A CONTROLLER DESIGN FRAMEWORK FOR BIPEDAL ROBOTS: TRAJECTORY OPTIMIZATION AND EVENT-BASED STABILIZATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Pranav Audhut Bhounsule

May 2012

A CONTROLLER DESIGN FRAMEWORK FOR BIPEDAL ROBOTS:
TRAJECTORY OPTIMIZATION AND EVENT-BASED STABILIZATION

Pranav Audhut Bhounsule, Ph.D.
Cornell University 2012

This thesis presents a model-based controller design framework for bipedal robots that combines energy-efficiency with stability.

We start with a physics based model for the robot and its actuators. Next, the parameters of the model are identified in a series of bench experiments. Then we formulate an energy-optimal trajectory control problem. Our energy metric is the total cost of transport (TCOT) and is defined as the energy used per unit weight per unit distance travelled. We solve the trajectory control problem using parameter optimization software and an adequately fine grid.

To implement the energy-optimal solution on the physical robot, we follow a two part approach. First, we approximate the converged optimal solution with a simpler representation that sufficiently captures the optimality. The resulting walking gait is called the nominal trajectory. Second, we stabilize the nominal trajectory using an event-based, discrete, intermittent, feed-forward controller. Our stabilizing controller tries to regulate heuristically chosen quantities in a step, like step length or step velocity, doing feedback on a few key sensor data values collected at key points in a step.

Using this control framework our knee-less 2D 1 m tall 9.9 kg 4-legged bipedal robot, Ranger, achieved two feats: one, Ranger walked stably with a TCOT of 0.19, which is the lowest TCOT ever achieved by a legged robot on level terrain and, two, Ranger walked non-stop for 65 km or 40.5 miles without battery recharge or touch by a human, setting a distance record for legged robots.

BIOGRAPHICAL SKETCH

Pranav Bhounsule was born on 24 February 1983 in Panjim, Goa (India). Pranav did his Bachelors in Mechanical Engineering at the Goa College of Engineering from 2000 to 2004. Next, Pranav did his Masters in Applied Mechanics at the Indian Institute of Technology Madras from 2004 to 2006. In fall of 2006, Pranav enrolled in the PhD program in Theoretical and Applied Mechanics at Cornell University. Since January 2007, Pranav is a student at the Biorobotics and Locomotion Laboratory at Cornell University, researching legged robots.

To my grandmother

Sushila J. Bhounsule

and

to my parents

Sushama A. Bhounsule and Audhut J. Bhounsule

ACKNOWLEDGEMENTS

I would like to thank my advisor, Andy Ruina, for his support, advice, and enthusiasm that helped me make steady progress in this thesis. Most of the ideas in this thesis were his, while I sweated the details. I am indebted to Jason Cortell for keeping the robot Ranger - the bipedal platform on which the ideas in this thesis were implemented - in good health and for his constant efforts to improve its hardware and software.

I wish to thank Professors Richard Rand and Mark Campbell for serving on my doctoral committee. Special thanks to Professor Tim Healey for his advice and encouragement earlier in my graduate studies that helped me over the hump.

I would like to thank the Ranger team comprising of post-docs, graduate students, under-graduates, visiting scholars and visiting students for their individual contribution to the Ranger project (see list towards the end). In particular, a special thanks to these people. Carlos Arango built the motor test setup and helped in motor testing. Bram Hendriksen measured the stiffness of the cable used for feet actuation. Jason Cortell, Anoop Grewal and Nicolas Williamson helped in walk testing. Madhusudhan Venkadesan provided a mexed 8th order explicit Runge-Kutta integrator that speeded up the MATLAB optimizations.

I would like thank fellow lab-mates; Jason Cortell, Gregg Stiesberg, Anoop Grewal, Petr Zaytsev and Atif Chaudhry for their company. I am thankful to Cindy Twardokus and Marcia Sawyer for their help in administrative work and to Dan Mittler for his help in teaching experimental courses. Finally, I would like to thank my wife Arti for her company, her sacrifices, her support and her constant encouragement.

My research work was partly supported by a Harriet Davis Fellowship, National Science Foundation grant on Robust Intelligence and various assistantships at Cornell teaching math and mechanics courses.

Ranger Team, 2006 - 2011: Ranger's development is due to efforts of many, including lab manager: Jason Cortell; these graduate students: Kang An, Leticia Rojas-Camargo, Atif Chaudhry, J.G. Daniël Karssen, Anoop Grewal, Bram Hendriksen, Pulkit Kapur, S. Javad Hasaneini, Rohit Hippalgaonkar, Ko Ihara, Sam Hsiang Lee, Feng Shuai, Gregg Stiesberg, Kevin Tang, Andrey Turovsky, Nan Xiao, Petr Zaytsev; these undergraduate students: Carlos Arango, Steve Bagg, Megan Berry, Saurav Bhatia, Sergio Biagioni, David Bjanes, Kevin Boyd, John Buzzi, Amy Chen, Brian Clementi, Alexis Collins, Matt Coryea, Stephane Constantin, Thomas Craig, Violeta Crow, Mike Digman, James Doehring, Gregory Falco, Hajime Furukawa, Alex Gates, Matt Haberland, Katie Hartl, Phillip Johnson, Kirill Kalinichev, Avtar Khalsa, John Kuriloff, Andrew LeClaire, Dapong Boon-Long, Emily Seong-hee Lee, Reubens Lee, Ming-Da Lei, Jehhal Liu, Hellen Lopez, Emily McAdams, Lauren Min, Alexander Mora, Andrew Mui, Andrew Nassau, Joshua Petersen, Nicole Rodia, Satyam Satyarthi, Andrew Spielberg, Yingyi Tan, Chen Kiang Tang, Kevin Ullmann, Max Wasserman, Nicolas Williamson, Denise Wong; these visitors: Dong Chun, Chandana Paul, Amur Salim and Li Peng Yuan; and one high school student: Ben Oswald. Apologies for accidental omissions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Most of the content of this thesis has been submitted as a paper with an extended online appendix [8] to the International Journal of Robotics Research. Parts of the thesis that are not part of the above paper are marked with the symbol *.

We start by listing the various contributions of this thesis. Next, we review controller design approaches for bipedal robots and introduce our approach in context. Unlike most approaches, our control algorithm integrates energy-efficiency and stability under a common framework. We finish up the chapter with an outline of the remainder of the thesis.

1.1 Thesis contributions *

The thesis contributions are listed here.

1. **Controller framework that combines energy-efficiency with stability.** Past control approaches (see section 1.2 in this chapter for a literature review), like passive dynamic walking are based on energy-efficiency or others, like zero-moment point and foot placement type controllers, are based on stability. Some frameworks like model predictive control and optimal feedback control combine both, energy-efficiency and stability, but are computationally expensive. This thesis presents a control framework that combines energy-efficiency with stability while being computationally tractable (overview of our control framework is in chapter 2).

2. **Validation of control framework on experimental test-bed.** We show two successful demonstrations of our proposed control approach on the custom-built 2-D 1m tall knee-less biped, Ranger (see chapter 7). One, Ranger walked with a **T**otal **C**ost **O**f **T**ransport (TCOT is defined as energy used per unit weight per unit distance travelled) of 0.19 stably and this is the lowest TCOT ever achieved by any legged robot on level ground. Two, Ranger walked non-stop for 40.5 mi or 65 km on a single battery charge to set a legged robot distance record (beating the earlier record by a factor of 3).

3. **Stabilizing a system with time delays longer than the characteristic time scale of the system.** Using our control framework and with a controller bandwidth artificially constrained to 0.5 second, we demonstrate balancing of a simple inverted pendulum with a characteristic time scale of 0.32 second (see chapter 2, section 2.3).

4. **DC Motor and gearbox model.** We present a DC motor and gearbox model (see chapter 3, section 3.3) and systematic experiments (see chapter 4, section 4.2) to estimate the parameters of the model of Ranger. In particular, we found two non-standard terms in the motor modeling: one, a brush contact resistance; and two, a load-dependent friction term that we modeled as a current dependent friction term.

5. **Energy-based control of walking.** Using a simple 2D point-mass model of walking and using step length control and ankle push-off control to modulate the mid-step kinetic energy, we demonstrate substantial increase in the basin of attraction of walking (see chapter 6).

6. **Benchmarks for passive dynamic walkers.** We provide two passive dynamic walker benchmarks: the rimless wheel and the simplest walker (see appendix B). These benchmarks can be used to validate various aspects of passive dynamic

models of locomotion like verifying equations of motion, the root finder and Jacobian of the linearized map.

7. **Benchmarks for optimal trajectory control for legged robots.** We present two optimal trajectory control problem benchmarks: passive dynamic walking and optimum level ground walking (see appendix D). These benchmarks can be used to check the proficiency of the optimization software on optimal trajectory control problems for legged robots.

1.2 Past control approaches

Here we review past control approaches. Our control approach is reviewed at the end of this section.

1.2.1 Passive dynamics

One approach to energy-effective control is based on purely mechanical periodic gaits, so-called passive dynamics, e.g., [28, 51, 70, 79]. A strictly passive-dynamic robot is a linkage with no sensors and no motors that can step stably down slight slopes. As a passive-dynamic robot 'ramp walker' moves down a slope γ, the mechanical energy lost to friction and collisions is recovered by the decrease in gravitational potential energy.

Cost of transport. One measure of effectiveness penalizes power use and gives credit for weight and speed:

$$\text{Cost of transport} = \frac{\text{power consumption}}{\text{weight} \times \text{speed}}$$

This measure (based on weight and not mass) is dimensionless (W/(N m/s) = 1). The smaller the cost of transport the more energy-effective the locomotion.

There are different costs of transport depending on what power is included. For a human walking, the total cost of transport, accounting for the full food energy used by a person as they walk, is about 0.3, e.g., [3, 13, 33]. However, an often-reported cost of transport for people of 0.2 is based on subtracting the resting metabolic cost, the energy a person uses to stand still. Finally, one can estimate a mechanical cost of transport (MCOT) based on the total positive work done by the muscles or actuators (and not subtracting out the negative work). This is about MCOT $\approx 0.05 = 0.2 \times 25\%$ for humans because muscles are about 25% efficient (work is about 25% of chemical energy used in humans [69]). For a passive-dynamic robot the energetic cost of transport (TCOT = total power used per unit weight and speed) is $\sin\gamma \approx \gamma$, e.g., [40]; a machine that walks down a slope of $\gamma = 0.05$ rad (3°) has a TCOT of $\sin\gamma \approx 0.05$. For these robots the mechanical cost of transport (MCOT), the actuator work per unit weight and distance, is the same as the TCOT because all of the gravitational energy is supplied as mechanical work. Typical passive-dynamic ramp walkers happen to use about the same amount of gravitational work as is performed by the muscles of a human walking on level ground (MCOT ≈ 0.05).

The first passive-dynamic robot to have a major impact on robotics was McGeer's "4-legged biped" which had 4 side-by-side legs with knees and no upper body. For most analysis purposes, this is a two-legged machine living in 2 spatial dimensions [70]. Despite the non-anthropomorphic leg layout, McGeer's quadruped had a gait that was inspirationally evocative of human walking. McGeer's 2-D concept was extended to 3-D by Collins et. al. [28]; they built a true two-legged passive dynamic walker which successfully demonstrated downhill walking.

Stability of purely passive-dynamic, or passive-dynamic-based powered robots is less encouraging, however. Although stability of several passive-dynamic robots has been numerically predicted by non-linear simulations, there is no qualitative analytic theory of passive-dynamic stability beyond noting some contributing mechanisms: 1) dissipation (e.g. the rimless wheel [22]); or 2) the non-holonomic nature intermittent foot contact [91]; or 3) the static stability of the splayed standing configuration that is intermittently visited in the walking cycle (as discussed in [25]). Thus there are no analytic recipes for enhancing stability, and so far there are no promising iterative numerical approaches either.

Stability of passive-dynamic robots. A primitive, but at least objective (not coordinate-system-dependent), measure of stability is given by the magnitude of the biggest eigenvalue of the Jacobian of the step-to-step map (McGeer's stride function) of the periodic cycle of the walker [70, 99]. The magnitude of this eigenvalue indicates how fast the disturbances would grow or shrink as the walker takes multiple steps after a small disturbance away from a periodic motion. If the biggest (possibly complex) eigenvalue has magnitude less than one, then the periodic cycle is stable. By this measure, if all eigenvalues are far inside the unit circle on the complex plane (far less than one in magnitude), then the robot is very stable. Typical passive-dynamic walkers are only mildly stable at best by this measure, with their biggest eigenvalues rarely less than about 0.6 [24, 41].

In the lab, the behavior of even the best passive-dynamic robots has been erratic and fussy. Similarly, passive-based powered walkers (see next section for review) which rely on passive dynamics for stability are also fussy, in our experience, and especially so in three dimensions. That passive-dynamic robots can be stable at all has been great for physical demonstrations, has made great videos, but has perhaps mis-inspired some into

pursuing passive strategies for stabilizing motorized robots. At present there seems to be little evidence that passive strategies can have anywhere near the reliability needed for practical robotics or for predicting the observed stability of walking humans.

1.2.2 'Powered' passive dynamics *

Inspired by the simplicity of passive dynamic walkers, there have been attempts to realize passive dynamic walking on level ground by adding a source of power while preserving the passive dynamics. Camp [15], in simulations, added ankle actuation to a 2-D knee-less passive dynamic walker. The ankle motors were turned on to a prescribed voltage during a prescribed time in the walking cycle. As parameters were varied, this walker exhibited stable and unstable limit cycles, period doubling and chaos as previously observed in fully passive walkers [41, 44]. Collins and Ruina [27] built a 3-D bipedal robot, the 'Collins' walker, that walked successfully on level ground. The Collins walker had a passive hip and powered ankles. During the single support phase, the ankle motor loads up an ankle spring. The ankle spring is released once the swinging foot hits the ground, thus generating an ankle push-off that powers walking. Robots based on passive dynamics are quite energy-effective. For example, Collins 12.7 kg walker used only 12 W to walk at 0.44 m/s on level ground. It had a TCOT of 0.2, which is two thirds of the power of a human scaled for weight and speed and apparently lower than that of any motor-driven legged robot before or since, with the exception of Ranger described here. However, stability-wise these robots have not been any better than the passive dynamic walkers on which they are based [48, 49].

1.2.3 Zero Moment Point

One prominent class of control ideas focuses on the position of the Zero Moment Point (ZMP), the point on the ground where the reaction force and couple have no horizontal moment component. In 2D, this is the point where the net reaction is a force with no couple (the so-called center of pressure, COP). ZMP controllers focus their attention on choosing ankle torques to keep the ZMP inside the foot contact polygon, thus keeping the foot flat on the ground [105, 106]. For standing still, balance of such robots is attained primarily by manipulating the robot center of mass (COM) location with the ZMP, in effect chasing the COM towards the center of the support polygon. For walking, the foot placement must be such that the ZMP can be kept inside the foot-contact polygon while the robot COM is moving on or near a desired trajectory. Although these robots may use foot placement in their balance control, the underlying principle is that of balance by ankle torques.

Robots that use ZMP control for walking, most famously Honda's ASIMO [93] series, seem to have various characteristic attributes: they walk with bent knees that allow the controllers to have authority over all the upper body degrees of freedom; they have flat-bottomed feet, and they consume lots of energy, perhaps because all the robot joint angles are carefully controlled at all instances of time. The TCOT of ASIMO in 2005 was estimated (from battery capacity, speed, weight and time) to be about 3.2 [26], which is 10 times the TCOT of a typical human.

1.2.4 Linear inverted pendulum based control *

Generally one thinks that linear systems are easier to understand and control than non-linear systems. This motivated Kajita to propose the 2-D linear inverted pendulum

model of walking. The idea is to use knee torques in single support phase to constrain the hip to move in a straight line [55, 56, 57] and hence the name, linear inverted pendulum based control. The resulting equation of motion are linear and one can use standard linear control theory to do control. Later, Kajita extended the linear inverted pendulum idea to control a three-dimensional robot [53, 54]. Because of the linearity of the equations, even in 3D, the sagittal motion is decoupled from the side-to-side motion. Thus he could apply the linear inverted pendulum control method separately to both, the lateral and fore-aft balance.

Calculations with a point-mass model and with work-based cost suggest the lack of energy optimality of such smooth level walking [92, 96]. This non-optimality has been confirmed through human experiments [78]. As this method of control relies on using large ankle torques, there is the possibility of the robot's overturning as the center of mass of the robot leaves the foot support polygon. The latter issue is mitigated by combining ZMP with the linear inverted pendulum walking by using preview control of ZMP [52]. The preview controller looks at the future reference ZMP and modifies current inputs ahead of time to do smooth tracking and thus preventing the overturning of the robot.

1.2.5 Balance by foot placement

Some more-dynamic feedback-controlled robots have had balance control based almost entirely on foot placement, with little or no thought of balance by reaction torques acting on flat feet. The best known of these are from Marc Raibert's MIT lab and his company Boston Dynamics. Originally these robots were 2D single-leg hopping robots with control based on the observation that hop height, forward speed and body orientation could

all be controlled by control of leg angle and leg length at appropriate flight or contact times [88]. These ideas were extended to 3D and multiple legs, e.g., [86]. Recently, balance based on foot placement has been used to make what seems to be a highly-reliable true 3D biped walker [32]. Assuming PETMAN weighs about 1000 N, moves at about 2 m/s and consumes about 10,000W (about 13.5 hp) of hydraulic pump power, it has a TCOT of about 5 (about 16 times that of a walking person).

One approach to foot placement is to step into an N-step capture region [60, 84], where a step can be taken with the knowledge that the robot will be able to come to a stop in n steps or fewer if desired. Speed control can be performed by stepping to a spot relative to the instantaneous capture point and influencing the dynamics of this point through moving the center of pressure on the foot. These techniques to date have utilized simple inverted pendulum models of walking in order to reduce computational complexity and applied on physical robots [34, 83].

1.2.6 Controller representation with neural nets and tuning via evolutionary algorithms or learning *

Artificial neurons (a computational analog of biological neurons) encode information and are linked to other neurons to make up a neural net. Typically, using neural nets, one defines a controller that maps the sensor inputs to the actuator outputs with tunable weights. Various performance measures can than be optimized by tuning the weights of the neural nets using either a learning algorithm or heuristic optimization either in simulation or on the physical robot. For example, Solomon et. al [95], in simulation, used evolutionary algorithms to minimize the mechanical cost of transport (defined as the positive actuator work done per unit weight per unit distance travelled). Paul [82],

also in simulation, used genetic algorithms to maximize the distance travelled by the biped. Manoonpong [67, 68] implemented a two-level neural net to do adaptive walking for the bipedal robot Runbot. At the lowest level in Runbot was a controller that used the joint angles and joint velocities as inputs while at the highest level was an adaptive controller that used an infra-red sensor that monitored the ground slope. The weights of the neural net were tuned by a learning algorithm.

Mostly, it is hard to extract any meaningful message from these optimized neural nets. Also, a basic problem with physical implementation is that the learning generally involves falling. While transferring well-working simulated results to physical robot is a possibility, a well modeled robot would be vital for such an approach to work successfully.

1.2.7 Central Pattern Generator *

Central Pattern Generators (CPG) are neural nets that generate rhythmic patterns without any feedback or control from higher control centers like the brain. There is some evidence of humans using CPG's to control walking [29] and this has inspired CPG based bipedal control. In CPG's, locomotion is thought to be an emergent behavior of coupled oscillation of the neurons and mechanics; a view-point quite similar to the passive dynamic paradigm. In simulations, CPG's have been used to control a 2-D bipedal robot model by Taga [100, 101] and a 3-D bipedal model by Righetti and Ijspeert [89]. It is likely that CPG based robots are not more stable than their passive dynamics counter parts unless supplemented by feedback control.

1.2.8 Optimal trajectory control *

In optimal trajectory control, one is interested in optimizing for one particular robot behavior (for example steady walk with minimal energy, walk at certain speed). Given a model of the robot, one finds control values as a function of time that minimize a given cost metric and which generates the desired robot behavior like for example, steady walking. Some common cost functions used are: the integral of torque squared [6, 7, 18, 20, 76]; impulse squared if the actuators can provide impulses; work based [20]; or a combination of these [14, 17, 90].

These optimal control problems are converted to a parameter optimization problems by discretizing the controls or the robot kinematics. Some common parameterization schemes include: controls parameterized as piecewise linear function of time [76, 90]; joint angles parameterized as truncated fourier series [14]; polynomial functions of time [17, 20, 80]; cubic splines [7]; bezier polynomials [108]. The resulting parameter optimization problem is solved by using some variant of Newton's method [17, 76] or using Monte-Carlo methods like genetic algorithms or simulated annealing [14, 80].

Optimal trajectory control is not concerned with stability. The stability depends on the particular representation chosen. For example, a trajectory will have a different stability depending on whether it uses time or one or other state variables as the independent variable in the controller. One could add stability into the cost metric or as an optimization constraint. For example in Mombaur et. al. [74, 75], the biggest eigenvalue is bounded by specifying it as an optimization constraint. A problem here is that the eigenvalues are sometimes non-smooth and this could hurt the rate of convergence of the optimization software, particularly those based on gradient methods.

1.2.9 Model predictive control *

In model predictive control (MPC) [39], optimal trajectory control is done on the fly in real-time. In MPC, at every state sampling instance, an optimization algorithm is formulated over a specified time period (moving horizon control) and solved online. Only the first step of the control policy is implemented and the state is sampled again. Constraints like actuator limits, joint angle limits, obstacles can be seamlessly incorporated in the optimization. As planning is done based on constraints and is done quite frequently, robot stability is implicitly integrated into this scheme. Also, by sampling the system and planning over a short time scale, one could potentially use such a scheme to do walking over rough terrain. MPC has so far been demonstrated in walking simulations [4] and on a simple 2 DOF robot [65]. With the advent of faster computers and faster optimization algorithms, such an approach looks promising.

1.2.10 Optimal feedback control *

Optimal feedback control [5], unlike optimal trajectory control, is concerned with finding optimal control policies for every conceivable start point to every conceivable goal states. The problem is typically solved by discretization. Consider a n dimensional state space with each dimension discretized with a grid size of g. The discrete version of the state space has a total of g^n grid points. To solve the optimization problem, one computes a value function which is associated with the optimal cost and the best strategy at each of this grid points. Thus, optimal feedback control finds the optimal way to move from one point in state space to another and thus stability is integrated into the problem solution. However, this method suffers from the curse of dimensionality. For example, for a 6 dimensional state space with a grid size of 10, one would need about 10^6 (1

million) numbers to store the optimal solution besides the added burden of computing the value function for this grid. This approach has so far been demonstrated only in simulations [66, 109].

1.2.11 Library of trajectories *

Optimal feedback control is computationally intractable for higher dimensions but provides a global optimum and incorporates stability. On the other hand, optimal trajectory control is computationally tractable but is concerned with only a single trajectory (it is a local method) and does not account for stability. A method which combines the advantages of both the above methods while offsetting the negatives, is to control the system via a library of optimal control trajectories that are solved using optimal trajectory control and have been either verified to be stable or stabilized by a linear controller. We discuss two implementations.

Atkeson and Morimoto [2] used optimal trajectory control to generate several key trajectories. Using ideas from dynamic programming, a policy and a value function for these key trajectories were estimated and updated. Tedrake et. al. [103] developed a linear stabilizing controller based control algorithm to stabilize the system over large regions of state space. One first starts by computing one optimal trajectory. Next, one picks random point in state space and finds a local linear feedback controller that enables one to get from the selected random point to the optimal trajectory. Finally, one estimates the basin of attraction or stable region of these local controllers. Proceeding in this fashion, one tries to fill the full state space with local linear controllers that are guaranteed to be stable. Though these approaches look promising, there has yet to be a physical implementation using such techniques.

1.2.12 Non-linear stabilization of pre-computed trajectories *

Control theorists have traditionally been interested in computing stabilization laws that stabilize pre-computed trajectories. This is a two stage approach. First, one computes nominal trajectory using optimal trajectory control or using one or another intuitive control scheme. Second, one finds feedback control laws that enable one to track these nominal trajectories.

