion from downward to upward during running, in the perspective of collisional energy loss. The collisional loss due to impact can be reduced by distributing impacts through multiple foot strikes. Aligned with this model, Bertram et al. [11] tried to explain fundamental difference in mechanics between hind-limb initiated gallop and front-limb initiated gallop, while explaining why galloping is effective at high speed in terms of metabolic cost.

Such simple models exhibit fundamental behaviors of lumped dynamics of legged locomotion, whereas more elaborate models with specific controllers explain the details of quadrupedal locomotion with various gait patterns.

A planar model consists of a rigid body and four mass-less springy legs described pronk and bound of quadrupeds [12] with *Energy-pumping* feedback controller. Similar models were derived in [13] and [14] to compare trot, bound and gallop and to investigate coordination of four limbs. Herr and McMahon developed a similar planar model with additional degrees of freedom in body and legs [15], [16]. Feedforward/feedback control were applied to impose the sequence of four legs with self-stability property. Stride length and stride frequency were related with forward running speed to predict animal behavior.

The concept of Central Pattern Generator (CPG) has been introduced as a way to generate gait patterns. Evidences in biology suggested that a neural network mechanism can generate gait patterns without any sensory input, and the rhythmic pattern of limbs and body dynamics are coordinated with sensory feedback [17], [18]. CPGs are often modeled as a set of nonlinear oscillators of which phases are coupled with each other, to mimic neural networks [19]. Smooth gait transition was possible by employing existence of the limit-cycle and bifurcation property of such oscillators

[20]. The stability and synchronization properties have been studied with *contraction analysis* in [21], [22], [23]. An open research topic in controllers using CPGs is to coordinate the robot body dynamics and the CPG signals with sensory feedback.

Controllers for quadrupeds

Animal studies provided the roboticists with a better understanding in locomotive mechanism and guides to possible control strategies. A controller based on the variation of the SLIP model was implemented on RHex robot to achieve stable locomotion on rough terrain [24]. The stability of the SLIP model also inspired hybrid controllers which combined the passive dynamics predicted by the SLIP model and higher level controller [25], [26]. Nonlinear oscillators that mimic biological CPG include Fukuoka model [27], Van-der-pol oscillator, [28], and Hopf-oscillator [29]. The Tekken [27] and the Kotetsu [30] achieved walking on a rough terrain, with sensory feedback from ground reaction forces and body attitude. The Cheetah-cub [31] showed a successful trot running using an open-loop CPG, depending on the passive stability given from compliance of the legs.

On the other hand, Raibert and his company Boston Dynamics, the pioneers of the legged robotics community, have developed control algorithms which are not directly inspired by biological findings [32], [33]. The controllers for their recent work are not published, although they have achieved remarkable performances. Yoneda et al. presented a stable walking platform which implemented a ZMP-based controller [34]. Trot and pronk gait was realized by a leg thrust regulation controller on the KOLT robot [35]. A combination of intuitive approaches for a quadrupedal robot controller achieved various performances in simulation, with a global state feedback [36].

Optimization techniques have been also used to design control algorithms for legged locomotion. Optimal motion planning algorithms were implemented on LittleDog platform to achieve a walking and a bounding gait [37], [38], with a machine learning technique [39], for slow locomotion on rough terrain where careful motion planning is required. Genetic algorithm (GA) has been also proposed for trajectory optimization problems for legged machines, because it is a suitable multi-objective

optimization tool for dynamical systems with a large number of degrees of freedom. These properties make GA fits to legged locomotion problem [40]. For example, foot trajectories were optimized for the metabolic cost of locomotion using GA [41].

In order to directly control body dynamics of the robot, there have been efforts to optimize forces exerted by four legs. Valenzuela and Kim showed the possibility of pronk and bound gait for planar quadrupeds by optimizing hip torques at each stride [42]. The StarlETH realized walk and trot by computing joint torques to generate optimal forces and torques for controlling posture of the body [43].

Dynamic simulation techniques

Dynamic simulation techniques for a rigid body with kinematic constraints have been developed in two different ways. One approach directly calculates forces to impose the constraints, and the other, *penalty method*, indirectly applies the forces through displacements. Barzel et al.[44] and Witkins et al.[45] proposed the former method. Frictional dynamics on non-penetrating rigid body was also simulated with the Coulomb friction model [46].

The constraints enforced by the former method are violated due to numerical errors accumulated during integration. Various methods have been extensively developed to eliminate this *drift* on the constraints, either in state-space formulation or projection methods [47], [48], [49], [50], [51].

Among various methods, Baumgarte's stabilization method, which creates a second-order dynamics for the constraints to dissipate numerical errors, has been widely used due to its simplicity and intuitive approach [52]. Optimal Baumgarte parameters for particular integration solvers with fixed time step size were investigated in [53], [54].

Collision has been another issue in modeling legged dynamics, since collisional event happens whenever one of the legs hits the ground. It causes the system states undergo an abrupt change due to large impulsive forces during a short period of time. Two different approaches have been studied to solve this problem: algebraic formulations and differential formulations [55]. Differential formulations separate the impact period into compression and restitution period. They define relationships

between normal impulses at contact point of the two objects before/after the impact according to the Poisson's hypothesis, and integrate the system equations of motion over the short compression and restitution period to compute states after the impact. On the other hand, algebraic formulations deal with impact as an instantaneous event. They define kinematic relation between normal velocities of the two objects at contact point before/after impact, and solve for the generalized velocities after the impact using momentum equation. Hurmuzlu et al. [56] proposed a systemic procedure to solve multi-link rigid body collision problems using both formulations, especially for the case when impact between a system and the other occurs at one point while they are already having contact with each other at multiple points. Advanced algorithms for the algebraic collision law were invented afterwards [55], [57], but these are not deeply studied in this thesis.

Dynamic simulations for bipeds and quadrupeds have been built based on the techniques above. Often, they are accompanied with constrained optimization to find a periodic solution while avoiding slip on the ground. Optimal controller exploiting hybrid-zero dynamics was studied with bipedal robot simulator in [58]. Direct collocation technique for legged locomotion to find optimal trajectories of the robot and optimal ground reaction forces were studied with bipedal and quadrupedal robot simulator in [59], [2], using MATLAB.

1.3 Our approaches and accomplishments

Approaches

Although many robots already proved their ability for stable walking, only a few robots show both stable and agile performance. To enable quadrupedal robots to achieve high-speed locomotion, our controller is designed to solve three challenges in quadrupedal locomotion controller: 1) stabilization of the robot, 2) control of ground reaction forces, and 3) modulation of gait pattern.

The control algorithm is developed based on three strategies to deal with each of

the three issues. Programmable leg compliance is achieved in order to exploit *self-stabilizing* property of passive dynamics. We modulate the ground reaction forces using *equilibrium-point hypothesis* by designing trajectories for four legs. A gait pattern modulator is developed which has proprioceptive touchdown detection as a sensory feedback. These three strategies are integrated in the hierarchical structure of the controller.

First, we hypothesized that a quadrupedal robot can self-stabilize without active posture stabilization using high bandwidth attitude feedback such as body pitch and pitch rate. The previous studies on the SLIP model proved the existence of the state space region where the self-stability in the sense of limit-cycle is passively achieved [9], [60]. It was also shown that this stable region could be enlarged with additional active actuation [24], and this approach was implemented on the RHex robot [61]. The self-stabilizing property have also been proposed for quadrupedal legged locomotion [62], [63], [16]. Inspired by the SLIP model, previous researches installed mechanical springs on the legged machines to exploit the self-stability [64], [31]. For the MIT Cheetah robot, we achieve the programmable virtual leg compliance by the *impedance control* [65] using our proprioceptive actuation system [66].

Second, we exploit *equilibrium-point hypothesis* to control the ground reaction forces. The leg compliance created by the impedance control enables modulation of the ground reaction force by adjusting its equilibrium position. This interaction force control mechanism was observed in animals [67] and successfully demonstrated using a planar manipulator [68]. Aligned with this hypothesis, we propose to design desired leg trajectories for stance phase to penetrate into the ground. The penetration depth of the trajectory is controllable to modulate the ground reaction force. The larger impulse is required, the deeper the trajectory is designed to be. Besides, this hypothesis provides an unified approach to control both motion and interaction force, thus no switching between stance/swing control for each leg is required.

Lastly, we propose a gait pattern modulator which coordinates four legs per stride, combined with a leg touch down event detection feedback. The *stride-to-stride* modulation is proposed to coordinate the robot body dynamics and the rhythmic motion

of the four legs. The touch down event of the *reference leg* triggers the pattern modulator to generate phase signals for four legs. The phases of the other three legs are coupled with that of the *reference leg* to accomplish the desired gait pattern. The touch down event is detected without any additional sensor on the foot. We call this *proprioceptive* touchdown detection, which is allowed by the leg actuation paradigm proposed in [66]. In the previous controllers using CPG, CPG signals are directly transformed to joint-level trajectories for each joint position controller [27], [30], [31]. This is distinguished from our approach of which stance phase trajectories are designed to be reference trajectory for compliance control.

Accomplishments

The controller was verified in simulation experiments with properly chosen control parameters. The running robot using trot gait reached the constant speed of 3.4 m/s in the simulation, without any global posture feedback. With a simple accelerating algorithm proposed for trot gait in this thesis, it also achieved acceleration, without explicitly changing any control parameter but the target speed. Furthermore, the gait transition algorithm proposed in this paper was accomplished in the simulation. To achieve stable strides after transition from trot to gallop, we had to design different reference trajectories for gallop from trot. This implies the necessity of a further research on gallop.

The controller was then verified on the MIT Cheetah robot. The robot equipped with the controller showed a remarkable performance in terms of both speed and energy efficiency. The Froude number (Fr) is one of indices to describe dynamic locomotion which was studied by Alexander [3]. He examined that quadrupeds change their gait from walk to run at Fr of 1, and transit their gait to asymmetric running at Fr of 2 or 3. While only a few quadrupedal robots have reached Fr of 1, the MIT Cheetah accomplished stable trot running up to the speed of 6 m/s, which corresponds to Fr of 7.34, outperforming the other quadrupedal robots according to [31]. Also, it achieved highly efficient trot gait rivaling real cheetah, though this result is not addressed in details in this thesis.

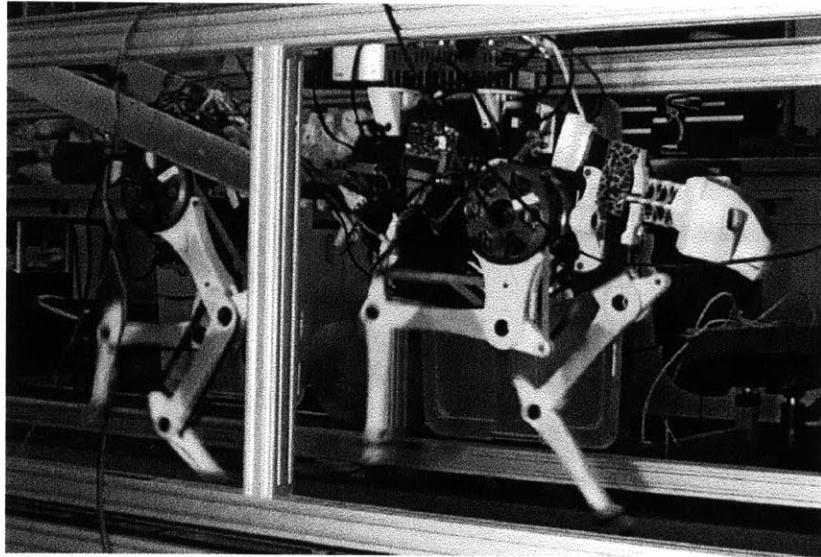


Figure 1-1: MIT Cheetah robot in the experiment, running with trot gait pattern on the treadmill while constrained on its sagittal plane.

1.4 Structure of this thesis

This introduction is followed in Chapter 2 by mathematics and algorithms in detail to construct the simulator. Hybrid dynamic system in legged locomotion is explained. MATLAB toolboxes to construct the simulator are briefly introduced as well.

The hierarchical controller is introduced in Chapter 3. Chapter 3 presents the structure of the controller and the set of control variables to parameterize quadrupedal locomotion. The underlying ideas and considerations for each level of controller are explained in detail.

Chapter 4 presents notable simulation results which convinced the authors in validity of the controller.

The controller is implemented on the MIT Cheetah robot, and the experimental setup and important raw data of the experiment are exhibited in Chapter 5. The data are post-processed, analyzed and discussed in Chapter 6 with description of remarkable performances of the MIT Cheetah robot.

The thesis is concluded in Chapter 7, where summary of the work and future research directions are presented.

Chapter 2

Modeling of the MIT Cheetah robot

A simulator is constructed using MATLAB in order to verify the controller before implement it on the MIT Cheetah robot. A planar model of the MIT Cheetah robot running on the flat, rigid ground is derived in the simulator.

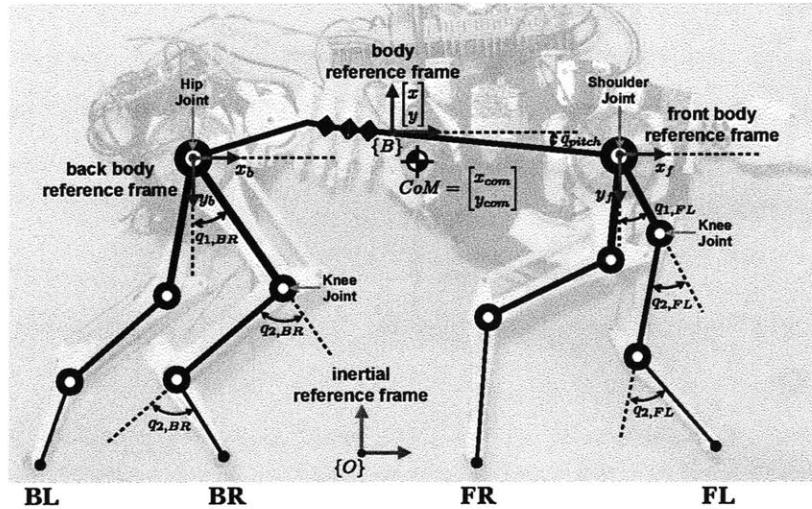


Figure 2-1: The planar-rigid model of the MIT Cheetah robot. A planar model is constructed since the robot is constrained in its sagittal plane for the current experimental setup. Leg indices are also listed. FR: front and right side. FL: front and left side. BR: back and right side. BL: back and left side.

The robot and its planar model is depicted in Fig.2-1 with generalized coordinates

and leg index $i \in \{FR, FL, BR, BL\}$. The proximal segment and the distal segment of each leg are parallel according to the pantographic leg design so that configuration of three-segmented-leg can be determined by two joint coordinates. Three independent coordinates (x, y, q_{pitch}) define the body coordinate with respect to the inertial frame. The flexible joints of the spine are designed to be dependent on hip angles of hind legs ($q_{1,BR}, q_{1,BL}$) through differential gears. Therefore, any additional coordinate for describing the motion of the spine is not necessary. The robot model has 11 degrees of freedom (DoF). Inertial and geometrical parameters are listed in Table A.1 in APPENDIX.A. The kinematic relationship between each joint of the segmented spine and hip joints of the hind legs is approximated in APPENDIX.B.

The distinct property of legged locomotion compared to wheeled locomotion is that it involves ‘Touch Down (TD)’ and ‘Lift Off (LO)’ events of all the legs. Before/after these boundary events, system dynamics changes and the recurrence of different dynamics on the system is called hybrid dynamics. Note that a floating-based model is derived in the simulator with appropriate constraint equations to describe all different dynamics.

MIT Cheetah robot is modeled using Lagrangian formulation with following additional assumptions.

- Each segment of the robot is perfectly rigid.
- The legs of the robot interact with the ground as point feet.
- The ground is perfectly rigid half-space.
- The interaction between each foot and the ground follows the Coulomb friction model.
- Impact between the foot and the ground is perfectly inelastic.
- Each joint is frictionless.

2.1 Hybrid dynamic systems

2.1.1 Hybrid dynamic systems in legged systems

Hybrid dynamic system exhibits both continuous and discrete dynamics. The system consists of multiple continuous dynamic systems, and when a certain condition is satisfied the system states jump from one to the other continuous dynamic system[58]. The discrete transition of the hybrid system can be modeled as a Finite state machine (FSM) as drawn in FIG.2-2. [69].

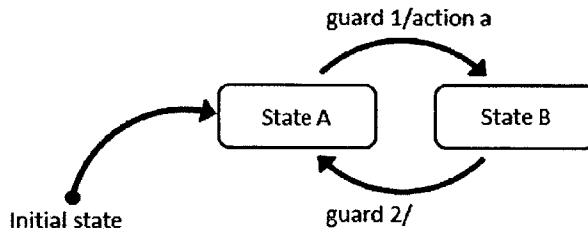


Figure 2-2: Typical schematic diagram for finite state machine

If ‘guard 1’ is satisfied, transition from state A to state B happens, and action a is executed. If ‘guard 2’ is satisfied in dynamics in state B, transition from state B to state A happens without executing any action.

Discrete footfall pattern of legged locomotion inevitably involves hybrid dynamics [2]. One should notice that the dynamics of the machine interacting with the ground (stance phase dynamics) is different from that of the machine in the air (flight phase dynamics). The transition happens whenever any leg of the machine *touches down* or *lifts off* the ground. In particular, when a leg touches down the ground, not only the dynamics changes from the moment but also instantaneous change in system states will occur due to impact.

As described in 2-3, flight phase, stance phase, leg *touch down*, leg *lift off* and *instantaneous jump in states due to impact map* correspond to state A, state B, ϕ_1 , ϕ_2 and action a , respectively. Typically, touch down event is detected with foot height and lift off event is detected with normal component of ground reaction force at the leg [70]. Mathematical description of the guard functions are as follows, for a leg

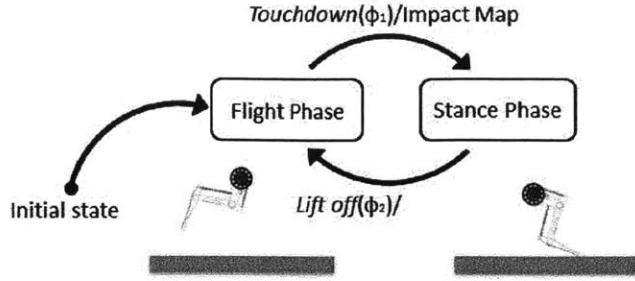


Figure 2-3: Finite state machine to model dynamics of legged mechanism

which is labeled as i :

$$\phi_1 : h_i(x) = 0 \wedge \dot{h}_i(x) < 0 \quad (2.1)$$

$$\phi_2 : F_{ext,i}^n(x) = 0 \wedge \dot{F}_{ext,i}^n(x) < 0, \quad (2.2)$$

where x is the state of the dynamic system, $h_i(x)$ is height of leg i and $F_{ext,i}^n(x)$ is the normal component of ground reaction force of leg i .

2.1.2 hybrid dynamic systems realized in MATLAB

To simulate dynamics in MATLAB one should define equations of motion (EoM) of the system. We choose to derive EoMs of the MIT Cheetah robot by floating-based model with appropriate constraint equations in MATLAB.

If the number of different dynamic systems is not large, for example hopping machine, it is desirable to derive EoMs for each state because it gives accurate dynamics satisfying the constraints. However, as the MIT Cheetah robot has four legs, it becomes much complicated and inefficient to derive equations of motion for all 256 dynamic states, combination of each leg condition:

case 1. not in contact with the ground

case 2. in contact with the ground and fixed to the contact position

case 3. in contact with the ground and slipping in one direction

case 4. in contact with the ground and slipping in the other direction.