Two common approaches are feedback linearization/computed torque control and sliding mode control. In feedback linearization/computed torque control [35, 72, 81], the stabilizing law has two parts, a part that cancels the non-linear terms like gravity and Coriolis forces and another proportional-derivative part that ensures that the system track the nominal trajectory. In a sliding mode control, the system is forced to slide along a hyper-plane in the state space using a controller that switches based on where the system is in the state space [16, 87]. Such controllers are probably not energy-efficient as they are based on canceling the natural robot dynamics, rather than working with them. Also trying to follow pre-computed trajectories exactly might lead to system chatter especially when the gains of the controller are high. Note that for these control schemes to work, the system needs to be fully actuated.

1.2.13 Hybrid zero dynamics

Hybrid zero dynamics (HZD) [46] is one of the most coherent approach towards developing a systematic control framework and that combines energy-efficiency with stability and was used by Westervelt [107] to control the 2-D robot Rabbit [19]. The central idea is to tightly control all internal degrees of freedom of the robot so as to effectively eliminate them as independent degrees of freedom; they are all slaved to the motion of the

uncontrolled ankle joint angles according to functions (the slave joints' angles are functions of the free ankle-joint angles) whose specification *is* the control. These functions can be chosen to minimize this or that cost function (say, energy use, [19, 107]). In HZD, stabilization occurs based on the interplay between energy lost on impact and energy gained during stance, thereby reaching a stable speed after a handful of impacts.

In HZD, stability can be increased three ways [21]. One is to redo the optimization with the same functional representation but adding a measure of stability to the optimization, using for example the biggest eigenvalues as a constraint or as a part of the objective function. A second less formal approach to the same idea is to change the functional representation, carry out the optimization again, check the stability of the resulting solution, and keep iterating till one finds a stable solution. A third way is to stabilize the system using an additional discrete event-based feedback controller [45]. While the first two of these seem similar to passive dynamics in philosophy and in practice, with the largest eigenvalues at about $0.7 \gg 0$, the event-based feedback controller could, in principle, confer much more stability.

There are a few potential issues: 1) HZD depends on high bandwidth control of the slaved degrees of freedom, and in practice such high-bandwidth, high-gain control seems to be energy consumptive, even when the pre-calculated mechanical work is small; 2) HZD depends on having a machine that is, after the HZD joint-position control is implemented, not compliant and thus perhaps not appropriately yielding to physical disturbances; and 3) HZD is perhaps not satisfyingly biomimetic in that it imposes tight control of possibly unimportant degrees of freedom, thus violating biologically-relevant ideas associated with, say, the uncontrolled manifold or with optimal control [63, 104]. Although HZD has not yet been used to make a robot with a low TCOT, the HZD approach still has potential to provide both stability and low energy use.

1.2.14 Control framework in this thesis

Our overall control design approach is similar to that of Miura and Shimoyama [73] (incidentally, Miura and Shimoyama were early, maybe only second to Formalsky [36] in discussing walking as a Poincaré map). Their controller was made up of two parts: an open loop time-based trajectory planner and a feedback controller to stabilize the nominal trajectory. Their stabilizing linear feedback controller used the measurements at the beginning of the step to drive the robot state at the end of the step to its nominal value. That is, instead of tracking a trajectory in the gait cycle, their controller tried to regulate the state only at the end of the step. The gains for the linear controller are calculated by doing a step-to-step eigenvalue calculation. Where the HZD and ZMP approaches constrain out most degrees of freedom at all times, the Miura and Shimoyama approach only worries about them once per step. We add two small changes to this control idea: 1) the minimization of an energy metric as the performance criterion for the nominal trajectory; and 2) allowing control at multiple times during a step using different control actions and control goals in each interval.

The framework was chosen so as to have various general features:

- it should allow simple implementation of simple controllers such as the Collins one-sensor-measurement-per-step controller [27] (e.g. see chapter 7);
- it should be able to implement intuitive control constructs (e.g., of the Raibert hopper type) (e.g. see chapter 7);
- it should gracefully handle sensor delays (e.g. see chapter 2, section 2.3);
- it should be able to come arbitrarily close to any continuous non-linear multi-variable feedback policy (e.g. see chapter 5);

- it should be of a form so that it has a relatively simple expression for a controller that is good enough (e.g. see chapter 5).

We found and implemented a control architecture that has these features. It is reflex-based; there are triggers (thresholds in dynamic variables or in elapsed time) and responses (motor programs). It is intermittently feed-forward in that there is no feedback (but for local motor-control feedback) during the motor programs that run between triggers. The control essentially does discrete trajectory tracking, but it is not based on an approximation of a continuous controller. It is not impulsive (and can be smooth). There is no tight control over any aspect of the robot pose or balance.

1.3 Outline of the remainder of the thesis *

In chapter 2, we present our model-based controller design algorithm and elaborate on some details. Next, we present a hi-fidelity model of the robot and its actuators (chapter 3). The parameters of the model are identified in a series of bench experiments (chapter 4). Our control design takes place in two stages. First, using optimal trajectory control and the hi-fidelity model, we find the nominal trajectory and approximate it (chapter 5). Second, we stabilize the nominal trajectory using a stabilizing controller. We motivate our stabilizing controller using a point-mass model of the robot (chapter 6). Next, we derive the stabilizing controller for the hi-fidelity model using the point-mass model idea, followed by implementation on the robot (chapter 7). Conclusions follow in chapter 8.

In appendix A, the equations of motion for robot are derived using Newton-Euler equations. In appendix B, two passive dynamic walking benchmarks are provided. These benchmarks help us to check and validate the equations of motion for the robot

in a limited sense. In appendix C, smoothings for various non-smooth functions used in this thesis are presented. In appendix D, the model and optimization software are checked against two legged robot walking benchmarks. In appendix E and F, we provide details on solving the energy-optimal control problem presented in chapter 5. Finally, in appendix G, we present the finite state machine that was used for control on the physical robot.

CHAPTER 2

CONTROLLER DESIGN FRAMEWORK *

Our controller design algorithm is as follows,

Stage 1 - Modeling: Define a physics based model for the robot and the actuators (chapter 3) and do a system identification to identify the parameters of the model (chapter 4).

Stage 2 - Trajectory Generator: Formulate and solve an optimal trajectory control problem (e.g. minimize energy per unit distance travelled) to get a sense of the optimal solution. Next, approximate the optimal control solution found earlier (chapter 5). See section 2.1 for more details.

Stage 3 - Stabilizing Controller: Stabilize the trajectory in step 2 using an event-based, discrete, intermittent, feed-forward controller (chapter 6 and chapter 7). See section 2.2 for more details.

In our controller framework, the control input U (e.g. torque, current, voltage, pressure) can be expressed as,

$$U = U_{\text{trajectory-generator}} + U_{\text{stabilizing-controller}}$$

The above form of control decomposition into a trajectory generator and stabilizing controller is not new in the controls community. However, the novelty here is the finer details of how the trajectory generation and stabilization is done. We elaborate on these in the next two sections.

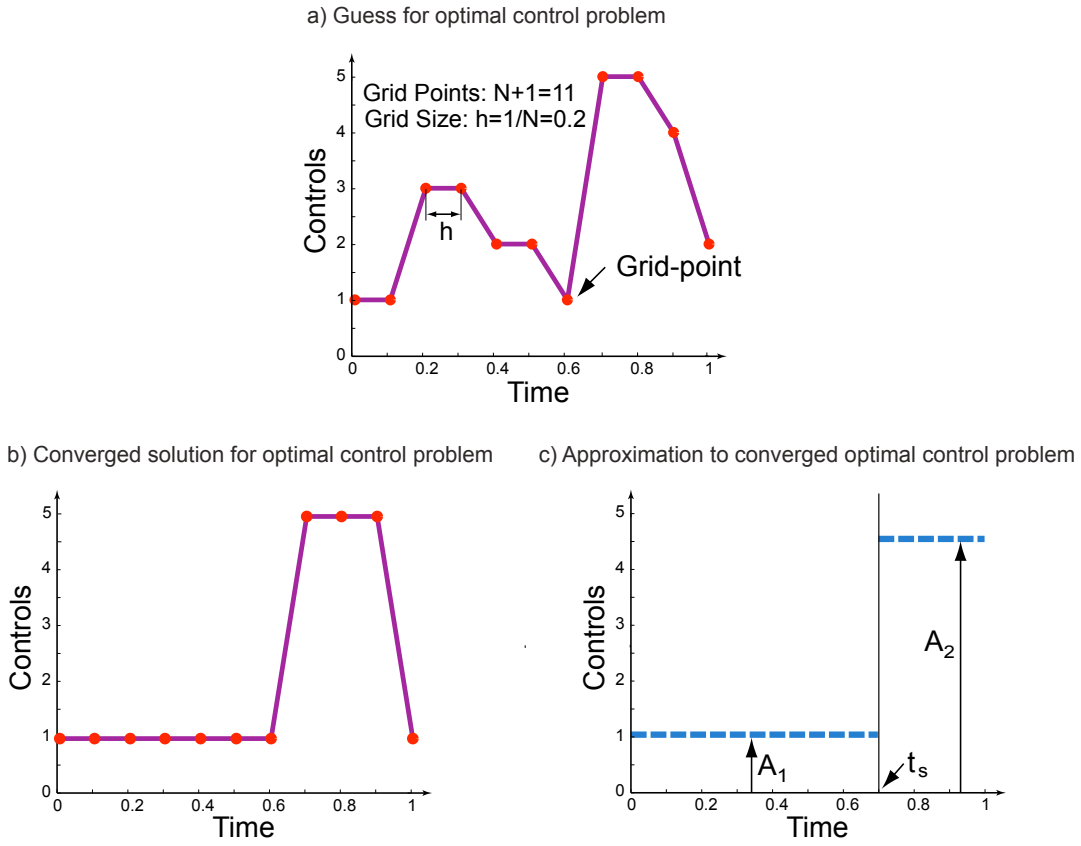

Figure 2.1: **A hypothetical example to illustrate the trajectory generation process.** (a) A piecewise linear control parameterization with grid size $N = 10$ involves $N + 1 = 11$ control parameters. This randomly chosen set of control values are used as a guess for the parameter optimization software. The grid size is $h = 1/N = 0.2$. (b) The converged solution (after running the optimization) for the grid size $N = 10$. (c) Approximation of the converged optimal control parameterization in b using three parameters; two amplitudes (A_1 and A_2) and a switching time (t_s). The switching of amplitudes is time based here, but could be made state based.

2.1 Trajectory Generator *

The trajectory generation proceeds in two stages. First, using a parameter optimization software and using fine grid size we solve an energy-optimal control problem to estimate the optimal cost. Second, informed by the converged solution we find an approximate

coarse-grid control parameterization that captures the essential structure of the optimal solution. We illustrate these two steps with a fictitious example.

Consider an optimization problem involving minimizing a certain cost with one actuator. To solve this optimization problem, we parameterized the controls as a piecewise linear function of time with grid size $h = 1/N = 0.2$ (see figure 5.2a). This implies that we have $N + 1 = 11$ control parameters at the grid points that can be tuned by the optimization program. Figure 5.2a shows our initial guess for the optimization. The optimization algorithm then finds the values of control variables at the grid points that optimizes the given cost. We repeat this process for increasing grid sizes (e.g. N, $2N$, $3N$ etc.), until the cost does not change appreciably between two successive grid sizes. Let us assume that at $N = 10$ we are sufficiently close to the energy-optimal solution and that further increase in N will give minimal improvement in the optimum. Figure 5.2b shows the converged solution for the grid size $N = 10$.

Next, we try to find a simple coarse-grid approximation to the optimal control solution obtained in 5.2b. In figure 5.2c, we have used three parameters, two amplitudes (A_1 and A_2) and one switching time (t_s), to represent the optimal solution obtained in 5.2b. We solve the optimal control problem again with this simple coarse-grid approximation of three parameters as our initial guess. We refine the approximation by adding parameters or changing the functional representation till the cost is within some percentage of the optimal cost obtained from the grid size $N = 10$ in 5.2b. The goal of this iteration is to maximize the simplicity of the coarse-grid control parameter representation, typically at a slight increase in the cost. On the physical robot, we implement this coarse-grid solution of the optimal control problem.

2.2 Stabilizing controller *

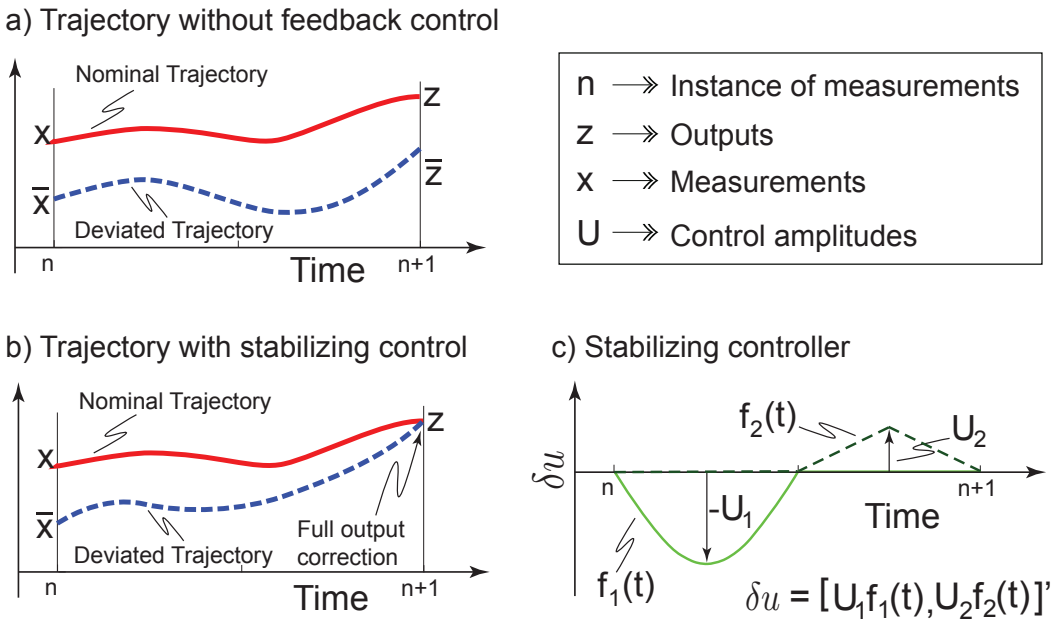

Figure 2.2: **Event-based, discrete, intermittent, feed-forward stabilizing control.** A schematic example. (a) Shows the nominal (solid red = periodic optimal) and deviated (dashed blue = disturbed by modeling errors, sensor errors or physical disturbances) trajectory for some dynamic variable x of interest which is measured at the start of a continuous interval, namely at section n. This is a generalized state in that it may contain redundant information such as average speed over the whole previous step. The goal of the stabilizing controller is to minimize the output variable error at the end of the next interval. (b) Shows the new deviated trajectory after switching on the feedback control show in c). (c) The feedback motor program has two control actions: a sinusoid for first half cycle and a hat function for the second half of the cycle. The amplitudes U_1 and U_2 of the the two functions are chosen at the start of the interval depending on the error $(x - \bar{x})$. By a proper choice of the amplitudes U_1 and U_2 deviations might be, for example, fully corrected in one step giving a 'deadbeat' controller, as shown in b).

The role of the stabilizing controller is to bring the robot back to its nominal trajectory. Commonly, trajectory tracking uses a high bandwidth, high gain continuous

control along the trajectory [37, 38, 47, 59, 64]. In our stabilizing control we try to track some key variables at key points of the trajectory, using state estimates only at those points, and try to track those key variables using as little sensor and little command bandwidth as possible.

We illustrate the idea with a schematic example. Consider the nominal trajectory of a second-order system shown as a solid red color line in figure 2.2. Let n and $n+1$ be instances of time at which we are taking measurements from sensors. The time interval between the measurements n and $n+1$ is typically on the order of the characteristic time scale of the system (say, leg swing time) and not the shortest time our computational speed allows. Let us assume that we take two measurements, $x = [x_1 \;\; x_2]'$ at time n (e.g., a position and velocity). We are interested in regulating two outputs: z_1 and z_2 at time $n+1$.

Due to external disturbances, the system has deviated from its nominal trajectory. This trajectory is shown as a dashed blue color line in figure 2.2a. Now, the sensors read $\bar{x}$ at time n. In the absence of any feedback correction, the output values would become $\bar{z} = [\bar{z}_1 \quad \bar{z}_2]'$.

The stabilizing controller measures deviations at time n ($\delta x_n = x - \bar{x}$) and uses actuation to minimize the deviations in output variables ($\delta z_{n+1} = z - \bar{z}$). For illustration we choose two control actions, $\delta u_n = [U_1 f_1(t) \;\; U_2 f_2(t)]'$, a half sinusoid and a hat function, each active for half the time between time $n+1$ and n. This is shown in figure 2.2c. We adjust the amplitudes of the two control functions U_1 and U_2, based on measured deviations δx_n, to regulate the deviated outputs δz_n. For example, with a proper selection of the amplitudes of the two functions it is possible to fully correct the deviations in the output variables, as shown in figure 2.2b.

We linearize the system about the nominal trajectory (actually, we only linearize the section to section map). The sensitivities of the dynamic state to the previous state and the controls $\delta U_n = [U_1 \;\; U_2]'$ are: $\mathbf{A} = \partial x_{n+1}/\partial x_n$, $\mathbf{B} = \partial x_{n+1}/\partial U_n$, $\mathbf{C} = \partial z_{n+1}/\partial x_n$ and $\mathbf{D} = \partial z_{n+1}/\partial U_n$. Thus we have

$$\delta x_{n+1} = \mathbf{A}\delta x_n + \mathbf{B}\delta U_n \tag{2.1}$$

$$\delta z_{n+1} = \mathbf{C}\delta x_n + \mathbf{D}\delta U_n. \tag{2.2}$$

Again, the x are a list of measured deviations, the z are a list of deviations which we wish to control, the U are the activation amplitudes (1 or 2 of them).

Two ways of using the above equations to derive a stabilization controller are: pole placement and discrete linear quadratic regulator (DLQR). We discuss these next.

1. Pole placement: We start by assuming a linear controller $\delta U_n = -\mathbf{K}\delta x_n$, where $\mathbf{K}$ is a constant gain matrix and substitute this in equation 2.2 to get,

$$\delta z_{n+1} = (\mathbf{C} - \mathbf{DK})\delta x_n \tag{2.3}$$

The goal of pole placement is to choose the gain $\mathbf{K}$ so that eigenvalues of the closed system $(\mathbf{C} - \mathbf{DK})$ are placed at the desired positions. For example to place the eigenvalues at the origin, making a dead-beat controller, the gain matrix would be, $\mathbf{K} = \mathbf{D^{-1}C}$.

2. Discrete linear quadratic regulator (DLQR): In DLQR [77], we seek to minimize the cost function J_{dlqr} defined as,

$$J_{\text{dlqr}} = \sum_{n=0}^{n=\infty} \left({\delta z_{n+1}}^T \mathbf{R}_{zz} \delta z_{n+1} + {\delta U_n}^T \mathbf{R}_{UU} \delta U_n \right), \tag{2.4}$$

where $\mathbf{R}_{zz}$ and $\mathbf{R}_{UU}$ are diagonal matrices that weight the different components of δz_{n+1} and δU_n. The weights $\mathbf{R}_{zz}$ and $\mathbf{R}_{UU}$ are design parameters picked to give reasonably

fast return to nominal values (for example more than 50% return within one step for walking) but without unduly high gains (which tend to lead to motor currents that are beyond safety limits).

Putting equation 2.2 in 2.4 and re-arranging gives

$$J_{\text{dlqr}} = \sum_{n=0}^{n=\infty} \left(\delta x_n{}^T \mathbf{Q} \delta x_n + 2\delta x_n{}^T \mathbf{N} \delta u_n + \delta U_n{}^T \mathbf{R} \delta U_n \right), \tag{2.5}$$

where $\mathbf{Q} = \mathbf{C}^T \mathbf{R}_{zz} \mathbf{C}$, $\mathbf{N} = \mathbf{D}^T \mathbf{R}_{zz} \mathbf{C}$ and $\mathbf{R} = \mathbf{D}^T \mathbf{R}_{zz} \mathbf{D} + \mathbf{R}_{UU}$. The solution to this optimization problem is linear state feedback

$$\delta U_n = -\mathbf{K} \delta x_n, \tag{2.6}$$

where the gain $\mathbf{K}$ is obtained by solving the Ricatti equation [77] which we do using the MATLAB control system toolbox (DLQR).

Some issues in the design of this controller (pole placement as well as DLQR) include:

- selecting a suitable section or instance of time to take measurements - this should be a time when the dynamic-state estimation is reasonably accurate, and also a time when dynamic-state errors which cause failure are evident;
- selecting measurement variables (x_n) that are representative of system failure;
- picking output variables (z_n) that also correlate with system failure; and
- picking actuator shape profiles that have large and relatively independent effects on the target variables.

2.3 Example of stabilizing controller: balancing a simple inverted pendulum *

We present a low bandwidth control of a simple inverted pendulum using our stabilizing controller idea presented in section 2.2. Our controller bandwidth is one half of a second and is slower than the characteristic time scale of one third of a second of the simple inverted pendulum. The low bandwidth controller presented here can also be interpreted as a time delay. Thus our stabilization method can be used to control a system with time delays.

2.3.1 Inverted pendulum model *

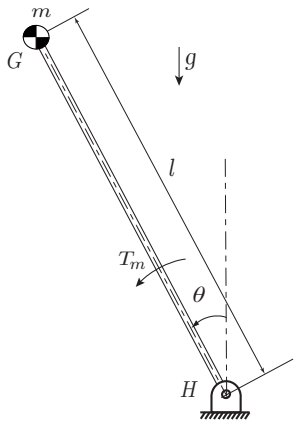

Figure 2.3: **Schematic of a simple inverted pendulum.** The simple inverted pendulum consisting of mass m at G and attached to a massless rod of length ℓ and controlled by a motor via a torque T_m at the hinge joint H.

Equations of motion. Figure 2.3 shows an inverted pendulum consisting of a mass m attached to massless rod of length ℓ at G and controlled by a motor at the hinge H by

applying a motor torque T_m. Gravity g points downwards and θ is the angle made by the pendulum with the vertical and measured in the counter-clockwise direction.

Using Newton-Euler's method and doing angular momentum balance about the hinge joint G, the equation of motion can be written as,

$$\ddot{\theta} = \frac{g}{\ell}\sin(\theta) + \frac{T_m}{m\ell^2} \tag{2.7}$$

Linearized equations of motion. We linearize the equations of motion about the vertical equilibrium position by setting $\sin(\theta) \approx \theta$. Further, we define $\omega^2 = \frac{g}{\ell}$ and $u = \frac{T_m}{m\ell^2}$ to get,

$$\ddot{\theta} = \omega^2\theta + u \tag{2.8}$$

State space model. In order to cast equation 2.8 into state space formulation we define $\theta = x_1$ and $\dot{\theta} = x_2$ to get,

$$\begin{bmatrix} \dot{x}_1 \\ \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \\ \omega^2 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} u \tag{2.9}$$

Further, we set $x = \begin{bmatrix} x_1 \\ \\ x_2 \end{bmatrix}$, $\mathbf{a} = \begin{bmatrix} 0 & 1 \\ \\ \omega^2 & 0 \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix}$ in equation 2.9 to get,

$$\dot{x} = \mathbf{a}x + \mathbf{b}u \tag{2.10}$$

2.3.2 Stabilizing Controller *

We now turn to our stabilizing controller (see section 2.2). We pointed out that successful application of our stabilizing control depends on proper selection of four key

quantities: a suitable section or instance of time to take measurements (n), measurement variables (x_n), output variables (z_n) and nature of controls actions (U_n). We discuss these next.