In this case it is more beneficial to derive one set of floating-body-based equations of motion and impose appropriate constraints according to each dynamic states. Furthermore, floating-based model can directly computes constraint forces such as ground reaction forces, which is required to detect ‘lift-off’ event. Therefore, it is common to choose floating based model in the legged robot dynamic simulation as in [58] or [2] and the author also follows the convention. However, it naturally induces accumulation of numerical error on constraints and therefore an effort to inhibit this drift is additionally required.

To realize the hybrid dynamics in the simulator one should evaluate the guard functions and provide proper action in transition while integrating the dynamics over time. The author uses MATLAB `ode45` solver with `odeevent` option. The event function looks for either foot height for aerial phase legs or normal component of ground reaction forces for stance phase legs. When one of these values crosses zero from positive to negative, the ode solver stops integration, the set of constraint equations is changed, and `ode45` restarts integrating with updated initial conditions, if necessary.

Note that, using the standard `ode` toolbox of MATLAB has advantages and disadvantages. Because it is a “black box” with *varying step size*, it turned out that first, it is difficult to choose proper parameters for numerical error inhibition, second, it is hard to have access and print dependent variables which are not defined as state of the dynamics, such as ground reaction forces or input efforts, and third, sometimes the event is not detected precisely without a clear reason. Nonetheless, MATLAB’s `ode` solvers has varying step size, which enables them to detect events more accurately than fixed time step solvers. The powerful aspect of this accurate event detection is desirable, thus we build the simulator on MATLAB `ode45`.

2.2 Constrained equations of motion

2.2.1 Multi-link rigid body equations of motion

Dynamics of the 11-DoF MIT Cheetah is governed by

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J_c(q)^T F_{ext} \quad (2.3)$$

where $q := [q_{pitch}, q_{1,FR}, q_{2,FR}, \dots, q_{1,BL}, q_{2,BL}, x, y]^T \in \Re^{11}$ is the generalized coordinates of the model. The front body is chosen as *reference floating body* with three coordinates $[q_{pitch}, x, y]^T$ to describe its posture and position. $D(q)$, $C(q, \dot{q})\dot{q}$, $G(q)$ and B are the inertial matrix, Coriolis and centrifugal terms, gravitational torque vector, and the input matrix, respectively. The matrices are calculated through Lagrange formulation,

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} - \frac{\partial L(q, \dot{q})}{\partial q_i} = \Xi_i, \quad (2.4)$$

where $L(q, \dot{q}) = T(q, \dot{q}) - V(q)$ is Lagrangian of the system which is difference between kinetic energy (T) and potential energy (V), and Ξ_i is generalized forces. $u \in \Re^8$ is leg joint actuation torques which are transformed from geared BLDC motor outputs. $J_c(q) = \frac{\partial p_c(q)}{\partial q}$ is a Jacobian matrix of $p_c(q) \in \Re^{2N_c \times 1}$, a stack of position vectors of each ground-contact foot with respect to the inertial frame. N_c is the number of ground-contact feet. $J_c(q)^T F_{ext}$ is the ground reaction forces(GRFs) transformed into the joint space, which is obtained by principle of virtual work.

To solve for \ddot{q} and GRFs, holonomic equality constraints on positions of the ground-contact legs have to be considered:

$$\Phi_h(q) = 0_{h \times 1}, \quad \dot{\Phi}_h(q, \dot{q}) = 0_{h \times 1}, \quad (2.5)$$

where h is the number of constraints. Note that, since the holonomic constraint Φ is

function of q only,

$$\dot{\Phi}_h(q, \dot{q}) = J_h \dot{q}, \quad \ddot{\Phi}_h(q, \dot{q}, \ddot{q}) = J_h \dot{q} + J_h \ddot{q}, \quad (2.6)$$

where $J_h(q) = \frac{\partial \Phi_h}{\partial q}$. The partial derivative terms with respect to time, $\frac{\partial \Phi_h}{\partial t}$, are ignored, since constraints are not explicitly dependent on time in this thesis. $\Phi_h = 0$, $\dot{\Phi}_h = 0$ and $\ddot{\Phi}_h = 0$ are named *legal position*, *legal velocity*, and *legal acceleration* and the state satisfying these constraints is named *legal state* in [45]. Equality constraints for each leg are applied from the moment when the leg makes contact with the flat ground.

2.2.2 Numerical error inhibition

If the initial condition of the dynamics satisfies Eqn.(2.5), and the dynamics is accurately integrated, the states will remain in the manifold which suffices the constraint equations, if we solve Eqn.(2.3) with $\ddot{\Phi} = 0$. In practice, however, feedback terms are required to bring the states back to the desired manifold from the numerical error, or, drift, which is accumulated during integration or initiated by impact map [45].

Various methods have been extensively studied to inhibit drift on the constraint either in state-space formulation or projection methods[47]. Particularly, the Baumgarte's stabilization method [52] is an intuitive and therefore widely studied method including this work. It creates a virtual damper and a virtual spring around the *legal state* so that even though small error occurs, the state of the equations converges to $\Phi = 0$ and $\dot{\Phi} = 0$. Baumgarte's stabilization method is formulated as

$$\ddot{\Phi}(q, \dot{q}) + \alpha \dot{\Phi}(q, \dot{q}) + \beta \Phi(q) = 0, \quad (2.7)$$

where α and β are called Baumgarte parameters. Ideally, any set of Baumgarte parameters will eventually make the system states converge to *legal state*. In practice, however, Baumgarte parameters should be carefully chosen since the stability of the discrete numerical integration is affected by choice of the parameters. Choosing proper

Baumgarte parameters have been studied for a particular type of numerical integrator with fixed time step [53], [54]. However, a rigorous parametric study on varying time step does not exist.

In our case, it was observed that Baumgarte's parameter values do not affect that much on the stability of the system. It is because constraints of a running legged system is frequently changing and therefore duration for any error to accumulate is very short. Therefore, we arbitrarily choose parameters to create an critically-damped 2nd order dynamics to inhibit oscillation which may induce inaccurate event detection and to employ the fastest convergence rate of the critically damped system: $\alpha = 2\zeta\omega_n$, $\beta = \omega_n^2$ while $\zeta = 1$, $\omega_n = 15\text{rad/s}$.

As a result, two bundle of equations Eqns.(2.3), (2.7) are solved for \ddot{q} and F_{ext} simultaneously with the Coulomb friction model and the impact map.

2.3 Friction model

Designing a controller which is able to prevent a robot leg from slipping is a very challenging task, even many studies have built optimal controllers which satisfy the no-slip assumption [58], [59]. The task itself is difficult even in the simulation, and moreover, because of inaccurate ground contact model, even if the controllers are implemented on a real robot, they cannot guarantee the robot will not fall off at all. Therefore, in the author's opinion, a simulator should allow a robot to slip with appropriate friction model, to provide more realistic environment to assess performance of a controller designed for a robot.

We decide to use the Coulomb friction model which is widely used in legged locomotion community, as [2], [38], and [58]. The Coulomb friction model defines a relation between tangential and normal component of the GRF, F_t and F_n respectively, namely friction cone: $|F_t| \leq \mu|F_n|$. Here μ is the friction coefficient between a foot and the ground. We set the same value for both static and kinetic friction coefficient, $\mu_s = \mu_k = 1$. Computed GRFs, F_{ext} , from Eqns (2.3), (2.7) should obey friction cone, and are highly related to the construction of equality constraints, Eqn.(2.5). Es-

pecially, if a foot slips, the magnitude of tangential force is determined as $|F_t| = \mu|F_n|$ and the number of unknown variables decreases.

Consequently, the construction of Jacobian J_c at each time step highly affects construction of a set of equations and solution of the problem. We classify a ‘non-slip ground-contact foot (NS)’ and a ‘slip ground-contact foot (S)’ and to distinguish legs having different states, *index vectors*, are created: Idx^{NS} is a 4×1 vector for non-slipping legs and Idx^S is a 4×1 vector for slipping legs. Sum of these two vectors yields Idx^{TD} , a 4×1 vector for ground-contact legs. Each boolean element of these vectors represents FR, FL, BR, and BL, and if the value of the element is 1 then corresponding leg is a element of the set. For example, if $Idx^{NS} = [0, 0, 1, 1]^T$ and $Idx^S = [1, 0, 1, 1]^T$, then FR leg is slipping, BR and BL legs are fixed to the ground, and FL leg is not contacting with the ground. Though these discrete states are not explained in detail in the following algorithm, they play an important role to write code in MATLAB.

In the simulator developed in this work, the frictional multi-contact problem is solved in following seven steps.

Step 1. Solve with non-slip assumption and check friction cone condition

As Baraff mentioned in [71], a good guess to begin iteration with is to assume each contact is maintained during integration . Additionally, we can assume each contact is maintaining non-slip contact, since this is most likely happen during legged locomotion with a well designed controller, and this assumption reduces computational burden. In that case, unknown variables \ddot{q} and F_{ext} are simply solved for. Let $H = C\dot{q} + G - Bu$ and $R = \ddot{\Phi}_h - J_h\dot{q} = -\alpha\dot{\Phi}_h - \beta\Phi_h - J_h\dot{q}$. Then, Eqns. (2.3), (2.7) can be rewritten as,

$$D\ddot{q} + H = J_c^T F_{ext} \quad (2.8)$$

$$J_h\ddot{q} = R \quad (2.9)$$

and then solutions are

$$\ddot{q} = D^{-1}(-H + J_c^T F_{ext}) \quad (2.10)$$

$$F_{ext} = (J_h D^{-1} J_c^T)^{-1} (R + J_h D^{-1} H). \quad (2.11)$$

Calculate ratio of tangential component to normal component of the ground reaction force for each leg i and check whether the ratio is not larger than friction coefficient, $\rho_i = \frac{|F_{t,i}|}{|F_{n,i}|} \leq \mu$. If no force vector disobeys friction cone, the iteration is terminated.

Step 2. Sort out a sliding leg for the next iteration

The leg of which ρ_i is maximum, among those whose values are larger than friction coefficient, is sorted as a leg sliding on the surface for the next iteration. Update slip index vector Idx^S and non-slip index vector Idx^{NS} for the next iteration. Here, the direction of slip is determined to be opposite of the tangential force computed in the previous iteration.

Step 3. Reconstruct new constraints set and corresponding Jacobians

The holonomic equality constraints are applied to the position of each ground-contact foot. Non-slip ground-contact foot position should be fixed during integration, whereas the constraint in tangential direction for slip ground-contact foot should be vanished to allow it to accelerate along the tangential direction.

$$\Phi_h(q) = \begin{bmatrix} p_c^{NS}(q) \\ p_{c,n}^S(q) \end{bmatrix} - \begin{bmatrix} p_c^{NS}(q_0) \\ p_{c,n}^S(q_0) \end{bmatrix} = \begin{bmatrix} 0_{2N_{NS} \times 1} \\ 0_{N_S \times 1} \end{bmatrix} \quad (2.12)$$

$$\dot{\Phi}_h(q) = \frac{\partial}{\partial q} \begin{bmatrix} p_c^{NS}(q) \\ p_{c,n}^S(q) \end{bmatrix} \dot{q} = \begin{bmatrix} J^{NS}(q) \\ J_n^S(q) \end{bmatrix} \dot{q} = \begin{bmatrix} 0_{2N_{NS} \times 1} \\ 0_{N_S \times 1} \end{bmatrix}, \quad (2.13)$$

where $p_c^{NS}(q)$ is a set of position vectors of non-slip contact feet, $p_{c,n}^S(q)$ is a set of normal component of position vectors of slip contact feet, and q_0 is the generalized coordinates at ‘TD’ event. N_{NS} is the number of non-slip ground-contact foot, and N_S is the number of slip ground-contact foot. Since the

constraint sets are modified, the Jacobian of holonomic constraint equations in Eqn.(2.9) also should be modified.

$$J_h = \begin{bmatrix} J^{NS} \\ J_n^S \end{bmatrix} \quad (2.14)$$

Step 4. Reconstruct necessary Jacobians for EoMs

For tangential component of GRF at the foot which slips, its direction is pre-defined from the previous iteration and its magnitude is function of normal component of the GRF ($|F_t^S| = \mu|F_n^S|$). Therefore, F_t is no longer an independent variable and the number of unknown variables should be reduced from \ddot{q} and $F_{ext} = [F_t^{NS}, F_n^{NS}, F_t^S, F_n^S]^T$ to \ddot{q} and $F_{ext} = [F_t^{NS}, F_n^{NS}, F_n^S]^T$. Consequently, Eqn.(2.8) should be reorganized as

$$\begin{aligned} D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) &= Bu + J_i^{NS}F_i^{NS} + (J_{n,i}^S \pm \mu J_{t,i}^S)F_{n,i}^S + \dots \\ &= Bu + \begin{bmatrix} J^{NS} \\ J_n^S + M J_t^S \end{bmatrix} \begin{bmatrix} F^{NS} \\ F_n^S \end{bmatrix} \\ &= Bu + J_c F_{ext}, \end{aligned} \quad (2.15)$$

where $M = \begin{pmatrix} \pm\mu & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \pm\mu \end{pmatrix}$ contains information of the direction of tangential force at corresponding leg.

Step 6. Solve for constraint forces and accelerations

With the updated J_c and J_h from Eqns.(2.14), (2.15), solutions \ddot{q} and $F_{ext} = [F_t^{NS}, F_n^{NS}, F_n^S]^T$ can be attained directly from Eqns.(2.10), (2.11). F_t^S is calculated as $F_t^S = \pm\mu F_n^S$.

Step 7. Terminate Iteration if all tangential forces are bounded.

Iteration from Step 2. through Step 6. is terminated when the ratios ρ_i are all bounded by μ .

Lastly, the outputs of the iteration are observed by the solver at each time step. Once any non-contact foot position becomes penetrating the ground, or any normal component of ground reaction force of contact foot becomes negative, the integrator is terminated and re-initiated with updated Idx^{TD} .

2.4 Impact map

When system dynamics make a transition from one to another with ‘TD’ event(s), the system states undergo an abrupt change due to large impulsive forces during a short period time of impact. Two different approaches have been studied to resolve this collision problem: algebraic formulations and differential formulations [56]. Differential formulations separate collision into compression and restitution period, define relation between normal impulses of two object at contact point before/after impact with Poisson’s hypothesis, and integrate the system EoM over the short duration to compute states after the impact. On the other hand, algebraic formulations deal with impact as an instantaneous event, and define kinematic relation between two objects using normal velocities at contact point before/after impact, and solve for generalized velocities after impact using momentum equation.

In this work, we use algebraic formulation to update the state of the system at impact. Mason and Wang [72] observed contradiction in using the algebraic formulation, such that kinetic energy of the system increased after the impact for certain cases. However, if perfect plastic collision was assumed, such problem did not occur. The perfect plastic impact is normally presumed in dynamic simulation of legged robots [58], thus we use algebraic formulation which is more stable, faster, and easier to implement than differential formulation.

Compliant ground model can be employed in order to avoid difficulties in rigid ground models such as inconsistency in impact law. It also provides continuity for gradient-based optimization, as well as simulation environment for both soft and rigid grounds [73]. Nonetheless, it requires additional states which increases computational cost and induces stiff problem due to high stiffness and damping of the ground model

[59]. Besides, we want to evaluate the performance of the controller under the existence of the discrete jump in the states, which occasionally happens in the real system.

Generalized positions are invariable before/after impact ($q^+ = q^-$). The generalized velocities after impact, \dot{q}^+ , can be solved via the algebraic impact law, Eqn.(2.17), with appropriate equality constraints, Eqn.(2.18).

$$q^+ = q^- \quad (2.16)$$

$$D(q)\dot{q}^+ - D(q)\dot{q}^- = J_c(q)^T \int_{t^-}^{t^+} F_{ext} dt \quad (2.17)$$

$$\begin{aligned} \dot{\Phi}_h(q) &= J_h \dot{q}^+ \\ &= 0_{h \times 1} \end{aligned} \quad (2.18)$$

The momentum equation is constructed by integrating the governing EoMs over a short instant ($\Delta t = t^+ - t^-$) before and after impact. Coriolis and centrifugal terms, gravitational terms and input terms are negligible compared to impulsive forces, thus ignored in the equation [58].

Multi-link collision problem for the legged locomotion is a complicate problem to solve. When collision occurs at a leg, some of the other legs are in unconstrained contact with the ground, and whether these legs rebound or slip after impact will strongly affect the result. Hence a systematic procedure is required to solve the problem, and we slightly modified that of [56].

Step 1. Similar to the multi-link frictional contact problem, we start the iteration with the simplest conditions. Solve with no-slip, no-bound assumptions for each ground-contact leg after impact, and check whether conditions are satisfactory with the assumptions. Let us define $I_{mp} = \int_{t^-}^{t^+} F_{ext} dt$, impulse. Then, solutions for Eqns. (2.17), (2.18) are

$$\dot{q}^+ = \dot{q}^- + D^{-1} J_c^T I_{mp} \quad (2.19)$$

$$I_{mp} = (J_h D^{-1} J_c^T)^{-1} J_h \dot{q}^- \quad (2.20)$$

In general, a leg having contact with the ground before impact can have either one of four different states after impact, and solution with each state should satisfy a certain condition. Possible leg states after impact and corresponding necessary conditions are listed in 2.1. If the solutions from Eqn. (2.19), Eqn.(2.20) satisfy all the conditions, terminate the iteration. Otherwise, proceed to the next step.

State at t^+ , after impact	Conditions to satisfy
Detaches without interaction : $I_{mp} = 0$	$v_n^+ > 0$
Sticks to the ground : $v_n^+ = v_t^+ = 0$	$I_{mp,n} > 0$ and $\frac{ I_{mp,t} }{ I_{mp,n} } \leq \mu$
Slides on the ground in the positive tangential direction : $v_n^+ = 0$ and $\frac{I_{mp,t}}{I_{mp,n}} = \mu$	$I_{mp,n} > 0$ and $\text{sign}(I_{mp,t}) \neq \text{sign}(v_t^+)$
Slides on the ground in the negative tangential direction : $v_n^+ = 0$ and $\frac{I_{mp,t}}{I_{mp,n}} = -\mu$	$I_{mp,n} > 0$ and $\text{sign}(I_{mp,t}) \neq \text{sign}(v_t^+)$

Table 2.1: Possible states of a leg after impact and corresponding necessary conditions

Step 2. Update Idx^{FS} , and Idx^{BS} .

Obtain set of combinations of leg states by reorganizing state vectors, Idx^{BS} and Idx^{FS} , those which slide in the positive and negative tangential direction, respectively. Construct M , J_h and J_c accordingly as in Eqn.(2.14), (2.15). If the solutions from Eqn. (2.19), (2.20) satisfy all the conditions, terminate the iteration. Otherwise, proceed to the next step.

Step 3. Update Idx^{NS} , and Idx^S .

Obtain set of combinations of leg states by reorganizing state vectors, Idx^{NS} and Idx^S , those which stick to the ground and slide on the ground, respectively. Construct M , J_h and J_c accordingly as in Eqn.(2.14), (2.15). If the solutions from Eqn. (2.19), (2.20) satisfy all the conditions, terminate the iteration. Otherwise, proceed to the next step.

Step 4. Update Idx^{TD} .