For n, we take measurements once every T seconds. We choose our output variables z_n to be the same as the measurement variables x_n, i.e. the joint angle and the joint velocity. We choose two control actions, $\delta u_n = [U_1 f_1(t) \;\; U_2 f_2(t)]'$, each active for half the time between time $n+1$ and n. Later we will make specific choices for the measurement time T, and the control functions $f_1(t)$ and $f_2(t)$.

To apply our control scheme we linearize the system about the nominal trajectory. Putting $\mathbf{A} = \partial x_{n+1}/\partial x_n$, $\mathbf{B} = \partial x_{n+1}/\partial U_n$, where $\delta U_n = [U_1 \;\; U_2]'$ and noting that $x_{n+1} = z_{n+1}$ we get,

$$\delta z_{n+1} = \delta x_{n+1} = \mathbf{A}\delta x_n + \mathbf{B}\delta U_n \tag{2.11}$$

For this specific example, it is possible to find the matrices $\mathbf{A}$ and $\mathbf{B}$ analytically. We got

$$\mathbf{A} = e^{\mathbf{a}T} \tag{2.12}$$

$$\mathbf{B} = \left[\int_{\tau=0}^{\tau=\frac{T}{2}} e^{\mathbf{a}(T-\tau)} \mathbf{b} f_1(\tau) d\tau, \;\; \int_{\tau=\frac{T}{2}}^{\tau=T} e^{\mathbf{a}(T-\tau)} \mathbf{b} f_2(\tau) d\tau \right] \tag{2.13}$$

Pole placement. We first assume a linear controller $\delta U_n = -\mathbf{K}\delta x_n$, where $\mathbf{K}$ is a gain matrix and put this in equation 2.11 to get,

$$\delta z_{n+1} = (\mathbf{A} - \mathbf{B}\mathbf{K})\delta x_n \tag{2.14}$$

The goal of pole placement is to choose the gain $\mathbf{K}$ so that eigenvalues of the closed system $(\mathbf{A} - \mathbf{B}\mathbf{K})$ are placed at the desired position. In particular, we place the eigen-

values at the origin (i.e. at zero). The gain matrix $\mathbf{K}$ needed to place the eigenvalues at zero is given by, $\mathbf{K} = \mathbf{B}^{-1}\mathbf{A}$.

This type of controller – that does full correction of deviation – is called a dead-beat controller. In dead-beat control, the system is brought to desired value in a finite number of time steps (see [1] page 201). The concept of dead-beat linear control is unique to discrete time controlled systems. Continuous control cannot give a dead-beat response with a linear controller (see [77] page 416-417).

2.3.3 Results *

We consider balance of a 1 m long rod with mass of 1 kg attached to its end. We put $g = 10$ m/s^2, $\ell = 1$ m, $m = 1$ kg, which gives $\omega^2 = 10$ /s^2. We choose the sampling time $T = 1$ s for control. Note that the characteristic time scale of this simple pendulum is about 0.32 s (time scale $= \sqrt{\ell/g} = \sqrt{1/10}s \approx 0.32s$), which about 3 times slower than the sampling time of 1 second that we will be using.

Eigenvalue of uncontrolled system. Putting $\mathbf{a} = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$ and $T = 1$ s in equation 2.12 gives, $\mathbf{A} = \begin{bmatrix} 11.8333 & 3.7286 \\ 37.2864 & 11.8333 \end{bmatrix}$. The biggest eigenvalues of the system is 23.6243. This implies that any disturbance to the system at time $t = 0$ will grow by a factor of 23.6243 in a time of 1 second.

Next, we consider dead-beat control of the inverted pendulum using two different pair of functions of time. First, we consider two sinusoids $f_1(t) = f_2(t) = \sin(2\pi t/T)$, and second, we consider constant functions $f_1(t) = f_2(t) = 1$.

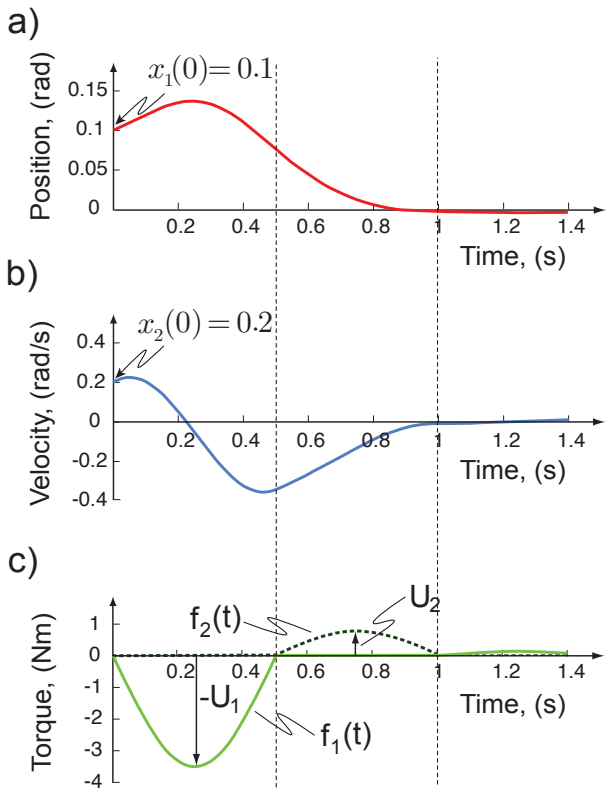

Figure 2.4: **Balance of a simple inverted pendulum using sinusoids for control functions.** (a) Joint angle vs time (b) Joint velocity vs time (c) Command torque vs time.

Example 1: Sinusoids for control functions. We put $\mathbf{a} = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $T = 1$, and $f_1(t) = f_2(t) = \sin(2\pi\tau/T)$ in equation 2.13 to get, $\mathbf{B} = \begin{bmatrix} 0.5670 & -0.0935 \\ 1.8244 & -0.4487 \end{bmatrix}$. Using the matrix $\mathbf{A}$ calculated earlier and value of $\mathbf{B}$ in equation $\mathbf{K} = \mathbf{B}^{-1}\mathbf{A}$, we calculated the gain matrix as $\mathbf{K} = \begin{bmatrix} 21.7536 & 6.7603 \\ 5.3498 & 1.1144 \end{bmatrix}$.

Figure 2.4 shows the joint angle, velocity, control vs time for the inverted pendulum with sinusoids for the control functions. The system is not quite dead-beat in simulation

because we used the linear equations to calculate the gain **K** while the simulations used the full non-linear equations.

Example 2: Constants for control functions. We put $\mathbf{a} = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $T = 1$, and $f_1(t) = f_2(t) = 1$ in equation 2.13 to get, $\mathbf{B} = \begin{bmatrix} 0.9300 & 0.1533 \\ 2.9927 & 0.7360 \end{bmatrix}$. Using the matrix **A** calculated earlier and value of **B** in equation $\mathbf{K} = \mathbf{B^{-1}A}$, we calculated the gain matrix as $\mathbf{K} = \begin{bmatrix} 13.2613 & 4.1212 \\ -3.2613 & -0.6794 \end{bmatrix}$.

Figure 2.5 shows the experimental set up of the inverted pendulum we used to demonstrate our stabilizing controller. There is a mass of about 1 kg at the end of the 1 m long rod. The carbon fiber rod itself has negligible mass. The joint angle were measured by an incremental encoder. The joint angular rate was calculated from the joint angles using numerical differentiation. The joint angle had a 1^o of hysteresis. The motors have a current dependent friction and constant friction (see chapter 3 section 3.3 and chapter 4 for more details on motor model). The gear backlash is about 2^o. Using the controller gain **K** calculated above – which did not account for backlash, friction, or angle hysteresis – we were able to balance the inverted pendulum upto a range of ±0.5 rad.

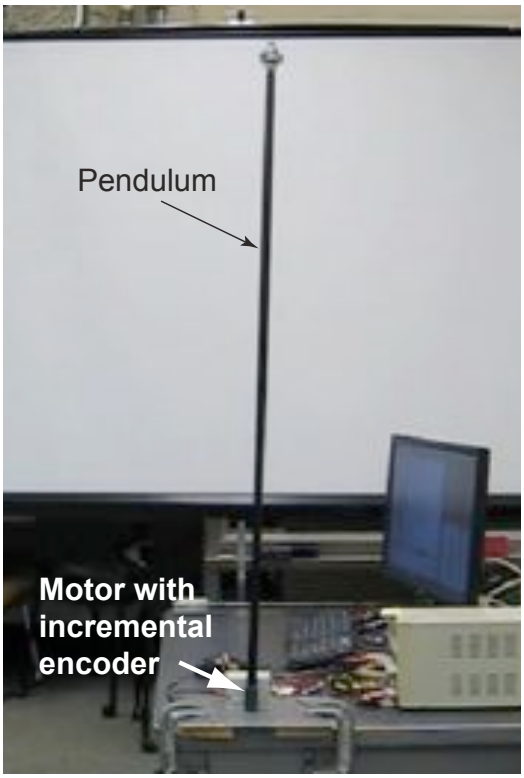

Figure 2.5: **Experimental verification of stabilizing control: balancing a simple inverted pendulum.** We measured the pendulum state – the angle and angular speed – once per second and used constant control functions active for half a second each. We were able to balance the inverted pendulum over a range of ±0.5 rad. Note that the bandwidth of control is 1 s and is slower than the characteristic time scale of 0.32 s of the simple inverted pendulum.

CHAPTER 3

MODELING

In this chapter, we present an overview of the legged robot called the Ranger. This robot served as the bipedal platform on which the ideas in this thesis were implemented. We also present a physics based hi-fidelity model of the robot and its actuators.

3.1 Ranger hardware

The Cornell Ranger is a four-legged knee-less biped (figure 3.1). It is about 1 m tall and has a total mass, including batteries, of 9.9 kg.

It is autonomous in that all sensing and computation is on board, batteries are on board, and it has no booms, tethers or cable connections. It is not autonomous in that, at least so far, it needs to be started manually, and steering is done with a model-plane type radio control.

Hardware. The robot has four legs in two pairs. The outer pair moves together, acting as one leg, as do the inner pair. Each leg has an ankle joint and a foot but no knee joint. For each pair of legs the ankle joints are mechanically connected. The robot has 3 main internal degrees of freedom (hip and two ankle pairs), which are all powered by brushed DC motors (46 W nominal). In parallel with the hip motor is a hip spring that tends to keep the pairs of leg parallel. The two ankle motors are near the hip axis and actuate the ankles via one-way (toe-off) cable drives. Foot lifting, for ground clearance, is powered by a return spring on each ankle. A small fourth motor (1 W nominal) twists the inner legs about a vertical axis with each step in order to steer the robot; the amount and

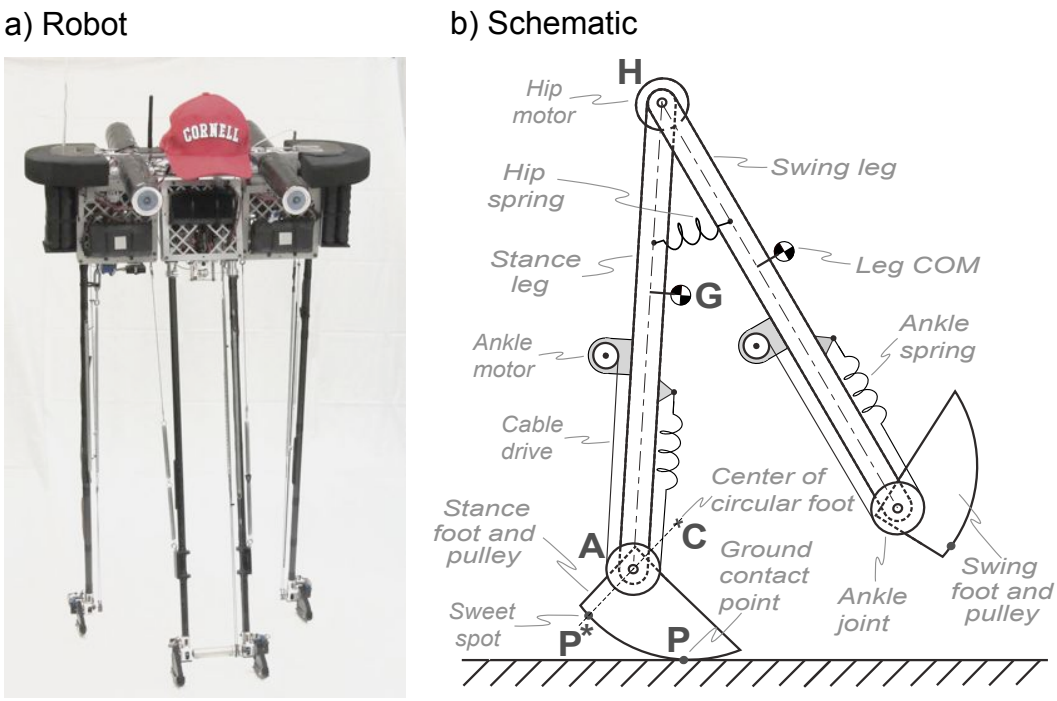

Figure 3.1: **Photo of Ranger and a schematic.** (a) Ranger. (b) 2D Schematic. The fore-aft cylinders with 'eyes' and the foam 'ears' (both visible in photo) are only for shock absorption in case of falls. The hat is decorative (hollow). The closed and rigid aluminum lace boxes, conceptually shown as point H, house all of the motors and gearing, various pulleys for the ankle cable drives, and most of the electronics (on the drawing the hip motor location is only schematic). There are two boxes connected by a hinge: an outer box, shaped like an upside-down U, rigidly connected to the outer legs, and an inner box, filling the space in the U, holding the inner legs (each of which can twist for steering) The hip spring, which aids leg swing, is shown schematically as symmetric between the two legs but shows as a diagonal cable and spring in the photograph. The feet are shaped so that toe-off is possible, so that no torques are needed during single stance, and so that ground clearance during swing can be achieved (by rotating the toe towards the hip).

direction of steering is governed by radio remote control.

Electronics. The main control loop runs, with no supervisory operating system, on an ARM9 microcontroller. Four ARM7 processors on custom boards monitor and control

the three main joints (the outer ankles are counted as a single joint) and the steering. Two more ARM7 processors supervise the on-board communications network (controller area network, CAN), the Bluetooth data reporting, and the onboard data display and lights. The Inertial Measurement Unit (IMU) also contains a proprietary microprocessor board. The multi-processor bus-based architecture (CAN) was chosen to facilitate design evolution, to simplify overall wiring (e.g., so a new sensor could be added without new wiring into the main processor) and to compartmentalize the control software (high-level on ARM9, low level on ARM7s). Sensors for each motor include an optical encoder, a voltage sensor, and a current sensor. In addition, each joint has an absolute angle sensor. Each foot has an optical strain gauge for measuring foot distortion (and hence foot contact). From the 3D IMU, Ranger's control only uses the sagittal plane angular rate sensor. The top-level control loop runs at 500 Hz on the ARM9 processor; data is sent to and from the satellite ARM7 processors once per loop execution; the motor current controllers, and their associated sensors operate at 2 kHz on the ARM7 processors. The motors and electronics are powered by seven 25.9 V lithium-ion batteries with a total capacity of about 493 watt-hours (1.77×10^6J).

Software. The total custom control code is about 10,000 lines of C and C++ code, the bulk of which is associated with low-level measurement, low-level control and communications protocols. The main control loop is based on hierarchical concurrent finite state machines (see chapter 5, section 5.3 and appendix G). Control and estimation tasks are coordinated by a simple cooperative-multitasking scheduler, while low-level input-output, such as from motor encoders, uses processor interrupts.

For debugging and development, walking data is viewed and logged via wireless system. Although control parameters can be adjusted wirelessly mid-walk, during attempts at walking distance records autonomy is maintained by sending to the robot only

steering signals and requests for data (e.g., cumulative number of steps, battery voltage) and not sending any walking control nor any changes of walking control parameters.

More design details. Description of more aspects of the hardware, electronics and low level control are in the appendix of the paper [8]. Even more details, including photos, videos and 100^{+} reports are on Cornell's www site [12]

However, as with any complex machine, success at control rests on a pyramid of such hardware developments and refinements. For example, success here depended on all of these things: the design of the foot shape to allow push-off, stance with low torque, and swing clearance; the design of the foot as a load cell (foot deformation is measured); a low-mass leg and foot; design of the single-cable drive (which is also a series elastic element which needs to have twice the stiffness for the inner leg-pair as for each of the outer legs); design of the body box for stiffness; selection of motors and gearing for torque and efficiency; design of the motor controllers (based on Pulse Width Modulation, PWM) for low dissipation; selection of energy-efficient electronics, i.e., the sensors and the microprocessors; determining overall state from sensor data, including the determination of predicted and actual ground collision time; design of a low-power leg-twist steering mechanism that would not interfere with the 2D dynamics; hip-spring design and placement; and dozens of other issues like the selection of glues, the protocol for washing off flux to prevent corrosion, methods for joint alignment and needed drive-train compliance, cable tensioners, shock absorbers for fall protection, etc.

3.2 Robot model

Our controller design development depends on a fast-running and reasonably accurate offline dynamical simulation of the physical robot. The model we used is two-dimensional; it only has dynamics in the sagittal plane (the steering control was not dynamically modeled).

Each 'leg' (one inner and one outer) is characterized by a mass, a center of mass (COM) location, and polar inertia about the COM. The rubber-bottomed feet are assumed to be rigid and massless with bottoms that make point contact with a rigid, flat and level ground (unless otherwise noted). The hip motor connects to the legs with gears so the drive is modeled as not compliant; i.e., in the model the hip does not have an independent degree of freedom from the hip motor. The spring at the hip adds a centering torque to the hip motor torque, in proportion to leg splay. The hip motor ('reflected') inertia is neglected because, after multiplying by the gear reduction squared ($66^2 = 4356$), the inertia of the hip motor is about 50 times less than that of the legs. In contrast, the ankle motors are in series with the elastic cable drive. So the ankle motors, with an associated rotary inertia, each have a degree of freedom independent from the ankles. The ankle drive cables are modeled as linear springs, as are the return springs. When in contact with the ground the feet are assumed to roll without slip. The ground collisions are assumed to be instantaneous with no bounce and no slip. The heel-strike collision is assumed to have no impulsive torques at the joints and no discontinuities in configuration. The robot is assumed to be symmetric with respect to inner and outer legs and hence only one step is needed to characterize a periodic gait. The two ankle motors do not participate in the collision because the ankle motors are isolated by the ankle springs (Achilles tendons). So the ankle motor positions and velocities are taken as continuous through heel-strike. In contrast, the hip motor has a collisional velocity

discontinuity but not collisional torque because its inertia is neglected. Although we model the ground contact as a point contact between rigid feet and rigid ground, we add a small contact-damping couple between foot and ground to damp out oscillatory rocking motions of the feet (observed in simulations if this damping is zero).

Altogether the robot has five internal degrees of freedom (one hip joint including the motor, two ankle joints, and two ankle motors). During single stance, there is one additional pose degree of freedom (rotation of the stance foot) making a total of six. During double stance there are only the five internal degrees of freedom.

Phases of motion. The motion of the robot consists of two smooth phases: 1) single-stance, when only one foot is on the ground, and 2) double-stance when both feet are on the ground. The two phases have different equations of motion. The two phases are separated by two instantaneous transitions: a) the heel-strike collision at the transition from single-stance to double-stance, and b) the (non-collisional) toe-off transition from double-stance to single-stance. A walking cycle consists of a single step. For example:

$$\underbrace{\ldots \text{ outer-legs toe-off}}_{\text{previous step}} \rightarrow$$

$$\underbrace{\overbrace{\text{inner-legs single-stance}}^{\text{smooth phase}} \rightarrow \overbrace{\text{outer-legs heel-strike}}^{\text{collisional transition}} \rightarrow \overbrace{\text{double-stance}}^{\text{smooth phase}} \rightarrow \overbrace{\text{inner-legs toe-off}}^{\text{smooth transition}}}_{\textbf{a single step = one cycle}}$$

$$\rightarrow \underbrace{\text{outer-legs single-stance} \rightarrow \ \ldots}_{\text{next step}} \tag{3.1}$$

Due to intermittent contact the robot may be viewed as non-holonomic, but no consideration is taken of this in the modeling because the system is holonomic within each phase [91].

Governing equations. Using the assumptions described above, the governing equations follow from momentum and angular momentum balance applied to the robot and its subsystems:

$$\text{Single stance (continuous):} \quad \mathbf{M}_s(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_s(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{K}_s(\mathbf{q}) = \mathbf{T} \tag{3.2}$$

$$\text{Heel-strike (instantaneous):} \quad \text{Angles continuous but velocities jump}$$

$$\mathbf{M}_c(\mathbf{q}^+)\dot{\mathbf{q}}^+ - \mathbf{M}_c(\mathbf{q}^-)\dot{\mathbf{q}}^- = \mathbf{J}_\mathbf{c}^\mathbf{T}\mathbf{P}^\star \tag{3.3}$$

$$\text{Double stance (continuous):} \quad \mathbf{M}_d(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_d(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{K}_d(\mathbf{q}) = \mathbf{T} + \mathbf{J}_\mathbf{d}^\mathbf{T}\mathbf{P} \tag{3.4}$$

$$\text{Toe-off (instantaneous):} \quad \text{Angles and velocities continuous} \tag{3.5}$$

where each equation is a 6-component vector equation. The six elements of $\mathbf{q}$ are the three robot joint angles, the two ankle motor angles and the absolute angle of one foot. The six elements of $\mathbf{T}$ are the three motor torques (including the gear box friction), and three foot-related torques (two ankle torques and one torque at the reference-foot ground contact). The foot torques are determined by joint friction and by the ground contact damping term. The $\mathbf{M}$s are the mass matrices (subscripts s, d and c are for the 'single', 'double' and 'collisional' phases), the $\mathbf{C}$s are the velocity squared terms (centrifugal and Coriolis) and the six components of the $\mathbf{K}$s include the gravity and spring related terms. The 2-element $\mathbf{P}$ and $\mathbf{P}^\star$ are trailing foot constraint force components during double stance and constraint impulse components during heel-strike respectively (in some formulations $\mathbf{P}$ and $\mathbf{P}^\star$ are Lagrange multipliers for constraints). Finally, the 6×2 matrices $\mathbf{J_c}$ and $\mathbf{J_d}$ are kinematically determined matrices that show the effect of trailing-foot constraint force components on the internal degrees of freedom when the kinematic chain is closed (double stance). See appendix A for derivation of equations of motion and appendix B for benchmarks to check the accuracy of derived equations of motion.

3.3 Motor and gearbox model

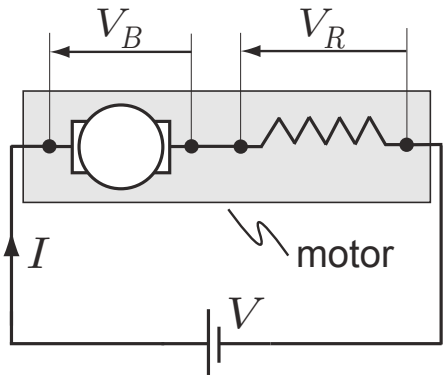

Figure 3.2: **Schematic of DC motor connected to a DC voltage source.** The DC motor consists of rotating part called the rotor and a resistive part. The sum of voltage drop across the rotor (V_B) and resistive part (V_R) equals the total voltage supplied by the DC source V. The current flowing in the circuit is I.

Although we run a tight feedback loop on motor current (2 kHz control loop), we do not run a tight feedback loop on the motor torque or angular velocity. So we cannot model the motors as pure torque, pure velocity or pure position sources and we need a model for motor torque in terms of motor current and angular velocity. Measuring torque, angular velocity, current and voltage during both positive and negative work on our own bench-test setup, we have found that standard motor models lack two major features: 1) a voltage drop across the brush contacts, and 2) a load dependent, velocity-direction dependent and roughly velocity-magnitude independent frictional torque. Although these phenomena are known, e.g., [31, 50] they are not commonly accounted for, so we review our motor model here.