Obtain set of combinations of leg states by reorganizing state vectors, Idx^{TD} , those which maintain contact after impact. Some of leg contacting with the ground may rebound after impact. Construct \mathbf{M} , J_h and J_c accordingly as in Eqn.(2.14), (2.15). If the solutions from Eqn. (2.19), (2.20) satisfy all the conditions, terminate the iteration. Otherwise, proceed to the next step.

Note that in the numerical computation, the conditions in 2.1 should have threshold values, for example, $v_n^t > 1e - 10$. The problem often issued about this algebraic approach is that sometimes increase in kinetic energy after is observed. However, with the perfect plastic assumption in this simulation, the problem has never been observed.

Chapter 3

Quadrupedal Locomotive Control Framework

This chapter presents a new locomotive control framework for quadrupedal robots developed in this thesis. The control algorithm is developed under guidance and lots of discussion with Dr. Dong Jin Hyun. We address three challenges in designing quadrupedal locomotion controller: 1) stabilization of the robot, 2) control of ground reaction forces, and 3) modulation of gait pattern.

The control algorithm is developed based on three strategies to deal with each of the three issues. Programmable leg compliance is achieved in order to exploit *self-stabilizing* property. We modulate the ground reaction forces using *equilibrium-point hypothesis* by designing trajectories for four legs. A gait pattern modulator is developed which has proprioceptive touchdown detection as a sensory feedback. These three strategies are integrated in the hierarchical structure of the controller. The controller is governed by a set of control parameters, such as target speed, target gait pattern, and the spatial properties of the leg trajectories.

The overall structure of the proposed controller is schematically represented in Fig.3-1. An operator specifies 1) a desired velocity v_d and 2) a target gait pattern expressed by $\Delta\vec{S}$. The gait pattern modulator and the leg trajectory generator constitute the high-level controller which manages and plans the motion and role of four legs. Proprioceptive touch-down detection on a *reference leg* provides the stride initi-

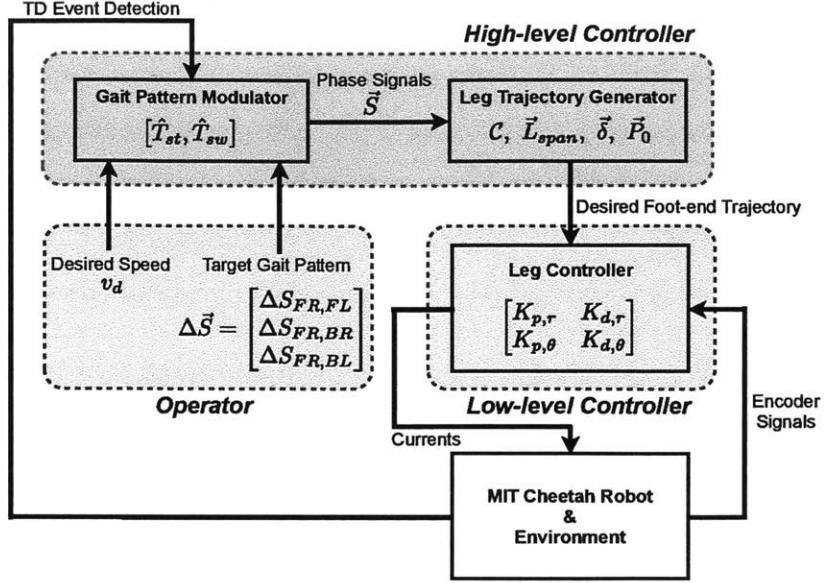


Figure 3-1: Schematic overall structure of the proposed hierarchical locomotive control framework. Each block represents each step and corresponding control parameters. In this schematic diagram, the *reference leg* is front right leg, FR.

ation timing. The phase signals generated by the gait pattern modulator are mapped to the desired trajectories of four legs. The low level individual leg controller controls each leg as planned. Table 3.1 shows the control variables to be managed in the proposed controller.

3.1 Gait pattern modulator with proprioceptive sensory feedback

The gait pattern modulator achieves a desired velocity and a target gait pattern by coordinating the robot's four limbs with the phase signals, \vec{S} . The phase signal assigned to each leg is a fraction of the desired swing and stance period, \hat{T}_{sw} and \hat{T}_{st} , respectively, which are determined in accordance with the commanded desired velocity.

Gait pattern is one of characteristics of quadrupedal locomotion. Animals tend to have desirable gait pattern according to their speed of locomotion. Energetic

Terminology	Definition
v_d	Desired speed
$\Delta S_{FR,i}$	the phase lag between <i>FR</i> leg and leg i
\hat{T}_{st}	Desired stance phase period
\hat{T}_{sw}	Desired swing phase period
$\mathcal{C} = [\mathcal{C}_F, \mathcal{C}_B]$	Bezier control points for swing phase trajectory
$\vec{L}_{span} = (L_{span,F}, L_{span,B})^T$	half of the stroke length
$\vec{\delta} = (\delta_F, \delta_B)^T$	penetration depth for stance phase trajectory
$\vec{P}_0 = (P_{0,F}, P_{0,B})^T$	reference point of the trajectory
$K_{p,r}$	radial stiffness of each leg
$K_{d,r}$	radial damping of each leg
$K_{p,\theta}$	angular stiffness of each leg
$K_{d,\theta}$	angular damping of each leg

Table 3.1: Nomenclature of the set of control parameters

advantages of choosing different gait patterns for different speeds have been observed and therefore it is believed that robots should also be capable of choosing different gait patterns. To impose a specific target gait pattern such as walk, trot, canter, and gallop, the gait pattern modulator assigns the phase lag between four legs, $\Delta\vec{S}$, so that the temporal limb coordination is achieved.

3.1.1 Phase signals

Generated phase signals are referred as parameters for designing stance/swing phase trajectories by the leg trajectory generator. Each signal $S_i^j \in [0, 1]$ is time-normalized by \hat{T}_{st} and \hat{T}_{sw} , where $i \in \{FR, FL, BR, BL\}$ is leg index and $j \in \{st, sw\}$ is leg state index(stance, swing).

Maes et al. [4] illuminated that the duration of leg swing remains steady over various speeds. Respecting the natural behavior of biological systems, the desired swing phase period, \hat{T}_{sw} , is prescribed as constant, 0.25 sec.

While maintaining the swing duration, animals decrease stride frequency and contact time as the locomotion speed increases [5], [7]. Therefore the desired stance phase period, \hat{T}_{st} , is determined according to the desired speed v_d as:

$$\hat{T}_{st} = \frac{2L_{span}}{v_d}, \quad (3.1)$$

where L_{span} is half of the stroke length as depicted in Fig.3-4. The stroke length approximates a distance traveled of the shoulder/hip joint during stance phase. This implies we only modulate stride frequency to accelerate, which is similar to [74].

Once the gait pattern modulator is triggered, it assigns stance phase S_i^{st} and then swing phase S_i^{sw} to each leg, which completes a whole stride.

3.1.2 Synchronization by proprioceptive sensory feedback

The phase signals generated by the gait pattern modulator have to be synchronized with the environment such that stance phase or swing phase is properly commanded to a leg when it touches down or lifts off the ground, respectively. Therefore, detection of ‘TD’ event and ‘LO’ event become crucial. In lots of legged robot researches, a force sensor is attached to the foot for the event detection, and furthermore, for ground reaction force feedback control and for stance/swing switching control.

The MIT Cheetah robot leg with low mechanical impedance enables the proprioceptive ‘TD’ detection without additional force sensor, by sensing abrupt change in the commanded force computed by the impedance control scheme. When a leg touches down the ground, radial force in outward direction is commanded by the impedance controller in order to impede the leg’s sudden flexion, which is induced by the ground impact. The value is comparatively higher than one commanded during the swing phase; therefore, the ‘TD’ event can be recognized in a way like a limit switch, by setting a threshold value on radial force, F_r .

Note that the leg may lose contact for a short instant during the stance phase due to uneven terrain, which leads to the controller’s mis-detection for ‘stance to swing transition’. To prevent this failure mode, we activate proprioceptive ‘TD’ detection after completion of 90% of the swing phase. 10% margin is given for the leg’s agile response to the early occurrence of the ‘TD’ event¹.

¹The touch down sensor thus acts similar to limit switch. This type of sensing might be vulnerable to chattering of leg but note that the chattering after 90% of the swing phase is mainly in the radially outward direction by the centrifugal force.

3.1.3 Imposing gait patterns

In order to impose a desired gait pattern, phase coupling between four legs is required. If all the leg phases are initiated by each ‘TD’ event, however, it is challenging to maintain the consistency of the specific gait pattern. A novel yet simple way is developed to solve this problem: *TD event-based stride to stride pattern modulation*. Detection of ‘TD’ event initiates generation of the phase signals of the reference leg, and the other three legs’ phases are coupled with that of the reference leg having predefined phase lag, $\Delta\vec{S}$. The proposed method can be versatilely applied to any symmetric/asymmetric gait pattern.

This ‘stride to stride’ method generates phase signals just for one stride and waits for the next trigger. Therefore, after a completion of one stride, all the legs become stationary until next ‘TD’ event on the reference leg is detected. This undesired stationary period is the result of minimal adaptation to the environment while maintaining a specific gait pattern, which is necessary for locomotion on rough terrain with unexpected external disturbances. Ideally, the undesired stationary period can be vanished if the pre-defined swing phase period, \hat{T}_{sw} , is exactly matching to the actual value, or feasibly estimated during stance phase in *priori*.

3.1.4 Phase signal generation according to TD elapsed time and phase differences

A mathematical formulation of the phase signal generation is presented here. The phase signals are created referring to *clocks* of each leg. The reference leg clock is the time elapsed after ‘TD’ event, and the other clocks have time delay with respect to the reference leg clock.

The proprioceptive ‘TD’ detection is represented as a boolean variable ‘TD’. It changes from FALSE to TRUE when the event is detected, and one stride cycles with the elapsed time after the event as following:

$$t_{ref}^{elapse} = \begin{cases} t - t_{ref}^{TD} \\ \hat{T}_{stride} & \text{if } t_{ref}^{elapse} > \hat{T}_{stride} \end{cases} \quad (3.2)$$

$$t_{ref}^{TD} = t \quad \text{if } (S_{ref}^{sw} > 0.9) \wedge (\text{'TD'}) \quad (3.3)$$

$$t_i = t_{ref}^{elapse} - \Delta S_{ref,i} \hat{T}_{stride}, \quad (3.4)$$

where t_{ref}^{elapse} is the elapsed time after t_{ref}^{TD} which is the moment of the TD event on the reference leg. t_i is the clock for each leg, where $i \in \{FR, FL, BR, BL\}$. $\Delta S_{ref,i}$ is the phase lag of the leg i with respect to the reference leg, represented as a fraction of the desired stride period. Note that $\Delta S_{ref,ref}$ is zero by definition. t_{ref}^{TD} is updated discretely when the next 'TD' event is detected, while 90% of swing phase of the reference leg (S_{ref}^{sw}) is completed.

S_i^{st} increases from zero to one during \hat{T}_{st} and subsequently S_i^{sw} repeats from zero to one during \hat{T}_{sw} . Each leg phase signal follows the reference time having time delay, or, phase lag with respect to the reference leg.

$$S_i^{st} = \frac{t_i}{\hat{T}_{st}} \quad 0 < t_i < \hat{T}_{st} \quad (3.5)$$

$$S_i^{sw} = \begin{cases} \frac{t_i + \hat{T}_{sw}}{\hat{T}_{sw}} & \text{if } -\hat{T}_{sw} < t_i < 0 \\ \frac{t_i - \hat{T}_{st}}{\hat{T}_{sw}} & \text{if } \hat{T}_{st} < t_i < \hat{T}_{stride} \end{cases} \quad (3.6)$$

Here we present phase signals generated by proposed phase modulator and corresponding foot fall pattern diagrams for trot and gallop gait pattern used in the experiment. A front right leg (FR) is chosen as the reference leg in the examples. To accomplish a specific gait pattern, appropriate phase lags are prescribed with respect to the FR leg. Trot gait is a symmetric gait pattern, of which phase for contralateral legs are the same.

$$\Delta \vec{S}_{\text{trot}} = \begin{bmatrix} \Delta S_{FR,FL} \\ \Delta S_{FR,BR} \\ \Delta S_{FR,FL} \end{bmatrix}_{\text{trot}} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix} \quad (3.7)$$

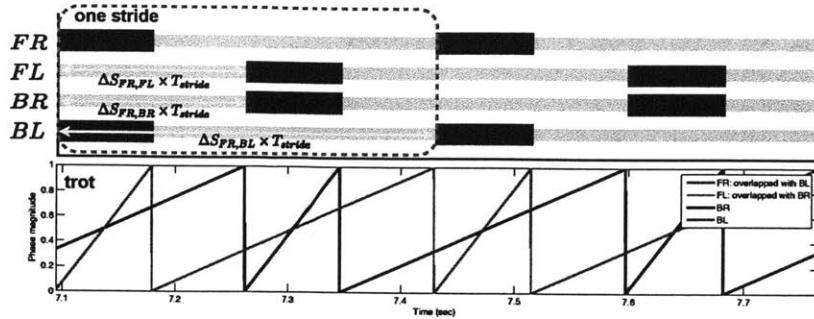


Figure 3-2: Sawtooth phase signals generated by phase modulator for trot gait and its corresponding foot fall pattern diagram. The phase difference is represented in the diagram.

The phase difference between each leg for a gallop, which is an asymmetric gait pattern, is not well defined but varies by literature. Here, following phase difference is chosen for gallop.

$$\vec{\Delta S}_{\text{gallop}} = \begin{bmatrix} \Delta S_{FR,FL} \\ \Delta S_{FR,BR} \\ \Delta S_{FR,BL} \end{bmatrix}_{\text{gallop}} = \begin{bmatrix} 0.2 \\ 0.55 \\ 0.75 \end{bmatrix} \quad (3.8)$$

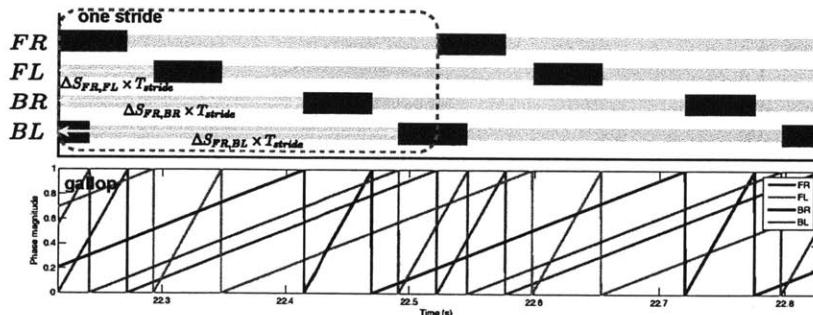


Figure 3-3: Sawtooth phase signals generated by phase modulator for gallop gait and its corresponding foot fall pattern diagram. The phase difference is represented in the diagram

3.2 Leg trajectory generator

The leg trajectory generator transforms the phase signals from the phase modulator to the desired trajectories for each foot-end which are designed with variable parameters. The leg trajectory generator plays a key role in our scheme; the robot is capable of managing its stability with adjustable trajectories while following a specific gait pattern commanded by the gait pattern modulator.

Swing phase and stance phase trajectories are designed individually in different perspectives while guaranteeing continuous and smooth transitions: pure position control and compliance force control, respectively.

Swing phase trajectories are designed by Bézier curve defined by a set of 12 control points, $\mathcal{C} = \{c_k\}$, and stance phase trajectories are spatially designed as sinusoidal wave of which period is twice of stroke length. Also, the trajectory is designed about the reference point, \vec{P}_0 with respect to the local shoulder/hip coordinate system. Fig.3-4 shows the designed reference gait trajectory.

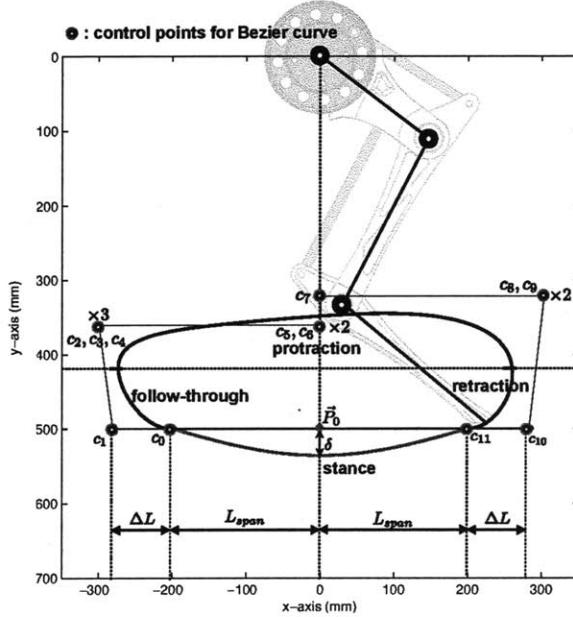


Figure 3-4: The reference desired foot end trajectory with control parameters. 12 control points of Bézier curve are shown which define the swing phase trajectory (black solid line). Multiple overlapped points are denoted as ‘ x^n ’. Stance phase trajectory as sinusoidal wave is also described in the figure (red solid line).

3.2.1 Design of the trajectory in the swing phase

The primal objective of designing the swing phase trajectory is to protract a leg with sufficient ground clearance to avoid obstacles and with sufficient leg retraction rate. Also, feasible swing phase trajectory, $p_i^{sw}(t)$, should require relatively low energy consumption for overall energy efficiency of the locomotion.

The three-segmented pantographic design of the MIT Cheetah leg enables protraction with low energy usage by exploiting its natural dynamics. Swing phase trajectory is designed by two dimensional Bézier curve constructed with 12 control points while approximating the protraction trajectory to around the natural behavior of the legs as well as satisfying geometrical requirements (ground clearance). It is parameterized by corresponding swing phase signal, $S_i^{sw} \in [0, 1]$ as:

$$p_i^{sw}(t) = p_i^{sw}(S_i^{sw}(t)) = \sum_{k=0}^n c_k B_k^n(S_i^{sw}(t)) \quad (3.9)$$

$$v_i(t)^{sw} = \frac{dp_i}{dS_i^{sw}} \frac{dS_i^{sw}}{dt} = \frac{1}{\hat{T}_{sw}} \frac{dp_i}{dS_i^{sw}}, \quad (3.10)$$

where $B_k^n(S_i^{sw}(t))$ is the Bernstein polynomial of degree n, $(n+1)$ is the number of control points, $c_k \in \mathbb{R}^2$ is a k-th control point and $k \in \{0, \dots, 11\}$.

Note that a r-th derivative of a Bézier curve is as below² and therefore it has the following useful properties:

$$p^{(r)}(s) = \frac{n!}{(n-r)!} \sum_{k=1}^{n-r} \Delta^{(r)} c_k B_k^{n-r}(s) \quad (3.11)$$

- $p_i^{sw}|_{S_i^{sw}=0} = c_0 \quad v_i^{sw}|_{S_i^{sw}=0} = n \Delta c_0 / \hat{T}_{sw}$
- $p_i^{sw}|_{S_i^{sw}=1} = c_n \quad v_i^{sw}|_{S_i^{sw}=1} = n \Delta c_n / \hat{T}_{sw}$
- Double-overlapped control points generate zero velocity.
- Triple-overlapped control points generate zero acceleration.