The electrical power consumption P of the motors is given by the voltage V across the motor times the current I through it. At this point in the modeling we think of a constant Direct Current (DC) source and do not consider losses due to the oscillations

in current from the Pulse Width Modulation (PWM) used by our motor controllers.

The total electrical power is given as follows (see figure 3.2).

$$\begin{aligned} P = VI &= V_R I + V_B I \\ &= \overbrace{(IR + V_c \operatorname{sgn}(I))}^{V_R} I + \overbrace{GK\omega}^{V_B} I \end{aligned} \tag{3.6}$$

where $\operatorname{sgn}(I) \equiv I/|I|$.

The motor voltage has a contribution from resistance V_R and from "back EMF" V_B. The resistance of a real motor has a contribution from the windings, R (the part reported in the motor specification sheets), and a part due to brush contact resistance (which the manufacturer does not mention). Our experiments (consistent with the literature [50]) show that the contact resistance leads to a more-or-less constant voltage drop V_c, which changes sign when the current is reversed. We used $V_c = 0.7$ V with a winding resistance of 1.3 Ω in simulation. The brush voltage drop has the biggest relative effect when the total motor current is close to zero.

We characterize rotation rate using the gearbox output angular velocity ω hence $V_B = KG\omega$ is used above, where K is the motor constant, G is the down gearing ratio, and $G\omega$ is the motor angular speed.

The output shaft torque (T) is given by the ideal-motor output torque after the gearbox (GKI) minus the frictional losses in the motor and gear box T_f.

$$T = \overbrace{T_{ideal}}^{GKI} - T_f \tag{3.7}$$

Our bench tests show friction with viscous, Coulomb and load-dependent parts for

which the following functional form is a reasonable fit to our bench tests:

$$\begin{aligned} |T_f(I,\omega)| &\leq C_0 + \mu GK|I| && \text{if } \omega = 0 \\ T_f(I,\omega) &= C_1\omega + C_0\ \mathrm{sgn}(\omega) + \mu\ \mathrm{sgn}(\omega)GK|I| && \text{if } \omega \neq 0 \end{aligned} \tag{3.8}$$

The viscous friction is given by $C_1\omega$, the coulomb friction by $C_0\ \mathrm{sgn}(\omega)$. A friction that scales approximately with torque, apparently due to load-dependent friction in the gears, is given by $\mu\ \mathrm{sgn}(\omega)GK|I|$ (Note that $|I|$ correlates with the magnitude of net torque and $\mathrm{sgn}(\omega)$ shows the resistance to motion).

Combining the equations above, when $\omega \neq 0$ we get:

$$T = GKI - C_1\omega - C_0\ \mathrm{sgn}(\omega) - \mu\mathrm{sgn}(\omega)GK|I|. \tag{3.9}$$

Again note that we have two nonstandard terms: in the power equation, there is a contact voltage drop; and in the torque equation, there are both Coulomb friction and current-dependent friction terms. Our current-dependent friction term is similar in effect to the load dependent friction term [31]. To make the solutions unique in numerical implementation at $\omega = 0$ and to ensure smoothness in the solution for better convergence of numeric optimizations, we replace the sgn (sign) function with a hyperbolic tangent function (see appendix C). The values we used in simulation were: $K = 0.018\,\mathrm{N\,m/A}$, $G = 66$ (hip), $G = 34$ (ankle, gear reduction is 43, taking account of pulley radii the ratio is 34), $C_1 = 0\ \mathrm{N\,m\,s/rad}$, $C_0 = 0.01\ \mathrm{N\,m}$ and $\mu = 0.1$. In this approximation the torque is mis-estimated by about 10% by not using the friction terms. See chapter 4, section 4.2 for more details on system identification.

CHAPTER 4

SYSTEM IDENTIFICATION

This chapter is divided into two parts. In the first part, we present system identification for the robot model. In the second part, we present system identification for the motor and gearbox model.

4.1 System identification for mechanical parameters

For numerical simulation we need to estimate the ten parameters shown in figure A.1. CAD drawings could have been used to estimate most of these. But because there were many modifications not in the original drawings (rubber feet, glue, tape, etc.) we took inertial parameter identification as a separate measurement project. The length parameters (ankle eccentricity d, the radius of feet r and leg length ℓ) are measured with a tape measure. Measurement of the inertial parameters is presented in section 4.1.2. Measurement of the spring parameters: the hip spring constant and the ankle spring constant is presented in section 4.1.3. However, first we show how we dynamically balance the the legs, thereby making the robot symmetrical and simplifying the controller design and simulations.

4.1.1 Dynamic balancing of legs

We would like the legs to be dynamically balanced so that a controller that treats the legs as equal does not lead to a limping gait. Assuming a balanced machine, we can then simplify both the simulations and the controller by only dealing with a single step (rather than a 2-step sequence) as the basic action.

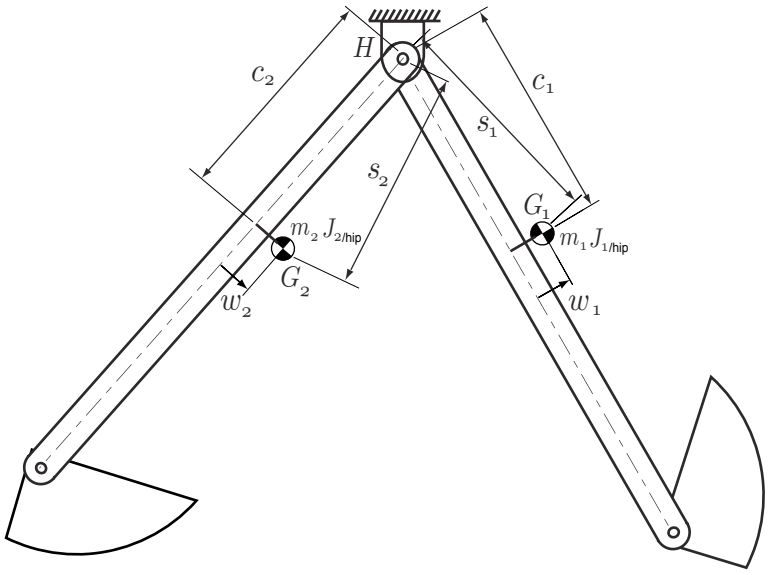

Figure 4.1: **Dynamic balance of the legs.** The two legs of the robot are dynamically symmetrical ('balanced') if they they have the same distances between the hip and ankle hinges, the same feet shapes and have 3 matching inertial properties. The two masses do not have to be the same, however.

Figure 4.1 shows that the robot that may have unbalanced legs; i.e., the masses (m_i), inertia about hip ($J_{i/hip}$) and the location of the COM (s_i) of the two legs may different. Here i = *inner* or *outer*.

Number of parameters. In side view the robot has 4 serial links (inner foot, inner leg, outer leg, outer foot). As detached planar rigid objects each of these has 4 inertial parameters: center of mass position relative to landmarks (e.g., hinges and corners) on the object (x and y positions using an object-based coordinate system); mass m; and moment of inertia about the center of mass J_G. Thus one could imagine up to 16 inertial parameters in the model. Neglecting the mass of the feet eliminates 8 of these (4 for each foot), reducing the number of inertial parameters to 8.

Redundancy of parameters. At every joint we can imagine adding and subtracting point masses. For example we could add a point mass m_{add} to the inner legs at the hip and simultaneously add a negative mass $-m_{add}$ to the outer legs (that such is non-physical does not detract from the argument). No term in any of our equations of motion are affected by this addition and subtraction. Thus no motion nor torque is altered. One could also make the claim by appeal to Lagrange's equations: the expression for the system kinetic and potential energies are unaltered by this addition/subtraction.

We have thus changed the masses of each of the links and the locations of the centers of mass of each of the legs, but we have not changed any of the dynamics of the linkage. [As an aside, with this mass addition/subtraction we have changed the reaction forces transmitted at the hinge, but these do not affect the motions or the joint torques.] Thus, the supposed 8-dimensional parameter space is indifferent to one dimension. That is, there must be a collection of 7 parameters which can predict all coefficients in the governing equations.

We claim that the following is such a set of 7 (c and w are local object-referenced x and y coordinates):

$$J_{1/hip},\ J_{2/hip}; \qquad m_1c_1,\ m_2c_2; \qquad m_1w_1,\ m_2w_2; \qquad \text{and} \qquad M_{tot} \equiv m_1 + m_2. \quad (4.1)$$

Each of these parameters is indifferent to (doesn't change with) the hip-mass addition and subtraction described above. They are also independent in that by appending the list with another single number, for example the mass of one leg, all 8 of the original inertial parameters can be found (m_i, J_{Gi}, c_i and w_i). In summary, the 7 mass parameters are: The first mass-moment of the mass distribution of each leg about the hip (2 numbers for each leg), the second polar mass moment of each leg about the hip (polar moment of inertia about the hip), and the total mass of the robot.

4.1.2 Measuring inertial parameters

Not coincidentally, the independent set of 7 parameters are exactly what it is possible to measure *without disassembling* the robot. Note, for example, that it is not possible to find the mass of one leg (m_1 or m_2). Nor is it possible to find the distance of the center of mass of one leg from the hip (can't find s_1 or s_2).

We find the 7 parameters as follows:

1) The total mass M_{tot} is found by weighing the robot.

2,3) The first moment of mass along each leg $c \cdot m$ is found for each leg by the experiment shown in figure 4.2.

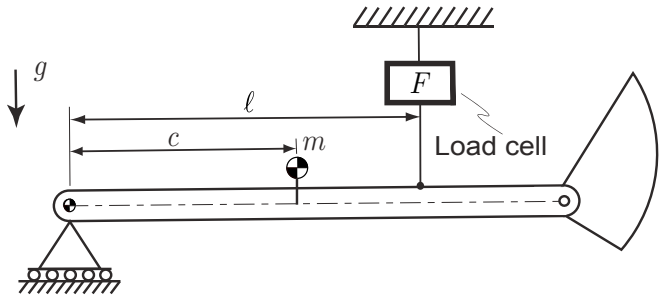

Figure 4.2: **Experiment to measure the first mass moment $c \cdot m$.** The robot is hinged at the hip joint and held such that the leg axis is perpendicular to gravity. Balance of moments about the hip hinge gives $cm = F\ell$.

4,5) The angle of the radial line from the hip on which the center of mass lies is found by hanging each leg from the hip hinge and measuring the angle θ of the leg (see figure 4.3). Because $\tan\theta = w/c$ we have for each leg that $wm = \tan\theta\, cm$.

6,7) The inertia of each leg about the hip joint can be found from timing the small oscillations of each leg freely swinging from a hip clamped in place.

$$T_{\text{pend}} = 2\pi \sqrt{\frac{J_{/hip}}{gsm}} \tag{4.2}$$

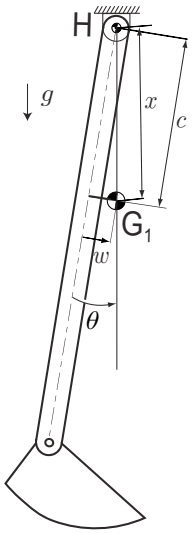

Figure 4.3: **Experiment to measure the fore-aft offset of the COM.** The robot is hinged at its hip joint H and held in the vertical plane. In equilibrium the center of mass of the leg G_1 is directly below the hinge point H. The angle θ can be measured.

where $sm = \sqrt{(cm)^2 + (wm)^2}$. So, for each leg,

$$J_{/hip} = \frac{T_{pend}^2}{4\pi^2} g \sqrt{(cm)^2 + (wm)^2}. \tag{4.3}$$

Thus all 7 independent inertial properties were measured without robot disassembly.

Balance of 3 inertial parameters. With these 7 independent parameters, symmetry of the two legs is achieved by making these 3 matches:

$$\begin{aligned} J_{1/hip} &= J_{2/hip} \\ m_1 c_1 &= m_2 c_2 \\ m_1 w_1 &= m_2 w_2 \end{aligned} \tag{4.4}$$

As mentioned, although we have four parameters for each leg ($J_{i/G}$, m_i, c_i and w_i) there are only three conditions that ensure dynamic balance of the two legs.

Physical balancing. We made the best balance we could by appropriate placement of the batteries on the outer legs. We chose not to add additional masses. Because we were constrained in our battery attachment points we could not get perfect dynamic balance (i.e., could not achieve equality within measurement accuracy of equation 4.4).

4.1.3 Measuring spring constants

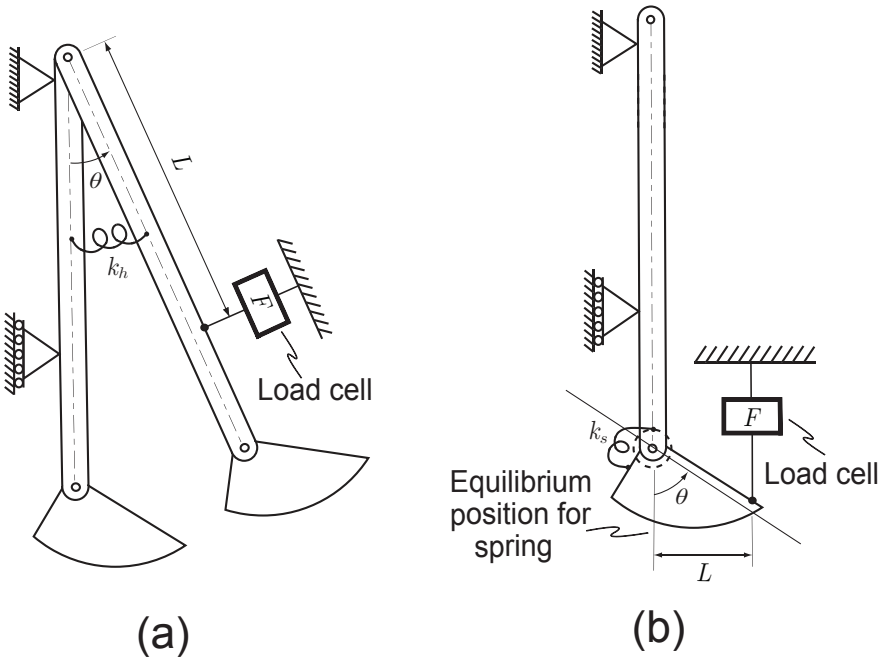

Figure 4.4: **Measuring the spring constants.** (a) The hip spring stiffness is determined by measuring the force to deflect the leg a given angle. (b) The ankle spring constants (the elasticities of the cable drive), are measured by locking the ankle motors and measuring the force to deflect the ankles.

Measuring the hip spring constant. Figure 4.4a shows the set up used to estimate the hip spring constant. In this experiment, the robot was placed in the horizontal plane and so gravity does not influence the experiment. The hip springs are designed so they

have no torque when the legs are parallel to each other. First, we checked this by noting that the spring torque is zero when the legs are parallel. Next, we pulled Ranger's legs with a digital 'fish' scale (a load cell with hooks) and noted the hip angle θ. The hip torsional spring constant is then

$$k_h = FL/\theta.$$

As predicted the spring is nearly linear in Ranger's operating region (hip angle ±0.5 rad).

Measuring the ankle spring constant. Figure 4.4b shows the ankle spring-constant measurement. The motor was put in position control (locked at a fixed angle) and the foot deflected with the fish scale (see Figure 4.4b) and the angle of ankle deflection θ measured. Thus the ankle spring constant

$$k_s = FL/\theta.$$

4.1.4 Summary of parameters estimated

We summarize the various parameters estimated in this section.

Parameter	Value
ℓ	0.96 m
r	0.2 m
d	0.11 m
wm	0.0
cm	0.72 kg m
J_{hip}	0.55 kg m^2
k_h	7.6 N m/rad
k_s	14 N m/rad

Table 4.1: **Values of robot parameters that were estimated in bench tests.**

4.2 System identification for motors and gearboxes

In chapter 3 section 3.3, we showed the motor and gearbox model we used for simulation and optimization. First we re-state these equations and follow that with a discussion on how the model was found and the parameters identified.

4.2.1 Power equation

The power equation gives the power consumption (P) as a function of the current (I) and the speed (ω). In the end we wrote this equation as

$$P = VI = \{IR + V_c \operatorname{sgn}(I) + GK\omega\} I \tag{4.5}$$

Here the gear ratio G is known based on manufacturer's specifications and we have to identify the three constants: resistance R, contact voltage V_c and torque constant K.

4.2.2 Torque equation

The torque equation gives the output shaft torque (T) as a function of the current and shaft speed. The function $T_f(I, \omega)$ in the expression below is the frictional torque and we have assumed that it depends on the current (I) and shaft speed (ω).

$$T = G(KI - J_m \overbrace{G\dot{\omega}}^{\text{motor acceleration}}) - T_f(I, \omega) \tag{4.6}$$

We found that the friction torque $T_f(I, \omega)$ can be reasonably decomposed into a constant friction term and viscous friction term.

$$\begin{aligned} |T_f(I, \omega)| &\leq C_{0s}(I) && \text{if } \omega = 0 \\ T_f(I, \omega) &= C_1\omega + C_{0d}(I) \operatorname{sgn}(\omega) && \text{otherwise} \end{aligned} \tag{4.7}$$

where C_1, C_{0s} and C_{0d} are coefficients of viscous, static and dynamic friction respectively and the latter two are assumed to be current dependent.

Next, we found reasonable fit using $C_{0s}(I) = C_{0s} + C'_{0s}|I|$ and $C_{0d}(I) = C_{0d} + C'_{0d}|I|$. We characterize the current dependent part of the constant friction by parameter μ as follows. We put $C'_{0s} = \mu_s GK|I|$ and $C'_{0d} = \mu_d GK|I|$. Thus our frictional torque becomes,

$$|T_f(I,\omega)| \leq C_{0s} + \mu_s GK|I| \qquad \text{if } \omega = 0 \qquad (4.8)$$

$$T_f(I,\omega) = C_1\omega + C_{0d}\ \text{sgn}(\omega) + \mu_d GK|I|\ \text{sgn}(\omega) \qquad \text{otherwise} \qquad (4.9)$$

In the above equations, the gear ratio G is known based on manufacturer's specifications. The torque constant K is known from the system identification on the power equation presented earlier. Thus in equation 4.8 and 4.9, we have to identify the five constants, C_{0s}, C_{0d}, μ_s, μ_d and C_1.

4.2.3 Cantilever test set-up for data collection

Details of set-up. Figure 4.5 shows the labeled photograph of our experimental set-up. The set-up consists of two motors; a Faulhaber test motor of the type used on the robot and a Maxon brake motor used as the motor load. Each motor can be individually controlled by DC power supplies. The test motor is fitted with a 14:1 gear box. The motors are connected to each other by two helical shaft couplings and a slender steel rod. A Hall-effect current sensor measures the current flowing through the test motor. A rotary encoder measures the angular position of the brake motor. Angular position can be converted to angular speed by finite differencing and low-pass filtering. Using the brake motor gear ratio and its measured angular speed, we can find the speed of the output shaft, the slender steel rod in the set-up. A load cell (Transducer Techniques, MLP-75) measures the test motor torque output. By varying the current flowing through

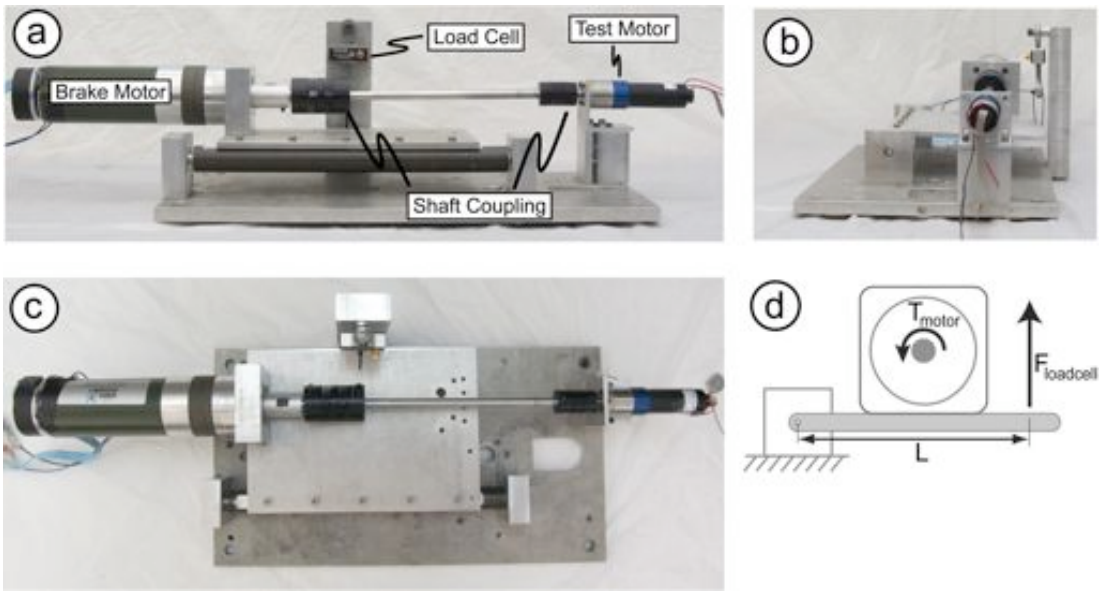

Figure 4.5: **Cantilever set up used for system identification for DC motors.** A 'brake' motor is mounted on a hinged plate so the torque acting on it can be measured. It is driven by the motor being tested, which is mounted on a solid workbench, via two helical shaft couplings and a slender steel rod. A DC power source is connected to the test motor that maintains the input voltage V. By varying the current in the brake motor, varying braking torques can be applied to the test motor. Test motor current I, output shaft speed ω, and output shaft torque T are measured and used for system identification. Note that the 'brake' motor can also be powered so that the test motor can be characterized in the negative-work regime (i.e., as a generator).

the braking motor, varying braking torques can be applied. Data is recorded by a National Instruments LabVIEW program which is interfaced with the sensors through a data acquisition system.

Data collection. The data collection was done as follows.

- A DC voltage (V) was set on the test motor. The braking torque was varied by varying the current to the brake motor. Test motor current (I), output shaft speed (ω) and output shaft torque (T) reading were noted for increasing and decreasing

braking torques.

- A different motor voltage was fixed and the test repeated. The test motor voltages chosen were $-6, -4, -2, 0, 2, 4$ and 6 V.

4.2.4 Fit the power equation

In order to fit the power equation 4.5 we need to find the constants R, V_c and K. First, we stall the motor and noting the current at various voltages, we fit the constants V_c and R. Next, using the data from the cantilever experiment we fit the constant K.

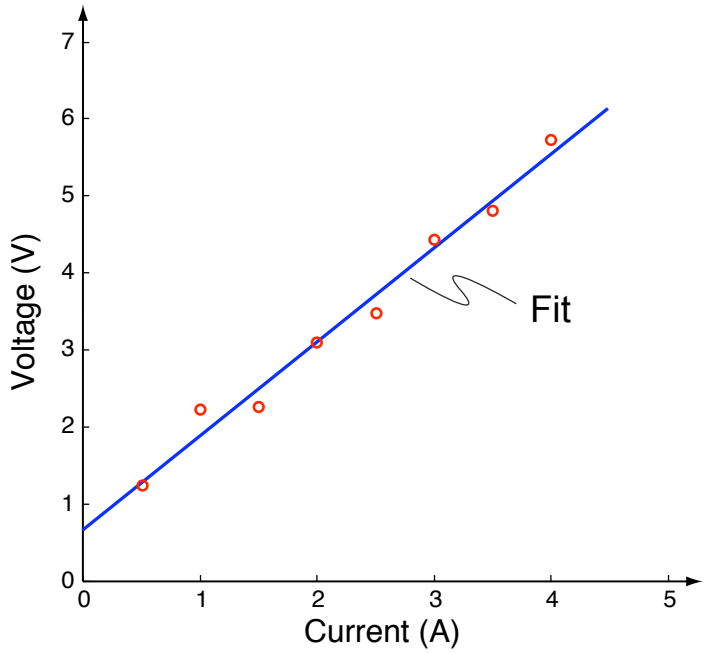

Figure 4.6: **Curve-fitting for contact voltage and terminal resistance.** Least squares curve-fitting voltage and current data for a stalled motor gives a terminal resistance $R = 1.3\ \Omega$ (slope) and contact voltage drop $V_c = 0.7\ V$ (y-intercept).

Fit the resistance (R) and contact voltage drop (V_c) by stalling the motor. In equation 4.5, we first set to identify the resistance R and contact voltage drop V_c. We stalled the motor and applied various currents I and measured the output voltage V. Putting speed $\omega = 0$ in equation 4.5 we have $V = IR + V_c \text{ sgn}(I)$. Using a least squares fit, we found $R = 1.3\ \Omega$ and $V_c = 0.7\ V$. Results of the fit are shown in figure 4.6.

Fit the the torque constant (K) from cantilever experiment. Having fitted the resistance R and contact voltage drop V_c in equation 4.5, we only need to fit the torque constant K. Using the DC voltage V, motor current I and output shaft speed ω data from the cantilever experiment and knowing that the gear ratio $G = 14$, we fitted the torque constant $K = 0.018$ as shown in figure 4.7.

4.2.5 Fit the torque equation

In order to fit the torque equations 4.8 and 4.9, we have to identify the five constants, C_{0s}, C_{0d}, μ_s, μ_d and C_1. First, using a series of pulley experiment as shown in figure 4.8 we identify all the five constants. Next, using the data from the cantilever experiment we check our fit and also point out the necessity of having the $|I|$ function in the frictional torque equations 4.8 and 4.9 .

Fit the dynamic friction at zero current (C_{0d}). We set the motor in figure 4.8 in open loop ($I = 0$) and increased the mass M in increments of 5 gram while gently tapping the pulley until it set into slight motion. At about $M = 40$ gram, the mass started to move downward (clockwise rotation of pulley when viewed from the right side). Using the radius of the pulley ($r = 2.5$ cm), mass M, gravity g and equation 4.6 and equation 4.9, we calculated the dynamic friction at zero current to be $T = Mgr = T_f = C_{0d} = 0.01$

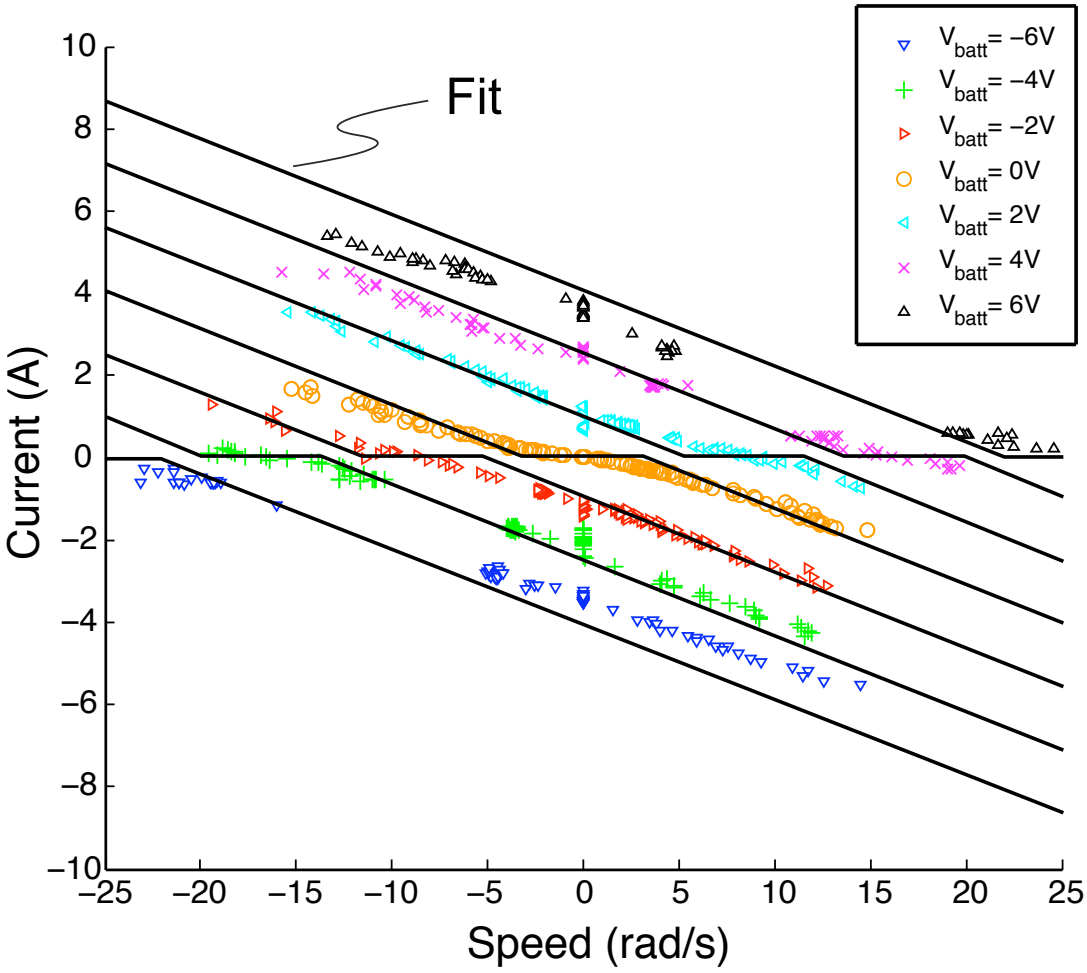

Figure 4.7: **Curve-fitting for motor torque constant.** Motor torque constant $K = 0.018$ V s/rad was curve-fitted from equation 4.5 using motor current I, output shaft speed ω and DC voltage V data obtained from the cantilever set-up. In equation 4.5 we used gear ratio $G = 14$ as per manufacturer's specification. Constants $R = 1.3\ \Omega$ and $V_c = 0.7\ V$ were obtained in an earlier experiment (see figure 4.6).

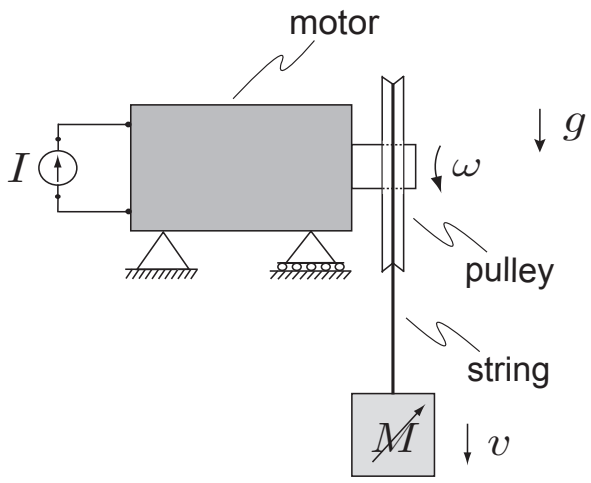

Figure 4.8: **Measuring friction coefficients.** In some experiments the motor was initially still, in others it is turning at known speed.

Nm. An identical value for the dynamic friction was calculated when the mass was hung on the other side of the pulley (counter-clockwise rotation of pulley when viewed from the right side).

Fit the static friction at zero current (C_{0s}). We set the motor in figure 4.8 in open loop ($I = 0$) and increased the mass M in increments of 5 gram till the motor-pulley, with no tapping, until it set into slight motion. At about $M = 45$ gram the mass began to move downwards. Using the radius of the pulley r, mass M, gravity g and equations 4.6 and equation 4.8, we calculated the static friction at zero current to be $T = Mgr = T_f = C_{0s}(0) = 0.01$ Nm. The same static friction value was calculated when the test was repeated with the mass hanging on the other side of the pulley.

Fit the viscous friction at zero current (C_1). We set the motor in figure 4.8 in open loop ($I = 0$) and increased the mass to M_1 until the motor-pulley set in motion. We noted that speed ω_1. Next we changed the mass to M_2 and repeated the experiment and

noted the new speed ω_2. From equation 4.6 and equation 4.9 we have

$$T_1 = M_1 gr = T_{f1} = C_1\omega_1 + C_{0d}$$

$$T_2 = M_2 gr = T_{f2} = C_1\omega_2 + C_{0d}$$

Subtracted the first equation from the second gives,

$$C_1 = \frac{(M_2 - M_1)gr}{\omega_2 - \omega_1} \tag{4.10}$$

Using equation 4.10, we calculated the viscous friction to be 3.3×10^{-3} N m s/rad. The same experiment when repeated with the mass hung on the other side of the pulley gave a similar value.

Fit the coefficient of dynamic friction (μ_d). We repeated the experiment used to identify C_{0d}, except that we set the current to a non-zero value. If M is the load at which the motor-pulley system just starts to move then, from equation 4.6 and equation 4.9, just at the onset of motion we have

$$T = Mgr = GKI - \mu_d GK|I| - C_{0d}$$

Solving for μ_d gives,

$$\mu_d = \frac{1}{\text{sgn}(I)}\left\{1 - \frac{Mgr + C_{0d}}{GKI}\right\} \tag{4.11}$$

We repeated this test for different current values and also by putting the mass on the other side of the pulley. The average value for μ_d across various tests was found to be 0.1.

Fit the coefficient of static friction (μ_s). We repeated the experiment used to identify C_{0s}, except that we set the current to a non-zero value. Again, if M is the load at which

the motor-pulley system just moves when tapped then, from equation 4.6 and equation 4.8, just at the onset of motion we have

$$T = Mgr = GKI - \mu_s GK|I| - C_{0s}$$

Solving for μ_s gives,

$$\mu_s = \frac{1}{\text{sgn}(I)} \left\{ 1 - \frac{Mgr + C_{0s}}{GKI} \right\} \tag{4.12}$$

We repeated this test for different current values and also by putting the mass on the other side of the pulley. The average value for μ_s across various tests was found to be 0.1.

Finally, we used the value of the constants C_{0s}, C_{0d}, C_1, μ_s and μ_d and checked the torque equation with the data obtained from the cantilever experiment. Figure 4.9 shows our model fit with the torque-speed-voltage data obtained from the cantilever experiment.

4.2.6 Summary of constants for motor model

Table 4.2 shows all the constants obtained in the motor equation.

4.2.7 Motor controller *

Our simulation and the robot's main computer assume that the motor controllers are current sources, i.e. they can provide the current we ask for. Our hardware motor controllers are voltage sources as they are driven by batteries and pulse width modulation (PWM). So, we convert them into current sources as follows. First, we use PWM to

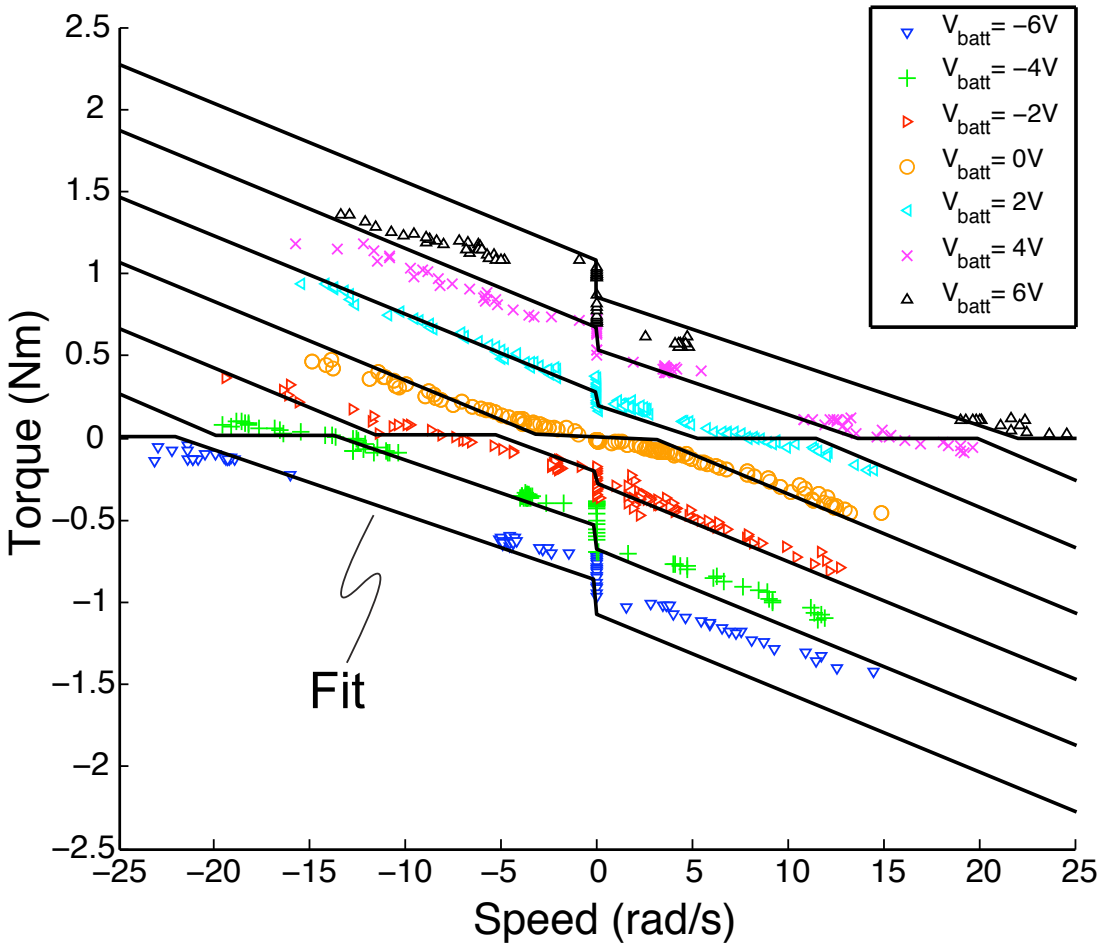

Figure 4.9: **Checking the friction model.** The frictional torque identified using a series of pulley experiments is checked with data obtained from the cantilever experiment. Various shaped dots are data and the curves are the fit. The positive work regimes are where the torque and angular velocity have the same signs (first and third quadrants). The worst data fits are for high braking torques (lower right and upper left on plots). The discontinuities at speed = 0 are from friction force reversals. The discontinuities near the torque = 0 axis are due to the reversing of the contact voltage drop when the current reverses.

Parameters	Symbol	Expts. (Specs.)
Terminal resistance (Ω)	R	1.3 (0.7)
Contact voltage drop (V)	V_c	0.7
Torque constant N m/A)	K	0.018 (0.017)
Viscous friction N m s/rad)	C_1	0
Constant friction (N m)	$C_0 = C_{0s} = C_{0d}$	0.01
Current-dependent const. friction	$\mu = \mu_s = \mu_d$	0.1

Table 4.2: **Comparison between experimental values with manufacturer's specification for the motor model.** The static and dynamic constant friction terms have the same value, i.e. $C_{0s} = C_{0d}$ and hence these are replaced with the term C_0. Similarly, static and dynamic current dependent friction terms have the same value, i.e. $\mu_s = \mu_d$ and hence these are replaced with the term μ. Note that the measured resistance is almost twice that reported in the specification sheet. Also, the brush-commutator contact voltage drop of the motor is not mentioned in the specification sheet

modulate the voltage supplied to the motor and next, we use a proportional-integral control on the motor current so that the motors can supply the needed current to drive the output shaft. In this section, we explain PWM followed by a note on how to choose the PWM frequency based on motor parameters to reduce energy losses.

Pulse width modulation to regulate voltage to the motor. Figure 4.10a shows an ideal DC motor connected to the DC motor source via an h-bridge. An h-bridge is an electronic device that is used to do pulse width modulation. The voltage applied by the DC source is constant and is V_B and is shown in figure 4.10b. The h-bridge applies pulses at a set frequency (here $1/t_p$). The duty cycle is the time the pulse is switched on. In the example above and as shown in figure 4.10c, the duty cycle is 40 % or the pulse is ON for 40 % of the pulse time t_p. The result is that the net voltage across the motor is 40 % of the constant DC voltage (or $0.4V_B$) as shown in figure 4.10d. Thus by varying the duty cycle it is possible to control the voltage applied to the motor.

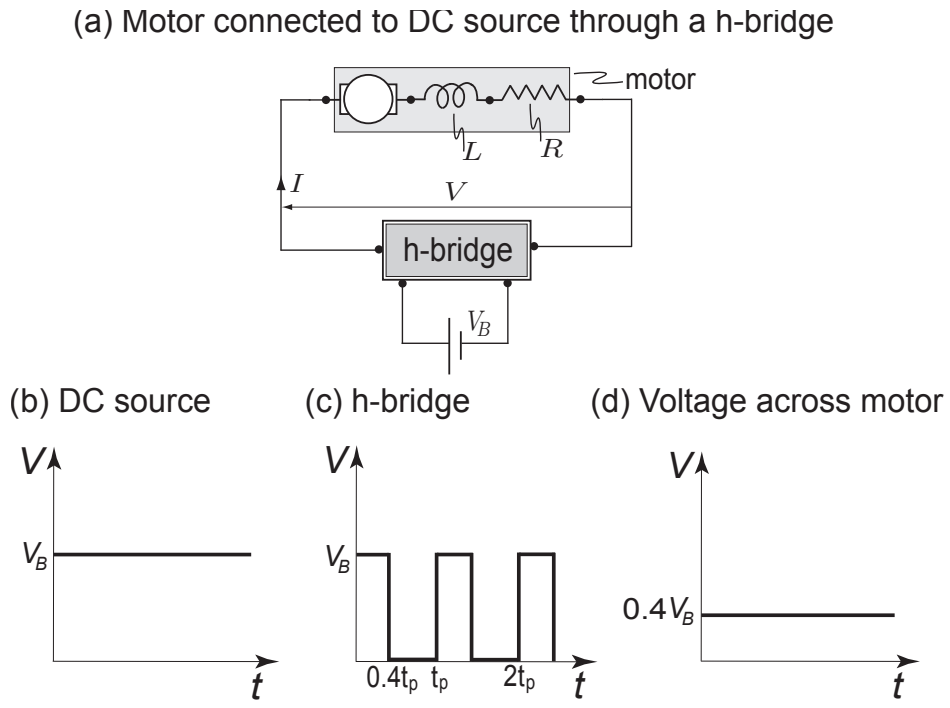

Figure 4.10: **Working of h-bridge that is used to do pulse width modulation.** (a) Motor connected to the DC source via h-bridge (h-bridge is an electronic chip that does pulse width modulation). (b) The voltage provided by the DC source is constant. (c) The h-bridge pulses the voltage at a set frequency. Here pulsing time is t_p. In this example, the pulses are on for 40 % of the pulsing time. (d) The net voltage across the motor circuit is 0.4 of DC source voltage.

Choosing the pulsing time t_p. Though we have ignored the motor inductance in our modeling in section 3.3, it plays an important role in current regulation and energy efficiency of the motors. The motor resistance (R) and motor inductance (L) pair (see figure 4.10a) acts as an integrator. From theory of R-L circuits, one can show that for the current to be more or less constant (good integrator action) we need to have $t_p \leq 0.2(L/R)$. This is shown in figure 4.11a. We illustrate the effect of using slower than ideal pulsing time with an example. Let us choose the pulsing time, $t_p = 0.5(L/R)$. The resulting current is shown in figure 4.11b and is clearly not constant. Note that average current in (b) is same as that in (a), however the power (voltage times current)

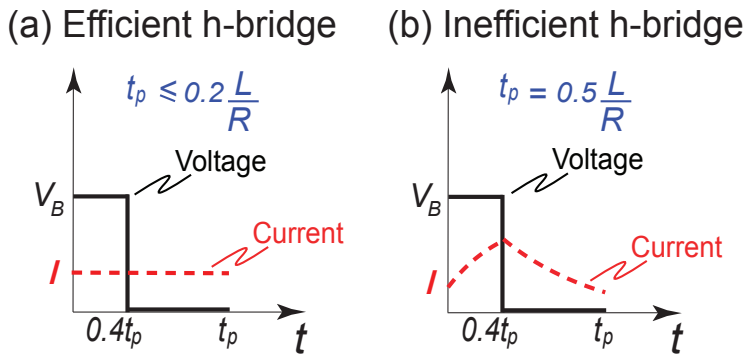

Figure 4.11: **Proper choice of pulsing time ensures power efficient working of the h-bridge.** The motor resistance (R) and motor inductance L pair act as an integrator. (a) For the current to be constant we need, $t_p \leq 0.2(L/R)$. (b) If the pulsing time is slower, here $t_p = 0.5(L/R)$, the current is non-constant. Note that although the average current in b) is that same as that in a), the power (voltage times current) usage in b) is more than that in a) because the root mean square value of the current in b) is more than that in a)

used in (b) is higher than that in (a). Thus (b) is power in-efficient as compared to (a).

For our motors, we found that the resistance R is non-constant and is given approximately as $R = R_{brush} + V_c/|I|$, where R_{brush} is the brush resistance, V_c is the contact voltage drop in the brush-commutator interface and I is the motor current. The motor resistance is high at low currents because of the second term but quickly asymptotes to R_{brush} as the current is increased. So we use $R = R_{brush}$ in the formulae for pulsing time. Our motors had a $R_{brush} = 1.3\Omega$ and motor inductance of $70\mu H$. The pulsing time for our motors should be atleast $t_p = 0.2L/R_{brush} = 0.1\mu s$. This gives a pulsing frequency of $1/t_p \approx 100$ kHz. We used this frequency to drive our h-bridge.

CHAPTER 5

TRAJECTORY GENERATOR

In this chapter, we first formulate an energy-optimal trajectory control problem. Next, we solve the optimal trajectory control problem using a parameter optimization software package. Informed by the converged optimization results, we approximate the optimal control solution. This approximate control representation will be implemented on the physical robot.

5.1 Energy-optimal trajectory

Our energy-optimal trajectory control problem is as follows.

Given the walking sequence in equation 3.1 and described by equations 3.2, 3.3, 3.4 and 3.5, with parameters constrained to those of our physical robot, we seek a control strategy that minimizes the specific total cost of transport

$$\text{TCOT} = \frac{\text{total battery energy used per step}}{\text{weight} \times \text{step length}} = \frac{\text{total battery power used}}{\text{weight} \times v_{\text{step}}} \tag{5.1}$$

where velocity v_{step} is the average velocity in a step. A trajectory is specified by hip (h) and ankle (ia & oa for inner and outer) motor currents versus time ($I_h(t)$, $I_{ia}(t)$ and $I_{oa}(t)$), and initial conditions at the start of single stance. Associated with these are angles versus time, the times of transitions to and from double stance, and the fraction of the time spent in double stance.

The numerical trajectory optimization is run with various constraints: we seek an optimal periodic gait, so we only optimize over one step and impose the periodicity constraint that the state (angles and rates) at the beginning of single stance should be equal to the state at the end of double stance; we demand that in single stance, the swing-

ing leg's foot should have sufficient ground clearance to prevent foot scuffing; motor currents must be within specified bounds to prevent overheating and damage to gear boxes; and tensional contact with the ground is forbidden. Barring numerical difficulties, we would recover identical optimization results if the trajectory was parameterized by motor angles, rates, accelerations or voltages.

5.2 Optimization sub-costs

We split the energy used per step (in equation 5.1) into three parts which we take to be decoupled:

- P_{fixed} is the power of the always-on electronics. It is the power for the non-motor on-board electronics (e.g., microprocessors, sensors, communications). We assume that P_{fixed} is independent of the control strategy. Thus the energy needed to run the electronics is given by $E_{\text{fixed}} = P_{\text{fixed}} t_{\text{step}}$, where t_{step} is the period of one step.
- $E_{\text{foot-flip}}$ is the energy to flip the swing foot up once and down once in single swing.
- E_{walk} is the energy needed to take a step, excluding the energy needed to flip the foot up and down during swing. This is the electrical energy to the hip motor and its controller and to the stance ankle motor and its controller.