Horizontal (x) and vertical (y) positions of control points are designed indepen-

² Δ^r is r-th forward difference operator: $\Delta^r c_k = \Delta^{r-1} c_{k+1} - \Delta^{r-1} c_k$

dently based on these properties, in the parasagittal plane with respect to hip/shoulder joint. In vertical direction,

1. double overlapped control points, $c_{0,y}$ and $c_{1,y}$, are used for zero velocity at 'LO'.
2. For transition from 'follow-through' to 'protraction', the acceleration direction is changed by triple overlapped control points.
3. During 'protraction', double overlapped control points change the direction of trajectory.
4. For comfortable 'TD' with 'swing leg retraction' in the y-direction, the acceleration direction have to be changed by triple overlapped control points.
5. double overlapped control points, $c_{10,y}$ and $c_{11,y}$, are used for zero velocity at 'TD'.

The total required 12 control points are shown in Fig.3-5 as red circles.

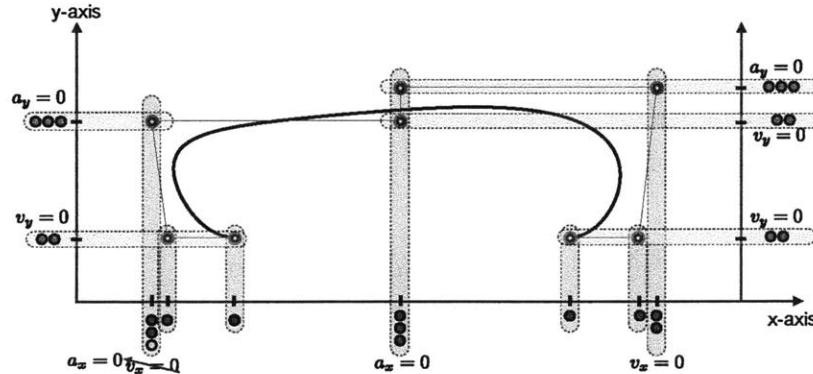


Figure 3-5: separation of 12 control points of Bézier curve for swing phase trajectory into vertical direction and horizontal direction

In horizontal direction, the leg initially has 'follow-through', changes the direction, has 'protraction', changes direction again and has 'swing-leg retraction' sequentially. Therefore,

1. The horizontal velocity at LO is defined as the difference between $c_{0,x}$ and $c_{1,x}$.

2. To change the direction after ‘follow-through’, double overlapped control points are positioned.
3. During protraction, acceleration direction is changed. Thus, triple overlapped control points are positioned.
4. Changing direction to have ‘swing-leg retraction’ requires double overlapped control points.
5. The velocity of ‘swing-leg retraction’ is defined by the difference in the x-direction between $c_{10,x}$ and $c_{11,x}$

Fig.3-5 shows 11 required control points as blue circles. In this thesis, 12 control points are used for the horizontal direction as well for ‘single instruction multiple data streams (SIMD)’, a fast and efficient computation algorithm, by coping with Bézier control points as 12×2 array.

The 12 control points for the designed gait trajectory are listed in Table. 3.2. Satisfying the requirements described, the swing phase leg-trajectory, of front right leg for example, is designed to create ground clearance of 0.15 m in the middle of ‘protraction’, swing leg retraction speed of 4.0 m/s with prescribed $\hat{T}_{sw} = 0.25$ ms, $L_{span} = 200$ mm, and $P_0 = (0 \text{ mm}, 500 \text{ mm})$ as Fig.3-4.

	x (mm)	y (mm)
c_0	-200.0	500.0
c_1	-280.5	500.0
c_2	-300.0	361.1
c_3	-300.0	361.1
c_4	-300.0	361.1
c_5	0.0	361.1
c_6	0.0	361.1
c_7	0.0	321.4
c_8	303.2	321.4
c_9	303.2	321.4
c_{10}	282.6	500.0
c_{11}	200.0	500.0

Table 3.2: 12 control points for Bézier curve for designing swing phase trajectory.

3.2.2 Design of the trajectory in the stance phase

The stance phase control of each leg directly affect the performance on quadruped locomotion via interaction with the ground. Therefore, the stance phase trajectory has to be designed in a different standpoint from a pure trajectory control as for swing phase. Rather, it should be understood as a reference trajectory to generate force with impedance controller as compliance force control scheme. A trajectory in horizontal direction is related to forward locomotion, and a trajectory in vertical direction is related to the vertical GRF applied to the robot, respectively.

The stance phase trajectory is proposed to simply be a sinusoidal wave with two parameters depicted in Fig.3-4: 1) half of the stroke length, L_{span} and 2) amplitude variable, δ . Based on the phase signals, S_i^{st} , the reference foot-end trajectory of each leg is generated during \hat{T}_{st} . Stance phase trajectories for each leg are parameterized by corresponding phase signal, $S_i^{st} \in [0, 1]$ as

$$p_{i,x}^{st}(t) = L_{span}(1 - 2S_i^{st}(t)) + P_{0,x} \quad (3.12)$$

$$p_{i,y}^{st}(t) = \delta \cos\left(\frac{\pi}{2L_{span}} p_{i,x}(t)\right) + P_{0,y} \quad (3.13)$$

$$v_{i,x}^{st}(t) = \frac{dp_{i,x}}{dS_i^{st}} \frac{dS_i^{st}}{dt} = -\frac{2L_{span}}{\hat{T}_{st}} \quad (3.14)$$

$$v_{i,y}^{st}(t) = \frac{dp_{i,y}}{dp_{i,x}} \frac{dp_{i,x}}{dt} = \frac{\delta\pi}{\hat{T}_{st}} \sin\left(\frac{\pi}{2L_{span}} p_{i,x}(t)\right) \quad (3.15)$$

The stance phase trajectory in horizontal direction is linearly designed under the assumptions that the foot does not slip and the relative velocity of the shoulder/hip to the foot-end is constant as the desired velocity during \hat{T}_{st} . L_{span} in stance phase means a half of the distance traveled of a hip/shoulder during each stance phase. To guarantee continuity at transition between stance and swing phases, L_{span} and P_0 are identically set to the values of the swing phase trajectory.

The trajectory in vertical direction is determined with δ which is the amplitude of the proposed sinusoidal wave. Again, the vertical trajectory should be interpreted in the compliance control scheme.

The robot body dynamics is resulting from net horizontal impulse and net verti-

cal impulse exerted by four legs. To achieve periodic motion of the center of mass of the body in vertical direction during running, the following vertical momentum conservation equation must be satisfied.

$$mgT_{stride} = \sum_{contact} \int_0^{T_{st}} F_{ext}^n(t) dt, \quad (3.16)$$

where m is the robot mass, T_{stride} and T_{st} is the actual stride period and actual stance phase period, and F_{ext} are external forces, i.e., GRFs. While maintaining the sum of vertical impulses, redistribution vertical impulses to each leg is necessary because the combination of net horizontal impulse and net vertical impulse induces pitch moment to the robot [75]. Therefore, stance phase trajectories of the front and the rear legs are designed separately to regulate pitch of the robot.

To directly control the robot body dynamics as described, full state sensory feedback is required to exactly compute the desired forces for legs at each moment considering the body acceleration. However, there is no doubt that measurement of the exact body states and proper distribution of the forces are challenging.

Nonlinear coupling between variables such as T_{st} , T_{stride} and F_{ext} in Eqn.(3.16) requires well defined model with minimal model uncertainties. Moreover, to compute and generate desired force vectors at each foot-end correctly, exact measurement of the robot orientation, or, reliable state estimation, which requires concurrent regulation for accumulated error, is necessary .

Due to these underlying difficulties, a stance phase control in this thesis is designed in local coordinate only with feasible sensory feedback such as kinematic data measured by joint encoders. As we mentioned briefly above, we create virtual compliance in four legs to employ possible advantage of intrinsic stability. This virtual compliance provides an effective way to solve the complex problem of contact force control, *equilibrium point hypothesis* [67]. The hypothesis proposes that a method animals might use to exert proper force is controlling the equilibrium point of the virtual compliant system of their limb to have a penetration depth into a contact sur-

face. Then, instantaneous difference between the actual position and the equilibrium position generate the requisite torques without complex inverse dynamics problem.

Trajectory tracking error in the stance phase is mainly induced by GRF due to the interaction with the ground. The impedance control realized by the leg controller generates forces at the foot-end in the reactive manner according to the displacement/velocity errors at the foot-end. The integral on the generated force by the contact legs during the stance period is the impulse as expressed in the right-hand-side of Eqn.(3.16). Especially at high speed, leg dynamics is comparatively faster than the body dynamics, and the shoulder/hip height y_e is almost stationary while the robot's leg is moving. When this assumption is effective, the shape of trajectory error in vertical direction follows that of the desired trajectory, so as the vertical force exerted to the ground according to the virtual impedance. If an implementable gains for the impedance control is predefined, the vertical impulse exerted on the ground during the stance period is possible to be adjusted through the design of the trajectory [76].

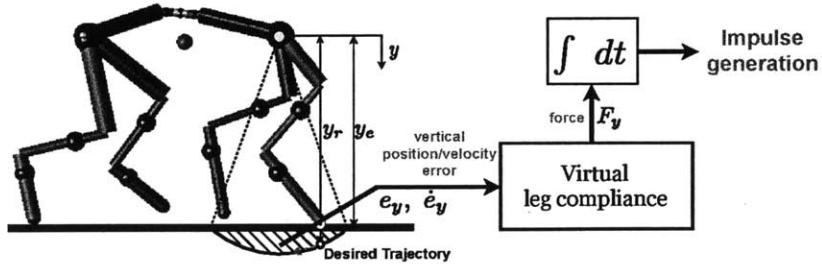


Figure 3-6: Stance trajectory design with equilibrium-point hypothesis: Along with vertical axis, the tracking error between the actual foot-end position and the designed foot-end trajectory is converted into the vertical force incorporated with the leg compliance. The time-integral of the force during the stance period is the vertical impulse.

Therefore, in order to generate the required impulse for running, the leg force exerted on the ground can be modulated through adjusting penetration depth of the virtual reference trajectory into the ground as shown in Fig.3-6 where y_r and y_e are the vertical component of a desired trajectory and the position of the ground with respect to the shoulder/hip joint, respectively.

Series of harmonic functions can create various shape of trajectories with a few parameters which can generate different impulses. In this thesis, only a sinusoidal stance phase trajectory is generated with the amplitude, δ , as a control variable for trot gait. We expect higher order series can be used for asymmetric gait pattern.

3.2.3 Scaling and translation of the designed trajectory

Once the reference trajectory is defined, additional 4 DoFs are permissible to the operator for adjusting the predesigned reference trajectory in order to adapt to the environment. The four parameters are translation of the reference point \vec{P}_0 in horizontal and vertical direction, scaling of the entire trajectory in horizontal direction, and penetration depth δ . The potential usage of $P_{0,x}$ is inspired by [77]. For trot, the value is set to zero, but further investigation is required for gallop where more aggressive acceleration and deceleration per step should be realized.

3.3 Low level individual leg controller

The impedance control is introduced to each leg controller to provide virtual compliance for intrinsic stability as well as to accomplish motion control in swing phase and interaction control in stance phase without solving computationally expensive inverse kinematics nor inverse dynamics. Since the impedance controller provides stable motion control and effective interaction control, we can avoid complex problem of switching between stance and swing control which involves exact event detection problem.

The agility of the mechanical and electrical leg system enables the low level leg controller to impose a virtual impedance despite of rapidly time-varying equilibrium point. The bandwidth of the mechanical parts of the robot's leg is measured to be 100 Hz with efforts to minimize mechanical impedance of the robot leg. For the legged robot, its legs inevitably has ground impact which generate wide frequency range of disturbance. In order to stabilize such a leg system and create virtual compliance, the fast overall closed loop is highly required and therefore a FPGA/RT based control

architecture is constructed to achieve 250 us loop time. Consequently, a virtual compliant leg system could be realized as the visco-elastic model of a muscle, without installing any mechanical spring and damper on the leg.

Virtual leg impedance is created in the polar coordinate as in Fig.3-7 when the ‘virtual leg’ is defined as the straight line from the shoulder/hip joint to the foot-end of each leg. By imposing this virtual spring and damper, one can adopt results from biological examinations, simplified dynamic model studies and previous robotic experiments with mechanical springs installed on legs, such as proper value for leg stiffness and angle of attack.

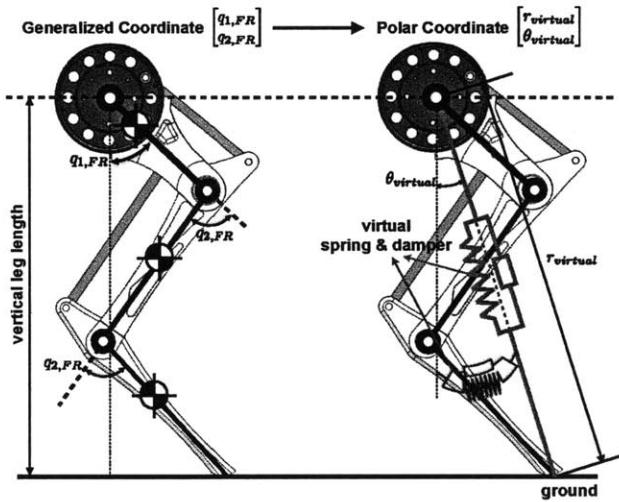


Figure 3-7: Front leg with the impedance control: virtual leg compliance realization with stiffness and damping for radial/angular directions

Fig.3-8 shows the block diagram for individual leg impedance controller implemented on the real robot. Measuring joint positions with the rotary encoders, leg forward kinematics and Jacobian return a foot-end position and velocity with respect to a shoulder/hip. The Cartesian position and velocity errors in trajectories are transformed to the polar coordinate and multiplied with the predefined impedance to calculate torque commands for each motor as:

$$u = J_{polar}^T \begin{bmatrix} K_{p,r}e_r + K_{d,r}\dot{e}_r \\ K_{p,\theta}e_\theta + K_{d,\theta}\dot{e}_\theta \end{bmatrix}, \quad (3.17)$$

where e_r , \dot{e}_r , e_θ , and \dot{e}_θ are radial position error, radial velocity error, angular position error and angular velocity error and J_{polar} is Jacobian from hip to foot end.

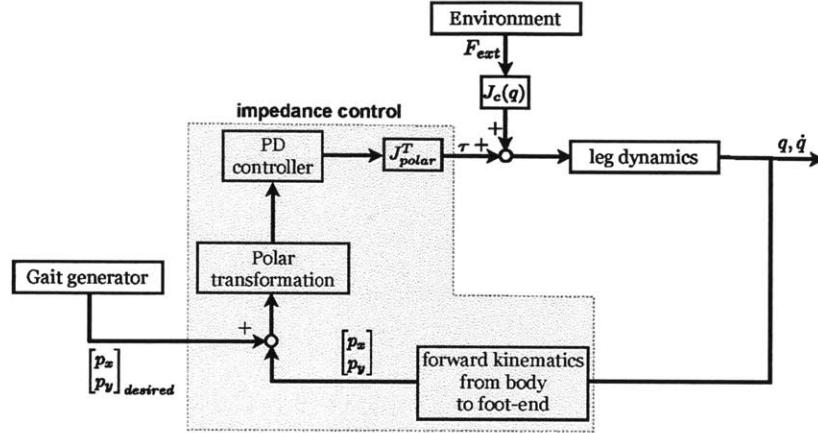


Figure 3-8: Block diagram for leg control; implementation of the impedance control to create virtual leg compliance.

Chapter 4

Simulation result

The control framework is assessed in the simulator. The simulation results are convincing enough for the author to verify the validity of the proposed control framework to enable fast and stable quadrupedal locomotion. It is also able to change its gait pattern from trot to gallop, followed by a few strides in the simulation. A still shot of the video is shown in Fig 4-1. The ground patch is adopted from the work of [78]. The simulator is constructed using MATLAB as developed in section 4. Equations of motion are integrated with the `ode45` solver, with the `RelTol` of 1e-6 and the `AbsTol` of 1e-6.

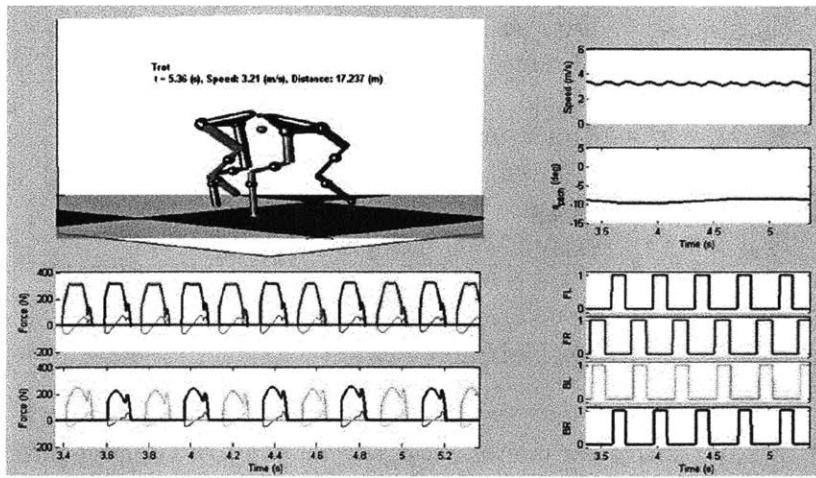


Figure 4-1: Animation interface constructed using MATLAB. 3-D animation of planar simulator, ground reaction forces, velocity, pitch variation, and leg phase(stance: 1 / swing: 0) are shown in the animation.

4.1 Trot running with constant speed

The robot achieved 3.4 m/s running with trot gait in the simulation experiment. A set of properly chosen control parameters for trot running are listed in Table 4.1. Corresponding leg trajectories for front and back legs are drawn with respect to local shoulder/hip joint of legs, in Fig. 4-2

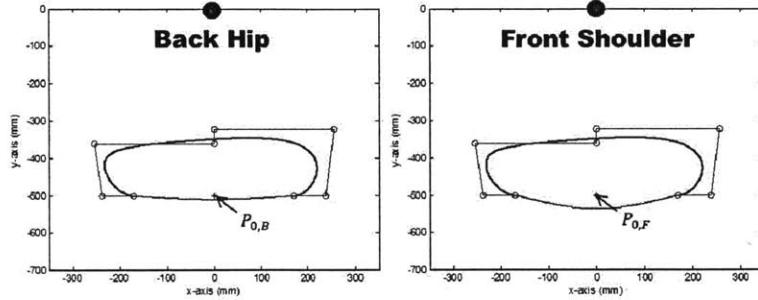


Figure 4-2: Leg trajectory designed for stable trot running. Both front and back leg trajectories are presented with respect to local shoulder/hip coordinates.

Terminology	Value
\hat{T}_{st}	determined according to Eqn.(3.1) with v_d
\hat{T}_{sw}	0.25 s
\mathfrak{B}_F	Bézier control points in Table. 3.2
$L_{span,F}$	170 mm
δ_F	36 mm
$P_{0,F}$	(0 mm, -500 mm)
\mathfrak{B}_B	Bézier control points in Table. 3.2
$L_{span,B}$	170 mm
δ_B	10 mm
$P_{0,B}$	(0 mm, -500 mm)
$K_{p,r}$	5,000 N/m
$K_{d,r}$	100 Ns/m
$K_{p,\theta}$	100 Nm/rad
$K_{d,\theta}$	4 Nms/rad

Table 4.1: Set of control parameters designed for the trot running simulation.