Given the total robot mass M_{tot}, gravitational acceleration g, and step length d_{step}, we can re-write equation 5.1 as

$$\text{TCOT} = \frac{P_{\text{fixed}} t_{\text{step}} + E_{\text{foot-flip}} + E_{\text{walk}}}{M_{\text{tot}}\, g\, d_{\text{step}}} \tag{5.2}$$

$$= \text{COT}_{\text{fixed}} + \text{COT}_{\text{foot-flip}} + \text{COT}_{\text{walk}} \tag{5.3}$$

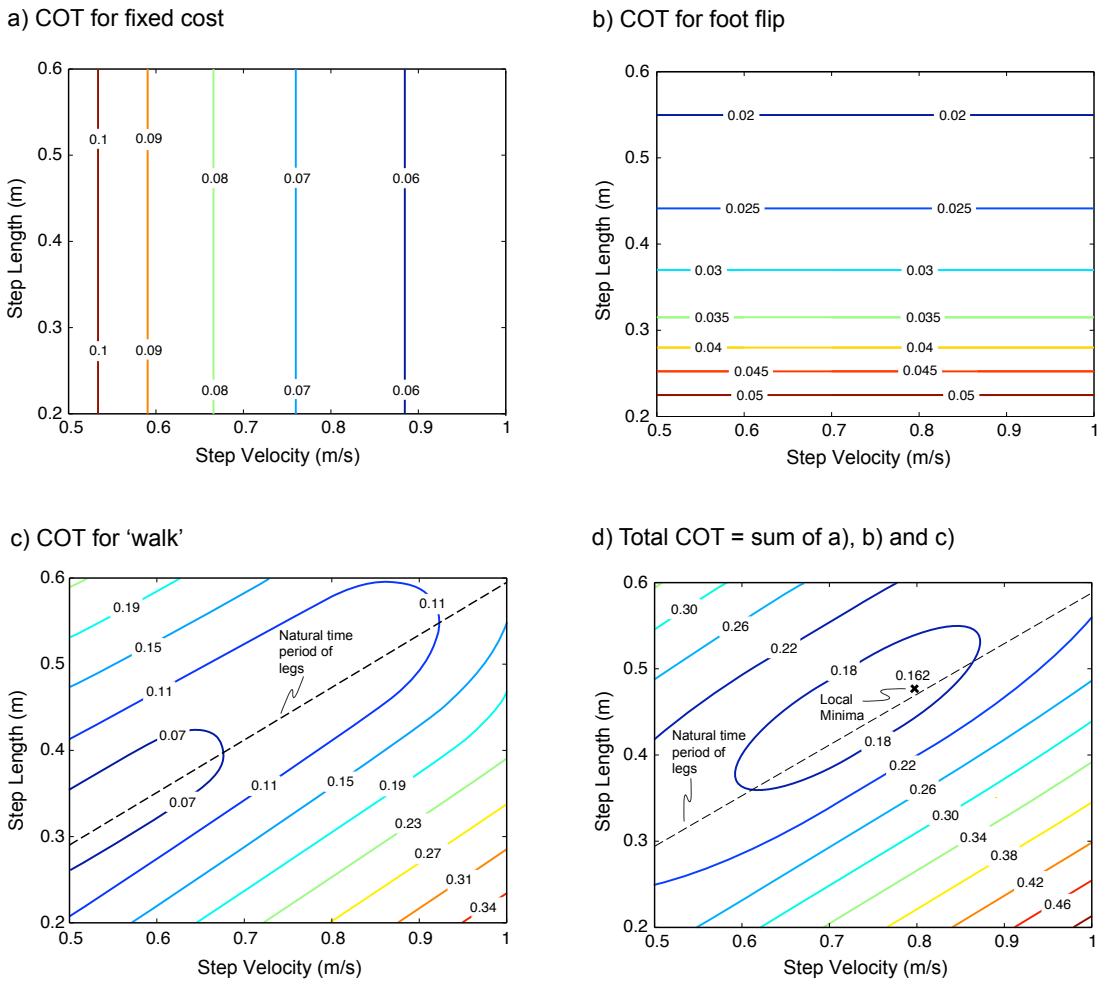

Figure 5.1: **Contour plots: COT versus step velocity and step length.** (a) COT for fixed cost (i.e., the electrical overhead from equation 5.4). This is a fixed cost per unit time. (b) COT for foot flip-up, flip-down after decoupled ankle optimization from equation 5.6. (c) COT for 'walk' (hip swing + stance ankle) from equation 5.7. The dotted line is a line of constant step frequency. (d) Total cost of transport (TCOT) given by the sum of the three constituent COTs (see equations 5.2 and 5.3). The local minimum comes from a trajectory optimization of all three costs summed, and without use of the independent trajectory optimization of COT_{walk} (i.e., it is not found using the contours from c).

where $\text{COT}_{\text{fixed}}$, $\text{COT}_{\text{foot-flip}}$, and COT_{walk} are COTs corresponding to fixed cost, flip the foot up and down in single stance, and motor use for push-off and leg-swing, respectively. The step time is given by $t_{\text{step}} = d_{\text{step}}/v_{\text{step}}$.

Figure 5.1 shows the contour plot of various constituent COTs versus step length and step velocity. As in [97], some independent optimizations are possible, as discussed below.

5.2.1 Fixed-cost (electrical overhead) COT

We can re-write the fixed cost COT from equations 5.2 and 5.3 as follows,

$$\text{COT}_{\text{fixed}} = \frac{P_{\text{fixed}}}{M_{\text{tot}}\, g\, v_{\text{step}}} \tag{5.4}$$

The overhead power for our robot is essentially independent of motor use, at $P_{\text{fixed}} = 5.15$ W. From equation 5.4 we see that $\text{COT}_{\text{fixed}}$ is inversely proportional to the step velocity and, for a given average speed v_{step}, independent of step length. Thus this 'basal energy' contribution is minimized by maximizing the overall robot speed v_{step} (as noted in [85] for example). Figure 5.1a shows the contours for the fixed-cost (electrical overhead) COT. As evident in equation 5.4 and from the figure 5.1a, the COT for fixed cost is inversely proportional to the step velocity and independent of step length. Thus, the fixed cost COT favors walking at fast speeds. This cost is hardware, and low-level control, limited. Despite care in hardware design, $\text{COT}_{\text{fixed}}$ was ultimately our biggest cost (49% of the TCOT in the final controller).

5.2.2 Foot-flip COT optimization

The truncated-heel foot shape was designed to give adequate ground clearance in swing if the swing toes are pointed up towards the shins. For robustness we use this full flip-up at every step, executing the flip-up quickly at the start of swing and lowering the foot quickly once the swing leg is sufficiently forward of the stance leg. We use the same flip-up and flip-down control independent of the walking speed or step length. We found that 0.25 s was fast enough for both flip-up and flip-down. Because the feet are light we assume that leg swing has no effect on ankle torques and currents, and vice versa.

Thus we did a decoupled optimization on the strategy for lifting (and, identically, for lowering) the foot the specified angle in the specified time (see appendix E, section E.1).

Figure 5.1b shows the contours for foot-flip COT after this optimization. As evident in equation 5.6 and from the figure 5.1b, the COT for foot-flip is inversely proportional to the step length and is independent of step velocity. Thus, the foot-flip COT favors walking at big step lengths.

Had we, instead, named a minimal foot clearance, the optimization would find a minimal motion that just makes that clearance. For example, if we only insist that the foot clear the ground the optimization finds a motion where the swing foot barely clears the ground for the whole swing phase. Here, there is a clear trade-off between energy use and robustness, and for this part of the control we choose robustness (big ground clearance by fast early foot lifting and fast late foot lowering).

The numerical flip-up and flip-down optimizations sought an ankle current as a function of time $I_a^f(t)$, which starts the ankle motors from rest, turns the feet an angle of 1.7 rad in 0.25 s, and returns to rest, while minimizing the electrical energy to do so. This

optimization used the full motor model, accelerating the motor rotor inertia, but neglected the mass of the foot, the return spring and the Achilles-tendon compliance (taking the tendon to be inextensible). The strategy for raising is the same as for lowering. Thus our cost function for flip-up-flip-down is,

$$\mathrm{E}_{\text{foot-flip}} = 2 \int_{t=0}^{t=0.25s} [P_{\text{foot-flip}}]^{+} \, dt \tag{5.5}$$

where $P_{\text{foot-flip}}$ is the power in the motor that turns the foot attached to the swing leg in single stance. The factor of 2 accounts for the inclusion of both flip-up and flip-down. To describe our approximation that the system is non-regenerative (the motors are not efficient at recharging the batteries when doing negative work) we define the function $[P]^{+}$ as follows, $[P]^{+} = P$, if $P \geq 0$ and $[P]^{+} = 0$, if $P < 0$.

The optimization is non-trivial because of the relatively complex motor equation 3.9. To solve the problem numerically, we use a discrete approximation for current versus time with 33 intervals of piecewise linearly-varying current (see appendix E, section E.1).

The result of the optimization is a foot-flip cost $\mathrm{E}^{\text{const}}_{\text{foot-flip}}$ that is independent of step length and speed.

Thus the foot-flip cost COT optimization problem from equations 5.2 and 5.3 is

$$\mathrm{COT}_{\text{foot-flip}} = \frac{\mathrm{E}^{\text{const}}_{\text{foot-flip}}}{M_{\text{tot}} \, g \, d_{\text{step}}}. \tag{5.6}$$

The optimized ankle flip-up current is shown between about 0.66 s and 0.9 s in the lower solid red curve in figure 5.2. The profile is approximately explicable in terms of what we would get by minimizing either of the standards for 'energy' optimization: $\int I^2 dt$ or $\int \textit{power} \; dt$. The former optimization would give current varying linearly in time and the latter a bang-bang (impulsive) control. Thus we see a ramp with a slight spike at

each end. In addition there is a slight bias that seems to counter mechanical friction. As mentioned, the mirrored current is used for flip-down.

Because the foot-flip energy cost is independent of speed or step length, from equation 5.6, we see that $\mathrm{COT}_{\text{foot-flip}}$ is inversely proportional to the step length and independent of step velocity. That is, the foot-flip contribution to the TCOT is minimized by maximizing step length. Taking this cost together with the fixed cost and optimizing would give the longest possible steps at the greatest possible forward speed. The optimal solution is not maximally fast with maximally long steps, however, because of the swing and push-off costs, both of which penalize speed and step length.

5.2.3 'Walk' COT optimization

The first two terms in the expression for energy cost, the electrical overhead fixed cost and the foot flip up/down cost, have been reduced to simple dependencies on step length and step velocity.

The remaining 'walk' cost is given by

$$\begin{aligned} \mathrm{COT}_{\text{walk}} &= \frac{\mathrm{E}_{\text{walk}}}{M_{\text{tot}}\, g\, d_{\text{step}}} \\ &= \frac{\displaystyle\int_{t=0}^{t=t_{\text{step}}} \sum [\mathrm{P}_{\text{walk}}]^{+}\, \mathrm{dt}}{M_{\text{tot}}\, g\, d_{\text{step}}} \end{aligned} \tag{5.7}$$

where P_{walk} are the various motor (plus controller) powers. The summation here is over all the motors involved in powering the walk excluding the foot flip-up and flip-down. Again we assume that our batteries cannot regenerate (see appendix E, section E.2 for more details).

From figure 5.1c, we note that the COT for walking has a strong frequency depen-

dence. Walking at frequencies other than those which are close to the 'natural' leg frequency are expensive. We found that the minimum for the 'walk' COT occurs at step length of 0.13 m and step velocity of 0.21 m/s (step frequency = 1.6 Hz). That the 'walk' COT is minimized for a non-zero step length came as a surprise because simple models of walking with work-based cost have shown that zero step lengths are optimum [40, 61, 62, 96].

Figure 5.1c shows a contour plot of the 'walk' COT (hip and stance-ankle motors) as a function of prescribed step length and speed. Note that step frequency is step velocity divided by step length. So, for a line through the origin, the reciprocal of the slope is the step frequency. The dotted line in the figure is one such line, a characteristic frequency of this machine. In this case it is the frequency of oscillation of the robot, a double pendulum with a spring at the intermediate joint, oscillating about vertical (but not the real root associated with the unstable falling motion).

5.2.4 Total COT optimization

The sum of the three costs are calculated and the TCOT is minimized with numerical optimal control using 33 mesh points in single stance and 9 in double stance for the hip and ankle.

Finally, we add the COTs for fixed cost, foot-flip and walking to get the total TCOT as a function of step length and step velocity. The TCOT as a function of step velocity and step length is shown in figure 5.1d. The optimum walking speed for the robot is 0.77 m/s at a rather large step length of 0.48 m and a step frequency of 1.6 Hz. The minimum TCOT is 0.163.

The resulting hip and ankle currents are the solid red curves in figure 5.2.

As expected from simple models e.g., [98] the optimization does use a pre-emptive push-off (big ankle extension current at 0.55 s). The shape of the hip current was not anticipated because we expected the hip motor would tend to speed the swing to reduce step size and collision costs. But it can be explained post-facto thus: the optimal leg swing is almost passive (perhaps balancing collisional cost reduction with small steps with flip-up costs). But motor friction losses need to be made up. The optimization chooses a motor current to most efficiently make up for frictional losses, and this involves using the motor when it is most efficient, namely when it is spinning fast.

5.3 Simplifying the optimal trajectory

The energy-optimal trajectory control problem presented in section 5.1 is infinite dimensional with an approximate numerical solution described by the current values at a finite set of discrete times. As the grid resolution is increased with more grid points, the numerical optimum is improved and the number of control parameters increases. We estimated reasonable convergence to the supposed optimum using a total of 126 parameters, which we consider too complex to implement on the robot given our goal of eliminating unnecessary complexity. Furthermore, the optimal solution is time-based and we require that portions of the current trajectories be synchronized to events, such as heel-strike.

Also, our energy-optimal solution is based on trajectory optimization and does not consider gait stability. Our energy-optimal trajectory is not necessarily stable at all.

To make the controller simpler, but still energy effective and stable, we first simplify

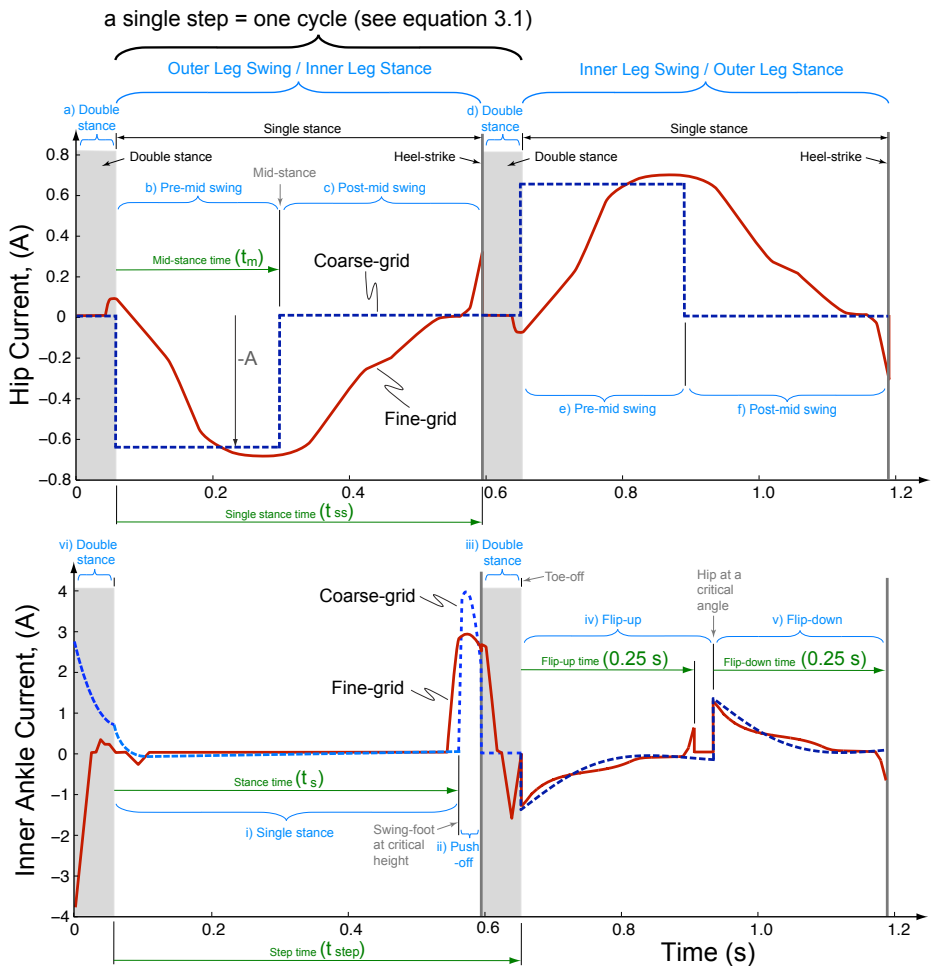

Figure 5.2: **Numerically fine-grid optimal (solid red) and coarse-grid optimal (dashed blue) motor currents.** Logical states of the state-machines (a-f, i-vi) are shown with respect to the coarse-grid (dashed blue) curves. Because the coarse-grid optimization has a different period than the fine-grid optimization (1.22 s versus 1.19 s) the red curve is slightly contracted to fit on the graph. The upper graph is the hip current; positive current swings the inner legs forwards. The lower graph is the inner ankle current, positive current extends the foot. In the left half of the graphs the outer leg is in swing and the inner in stance; in the right half the roles are reversed. The gray bands are periods of double support. For the hip, the current in the right half is the negative of the left half. But the inner ankle current lacks this symmetry because the ankle has a different job in stance (left half) than swing (right half). Amongst the parameters tuned in the coarse-grid (dashed blue) solution are the times t_m, t_s, t_{ss} and t_{step}. The fine-grid solution (solid red) is described with about 126 parameters. The coarse-grid solution is described with 15 parameters. The fine-grid and coarse-grid optimized currents predict TCOT = 0.167 and 0.18 respectively.

the description of the near-optimal trajectory (this section), and then add a reflexive (event-based) feedback controller (see chapter 7).

Optimizing the coarse-grid description. First, according to the KISS (keep it simple stupid) design principle, we use a simpler coarse-grid representation that has few parameters. We did this by looking at the general shape of the fine-grid optimization (red solid lines in figure 5.2) and choosing intervals where we could use a simpler controller. The coarse-grid form is chosen manually so as to best catch the features of the fine-grid optimization but with fewer terms. The values of the parameters (durations, and constants in the impedance controls) are chosen by again doing a trajectory optimization, but now parameterized with the simplified parameters. Our goal is to achieve a gait with close to the same TCOT, informally doing a simultaneous optimization of both energy use and simplicity.

For example for the foot-flip we replace the 33 piecewise-linear currents at the grid points with just 3 parameters of a proportional-derivative (P-D) controller. The three parameters are the proportional gain, derivative gain and the reference angle. The optimal cost is 1.06 J while the cost with the coarse-grid representation is 1.1 J. We note that in going from the fine-grid solution to the coarse-grid control parameterization for the foot-flip optimization we have increased the cost by 4% while decreasing the number of parameters from 33 to 3.

Similarly, we simplified the controls for the hip and ankle motors for the walk optimization. For the energy-optimal trajectory control problem we presented earlier, we had 126 parameters in all and a TCOT of 0.167. With our coarse-grid representation we reduced the parameters to 15 while increasing the TCOT to 0.18, a 7% increase in cost. Table 7.1, column 1 and 2 compare the cost obtained for the optimal gait with the

coarse-grid representation.

Concurrent augmented state machines. Each joint is controlled by its own concurrently running augmented finite state machine, traversing a set of logical states (see appendix G). The *logical* state describes which portion of control code is active. This is augmented by the *dynamical* state which describes the measured and estimated aspects of the robot (angles, angular rates, elapsed time since last transition, binary variables for whether each foot is on the ground or not, etc.). Each joint's state machine has access to the full dynamical state and is also informed by global commands (such as for steering, starting, shutdown). For each joint, a gait cycle (2 steps) traverses a circle of states. The transitions are triggered by events which are thresholds in the dynamic state or time (e.g., 'change logical states when ankle angle has reached 0.126'). The individual state machines do not communicate with each other explicitly but are synchronized through the shared data, and shared dynamic state estimation. Within one state there is a tight (2 kHz) feedback loop that controls, say, the current to the inner ankle (ia):

$$\begin{aligned} I_{\mathrm{ia}} &= f(\text{full dynamical state of the robot}) \\ &= A + C_1\theta_{\mathrm{ia}} + C_2\dot{\theta}_{\mathrm{ia}} + D \end{aligned} \tag{5.8}$$

Although the architecture allows any function f of the full dynamical state, we only use simple functions with dependence on only some dynamical-state variables; at any instant most joints are in a 1-degree-of-freedom impedance control mode, as expressed by equation 5.8. The redundant constant D expresses a dependence of the within-state control on the dynamical state at the start of the state. In most cases the within-state control is fully local. However, couplings more general than equation 5.8 are allowed and sometimes used. For example, during single stance the ankle joint current is based on the absolute angle of the foot (which is a state estimation based on data from several sensors). And one of the dynamical states is the binary variable saying whether the robot

is, overall, going too fast or too slow (see chapter 7 for more details).

We call this control architecture "reflex"-based because the change of state is triggered by an event. Because each state has its own motor program, in effect motor programs are triggered by events. If the triggers are defined to occur at minor changes in dynamical state, say the passage of 10 ms, and the changes of the within-state parameters are small, and if complex forms of f are allowed within a state, then the machine can be a close approximation of continuous gain scheduling. Similarly, if the functions f within a state are allowed to be arbitrarily complicated, then the control can be arbitrarily close to any non-linear state feedback controller.

The parameters for this controller (thresholds, numbers of states, gains within a state) are developed by a sequence of optimizations and human decisions: fine-grid trajectory optimization; coarse-grid trajectory definition and optimization; and stabilizing controller definition and optimization. In the final implementation, the resulting controller had this form (these logical states are shown in figure 5.2):

Hip logical states. The hip has six logical states per two-step cycle, three states for each step.

a) Double stance. Starts at heel-strike; ends at toe-off. Hip current = 0.

b) Pre-mid swing. Starts at toe-off; ends when stance leg is vertical (mid-stance). Hip current = constant.

c) Post mid-swing. Starts at mid-stance; ends at heel-strike. When on the nominal trajectory hip current = 0.

d,e,f) Repeat. The same 3 states are visited again but with the roles of the legs reversed.

Inner ankle logical states. The outer ankle has an identical (mirrored) set of states. There are 6 logical states for a 2-step cycle. Unlike the case for the hip, all 6 logical states need description because the foot has both stance and swing roles in a two-step gait cycle.

i) Inner-leg single stance. Starts at outer leg toe-off; ends when outer ankle reaches, from above, a prescribed height above the floor. Absolute foot angle, not ankle joint angle, is controlled (with the standard impedance controller, equation 5.8).

ii) Inner-leg push-off. Starts when the outer (swing) foot is a critical distance above the ground; ends at outer-leg heel-strike. It is the same controller structure as inner-leg single stance (above), but with a different target absolute angle and different compliance (and with damping of zero).

iii) Double-stance after inner-leg single stance. Starts at outer-leg heel-strike; ends at inner-leg toe-off. Ankle current is zero (and foot tends to lift because of tensioning spring).

iv) Inner-leg flip-up. Starts at toe-off; ends at mid-stance. A compliant controller tries to put the foot at its uppermost position (target ankle-joint angle).

v) Inner-leg flip-down. Starts when stance leg is at a critical angle; ends at inner-leg heel-strike. Compliant controller tries to put foot at a specified absolute angle.

vi) Double stance after heel-strike. Starts at inner-leg heel-strike; ends at outer-leg toe-off. Compliant control aims for target absolute angle.

Note that the simple command profiles generated by this state machine are far from those from the optimization (i.e., compare the solid red lines with blue dashed lines in figure 5.2), but they are close in their net energy cost.