Fig. 4-3 shows the transient response of the simulation with the initial perturbation. The robot successfully reach steady-state running of approximately 3.4 m/s with

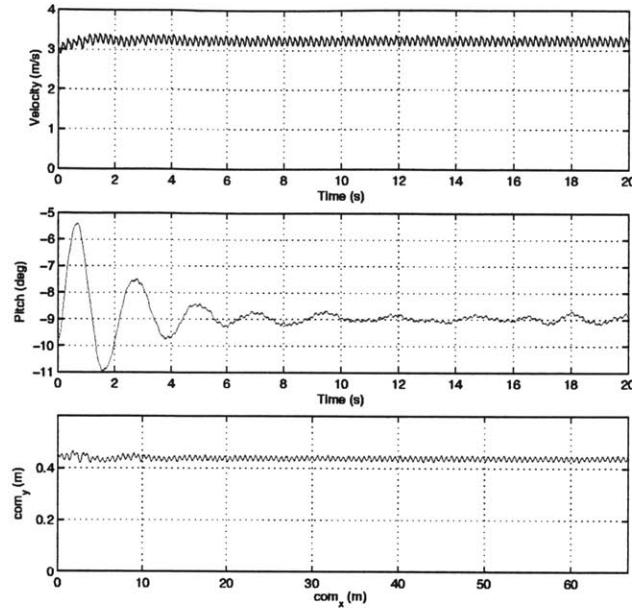


Figure 4-3: Trot running at 3.4 m/s. Velocity (top) and pitch of the front body (middle) are plotted over time. Height variation over distance traveled is also plotted (bottom). Pitch variation and height variation are damped out as the robot approaches to the steady state running.

stable trot gait; which was a bit faster than the target velocity, 3.0 m/s. Note that the steady-state running is achieved without global orientation information, resulting from the robot's intrinsic self-stability.

Fig. 4-4 shows GRFs solved in the simulator and GRFs estimated from torque command at each joints. The symmetric trot gait organized by the gait pattern modulator can be observed from the ground reaction force profiles. Integration of the vertical forces over one stride, i.e., from $t = 13.06$ s to $t = 13.44$ s, matches to the gravitational impulse over the same period, mgT_{stride} . Relatively flat portion of the force plot is due to saturation of knee joint torques for front legs. In addition, the comparison supports the concept of the proprioceptive feedback; the computed ground reaction forces and the estimated ground reaction forces are deviated from each other less than an order of magnitude.

Lastly, snapshots from the trot running of the planar MIT Cheeta robot simulation is shown in Fig.4-5.

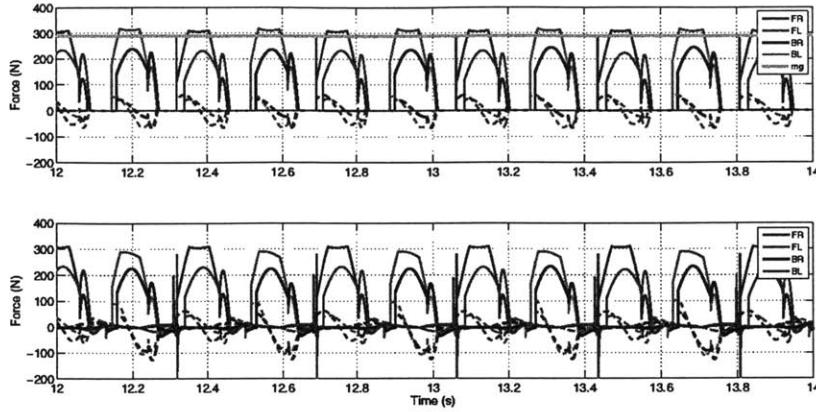


Figure 4-4: Measured ground reaction forces (top) and estimated ground reaction forces at each leg (bottom) between time $t = 12$ s to $t = 14$ s. Vertical ground reaction forces (solid lines) and horizontal ground reaction forces (dotted lines) are plotted together. Solid-dotted horizontal line indicates the weight, mg , of the robot, where $m = 29.443\text{kg}$ in the simulation. Input joint torques are coordinate transformed to estimate ground reaction forces.

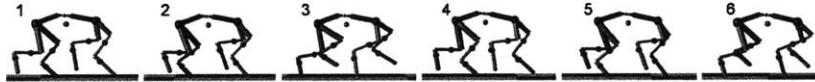


Figure 4-5: Snapshots of trot running at 3.4 m/s in the simulation

4.2 Trot running with acceleration and gait transition to gallop

The robot achieved accelerating trot running and subsequent gait transition to gallop in the simulation experiment. The acceleration is done by adjusting desired stance phase period according to Eqn.(3.1), while keeping the same trajectories for both front and back legs.

A gait transition algorithm compatible with the gait pattern modulator is also proposed, which is implemented in the experiment as well. Biological principle of gait transition in animal is unveiled yet, but it is both hypothesized and observed that animals have desirable gait pattern for different speed and therefore gait transition is necessary while accelerating or decelerating. To accomplish the gait transition we propose to change the phase lag vector $\Delta\vec{S}$ linearly with time within the predefined number of strides. As a gait pattern is defined with the set of the phase lag vector

$\Delta\vec{S}$, smooth and continuous gait transition is achieved by linear transition of $\Delta\vec{S}$ over a multiple period of \hat{T}_{stride} .

We first tried to change gait from trot to gallop, while maintaining the leg trajectory parameters. It is revealed that the leg trajectories for back leg have to be designed in a different way. The phase lag vector $\Delta\vec{S}$ and the leg trajectory control parameters are both changed linearly in time during gait transition. The modified equation for stance phase trajectory is as follows. Still, the trajectory is combination of harmonic waves to create desired shape with a few control parameters [3]. The back leg trajectory is designed considering that back legs provide strong propulsion forces to the quadrupeds in general. The additional variable GL represents *gallopization*, such that $GL = 1$ will modulate trajectory from sinusoidal shape for trot gait pattern.

Also, note that the reference point is moved backwards by 50 mm. This is speculated by the reasoning in [77] so that it can produce nose-down pitch and propulsion.

$$p_{i,y}^{st}(t) = \delta(\cos(\frac{\pi}{2L_{span}}p_{i,x}(t)) - \frac{GL}{2}\sin(\frac{2\pi}{2L_{span}}) + P_{0,y} \quad (4.1)$$

$$v_{i,y}^{st}(t) = \frac{dp_{i,y}}{dp_{i,x}} \frac{dp_{i,x}}{dt} = \frac{\delta\pi}{\hat{T}_{st}}(\sin(\frac{\pi}{2L_{span}}p_{i,x}(t)) + GL\cos(\frac{2\pi}{2L_{span}})) \quad (4.2)$$

A set of properly chosen control parameters for gallop running is listed in Table 4.2. Corresponding leg trajectories for front and back legs are drawn with respect to local shoulder/hip joint of legs, in Fig. 4-6

In Fig. 4-7, the stance/swing states of all four legs are plotted between time $t = 6$ s and $t = 9$ s. The gait transition happened when all four legs are in the swing phase, or, the robot has aerial phase. A clear distinction between trot gait and gallop gait is possible. Also, note the decrease in duty factor, a fraction of stance period in a stride, during acceleration in trot running.

Fig. 4-8 shows the velocity, pitch variation and center of mass height variation during steady state trot running, accelerating trot running, and gait transition to gallop. The pitch of the gallop running gait fluctuates much more than that of trot

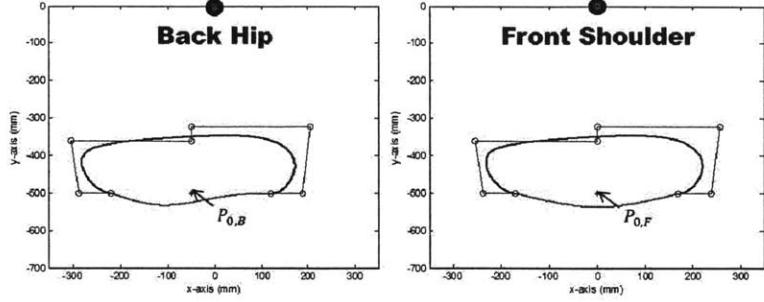


Figure 4-6: Leg trajectory designed for gallop running. Both front and back leg trajectories are presented with respect to local shoulder/hip coordinates.

running gait.

The measured ground reaction forces for four legs are also plotted in Fig. 4-9. It can be seen that the peak value for ground reaction forces increases as the robot accelerates. Note that rate of increase is more distinguishable in back legs than front legs. This can be explained by as robot accelerates, a nose-up pitch is induced, and back leg forces increase to cancel the pitch moment, which corresponds to the biological examination explained in [75]. Note the clear difference in coordination of four legs between trot and gallop. During gallop in the simulator, the forces are distributed almost equally to four legs.

Lastly, Fig.4-10 shows the simulated gait transition snapshots in one stride. In the simulation, the gait transition induces momentary instability which is desired to be stabilized in the gallop gait. Even though the stable gallop gait is designed with the proposed controller on the simulator, it is left for the future work.

Terminology	Value
\hat{T}_{st}	determined according to Eqn.(3.1) with v_d
\hat{T}_{sw}	0.25 s
\mathcal{B}_F	Bézier control points in Table. 3.2
$L_{span,F}$	170 mm
δ_F	36 mm
$P_{0,F}$	(0 mm, -500 mm)
\mathcal{B}_B	Bézier control points in Table. 3.2
$L_{span,B}$	170 mm
δ_B	25 mm
$P_{0,B}$	(0 mm, -500 mm)
GL	1
$K_{p,r}$	5,000 N/m
$K_{d,r}$	100 Ns/m
$K_{p,\theta}$	100 Nm/rad
$K_{d,\theta}$	4 Nms/rad

Table 4.2: Set of control parameters designed for the gallop running simulation.

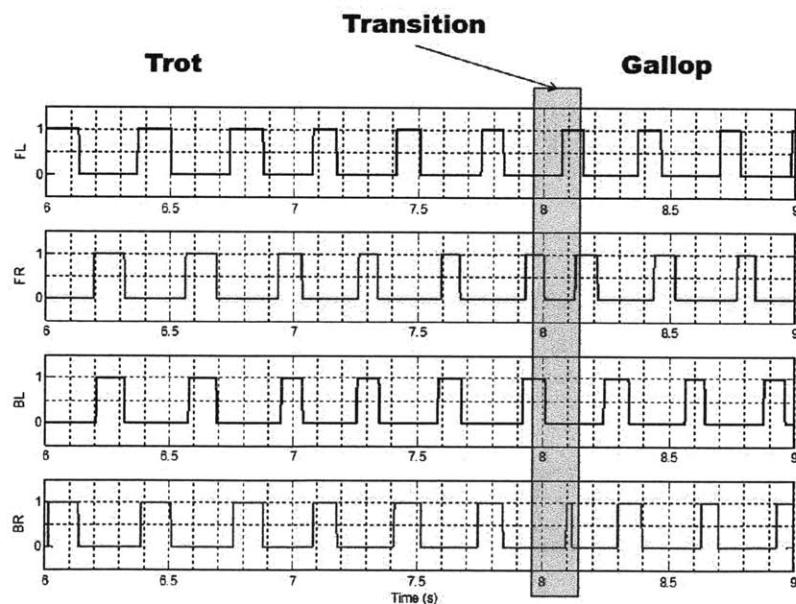


Figure 4-7: The state of all four legs from $t = 6$ s to $t = 9$ s. Gait transition happened at $t = 8$ s.

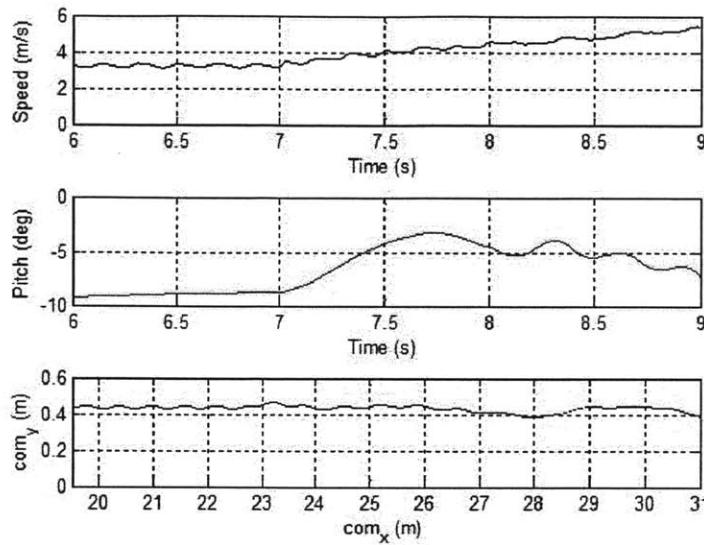


Figure 4-8: Trot acceleration and gait transition to gallop running. Velocity (top) and pitch of the front body (middle) are plotted over time. Height variation over distance traveled is also plotted (bottom). From $t = 7$ s to $t = 8$ s, the robot accelerates and at $t = 8$ s, transits its gait from trot to gallop.

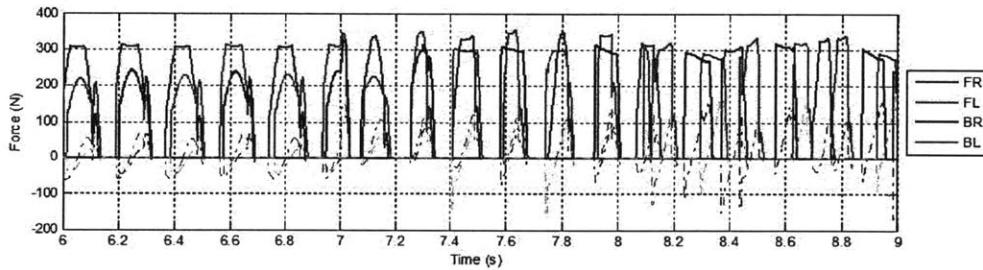


Figure 4-9: Measured ground reaction forces between time $t = 6$ s to $t = 9$ s. Gait transition happened at $t = 8$ s. Solid lines are vertical forces, and dotted lines are horizontal forces.

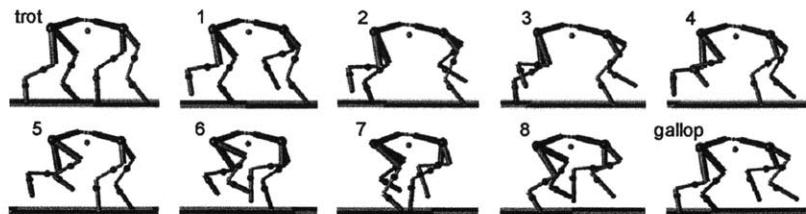


Figure 4-10: Snapshots of gait transition from trot to gallop by varying gait pattern parameters $\Delta \vec{S}$ in linear over time

Chapter 5

Experiments on the MIT Cheetah robot

The experiment demonstrated the performance of the proposed controller implemented on the real robot. During the experiments, the sagittally-constrained MIT Cheetah robot achieved fast and stable trot running up to the speed about 6 m/s, and transited to gallop running. The control system designed with National Instrument devices on the MIT Cheetah robot is largely the work of Sangok Seok. Readers are referred to Appendix.E for the video of the experiment.

5.1 Experimental setup

The MIT Cheetah robot is constrained on its sagittal plane, running on a modified commercial treadmill (SOLE TT8) as shown in Fig.5-1. The treadmill motor (3.5 hp TURDAN DC.PM 90V-25A) is manually controlled by a NI sbRIO-9642 with a motor driver (Apex Microtechnology MSA260KC) to synchronize its speed with the robot running speed. The maximum speed of the treadmill is measured to be 6 m/s. A minimal friction sliding guide is connected to the CoM of the robot through a revolute joint in order to prevent the roll and yaw motion of the robot. The effective mass of the guide on the robot, 3 kg, is equivalent to the mass of the four 22.2 V serially-connected LiPo batteries (465 Whrs) which is in the explosion proof box outside of

the robot for safety.

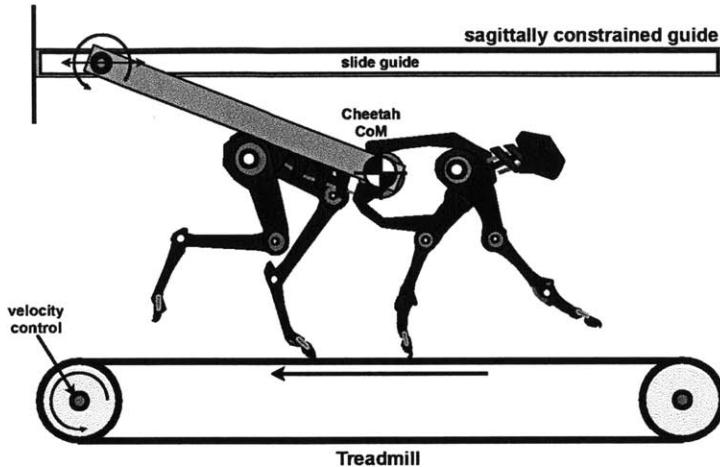


Figure 5-1: Experimental setup: the MIT Cheetah robot runs on the speed-controlled treadmill. Minimal friction sliding guides constrain the robot on its sagittal plane.

The feasible sensory feedback on the controller are only positional data for each leg joint given from the 13-bit rotary magnetic encoders installed at dual-coaxial BLDC motor units of each leg. Other state feedback of the robot are not available in the experiment. For the ‘TD’ detection of the reference leg the proposed proprioceptive feedback is set with the value of the threshold force, 2N, which is obtained in the preliminary experiment. Four Dynamixel EX-106+ smart motors are used to prevent ab/adduction of each leg. Also, the customized motor driver (MCU: Microchip dsPIC30F6010) handles the current control of each motor at 20 kHz.

The developed control framework is implemented with 4 kHz overall closed-loop control speed on the FPGA/RT based compactRIO (cRIO-9082) with the LabVIEW software. A wireless network (802.11 protocol) with the robot is established and the controllable parameters can be wirelessly adjusted by the operator through the LabVIEW front panel. All the data such as measured joint angles and commanded currents are saved with the sampling rate 250 μ s through the RT FIFO memory to the embedded flash memory during experiment.

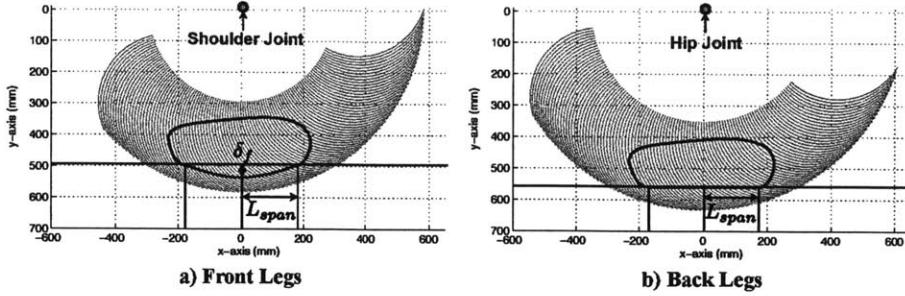


Figure 5-2: Gait trajectories implemented on the experiment with leg workspaces

5.2 Control parameters for the experiment

The predefined gait trajectories for the experiment are shown with respect to the shoulder joint for the front legs and the hip joint for back legs in Fig.5-2. Shaded regions are work space of each leg based on each mechanical joint limits. Control parameters for front and back legs, denoted as F and B respectively, are listed in Table.5.1.