5.4 Other Results

5.4.1 Optimization validation

To verify and develop our simulation and optimization, we carried out two optimizations whose solutions we already knew. First, we locked the ankles of the robot, simulated motion on a shallow ramp, used absolute value of the hip motor work in the TCOT equation and ran the optimization. The optimization discovered two passive solutions for different initial guesses consistent with earlier findings [41, 70]. Second, we put most of the mass in the hips, made the legs light, used the absolute value of mechanical work done by the ankles and hip in the TCOT equation and ran the optimization. The optimization discovered classical inverted pendulum walking with impulsive push-off before heel-strike for reasonable step length and step velocity combinations, consistent with earlier findings [96, 98]. Please see the appendix D for more details.

5.4.2 Optimal trajectory control solution

In early trials the heel-strike collisions made sounds, due to the large steps that the optimization requested, that made us fearful of damage. To avoid such large collisions, the step length was constrained in the fine-grid optimizations to 0.38 m, which, after optimization, led to a step velocity of 0.64 m/s, and TCOT= 0.167. The fixed costs accounted for 49% of the TCOT. In this fine-grid simulation, the constant cost per step associated with flipping the foot in single stance accounted for 23% of the TCOT. The remaining 28% was used by the hip swing motor (11%) and stance ankle motors (17%). Thus, almost half of the energy budget is spent powering the electronics and about a quarter is spent in accelerating and decelerating the swing ankle motors to do flip-up

and flip-down.

5.4.3 Comparison with alternative optimizations

Our primary optimization is of the total energy budget as estimated by a high-fidelity model. Traditionally in optimization one of two other proxy objectives are used: 1) $\int \text{torque}^2\, dt$ (often, unfortunately, called 'power'), or 2) Positive mechanical work.

Using the mechanical properties (masses, inertias, geometry) of our robot we can also optimize the gait using these objectives. Then we can find the currents and actual energy cost to implement the resulting trajectories.

Minimizing positive mechanical work we find a gait that has a step length of 0.57 m and step velocity of 0.97 m/s and with a TCOT of 0.3, nearly twice the TCOT of the optimal trajectory based on actual electrical cost.

Minimizing $\int \text{torque}^2\, dt$ we find a gait that has a step length of 0.67 m and step velocity of 1.1 m/s and with a TCOT of 0.23, about 40% greater than the TCOT of the optimal trajectory used as the basis of our control.

Finally we minimized $\int I^2 R\, dt$ for the hip and ankle motors, but using our full motor models (with a brush voltage, load-dependent friction and motor inertia). This minimizes the non-work parts of the actual electrical costs, neglecting the 'fixed' overhead costs. The resulting current profiles yielded a walk with a step length of 0.56 m, a step velocity of 0.92 m/s and a TCOT of 0.18, surprisingly not much bigger than the 0.167 of our full electrical cost optimization.

That is, all three optimizations yield coordinations that are somewhat reasonable,

given that at this point in the development of legged robots, robots differ by factors of 10 or more in energy effectiveness. However, once one gets to the margin where a 10 percent improvement is substantial (as for most any commercial machine or transportation device), the only proxy that is a reasonable substitute for the full energy optimization we used is $\int I^2 R\, dt$, while using the full non-ideal motor and transmission models in the simulation.

CHAPTER 6

ENERGY BASED CONTROL OF A 2-D POINT-MASS WALKING MODEL *

In this chapter, we present an energy based control of a 2-D point-mass model of walking. The ideas developed here will be extended to stabilize Ranger's nominal gait (see chapter 7).

6.1 Point-mass model of walking *

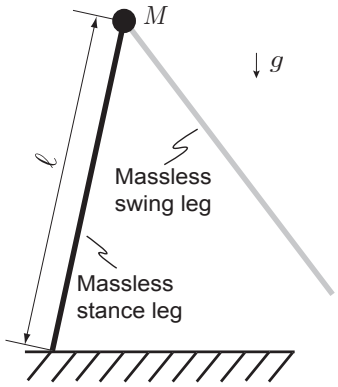

Figure 6.1: **2-D point-mass walker.** The walker consists of two massless legs of length ℓ with a point-mass M at the hip joint. There are two actuators; an actuator at the hip that controls the step length and an actuator on the stance leg that can generate an impulsive push-off along the stance leg.

Figure 6.1 shows a simplistic model of walking. The model has mass-less legs of length ℓ and a point-mass M at the hips. Gravity g points down as shown. The model has two actuators; a hip actuator and a stance leg actuator. As the legs are massless, we assume that the hip actuator can place the swing leg at the desired position instantaneously fast. The stance leg linear actuator is used to generate an impulsive push-off along the stance leg. We assume perfect state estimation, no saturation limits on actuators, ignore

scuffing of the swing leg with the ground, and assume that the grounded leg does not slip.

6.2 Methods *

6.2.1 Nominal and deviated trajectory trajectory *

Figure 6.2 shows a typical step for this model. The point-mass model starts in the upright position or the mid-stance position as shown in (a) with nominal velocity $\dot{\theta}_0$. The model moves passively to state just before heel-strike in (b) with hip velocity $\dot{\theta}_0^-$. Here, the stance leg applies an impulsive push-off P_0 just before heel-strike (we assume that the point-mass does not take-off during the impulsive push-off and this assumption is justified because the impulsive push-off is immediately followed by heel-strike in (c)) and the hip actuator position the swing leg at the angle $2\theta_0$. Next, in (c) the swing leg collides with the ground and becomes the new stance leg. The hip velocity after collision is $\dot{\theta}_0^+$. Finally, the new stance leg moves passively to the state (d) where the model ends up in the upright position or the mid-stance position with a velocity of $\dot{\theta}_0$.

Figure 6.3 shows a typical step due to the effect of disturbance. Due to a disturbance, the point-mass model starts at (a) in the upright position or the mid-stance position with velocity $\dot{\theta}_d \neq \dot{\theta}_0$. The model moves passively to state just before heel-strike in (b) with hip velocity $\dot{\theta}^-$. Here the stance leg actuator applies an impulsive push-off just before heel-strike and the hip actuator positions the swing leg at the angle 2θ. Next, in (c) the stance swing leg collides with the ground and becomes the new stance leg. The hip velocity after collision is $\dot{\theta}^+$. Finally, the new stance leg moves passively to the state (d) where the model ends up in the upright position or the mid-stance position.

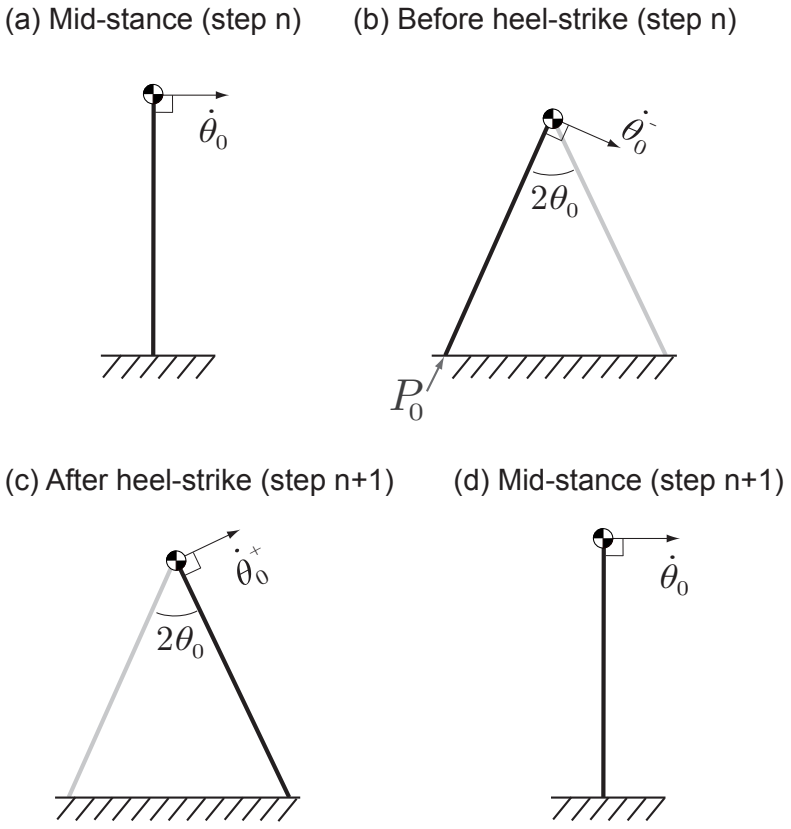

Figure 6.2: **Nominal gait for the walker.** The walker starts in the upright or mid-stance position in (a). Next, just before heel-strike in (b) the stance leg applies an impulsive push-off P_0 and the hip actuator positions the swing leg at an angle $2\theta_0$. Next, after heel-strike in (c) the swing leg becomes the new stance leg. Finally, the walker ends up in the upright position or mid-stance position on the next step in (d).

The control problem is to use the measurement at mid-stance $\dot{\theta}_d$ at step n, to decide a step length 2θ and the impulsive push-off P just before heel-strike to end up with the nominal velocity $\dot{\theta}_0$ at the next mid-stance at step n+1.

Equations of motion. Here we present equations of motion for the deviated trajectory (see figure 6.3) and later use these to derive a stabilizing controller.

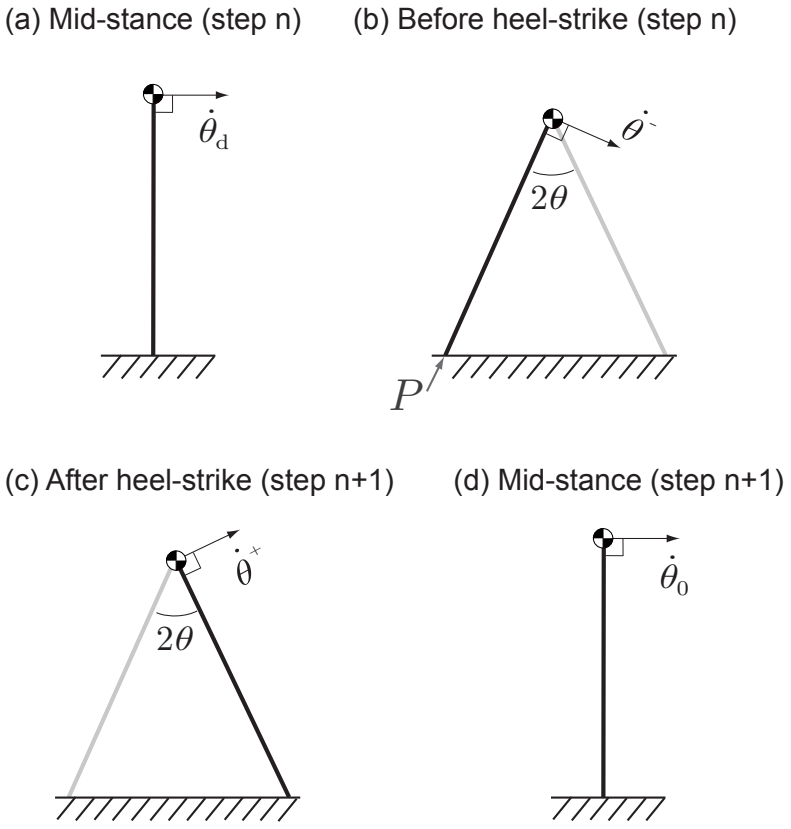

Figure 6.3: **Effect of disturbance.** Due to a disturbance, the point-mass model starts in the upright position or the mid-stance position with velocity $\dot{\theta}_d$ in (a). Next, just before heel-strike in (b) the stance leg applies an impulsive push-off P and the hip actuator positions the swing leg at an angle 2θ. Next, after heel-strike in (c) the swing leg becomes the new stance leg. Finally, the walker ends up in the upright position or mid-stance position on the next step in (d). The control problem is to use the measurement $\dot{\theta}_d$ and to control the impulse P and step length 2θ to end up with the nominal velocity $\dot{\theta}_0$ at the next mid-stance

Mid-stance position at step n (a) to instance before heel-strike at step n (b). Let the mid-stance velocity at step n be $\dot{\theta}_d$. We non-dimensionalize time with $\sqrt{\ell/g}$ here. The non-dimensional kinetic energy E_d corresponding to this velocity is given by $E_d = 0.5(\dot{\theta}_d)^2$. Let the step angle be 2θ and the velocity just before heel-strike be $\dot{\theta}^-$. Using

conservation of energy and from figure 6.3 (a) - (b) we have,

$$E_d = \frac{(\dot{\theta}_d)^2}{2} = \frac{(\dot{\theta}^-)^2}{2} + \cos\theta - 1 \Rightarrow \dot{\theta}^- = \sqrt{2(E_d + 1 - \cos\theta)} \tag{6.1}$$

Instance before heel-strike at step n (b) to instance after heel-strike at step n+1 (c). In order to relate angular velocities before heel-strike $\dot{\theta}^-$ to that after heel-strike $\dot{\theta}^+$, we do an angular momentum balance about the impending collision point. For the non-dimensional impulsive push-off P (impulse is non-dimensionalized with $M\sqrt{g\ell}$) before or after heel-strike we have from figure 6.3 (b) - (c),

$$\dot{\theta}^+ = P\sin 2\theta + \dot{\theta}^- \cos 2\theta \tag{6.2}$$

Instance after heel-strike at step n+1 (c) to mid-stance position at step n+1 (d). Let the mid-stance velocity on step n+1 be $\dot{\theta}_0$. The kinetic energy associated with this mid-stance velocity is given by $E_0 = 0.5(\dot{\theta}_0)^2$. Using conservation of energy we can relate the angular velocity after heel-strike ($\dot{\theta}^+$) with the angular velocity at mid-stance. From figure 6.3 (c) - (d) we have,

$$\frac{(\dot{\theta}^+)^2}{2} + \cos\theta - 1 = \frac{(\dot{\theta}_0)^2}{2} = E_0 \Rightarrow \dot{\theta}^+ = \sqrt{2(E_0 + 1 - \cos\theta)} \tag{6.3}$$

Relating energy from mid-stance position at step n to energy at mid-stance position at step n+1. To relate the energy at mid-stance at step n with that at mid-stance at step n+1, we put equation 6.1 and 6.3 in equation 6.2 and solve for E_0,

$$E_0 = f(E_d, \theta, P) = 0.5\left(P\sin 2\theta + \sqrt{2(E_d + 1 - \cos\theta)}\cos 2\theta\right)^2 - (1 - \cos\theta) \tag{6.4}$$

In the next section we present the control problem and its solution based on equation 6.4.

6.2.2 The control problem and its solution *

We want to regulate the mid-stance velocity at any step to its nominal value $\dot{\theta}_0$ and subsequently the kinetic energy $E_0 = 0.5\dot{\theta}_0^2$. Consider figure 6.3a where the mid-stance velocity is $\dot{\theta}_d \neq \dot{\theta}_0$. From equation 6.4 we see that using a combination of push-off impulse P and the step length 2θ it is possible to modulate the angular velocity and subsequently the kinetic energy on the subsequent step as shown in figure 6.3d. That is, we try to drive the system to its nominal value in one step. This kind of full correction of deviations is called a dead-beat control (see [1] page 201). We call this type of control as a one-step dead-beat control.

6.3 Results *

6.3.1 Nominal gait *

The nominal gait is characterized by a nominal step length $2\theta_0$, nominal mid-stance velocity $\dot{\theta}_0$, and a nominal impulsive push-off P_0. To calculate the nominal impulse, we put $E_d = E_0$, $P = P_0$, and $\theta = \theta_0$ in equation 6.4 and solve for impulse P_0,

$$P_0 = \sqrt{2\left[E_0 + 1 - \cos(\theta_0)\right]}\tan(\theta_0) \tag{6.5}$$

We present results for a step length of $2\theta_0 = 0.4$ and mid-stance velocity of $\dot{\theta}_0 = 0.2$ (Note that this corresponds to a step length of about 0.4 m and mid-stance velocity of 0.64 m/s for Ranger). Solving for nominal impulse using equation 6.5 gives, $P_0 = 0.0573$.

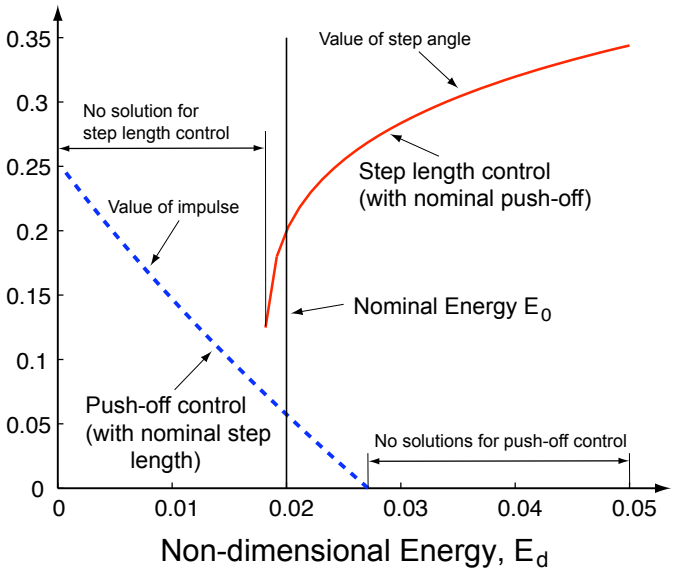

Figure 6.4: **Push-off control and step length control considered separately to do a one-step dead-beat control.** Push-off control with step length maintained at the nominal value (blue dashed line). Push-off control works best to inject energy into the system and is evident from the fact that there are no solution in the high disturbance energy range. Step length control with push-off maintained at the nominal value (red solid line). Step length control works best to extract energy out of the system and is evident from the fact that there are no solution in low disturbance energy range.

6.3.2 Push-off control *

In the push-off control, we maintain the step length at $2\theta_0$ and given the disturbance kinetic energy E_d, we try to solve for a push-off impulse P using equation 6.4 to get back to the nominal velocity $\dot{\theta}_0$ and hence nominal kinetic energy E_0 at the next step.

$$E_0 = f(E_d, \theta_0, P) = 0.5\left(P \sin 2\theta_0 + \sqrt{2(E_d + 1 - \cos\theta_0)}\cos 2\theta_0\right)^2 - (1 - \cos\theta_0) \quad (6.6)$$

Figure 6.4 (blue dashed line) gives the non-dimensional impulse P (holding the step length at $2\theta_0$) needed to get back to the nominal energy E_0 from the disturbance characterized by the energy E_d. We are able to find an impulse for all energy ranges, $E_d < E_0$.

However, we are not able to find an impulse solution for certain ranges $E_d > E_0$ as shown in the figure. Note that we do not allow for negative impulse which correspond to suction from the ground. Thus the push-off control works best to inject energy in to the system.

6.3.3 Step-length control *

In the step-length control, we maintain the push-off at P_0 and given the disturbance kinetic energy E_d, we try to solve for the step length 2θ using equation 6.4 to get back to the nominal velocity $\dot{\theta}_0$ and hence nominal kinetic energy E_0 at the next step. Thus from equation 6.4 we get,

$$E_0 = f(E_d, \theta, P_0) = 0.5\left(P_0 \sin 2\theta + \sqrt{2(E_d + 1 - \cos\theta)}\cos 2\theta\right)^2 - (1 - \cos\theta) \quad (6.7)$$

Figure 6.4 (red solid line) gives the step length θ (holding the push-off impulse at P_0) needed to get back to the nominal energy E_0 from the disturbance characterized by the energy E_d. We are able to find a step length for all energy ranges, $E_d > E_0$. However, we are not able to find a step length for certain ranges $E_d < E_0$ as shown in the figure. Thus the step length control works best to extract energy out of the system.

6.3.4 Two one-sided controllers: switching between push-off control and step-length control in different energy regimes *

Informed by the result above we note that the push-off control works best to inject energy in to the system and the step length control works best to extract energy out of the system. Hence, we propose two one-sided controllers that cover the entire range of

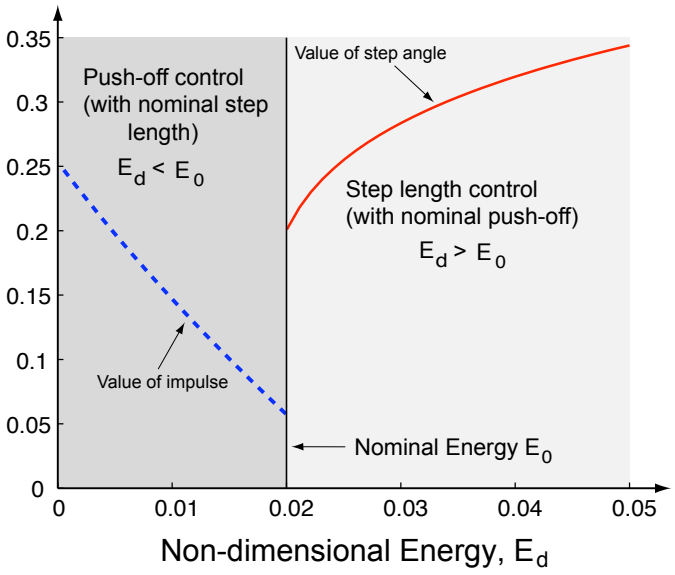

Figure 6.5: **Two one-sided controllers to do a one-step dead-beat control.** If the energy of the system is less than the nominal energy than a push-off control is used to inject energy into the system (blue dashed line). If the energy of the system is more than the nominal energy than a step length control is used to extract the excess energy from the system (red solid line). Such a controller works on the entire range of disturbances that we have considered here.

disturbance energies E_d we considered here. If the kinetic energy of the system is less than the nominal energy i.e. $E_d < E_0$, we use a push-off control. If the kinetic energy of the system is more than the nominal energy i.e. $E_d > E_0$, we use the step length control. Control using this form of the two one-sided controllers is shown in figure 6.5.

CHAPTER 7

STABILIZING CONTROLLER

The nominal trajectory generated by simplified control representation presented in chapter 5 in section 5.3 is not stable. In this chapter, we apply our event-based, discrete, intermittent, feed-forward controller to stabilize Ranger's nominal trajectory.

The flow of this chapter is as follows. First, we present an intuitive control scheme for stabilizing Ranger's nominal gait. The scheme is based on the energy-based control idea presented in chapter 6. We noted in section 7.1 that for successful application of our stabilizing controller it is important to properly select the four key quantities; the section or instance of time to collect sensor data, the measurements to be made, the output variables to regulate, and type of control actions to take. So next, we identify the four key quantities for the control scheme. Finally, using the linearization about the nominal trajectory and using a discrete linear quadratic regulator (DLQR) we implement our stabilizing controller on Ranger.

7.1 The control heuristic behind stabilization

The key heuristic idea behind stabilization was presented for a point-mass model of walking in chapter 6. Here, we adapt this idea to stabilize Ranger's nominal trajectory.

Mid-stance dynamic state evaluation. The angular speed of the stance leg at mid-stance is key for evaluating stability: the robot has to make it to that point at a positive, but not-too-fast, speed. It is also a good time for state estimation because it is not near in time to collisional vibrations which contaminate the dynamic-state estimation. Thus we use that 'mid-stance' time as a key time for evaluating the dynamical state.

This contrasts with typical numerical simulations of passive-dynamic robots that take (Poincaré) sections just before or just after heel-strike, an apparently natural transition time. The top-dead-center (mid-stance) position is sufficiently useful, however, that it is worth introducing an artificial logical-state boundary there.

How can the robot can fall down? Given that we do all we can to preclude tripping with our foot-lifting plan, the only possible falls are falling forwards and falling backwards. Because leg-swing speed is limited, if the robot is going too fast the foot will not swing forwards enough and the robot will fall forwards. On the other hand, if the robot is going too slowly the stance leg will not make it over the top-dead-center position and it will fall backwards. Thus the essential control needs to get the center-of-mass through the top-dead-center position at a positive speed close to nominal. It must speed up or slow down the walking, as needed, to keep this top-dead-center speed positive but slow.

Push-off and step-length regulation. There are two main ways to regulate the overall center-of-mass speed. One is regulation of the ankle push-off: the size and timing of the push-off affect the robot speed (bigger push-off leads to bigger speed of the next step). The other is by changing step length (increasing step length decreases speed of the next step). For simple walking models, it is known that the energy lost during heel-strike collision scales with the square of step length at a given forward speed [40, 41, 61, 92, 96]. Increasing the step length increases collisional loss and slows the robot. No other motor actions (leg swinging, foot motions) have a significant effect on robot speed.