Terminology	Value
\dot{T}_{st}	Varies according to Eqn.(3.1)
\dot{T}_{sw}	0.25 s
\mathfrak{B}_F	Bézier control points in Table. 3.2
$L_{span,F}$	170 mm
δ_F	36 mm
$P_{0,F}$	(0 mm, -500 mm)
\mathfrak{B}_B	Bézier control points in Table. 3.2
$L_{span,B}$	170 mm
δ_B	0 mm
$P_{0,B}$	(0 mm, -550 mm)
$K_{p,r}$	5,000 N/m
$K_{d,r}$	100 Ns/m
$K_{p,\theta}$	100 Nm/rad
$K_{d,\theta}$	4 Nms/rad

Table 5.1: Set of control parameters designed for the experiment.

The swing trajectories for four legs are set to be identical by sharing the same set of control points for the Bézier curve. During trot-walking at low speed in the

initial experiment, however, δ_F and δ_B are adjusted separately to stabilize the body pitch. Due to the coupling effect between the semi-active spine and back legs, the stance trajectory design of the back legs is conservatively designed just as flat simply by describing $\delta_B = 0$, which is discussed in 6. $P_{0,F}$ and $P_{0,B}$ are designed considering the stable attitude of the robot in the static posture. The desired swing leg retraction speed with $L_{span}=170$ mm is determined as 3.4 m/s by the property of the Bézier curve.

Referring to the dimensionless analysis of ‘relative leg stiffness’ of running animals [79], $k_{rel,leg}$, the virtual radial stiffness of each leg of the robot is set to be 5,000 N/m. The gain values for radial damping ($K_{d,r}$), angular stiffness ($K_{p,\theta}$) and angular damping ($K_{d,\theta}$) are obtained with preliminary tests.

To increase the speed of trot running, the stride frequency is modulated in accordance with $v_d = \frac{2L_{span}}{\hat{T}_{st}}$ as described in Chapter.3. , while each leg compliance and trajectories are fixed during the experiment for a range of speeds. Biological observations and previous robotic experiment inspired this algorithm [6]. Farley et al. observed that leg stiffness(k_{leg}) is independent on locomotion speed and examined that animals such as dogs and horses increase their stride frequency linearly with speed in their treadmill running experiment. However, the statement itself is controversial and a further investigation is required.

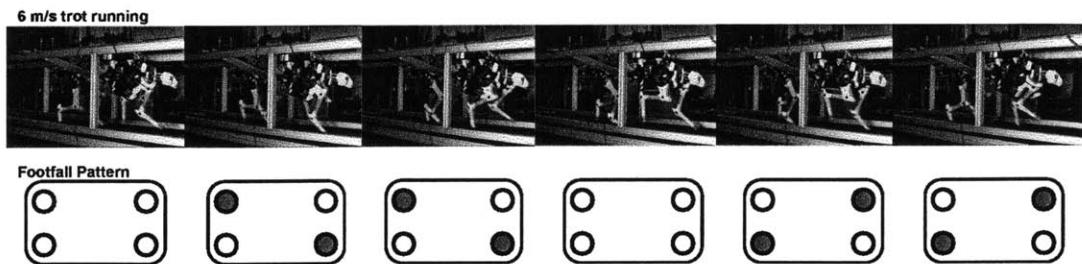


Figure 5-3: Snapshot for the trot running gait of the MIT Cheetah robot with each footfall diagram: the existence of the aerial phase is observed between each paired legs’ step in trotting gait

5.3 Experimental results

It is demonstrated that the robot can accelerate up to 6m/s with the trot gait pattern according to the operator's forward speed command. Stable running in the sense of the limit cycle is observed over all the range of recorded velocity with the trot footfall pattern as in Fig.5-3 in the experiment.

At low speed the robot walked with trot gait, but as speed goes up, it started running, having the aerial phase which all four legs are off the ground.

5.3.1 Estimated locomotion speed

Fig.5-4 shows the estimated forward speed of the MIT Cheetah robot based on the measured kinematic data during the experiment because the controlled stance phase period (\hat{T}_{st}) for each leg is determined according to the desired speed. Comparing the

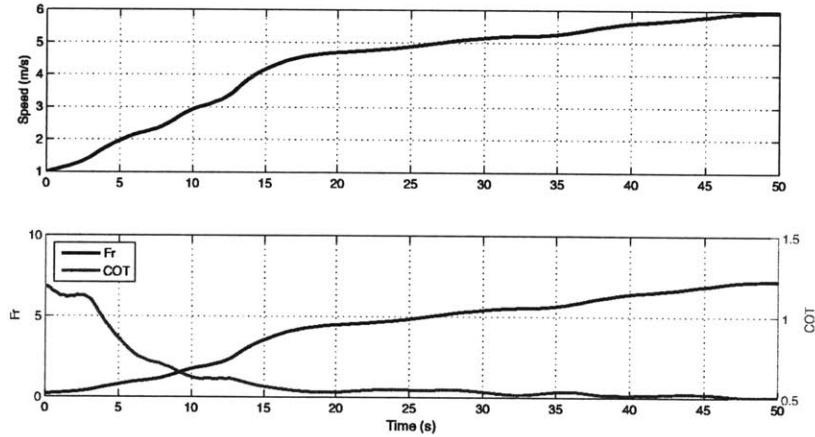


Figure 5-4: Estimated forward velocity (top) of the MIT Cheetah robot: ± 0.15 m/s bounded difference was observed between the estimated speed and the treadmill speed measured by a rotary encoder (CUI Inc. AMT 102-V) directly attached to the DC motor main axis. Froude number (middle, blue solid line) and cost of transport (middle, red solid line) of the robot are plotted together.

observed speed of the treadmill, the difference is smaller than 0.15 m/s. As shown, the robot accelerated up to approximately 6m/s in 50 seconds. The nondimensional metrics Froude number (Fr), cost of transport (CoT) shown together in the graph below. Fr was a record-breaking high number as 7.34 and CoT was impressively

low number as 0.52 to rival legged animals around the robot's maximum estimated speed, 6 m/s.

5.3.2 Measured leg trajectories

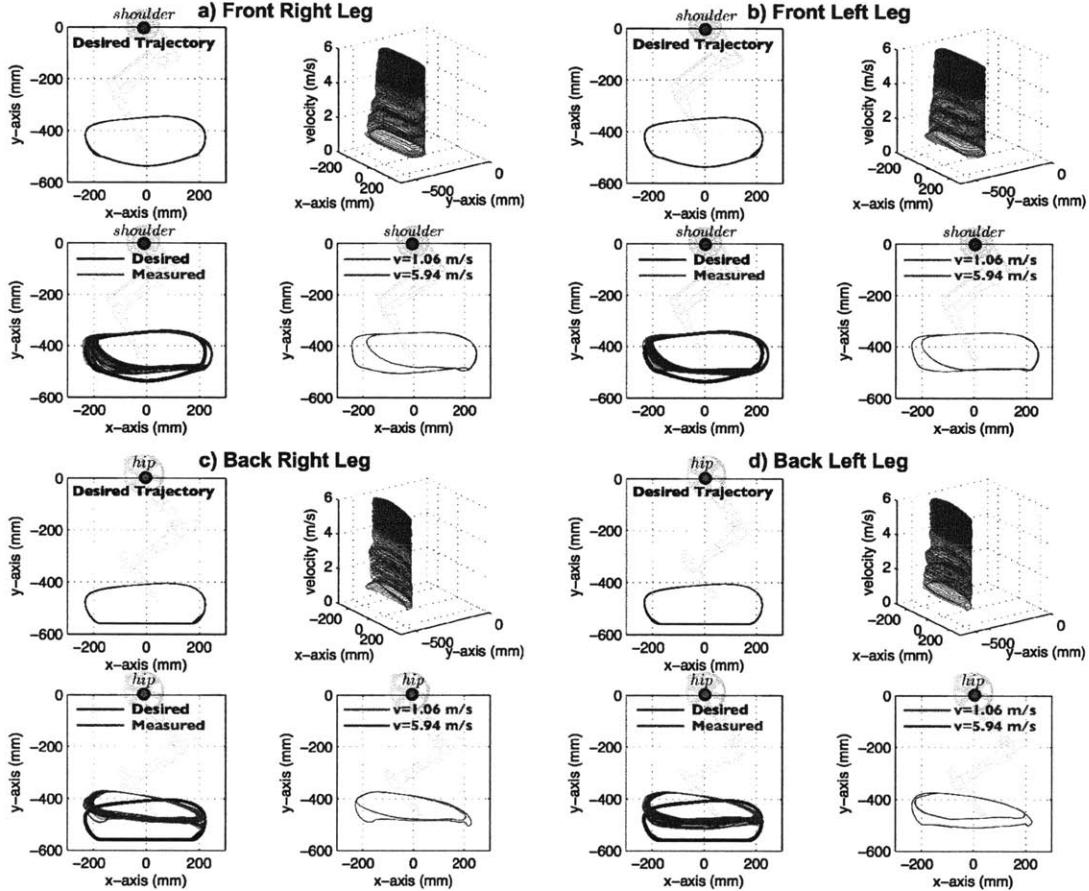


Figure 5-5: Graph set of generated desired trajectories by the leg trajectory generator and measured actual gait trajectories during trotting gait from 1 to 6 m/s for a) front right leg, b) front left leg, c) back right leg, d) back left leg. In each leg graph, *top left*: desired trajectories, *top right*: actual trajectories with respect to speed, *bottom left*: comparison between desired/actual trajectories, *bottom right*: comparison between actual trajectories at low/high speeds, respectively

Fig.5-5 shows trajectories of all four legs measured over various speeds during the experimentation. Trajectories are plotted with respect to each shoulder or hip joint in the body local coordinate since access to global orientation is precluded in this experimental setup. Nevertheless, trajectories are consistent with the desired trajec-

tory over a large speed variation. The data supports that both the pitch and height of the frontal body fluctuate a little during the experiment. Slight pitch variation at the beginning and the end of the swing phase can occur due to effect of the coupling mechanism between hind legs and the spine. The top left subplot shows the desired trajectory. Again, the shape of the desired trajectory remained the same over different target speeds. Note the short straight line connecting the end of the swing phase trajectory and the beginning of the stance phase trajectory. This short-cut transition in commanded trajectory was due to early TD detection. The top right subplot expands the actual trajectories with respect to its achieved speeds. As speed goes up, the vertical width of the trajectory closed-loop decreases such that shoulder/hip heights in the stance phase decreases. The shorter stance time requires the higher vertical force to provide the required vertical impulse for running as explained in Eqn.(3.16). The bottom left compares the desired trajectory and the measured trajectory. We can observe the tracking performance of the impedance control for the swing phase; the trajectory converges within a fourth of duration of the entire swing phase. On the other hand, for the stance phase, the impedance control exerted high enough forces to make the robot run, due to the errors in stance phase trajectories. Stance phase performance will be discussed with force plot. The bottom right plot compares the measured trajectory at low speed and high speed. The relatively flat portion of the trajectory indicates ground-contact. It can be concluded that the distance traveled of the front right shoulder was shorter than we designed as the target speed increases.

A coupling effect of back legs with the spine through the differential gear was observed in the measured back leg gait trajectories, which leads to the deviation of the actual trajectories away from the desired trajectory even during the swing phase where there was no interaction with the ground. Therefore, in order to improve the performance of back legs, which are coupled with the spine, understanding of the back legs-spine system is required.

5.3.3 Estimation of generated leg-force based on applied currents

To investigate interaction between the robot and the environment, forces generated at foot-end of the robot should be analyzed. However, the experiment was done without a force plate on/under the treadmill or any force sensor at the foot to measure GRFs. Nevertheless, the forces exerted at foot-end can be accurately estimated by joint torques generated from the impedance control law, which has been demonstrated in the simulation and in [66]. Each joint torques can be estimated by currents measured at the customized motor driver for the MIT Cheetah robot considering actuation transmission and pantographic leg design.

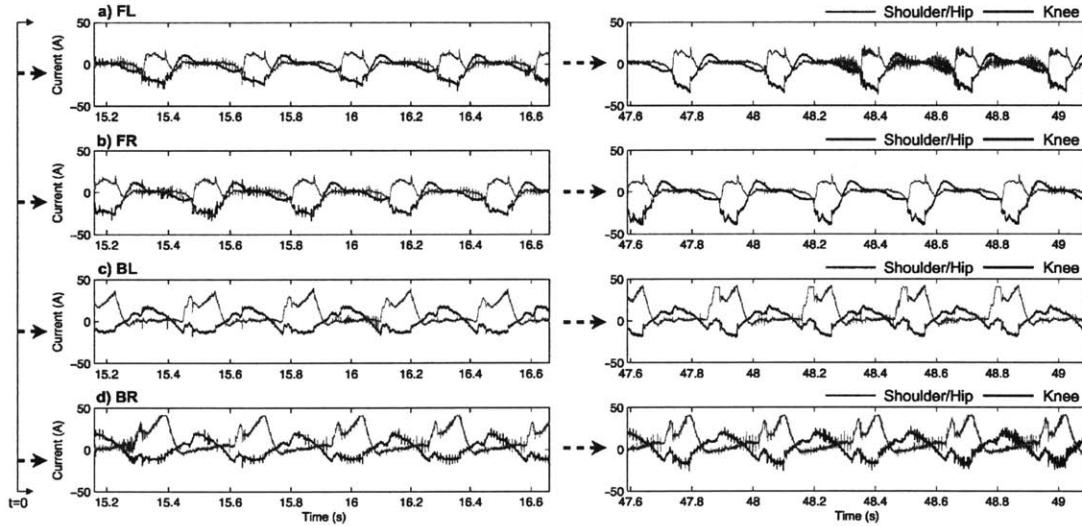


Figure 5-6: Commanded currents applied to each EM motor for shoulder/hip and knee joints at leg $i \in \{FL, FR, BR, BL\}$: The motor driver supplies the commanded current to the EM motor which proportionally generate torque by the torque constant K_τ .

The currents at each actuator are precisely measured for the entire experiment. Fig.5-6 shows measured currents provided to the dual-coaxial electric motors at the robot's shoulder/hip, for 1.5 second around different speeds (4.1 m/s and 5.9 m/s). These two sections are selected for fair comparison between different speed, because the robot CoT, an indicator of efficiency, was similar.

Note the difference between currents at low speed and high speed, as well as for

front legs and back legs. We can observe contralateral symmetry in the graphs, which is the characteristic of symmetric gait pattern, trot. Increase in peak current and decrease in duration of high-current region are observed at higher speed. Currents measured at back legs are generally much higher than that at front legs. This is because 1) the back legs are longer than the front legs and therefore higher torques are required to exert forces at foot end, given similar hip/shoulder joint height, and 2) the currents used in back legs also contribute to rotating spine.

5.3.4 Gait transition

The gait transition method proposed in the Chapter 4 is implemented on the MIT Cheetah robot. At the maximum speed 6 m/s in the experiment, the robot was commanded to change its gait from trot to gallop and it was recorded by the high-speed camera. The Fig.5-7 shows the phase signals for all legs when the gait pattern modulator linearly changes its phase lag parameters from trot to gallop in the predefined period, $6 \times \hat{T}_{\text{stride}}$. The desired smooth and continuous gait transition was observed and the robot could keep running a few strides with the gallop gait pattern. However, this is the preliminary result and further research will enlighten stable gallop with smooth gait transition.

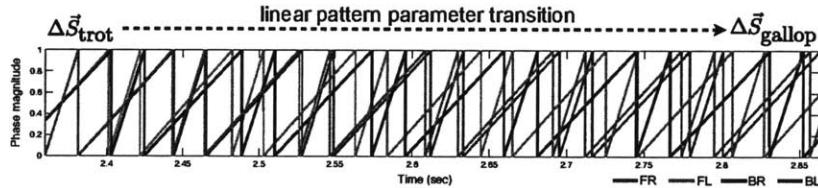


Figure 5-7: Sawtooth phase signals generated by phase modulator for gait transition from trot to gallop. The phase signal converges from one to another, linearly with time. Gait transition through linear convergence from the trot parameters to the gallop parameters

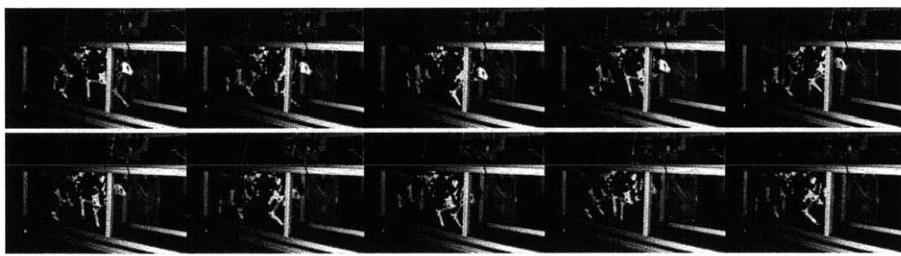


Figure 5-8: Snapshots of gait transition from trot to gallop recorded by a high speed video captured at 500 fps depicting 150 ms

Chapter 6

Discussion

To the best of the author's knowledge, the MIT Cheetah robot is the fastest and the most efficient quadruped robot among those which are actuated by electro-magnetic motors. The FPGA/RT based system architecture enables the fast closed-loop control including data logging for desired/measured leg joint positions and commanded currents. Based on these logged data, key features of the experiment are discussed in this chapter.

6.1 Estimated pitch variation

Effective indices for locomotive stability in the sense of limit-cycle can be body height and pitch variations. We hypothesizes the stable gait has small repetitive fluctuation of body height and pitch. These height and pitch fluctuations can be estimated with measured foot-end trajectory data for four legs as showin in Fig.6-1. The flat regions of the measured foot-end trajectories is due to interaction with the ground and therefore represents the stance phase. The height of each shoulder/hip joint is equivalent to the distance between shoulder/hip joint and foot-end positions in the stance phase at instant. Therefore, pitch variation can be deduced from difference between front and back height data, considering the shoulder-to-hip horizontal distance, 66 cm.

During trotting, the heights of both shoulder/hip joints are observed to remain around 0.5 m. Maximum estimated height difference between a pair of legs for trot is

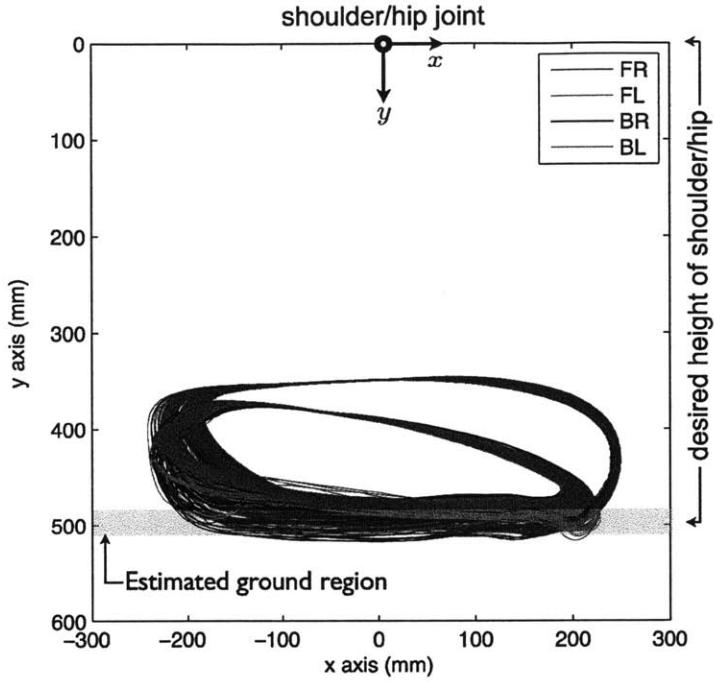


Figure 6-1: Actual measured trajectories of four legs with respect to each shoulder/hip joint, where all the trajectories for four legs are overlapped with respect to each shoulder/hip joint. The data allow us to approximate the height/pitch variation of the robot during trot experiment.

roughly around 4 cm at low speed and around 2.5 cm at high speed. The estimated pitch variation decreases as locomotion speed increases. Maximum estimated pitch variation is estimated to be 4 deg at low speed and only 2.2 deg at high speed. This small pitch variation relies on intrinsic stability. Since virtual compliance exerts force proportional to error, if a large pitch motion occurs, force exerted by front/back legs naturally adapt to induce pitch moment to cancel out the disturbance.

For the same reason, as speed increases, the height of the body gets lower as shown in the trajectory vs speed plot of Fig.5-5. In the experiment, each designed leg trajectory is identically applied to front/back legs over different speed range. At higher speed, larger vertical forces are required within shorter stance period to satisfy a law of momentum conservation in Eqn.(3.16). Therefore, by larger exerted forces on compliant legs the height of body is observed to become slightly lower as speed goes up.

6.2 Fr and COT

Nondimensional metrics such as Froude number (Fr) [74][31] and cost of transport (COT) [80] are often used to appropriately compare different sized legged robots and animals. Fr is size-independent locomotion speed, defined as below

$$Fr = \frac{v^2}{gh}, \quad (6.1)$$

where v is a characteristic (forward) speed of the locomotion, g is the gravitational acceleration, h is a characteristic length [3].¹. For legged robots and animals, h is usually chosen as the shoulder/hip height; in case of the MIT Cheetah robot, it is assumed to be 0.5 m from Fig.6-1.

The maximum Fr of the MIT Cheetah robot is 7.34. In comparison to other quadrupedal robots, this is a remarkably high value according to the data found in [31]. Biological data showed that quadrupedal animals normally transit its gait from walk to trot at $Fr = 1$ and trot to gallop at around $Fr = 2$ or 3 [3], and dogs are observed to use trot gait up to the speed which corresponds to $Fr \approx 5.5$ [4]. Interestingly, we could achieve much higher Fr using trot gait pattern with MIT Cheetah robot. We tentatively speculate that the difference between animals' muscle actuation system and the robot's electrically powered actuation system may enable the accomplishment, but we withhold the conclusion.

Indeed, we believe the maximum speed of the robot could be even higher because 1) the recorded maximum speed of the robot was limited by the maximum speed of the treadmill and 2) the robot's maximum power consumption was not reached yet; At 6 m/s, the consumed power was 1 kW, whereas the maximum power rating of the robot is assumed to be 3 kW.

COT is used as a mass-independent energy/power consumption, which measures energy efficiency of the locomotion. COT is defined as

¹Some use different notation, $\frac{v}{\sqrt{gh}}$

$$COT = \frac{P}{Wv}, \quad (6.2)$$

where P is power used, W is weight of the robot and v is locomotion speed [80]. Computed COT of the robot is surprisingly low by virtue of its mechanical and electrical design. For detailed description of enabling technology applied to the robot to achieve high energy efficiency, the reader is referred to [81].

6.3 Evidences for the MIT Cheetah running

It is estimated that the MIT Cheetah robot ran around 2.2 m/s, by taking the energetic criterion of running, which was introduced in [82]. Fig.6-2 shows the vertical length of the virtual front right leg and the estimated speed in the time interval 2 ~ 10 s. The shoulder height can be estimated by the vertical length of the virtual leg in

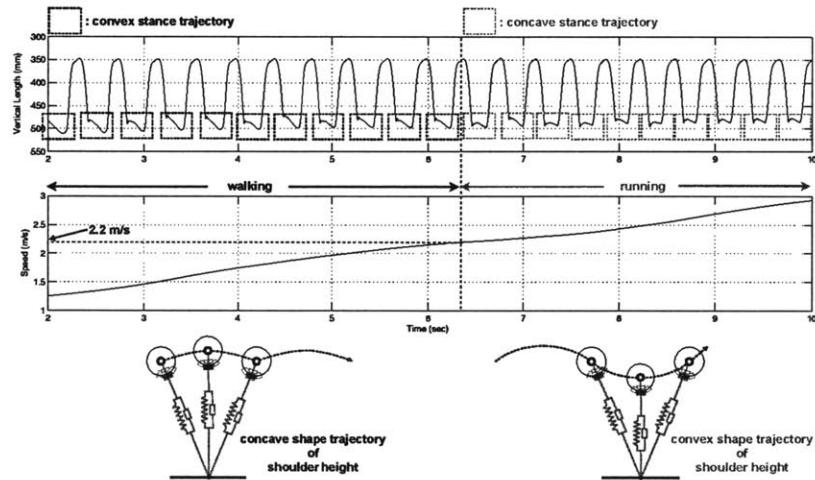


Figure 6-2: Estimated shoulder height trajectories: convex shape of shoulder height trajectory as a typical representation of running is observed after 2.2 m/s.

the stance phase. The convex shape of the vertical length matches to the concave shape of the shoulder height and vice versa. Therefore, the typical concave/convex shoulder trajectory for walking/running can be drawn before/after 2.2 m/s which is

matched to the Froude number 1 in Fig.5-4. Interestingly, this result is aligned with the biological studies which have shown that quadrupeds start running at $\text{Fr} = 1$ [3].

The estimated duty factor D_{est} in Fig.6-3 provides another evidence that the MIT Cheetah robot could run. Duty factor of a foot is defined as a fraction of the stride

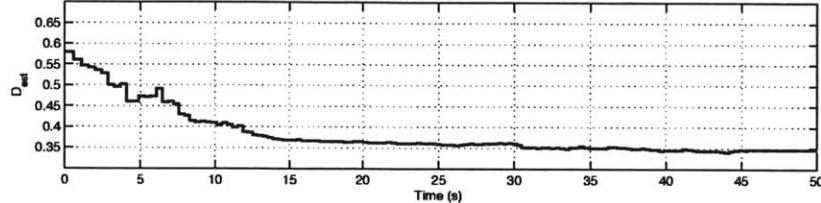


Figure 6-3: Estimated duty factor D_{est} for the front right leg: measured kinematic data for virtual leg is used to estimate duration of the stance/swing phases.

for which the foot is in contact with the ground [3]. Therefore, duty factor < 0.5 implies the animal/ robot has aerial phase. Even though the definition of running is controversial and animals showed running with duty factor > 0.5 , [4], [83], [82], it is clear that having aerial phase is sufficient condition for running. Because of the absence of force plate in the experiment, we estimated duty factor, D_{est} , based on kinematic data of each leg as we did for pitch estimation. From the duty factor plot, we can say that the robot had the aerial phase after time $t = 4$ s, where the speed estimation was 1.4 m/s.

6.4 Estimated force exerted on foot end-effectors

To analyze the performance and characteristic of the trot running result in Cartesian coordinate, we estimate the interaction forces between the robot and the ground by measured current in Fig.5-6. The motor driver is devised to compensate the nonlinear effects such as cogging torque of the BLDC motor in a feed-forward manner and therefore guarantees linear relation between motor torque and applied current.

Therefore, torque outputs from BLDC motors are calculated based on measured currents and measured torque constant, $K_T = 0.27 \text{ Nm/A}$. Assuming frictionless joints and rigid leg segments and considering actuator transmission and leg configuration at each moment, forces exerted by the foot-end are estimated as below:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix}_{foot} = (J_{Cart}^T)^{-1} \tau_{2 \times 1}, \quad (6.3)$$

where J_{Cart} is Jacobian from shoulder/hip to foot-end. In the preliminary test, the bandwidth of the leg system was observed around 400 Hz. Therefore, the estimated forces are filtered with the cut-off frequency of 400 Hz. The vertical force F_y and the horizontal force F_x exerted to the ground by the robot are estimated with respect to the local coordinate at each shoulder/hip joint of the leg, which is described in Fig.2-1. Nonetheless, we believe this can be an effective approximation for GRFs with respect to the inertial frame, since the pitch variation was a little.

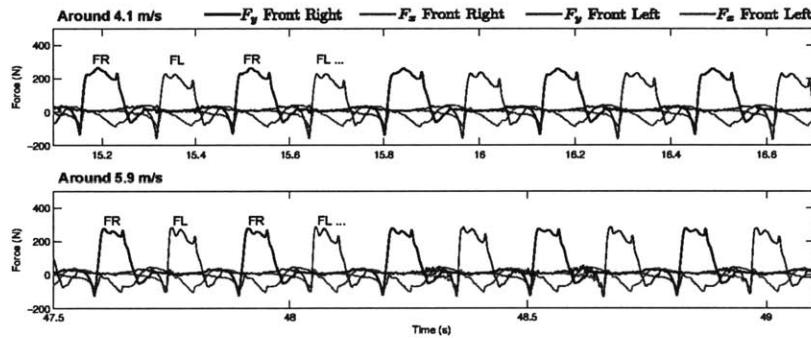


Figure 6-4: Estimation of generated forces in Cartesian coordinate at foot-end of front legs. Stance phase (high vertical force region) and swing phase (low vertical force region) can be approximated from the graph. Increase in peak force value is observed with decrease in duty factor.

6.5 Force analysis on the front legs

Fig.6-4 shows horizontal and vertical force estimations, F_x , F_y , at both front legs for each duration. High vertical force regions represent stance phases of each leg. Characteristics of symmetric gait pattern, trot, are clearly observed in the graphs. Identical forces produced by left and right side of legs are rhythmically alternating each other over time, peak value and frequency of vertical forces increase as desired speed increases. High propulsion force, expressed as negative F_x , accompany high

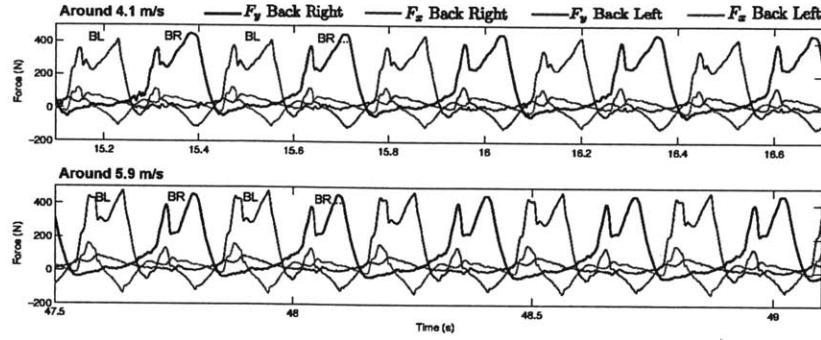


Figure 6-5: Estimation of generated forces in Cartesian coordinate at foot-end of back legs. Increase in peak force value is observed with decrease in duty factor

vertical forces during stance phase to compensate impact loss. For swing phase while protracting legs, forces generated are small compared to that for stance phase, due to leg design and swing phase trajectory design.

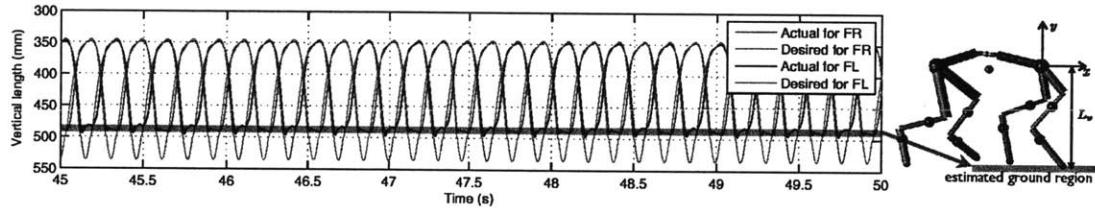


Figure 6-6: Desired/actual vertical virtual leg lengths L_v

Fig.6-6 shows the vertical motion of front legs with corresponding desired trajectories at 6 m/s. The desired foot-end vertical trajectories penetrate into the ground in the short stance period, but the actual foot-end vertical trajectories L_v cannot track the desired trajectory due to the ground contact and remains in the estimated ground region (around 0.5 m), which enables the distinct separation for the stance and air phases. Therefore, the hypothesis for stationary height of the shoulder joint during the short stance period is effective such that the height variation is smaller than 1 cm around 5.9 m/s. When the leg contacts with ground, the virtual leg compliance generates vertical forces according to kinematic error intentionally induced by the proposed sinusoidal trajectory for the stance phase and the time-integration of the forces during the stance period is the vertical impulse required for the trot running.

6.6 Contribution of virtual stiffness and virtual damping to leg compliance

In order to quantify significance of virtual damping in the stance control, Fig.6-7 plots contributions of each component of virtual leg compliance (virtual stiffness/damping) to force exerted by the foot-end at two different speeds. Not only rejecting force ripple in trajectory tracking control in the swing phase, virtual damping generates significant compensating force in the stance phase at high speed.

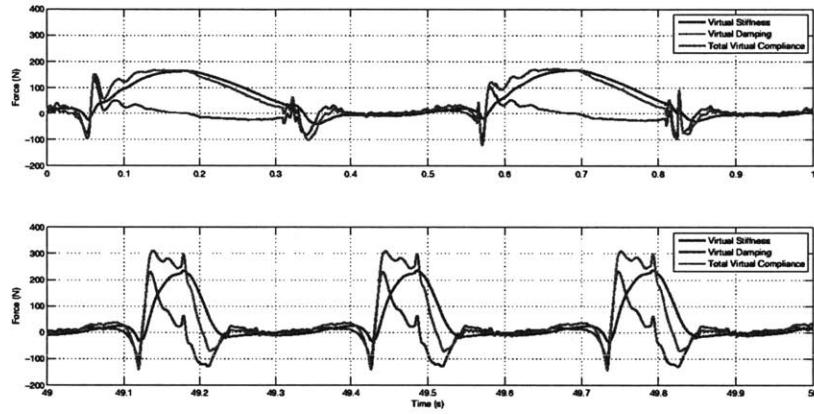


Figure 6-7: Virtual stiffness and damping contribution to create vertical force at the foot-end

The two distinctive phenomena observed in the figure are following: 1) virtual damping creates abrupt high forces at ‘TD’ event to deal with impact, and 2) the contribution of virtual damping increases as duration of stance phase decreases, or, as desired speed increases. Both features can be clearly explained by the fact that force created by virtual damping is reactive to velocity error of the foot-end. A sudden impediment due to the ground induces large velocity error which creates a peak force at impact. As target speed increases, desired stance period decreases and therefore the commanded velocity becomes faster. Naturally, it results in a huge velocity errors accompanied with large force exertion. Therefore, virtual stiffness is dominant at low speed but two compliant elements are equally dominant at high speed. By employing this effect, we could create very stable fast locomotion with a little pitch fluctuation, without variable compliance or variable trajectory over speed.

6.7 Force analysis on the back legs

Forces exerted at foot-end of the back legs are similarly calculated as shown in Fig.6-5. As mentioned in chapter 5, however, these values are not good approximation of GRFs because the back leg's actuating forces are spent for generating GRFs as well as for bending spine through a differential gear. Nonlinear stiffness of spine is not accurately modeled yet, thus analysis of GRFs exerted by back legs is complicated. Still, general trends shown in the front legs are observed in the back legs. similar. It is possible to observe contralateral symmetry between left and right legs, and legs exert higher peak forces at higher speeds. To improve tracking/compliance control performance of the back legs, further consideration of the system identification for spine is required.

Chapter 7

Concluding Remarks and Future Research Topics

A novel hierarchical controller for quadrupedal robots is presented in this thesis. The controller is based on three key strategies: self-stability, equilibrium point hypothesis, and gait pattern modulator combined with touch down event detection. Virtual compliance of each leg is created by low level impedance controller, providing self-stability to the robot. Leg trajectories are designed to control the motions and ground reaction forces, exploiting the virtual compliance in a manner suggested by the equilibrium-point hypothesis. Phase coupling between four legs is imposed by the high level gait pattern modulator to generate specific quadrupedal gait patterns. Stride-to-stride pattern modulation is proposed to adapt to the environment while maintaining the target gait pattern.

A dynamic simulator is constructed based on the algebraic impact law and the Coulomb friction model using MATLAB. In the simulator, the proposed controller demonstrates capability of achieving a stable trot running, and achieving gait transition from trot to gallop. Simulation experiments conclude that the desired leg trajectories for gallop should be differently designed from that of trot, while a range of speeds could be achieved without modulating spatial parameters for the leg trajectory in trot running.

The proposed controller is implemented on the MIT Cheetah robot running on

the treadmill, without any force sensor or attitude measurements. The performance of the robot is remarkable in terms of both speed and energy efficiency, which are noticeable from corresponding Fr and CoT.

A set of control parameters is presented in the thesis with additional degrees of freedom permitted to the operator, but basically the configuration of the set remains flexible. According to the simulation experiments, the construction of this set of parameters highly influences the performance of the robot. The hierarchical structure of the controller and the flexibility in designing relationship between control variables open various possibilities on further investigation.

1. Achieving stable gallop at even higher speed by using the proposed control framework remains an ongoing work. Proper design of front/back leg trajectories and proper temporal coordination of four legs should be investigated.
2. The relationship between control parameters will be further investigated to reduce the DoF permitted to the operator. We expect to extract and exploit suitable principles from biological findings so that the operator ends up with modulating only the target speed and all the control parameters are automatically adjusted.
3. The ground reaction forces can be more accurately modulated by the force tracking impedance control. The force tracking impedance control has been developed for manipulators fixed to the base. This approach can be extended to legged machines if full-state feedback is feasible.
4. The spine will be modeled so that a force compensation for the effect of the spine becomes possible in a feed-forward manner. We speculate that this flexible spine will play an important role in galloping by storing and releasing elastic energy.
5. An intelligent algorithm for active posture control using attitude sensory feedback will be investigated for stable running on a rough terrain. Active stabilizing on top of self-stabilizing will show a better performance on irregular terrain while maintaining simplicity in control.

6. A stabilizing controller for outdoor running using attitude sensory feedback will be investigated. Roll and yaw stabilization have be to integrated with the current control framework through additional actuation in ab/adduction of each leg.

Appendix A

Model Parameters

Characteristics	Value
$m_{body,F}$	10.05 kg
$l_{body,F}$	0.2 m
$m_{scapula,F}, m_{humerus,F}, m_{radius,F}$	2.63 kg, 0.16 kg, 0.12 kg
$l_{scapula,F}, l_{humerus,F}, l_{radius,F}$	0.16 m, 0.22 m, 0.2222m
m_{spine}	0.225 kg
l_{spine}	0.144 m
$m_{body,B}$	7.13 kg
$l_{body,B}$	0.187 m
$m_{femur,B}, m_{tibia,B}, m_{metatarsals,B}$	2.70 kg, 0.16kg, 0.085 kg
$l_{femur,B}, l_{tibia,B}, l_{metatarsals,B}$	0.245 m, 0.22 m, 0.165m

Table A.1: Model Parameters

Table A.1 shows the characteristic model parameters of the MIT Cheetah robot. Note that, mass of the proximal segments($m_{scapula,F}, m_{femur,B}$) include motor rotor mass.

Appendix B

Back spine coordination

$$q_{spine} = C_s(q_{humerus,BL} + q_{humerus,BR}) \quad (\text{B.1})$$

$$q_{backpitch} = q_{frontpitch} + 4q_{spine} \quad (\text{B.2})$$

Where $C_s = \frac{393}{6400}$ is the effective differential gear ratio.

Appendix C

Leg compliance

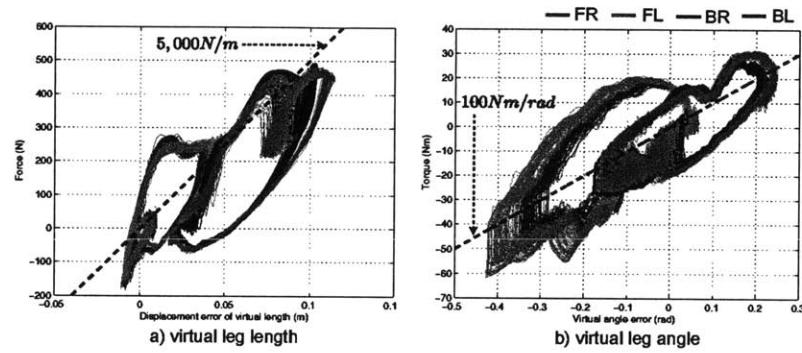


Figure C-1: Measured virtual leg compliance in the polar coordinate (r, θ) : a) longitudinal stiffness b) angular stiffness

Appendix D

Inverse Kinematics

The derivation of the inverse kinematics for three-segment leg is adopted from [84].

Typo in the reference is modified and rewritten in this work.

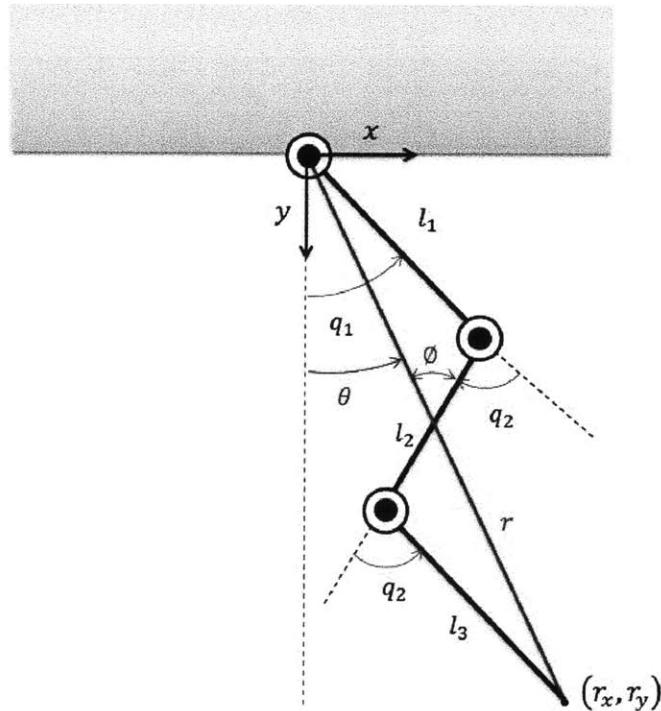


Figure D-1: Inverse kinematics of three-segment leg. The proximal and the distal segment are parallel due to pantographic leg design.

$$\theta = \arctan\left(\frac{r_x}{r_y}\right) \quad (\text{D.1})$$

$$r = \sqrt{r_x^2 + r_y^2} \quad (\text{D.2})$$

$$q_2 = q_3 = \arccos\left(\frac{(l_1^2 + l_2^2 + l_3^2 + 2l_1l_3) - r^2}{2l_2(l_1 + l_3)}\right) \quad (\text{D.3})$$

$$\phi = \arctan\left(\frac{(l_1 + l_3)\sin q_2}{l_2 - (l_1 + l_3)\cos q_2}\right) \quad (\text{D.4})$$

$$q_1 = \pi + \theta - q_2 - \phi \quad (\text{D.5})$$

$$\dot{\alpha} = \frac{r_y v_x - r_x v_y}{r_x^2 + r_y^2} \quad (\text{D.6})$$

$$\dot{r} = \frac{v_x r_x + v_y r_y}{\sqrt{r_x^2 + r_y^2}} \quad (\text{D.7})$$

$$\dot{q}_2 = \dot{q}_3 = \frac{r \dot{r}}{l_2(l_1 + l_3) \sin q_2} \quad (\text{D.8})$$

$$\dot{\rho} = \frac{(l_1 + l_3)(l_2 \cos q_2 - (l_1 + l_3)) \dot{q}_2}{l_2^2 + (l_1 + l_3)^2 - 2l_2(l_1 + l_3) \cos q_2} \quad (\text{D.9})$$

$$\dot{q}_1 = \dot{\theta} - \dot{q}_2 - \dot{\phi} \quad (\text{D.10})$$

Appendix E

Supplementary Video

A video is provided along with this thesis to show the results of the experiment.

Experimental demonstration 1 <http://www.youtube.com/watch?v=L-OXhfp3tWk>

Experimental demonstration 2 <http://www.youtube.com/watch?v=UBHJqnM8RTU>

Table E.1: Supplementary videos and urls are listed in the table

Bibliography

- [1] J. E. Bares and W. L. Whittaker, “Configuration of autonomous walkers for extreme terrain,” *The International Journal of Robotics Research*, vol. 12, no. 6, pp. 535–559, 1993.
- [2] C. D. Remy, “Optimal exploitation of natural dynamics in legged locomotion,” Ph.D. dissertation, ETH, 2011.
- [3] R. M. Alexander, “The gaits of bipedal and quadrupedal animals,” *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 49–59, 1984.
- [4] L. D. Maes, M. Herbin, R. hacker, V. L. Bels, and A. Abourachid, “Steady locomotion in dogs: temporal and associated spatial coordination patterns and the effect of speed,” *The Journal of Experimental Biology*, vol. 211, pp. 138–149, 2008.
- [5] S. J. Wickler, D. F. Hoyt, E. A. Cogger, and G. Myers, “The energetics of the trot–gallop transition,” *The Journal of Experimental Biology*, vol. 206, no. 9, pp. 1557–1564, 2003.
- [6] C. T. Farley, J. Glasheen, and T. A. McMahon, “Running springs: speed and animal size,” *The Journal of Experimental Biology*, vol. 185, no. 1, pp. 71–86, 1993.
- [7] J. A. Vilensky, J. N. Libii, and A. M. Moore, “Trot–gallop gait transitions in quadrupeds,” *Physiology & Behavior*, vol. 50, pp. 835–842, 1991.
- [8] C. T. Farley and C. R. Taylor, “A mechanical trigger for the trot-gallop transition in horses,” *Science*, vol. 253, no. 5017, pp. 306–308, 1991.
- [9] R. Blickhan, “The spring-mass model for running and hopping,” *Journal of Biomechanics*, vol. 22, no. 1112, pp. 1217 – 1227, 1989.
- [10] A. Ruina, J. E. Bertram, and M. Srinivasan, “A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition,” *Journal of Theoretical Biology*, vol. 237, no. 2, pp. 170–192, 2005.

- [11] J. E. Bertram and A. Gutmann, “Motions of the running horse and cheetah revisited: fundamental mechanics of the transverse and rotary gallop,” *Journal of The Royal Society Interface*, vol. 6, no. 35, pp. 549–559, 2009.
- [12] M. D. Berkemeier, “Modeling the dynamics of quadrupedal running,” *The International Journal of Robotics Research*, vol. 17, no. 9, pp. 971–985, 1998.
- [13] J. P. Schmiedeler and K. J. Waldron, “The mechanics of quadrupedal galloping and the future of legged vehicles,” *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1224–1234, 1999.
- [14] P. Nanua and K. Waldron, “Energy comparison between trot, bound, and gallop using a simple model,” *Journal of biomechanical engineering*, vol. 117, no. 4, pp. 466–473, 1995.
- [15] H. M. Herr and T. A. McMahon, “A trotting horse model,” *The International Journal of Robotics Research*, vol. 19, no. 6, pp. 566–581, 2000.
- [16] ——, “A galloping horse model,” *The International Journal of Robotics Research*, vol. 20, no. 1, pp. 26–37, 2001.
- [17] S. Grillner and P. Wallen, “Central pattern generators for locomotion, with special reference to vertebrates,” *Annual review of neuroscience*, vol. 8, no. 1, pp. 233–261, 1985.
- [18] A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: a review,” *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [19] M. Golubitsky, I. Stewart, P.-L. Buono, and J. Collins, “A modular network for legged locomotion,” *Physica D: Nonlinear Phenomena*, vol. 115, no. 1, pp. 56–72, 1998.
- [20] S. H. Strogatz, “From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators,” *Physica D: Nonlinear Phenomena*, vol. 143, no. 1, pp. 1–20, 2000.
- [21] W. Wang and J.-J. E. Slotine, “On partial contraction analysis for coupled nonlinear oscillators,” *Biological cybernetics*, vol. 92, no. 1, pp. 38–53, 2005.
- [22] Q.-C. Pham and J.-J. Slotine, “Stable concurrent synchronization in dynamic system networks,” *Neural Networks*, vol. 20, no. 1, pp. 62–77, 2007.
- [23] K. Seo, S.-J. Chung, and J.-J. E. Slotine, “Cpg-based control of a turtle-like underwater vehicle,” *Autonomous Robots*, vol. 28, no. 3, pp. 247–269, 2010.
- [24] J. Seipel and P. Holmes, “A simple model for clock-actuated legged locomotion,” *Regular and chaotic dynamics*, vol. 12, no. 5, pp. 502–520, 2007.

- [25] I. Poulakakis, E. Papadopoulos, and M. Buehler, “On the stability of the passive dynamics of quadrupedal running with a bounding gait,” *The International Journal of Robotics Research*, vol. 25, pp. 669–687, 2006.
- [26] L. R. P. III and D. E. Orin, “Force redistribution in a quadruped running trot,” *IEEE International Conference on Robotics and Automation*, April 2007.
- [27] Y. Fukuoka, H. Kimura, and A. H. Cohen, “Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts,” *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, 2003.
- [28] C. Liu, Q. Chen, and J. Zhang, “Coupled van der pol oscillators utilised as central pattern generators for quadruped locomotion,” in *Control and Decision Conference, 2009. CCDC'09. Chinese*. IEEE, 2009, pp. 3677–3682.
- [29] L. Righetti and A. J. Ijspeert, “Pattern generators with sensory feedback for the control of quadruped locomotion,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 819–824.
- [30] C. Maufroy, H. Kimura, and K. Takase, “Integration of posture and rhythmic motion controls in quadrupedal dynamic walking using phase modulations based on leg loading/unloading,” *Autonomous Robots*, vol. 28, no. 3, pp. 331–353, 2010.
- [31] A. Sprówits, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, “Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot,” *The International Journal of Robotics Research*, vol. 32, pp. 932–950, 2013.
- [32] M. Raibert, *Legged robots that balance*. MIT press Cambridge, MA, 1986, vol. 3.
- [33] M. Railbert, K. Blankespoor, G. Nelson, R. Playter, and B. Team, “Bigdog, the rough-terrain quadruped robot,” in *Proceedings of the 17th World Congress The International Federation of Automatic Control (IFAC)*, 2008.
- [34] K. Yoneda, H. Iiyama, and S. Hirose, “Sky-hook suspension control of a quadruped walking vehicle,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 999–1004.
- [35] J. Estremera and K. J. Waldron, “Thrust control, stabilization and energetics of a quadruped running robot,” *The international Journal of Robotics Research*, vol. 27, no. 10, pp. 1135–1151, 2008.
- [36] S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. van de Panne, “Locomotion skills for simulated quadrupeds,” *ACM Trans. Graph.*, vol. 30, no. 4, p. 59, 2011.
- [37] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, “A control architecture for quadruped locomotion over rough terrain,” in *Robotics and Automation (ICRA), 2008 IEEE International Conference on*. IEEE, 2008, pp. 811–818.

- [38] A. Shkolnik, M. Levashov, I. R. Manchester, and R. Tedrake, “Bounding on rough terrain with the littledog robot,” *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 192–215, 2011.
- [39] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Learning, planning, and control for quadruped locomotion over challenging terrain,” *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.
- [40] D. Gong, J. Yan, and G. Zuo, “A review of gait optimization based on evolutionary computation,” *Applied Computational Intelligence and Soft Computing*, vol. 2010, 2010.
- [41] K. Chae and J. Park, “Trajectory optimization with ga and control for quadruped robots,” *Journal of Mechanical Science and Technology*, vol. 23, no. 1, pp. 114–123, 2009.
- [42] A. K. Valenzuela and S. Kim, “Optimally scaled hip-force planning: A control approach for quadrupedal running,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1901–1907.
- [43] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, “Control of dynamic gaits for a quadrupedal robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*.
- [44] R. Barzel and A. H. Barr, “A modeling system based on dynamic constraints,” *SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 179–188, 1988.
- [45] A. Witkin, M. Gleicher, and W. Welch, “Interactive dynamics,” *SIGGRAPH Comput. Graph.*, vol. 24, no. 2, pp. 11–21, 1990.
- [46] D. Baraff, “Coping with friction for non-penetrating rigid body simulation,” *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 31–41, 1991.
- [47] U. M. Ascher, H. Chin, L. R. Petzold, and S. Reich, “Stabilization of constrained mechanical systems with daes and invariant manifolds,” *Journal of Structural Mechanics*, vol. 23, no. 2, pp. 135–157, 1995.
- [48] K. Park and J. Chiou, “Stabilization of computational procedures for constrained dynamical systems,” *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 4, pp. 365–370, 1988.
- [49] W. Blajer, “Elimination of constraint violation and accuracy aspects in numerical simulation of multibody systems,” *Multibody System Dynamics*, vol. 7, no. 3, pp. 265–284, 2002.
- [50] M. Cline and D. Pai, “Post-stabilization for rigid body simulation with contact and constraints,” in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 3, 2003, pp. 3744–3751.

- [51] R. Weinstein, J. Teran, and R. Fedkiw, “Pre-stabilization for rigid body articulation with contact and collision,” in *ACM SIGGRAPH 2005 Sketches*. ACM, 2005, p. 79.
- [52] J. Baumgarte, “Stabilization of constraints and integrals of motion in dynamical systems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 1, no. 1, pp. 1–16, 1972.
- [53] S.-T. Lin and J.-N. Huang, “Stabilization of baumgartes method using the runge-kutta approach,” *Journal of Mechanical Design*, vol. 124, p. 633, 2002.
- [54] P. Flores, M. F. Machado, E. Seabra, and M. T. Silva, “A parametric study on the baumgarte stabilization method for forward dynamics of constrained multibody systems.” ASME, 2011.
- [55] A. Chatterjee and A. Ruina, “A new algebraic rigid body collision law based on impulse space considerations,” *Journal of Applied Mechanics*, vol. 65, pp. 939–951, 1998.
- [56] Y. Hurmuzlu and D. B. Marghitu, “Rigid body collisions of planar kinematic chains with multiple contact points,” *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.
- [57] M. Payr and C. Glocker, “Oblique frictional impact of a bar: Analysis and comparison of different impact laws,” *Nonlinear Dynamics*, vol. 41, no. 4, pp. 361–383, 2005.
- [58] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.
- [59] M. Posa and R. Tedrake, “Direct trajectory optimization of rigid body dynamical systems through contact,” in *Algorithmic Foundations of Robotics X*. Springer Berlin Heidelberg, 2013, vol. 86, pp. 527–542.
- [60] R. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, “A simply stabilized running model,” *SIAM Journal on Applied Dynamical Systems*, vol. 2, no. 2, pp. 187–218, 2003.
- [61] J. Seipel and P. Holmes, “Three-dimensional translational dynamics and stability of multi-legged runners,” *The International Journal of Robotics Research*, vol. 25, no. 9, pp. 889–902, 2006.
- [62] R. Blickhan, A. Seyfarth, H. Geyer, S. Grimmer, H. Wagner, and M. Günther, “Intelligence by mechanics,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 199–220, 2007.

- [63] R. Ringrose, “Self-stabilizing running,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1, 1997, pp. 487–493 vol.1.
- [64] S. Cotton, I. Olaru, M. Bellman, T. van der Ven, J. Godowski, and J. Pratt, “Fastrunner: A fast, efficient and robust bipedal robot. concept and planar simulation,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 2358–2364.
- [65] N. Hogan, “Impedance control-an approach to manipulation. i-theory. ii-implementation. iii-applications,” *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, 1985.
- [66] S. Seok, A. Wang, D. Otten, and S. Kim, “Actuator design for high force proprioceptive control in fast legged locomotion,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1970–1975.
- [67] E. Bizzi, N. Hogan, F. A. Mussa-Ivaldi, and S. Giszter, “Does the nervous system use equilibrium-point control to guide single and multiple joint movements?” *Behavioral and Brain Sciences*, vol. 15, pp. 603–613, 1992.
- [68] N. Hogan, “Stable execution of contact tasks using impedance control,” in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 1047–1054.
- [69] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: a cyber-physical systems approach*. Lee & Seshia, 2011.
- [70] M. W. Spong, J. K. Holm, and D. Lee, “Passivity-based control of bipedal locomotion,” *Robotics & Automation Magazine, IEEE*, vol. 14, no. 2, pp. 30–40, 2007.
- [71] D. Baraff, “Analytical methods for dynamic simulation of non-penetrating rigid bodies,” *SIGGRAPH Comput. Graph.*, vol. 23, no. 3, pp. 223–232, 1989.
- [72] M. T. Mason and Y. Wang, “On the inconsistency of rigid-body frictional planar mechanics,” in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, vol. 1, 1988, pp. 524–528.
- [73] J. B. Keller, “Impact with friction,” *Journal of applied Mechanics*, vol. 53, no. 1, pp. 1–5, 1986.
- [74] S. Kim, J. E. Clark, and M. R. Cutkosky, “isparawl: Design and turning for high-speed autonomous open-loop running,” *The International Journal of Robotics Research*, vol. 25, pp. 903–912, 2006.

- [75] D. V. Lee, J. E. Bertram, and R. J. Todhunter, “Acceleration and balance in trotting dogs,” *The Journal of Experimental Biology*, vol. 202, no. 24, pp. 3565–3573, 1999.
- [76] S. Jung, T. Hsia, and R. Bonitz, “Force tracking impedance control of robot manipulators under unknown environment,” *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 3, pp. 474–483, 2004.
- [77] J. K. Hodgins and M. Raibert, “Adjusting step length for rough terrain locomotion,” *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 289–298, 1991.
- [78] C. D. Remy, K. Buffinton, and R. Siegwart, “A matlab framework for efficient gait creation,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 190–196.
- [79] R. Blickhan and R. Full, “Similarity in multi legged locomotion: Bouncing like a monopede,” *Journal of Comparative Physiology A*, vol. 173, pp. 509–517, 1993.
- [80] V. A. Tucker, “The energetic cost of moving about: Walking and running are extremely inefficient forms of locomotion. much greater efficiency is achieved by birds, fishand bicyclists,” *American Scientist*, vol. 63, no. 4, pp. 413–419, 1975.
- [81] S. Seok, A. Wang, Chuah, M. Y. (Michael), D. Otten, J. Lang, and S. Kim, “Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 3292–3297.
- [82] T. A. McMahon, G. Valiant, and E. C. Frederick, “Groucho running,” *Journal of Applied Physiology*, vol. 62, no. 6, pp. 2326–2337, 1987.
- [83] T. McGeer, “Passive dynamic walking,” *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.
- [84] C. Maufroy, “Generation and stabilization of quadrupedal dynamic walk using phase modulations based on leg loading information,” Ph.D. dissertation, The University of Electro-Communications, 2009.