However, push-off control and step length control each have limitations. To decrease speed, push-off can only be decreased to zero. Thus push-off regulation has limited effectiveness for slowing the robot. On the other hand, decreasing the step length is also bounded in effectiveness because the minimum step length is zero. And, more

practically, the step time has to be long enough for the foot to flip up and flip down, so very short steps are not possible.

To circumvent these limitations we use each mode where it is most effective. If the robot is going too fast we use increased step length to slow the robot. If it is going too slow we use increased push-off to increase speed. In this way, we expect our quasi-linear controller to have a larger controllable region.

7.2 Design of two one-sided controllers

To implement this intuitive control scheme with our feedback control framework (see chapter 2, section 2.2), we need to decide the four key variables; a suitable section or instance of time to take measurements (n), measurements (x_n), output variables to be regulated (z_n) and nature of controls actions (U_n). We choose our measurement section n, as the instance of time when the stance leg is vertical, henceforth called as the mid-stance position. At the mid-stance position we measure the stance leg rate x_{sr}, hip angle x_{ha} and hip rate x_{hr}, i.e. $x = [x_{sr} \;\; x_{ha} \;\; x_{hr}]'$. We select different outputs z_n and control actions U_n depending on the controller used and is discussed next.

7.2.1 Regulate fast speeds using step length control

If the robot's mid-stance velocity is greater than the nominal, we alter the foot placement while maintaining the nominal push-off. We try to regulate the stance leg angular rate z_{lr} at the next midstance as well as the falling rate z_{ar} of the foot just before the next heel-strike (to prevent mis-timing the next push-off). Thus the regulated variables are $z^f = [z_{lr} \;\; z_{ar}]'$.

Relative to the nominal trajectory we have

$$\delta x_{n+1} = \mathbf{A}^f \delta x_n + \mathbf{B}^f \delta U_n^f \tag{7.1}$$

$$\delta z_{n+1}^f = \mathbf{C}^f \delta x_n + \mathbf{C}^f \delta U_n^f \tag{7.2}$$

Using DLQR cost given in equation 2.5 and equations 7.1 and 7.2 we set up a DLQR problem for determining the 2×3 gain matrix $\mathbf{K}$, which permits calculation of the U values from the x values (equation 2.6). In the DLQR we use the weights $\mathbf{R}_{zz}^f = \text{diag}\{1/\sigma_{sr}^2, 1/\sigma_{lr}^2\}$ and $\mathbf{R}_{UU}^f = \rho_{hip}\text{diag}\{1, 1\}$, where $\sigma_{ar} = 0.2$ is a user-selected characteristic foot falling rate deviation, $\sigma_{lr} = 0.2$ is the user-selected characteristic deviation in the leg rate (both in consistent units), and $\rho_{hip} = 0.5$ weights the effort by the hip actuators.

7.2.2 Regulate slow speeds using push-off control

If the robot's mid-stance velocity is less than the nominal then we increase push-off while maintaining the same step length. So we try to regulate, back to nominal, the values of three dynamic state variables: the stance leg-rate z_{lr} at the next step, the downward velocity z_{ar} of the ankle just before the next heel-strike, and the step length z_{sl} at heel-strike. These three quantities $z^s = [z_{lr}\ \ z_{ar}\ \ z_{sl}]'$ are affected by three actions: two constant-in-time hip torques and the reference angle of the proportional-derivate control on the foot during push-off. The discrete linear equations have the standard form:

$$\delta x_{n+1} = \mathbf{A}^s \delta x_n + \mathbf{B}^s \delta U_n^s \tag{7.3}$$

$$\delta z_{n+1}^s = \mathbf{C}^s \delta x_n + \mathbf{D}^s \delta U_n^s. \tag{7.4}$$

Again using DLQR cost given in equation 2.5 and equations 7.3 and 7.4 we set up a DLQR problem to determine a 3×3 gain matrix $\mathbf{K}$. We choose $\mathbf{R}_{zz}^s =$

diag$\{1/\sigma_{sr}^2, 1/\sigma_{ar}^2, 1/\sigma_{sl}^2\}$ and $\mathbf{R}_{UU}^s = \text{diag}\{\rho_{hip}, \rho_{hip}, \rho_{ankle}\}$, where $\sigma_{sr} = 0.2$ is the user-specified characteristic stance leg angular rate, $\sigma_{ar} = 0.2$ is the characteristic falling rate of the foot just before heel-strike, $\sigma_{sl} = 0.2$ is the characteristic step length, and $\rho_{hip} = 0.5$ weights the hip actuator effort relative to the deviation of outputs while $\rho_{ankle} = 2$ weights the feet actuators efforts relative to the deviation of the output variables.

In summary, the stabilizing (discrete trajectory tracking) controller has 15 gains. Six for 'too fast' (two actuations $\times$ three sensors) and nine for 'too slow' (three actuations $\times$ three sensors).

7.3 Results

7.3.1 Robustness *

Figure 7.1 and 7.2 show the effect of perturbing the robot from its nominal trajectory in simulation.

We applied a pushing disturbance (see figure 7.1) to the robot just before mid-stance (note that we measure the robot state at mid-stance). The push caused the robot's mid-stance velocity to increase by 50 % from its nominal value. The robot recovers from the push in about 2 steps by taking bigger than nominal steps while maintaining the nominal push-off.

We applied a pulling disturbance (see figure 7.2) to the robot just before mid-stance (note that we measure the robot state at mid-stance). The pull caused the robot's mid-stance velocity to decrease by 50% from its nominal value. The robot recovers from

the pull in about 4 steps by increasing the ankle push-off from its nominal value while maintaining the nominal step length.

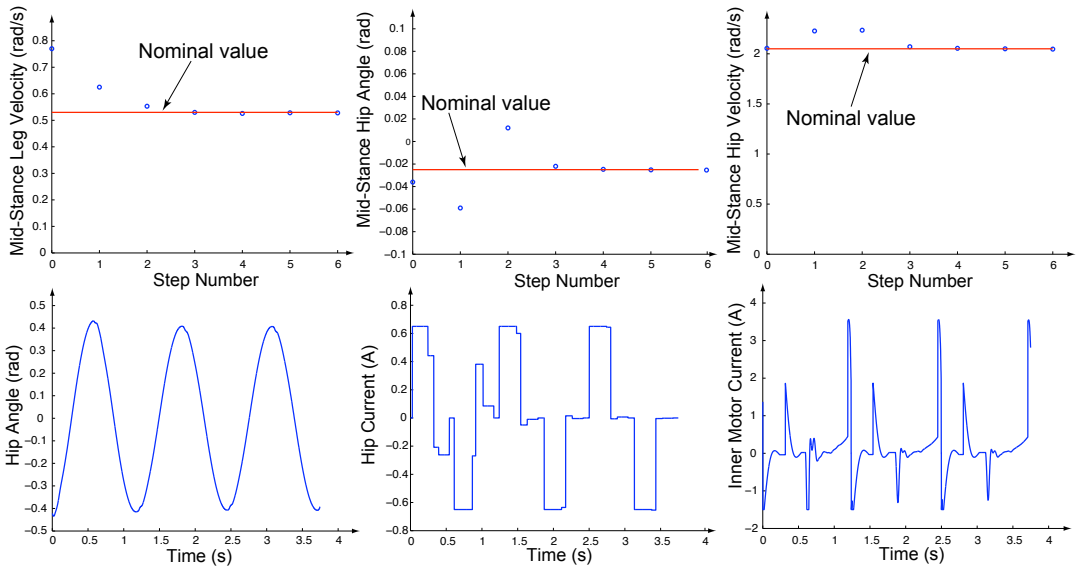

Figure 7.1: **Recovering from pushes.** At the 0th step the mid-stance velocity is increased by 50% by a pushing disturbance. The robot is back on the nominal trajectory in about 2 steps.

7.3.2 Implementation on the physical robot

The robot control was based on approximating (coarse-grid optimization) the fine-grid optimization of total energy use (see chapter 5, section 5.3). We tried implementing the coarse-grid control representation on the physical robot. Unsurprisingly we found that in this nearly open-loop mode the robot could not walk reliably; it always fell down in a few steps. A look at the sensor outputs revealed that the outer legs swung slower than the inner ones. This was confirmed by further physical measurements that showed that the outer legs have a higher moment of inertia than the inner legs. Note that the same commands are given to the outer and inner legs. The result was that the outer legs

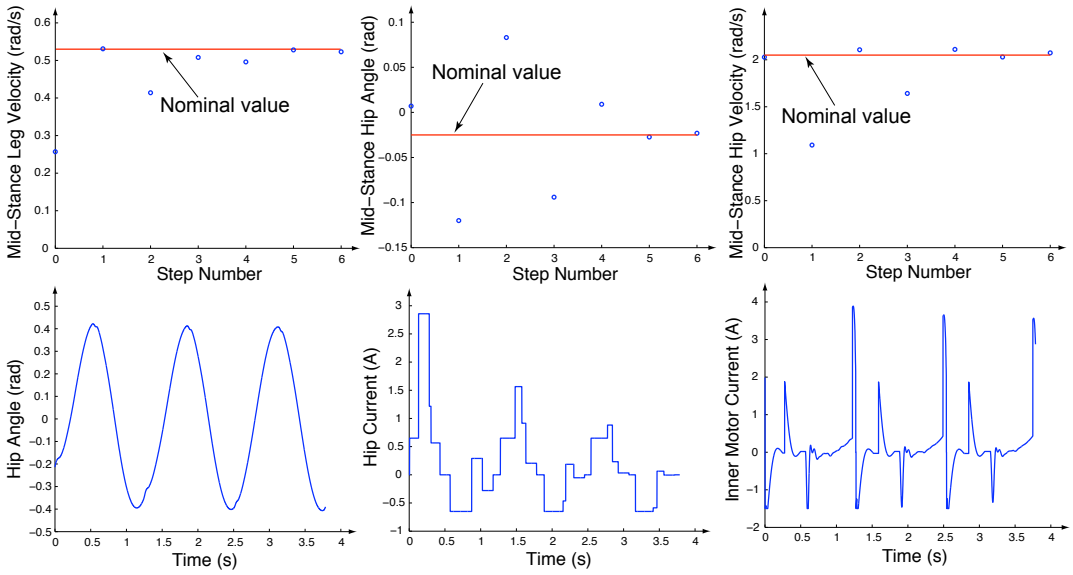

Figure 7.2: **Recovering from pulls.** At the 0 th step the mid-stance velocity is decreased by 50% a pulling disturbance. The robot is back on the nominal trajectory in about 4 steps

did not swing fast enough to break the forward fall of the robot. Next, we switched our discrete intermittent feedback controller on and tried walking the robot. The feedback controller compensated for the slow speed of the outer hips and the robot realized successful walking.

Figure 7.3 compares the joint angle, motor current and motor power predicted by the open loop coarse grid optimization (solid blue) with those obtained on the physical robot with feedback (red dashed). The spikes in figure d and g are from the feedback controller compensating for the robot asymmetry. Of course, we could have tried to learn this or precompensate, but we chose to let the feedback control do what it is supposed to do, compensate for errors, including model errors.

The robot's total power of 10.7 W was used by sensors, processors and communications (49%), motor dissipation (30%) and positive mechanical work (21%). In this final

implementation the TCOT was 0.19 (compared to a human's TCOT of about 0.3) or, subtracting the electrical overhead, the COT was 0.1 (compared to a COT for humans of about 0.2 if the resting metabolic costs are subtracted). Ranger's *mechanical* cost of transport, based on positive motor work, was MCOT = 0.04 (compared to MCOT $\approx$ 0.05 for the Collins robot and for humans). For comparison, recall that the TCOT of most other legged robots is well over 1, at least 3 times more.

7.3.3 Comparison between fine-grid simulation, coarse-grid simulation, and experiment

Table 7.1 lists the various gait parameters and energetics for the fine-grid trajectory control problem, the coarse-grid control representation and the experimental robot data.

Gait Parameter	Fine-grid	Coarse-grid	Experiment
Total COT	0.167	0.180	0.190
Motor COT	0.087	0.100	0.110
Overhead COT	0.083	0.080	0.080
Hip COT	0.019	0.018	0.030
Ankle COT (push-off)	0.029	0.052	0.046
Ankle COT (foot-flip)	0.039	0.029	0.034
Step Length	0.38	0.39	0.38
Step Velocity	0.64	0.66	0.62
Step Time	0.60	0.60	0.61
Double Stance (% of cycle)	9.5	5.0	3.0
Number of Control Parameters	126	15	30

Table 7.1: **Comparisons between fine-grid optimization, coarse-grid control optimization and experiment (mean values).** The energetics, controls and gait parameters.

The energetic cost of control. We see that in going from the optimal trajectory control of a simulated robot to our coarse-grid control with stabilizing control of a physical

robot, we have reduced the parameters from 126 to 30 and added gait reliability, but at the cost of increasing the TCOT from 0.167 to 0.19, a 14% increase. That is, the incremental cost of simplifying the trajectory description over the fine-grid calculated trajectory was about 8%, as calculated in simulation. The cost of stabilization was about 6%, as calculated by comparing the physical robot with the simulation. The latter is a less reliable estimate because it also includes modeling errors. Nonetheless, for this simple walking task the energetic cost of simplifying the controller is small. And the energetic cost of adding stability seems to be not far from the theoretical prediction that stability (in the absence of disturbances) should have no cost.

The TCOT of the physical robot of 0.19 makes our robot probably a little more energy-effective than the Collins walker, which had a measured TCOT of about 0.2.

Robustness. One issue noted in testing resulted in a change in our description of the swing-foot logical-state transition. The fine-grid optimization is always time-based. In the conversion to the coarse-grid optimization we thus had a parameter for the time when push-off should start. From theory we know that the timing of push-off has a huge effect; a factor of 4 difference in simple models, depending on the exact timing [62, 92]. Thus we originally used as a trigger the time until the extrapolated time of heel-strike (based on velocity and height of the foot). Because the optimal gait had a nearly grazing collision, on the robot this led to a high sensitivity to sensor errors and to ground height fluctuations; a small ground-height change would substantially change the time of push-off relative to heel-strike and thus dramatically affect the speed of the next step. Thus to increase the robustness we changed the push-off transition to be at a critical height of the swing foot. When this was given as a parameter to the optimization the optimization chose a gait with swing-leg retraction prior to heel-strike and thus a higher vertical velocity of the swing foot before heel-strike. Note that this robustness

from leg retraction is unrelated to the stability that leg retraction can provide in open loop control, as in [94, 110].

The final machine was robust enough to work reliably on a running track where the maximum slopes were about 1° and maximum step-to-step variation was a few mm, but not much more.

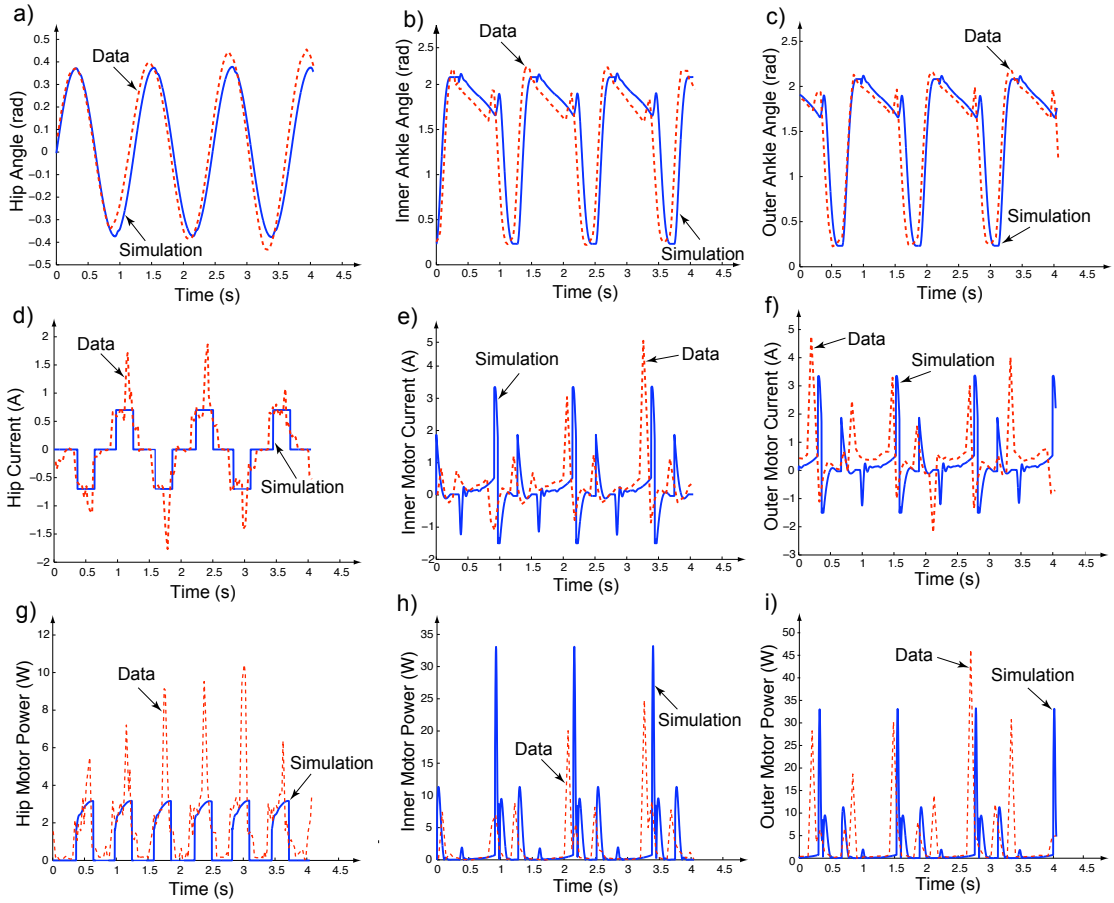

Figure 7.3: **Comparing forward simulation of the robot with experimental data for the same controller**. The simulation (solid blue) is periodic, the data (dashed red) is not. The biggest discrepancies are the spikes in hip current and hip power. These are from the stabilizing controller attempting to compensate for differences between model and machine. The horizontal offsets visible above are because the step period of the machine does not exactly match the period of the model.

7.3.4 Long distance walking record

The broad goal of the Ranger project [12] is to develop a reliable robot capable of walking long distances on minimal amounts of energy. We set ourselves the goal of making Ranger walk a marathon distance of 26.2 miles or 42.2 kilometers, without falling down, without stopping, and without recharging.

The feat of walking a marathon was achieved in stages. Ranger was built in the fall of 2006 [58]. In December 2006, it walked 1 km or 0.6 mi non-stop with a TCOT of 1.6 [9], setting a legged distance record then. Further improvements in the hardware and walking controller led to longer walks. In April 2008, it walked 9 km or 5.6 mi with a TCOT of 0.6 [10]. In July 2010, it walked 23 km or 14.3 mi with a TCOT of 0.49 [11].

On 1-2 May 2011, before we had implemented all steps of the optimization in the energy-effective controller presented earlier in this paper, Ranger walked 40.5 miles or 65 kilometers, non-stop, and on a single battery charge (beating BigDog's record of 12.6 mi set in 2008 by more than a factor of 3). Ranger took 186,076 steps at a leisurely pace of 2.12 kilometers per hour or 1.32 miles per hour to set this distance record. The total energy consumption for Ranger for this walk was 493 watt-hours[1]. For this ultra-marathon Ranger had a TCOT of 0.28. As noted, this was later reduced to TCOT = 0.19. Thus, we believe Ranger could now walk (0.28/0.19)*65 km = 95 km on a single charge, well over two full marathons.

[1]Electricity costs 11.2 cents per kilo-watt-hour in United States for the year 2011. Ranger's 493 watt-hours of battery charge can be bought for only 5.5 cents ($0.493 \times 11.2 = 0.55$) in the United States.

Figure 7.4: **Ranger's ultra-marathon walk.** On 1-2 May 2011 [12], Ranger walked non-stop for 40.5 miles (65 km) on Cornell's Barton Hall track without recharging or being touched by a human. Some of the crew that worked on Ranger are shown walking behind Ranger during the 65 km walk. Basic data are in the table below.

Total steps	186,076
Total time	110,942 s (= 30 hrs 49 min 2 sec)
Number of laps	307.75
Lap distance	212 m (= 0.132 miles)
Total distance	65,243 m (= 65.24 km = 40.54 mi)
Average time per step	0.6 s
Average distance per step	0.35 m (= 13.78 in)
Average speed	0.59 m/s (= 2.12 km/h = 1.32 mph)
Total power	16 W
Power used by motors	11.3 W
Power used by computers and sensors	4.7 W
Total energy used	493 watt-hours
Battery	25.9 V Li-ion
Total robot mass	9.91 kg (= 21.85 lb)
Battery mass	2.8 kg (= 6.3 lb)
Total cost of transport (TCOT)	0.28 (later lowered to 0.19)

Table 7.2: **Statistics of Ranger's 40.5 mile ultra-marathon walk on 1-2 May 2011.**

CHAPTER 8

CONCLUDING REMARKS

8.1 Thesis summary *

This thesis presented a model based control design framework for bipedal robots that combines energy efficiency with stability and demonstrated its application on the custom built bipedal robot called the Ranger.

First in chapter 2, we give a peek at our control design algorithm that starts with a hi-fidelity robot model, and proceeds with a trajectory generator to get the nominal gait and finally a stabilizing controller that stabilizes the nominal gait. We applied our stabilizing controller idea to balance of a simple inverted pendulum with a controller bandwidth slower than the characteristic time scale of the system.

In chapter 3, we presented a model for the robot and its actuators. In chapter 4, we presented bench experiments that helped fit the parameters of the assumed model. In particular, we found that the simple ideal DC motor description inadequate. The motor brush resistance was almost twice than what was reported by the manufacturer and there was substantial brush-commutator contact resistance. The gear-box had a load dependent friction which we approximated as a current dependent friction.

In chapter 5, we formulated an energy-optimal trajectory control problem. Our energy metric was the total cost of transport (TCOT) and is defined as the energy used per weight per unit distance travelled. We decoupled the motion of the foot of the swinging leg in single stance phase (that does the ground clearance) from the rest of the walk. These helped us write the TCOT as a sum of COT to power the computers and others electronics, COT for foot-flip and COT for walking. We considered minimization of

the various COT's as a function of step length and step velocity. The COT to run the computers and electronics is step velocity dependent and favors fast speeds. The COT for foot-flip is step length dependent and favors big steps. The COT for walking has a strong step time dependence and favors walking at the natural frequency of the swing legs. After summing the COT we found that minimum TCOT is 0.163 (Total Watts = 10.2) and occurs at a step length of 0.48 m and step velocity of 0.77 m/s. In the optimal solution, 49 % of the energy goes to power the computer and electronics, 23% is used to do the foot-flip for ground clearance and the rest 28 % is used in taking a step.

Next, we turn to implementation on the robot. In the later half of chapter 5, we re-parameterized the optimal solution with the goal of simplifying the control representation. Our approximate representation yields the nominal trajectory and reduces the original 126 parameters in the fine grid solution to about 15 parameters while increasing the cost by about 7%. In order to stabilize the nominal trajectory, we first motivated an energy based control of a 2-D point-mass model of walking in chapter 6. The central idea here is that we are interested in regulating the kinetic energy of the robot's center of mass in the upright position. We identify two means of doing so; using an ankle push-off control and using step length control. Further analysis reveals that ankle push-off control works best to regulate slow walking while step length works best to regulate fast walking.

Using the approximate coarse-grid controller representation (chapter 5) and the stabilizing control ideas (chapter 6), we implement the simulation based controller on the biped platform Ranger. We show that using our control framework Ranger walks stably with a TCOT of 0.19 (about 14 % more than the fine grid optimal solution), more energy-efficient than any legged robot built-to-date. Also, using a slightly less energy-efficient version of approximate controller than the one presented here, Ranger walked

non-stop for 40.5 miles or 65 km on a single battery charge, setting a legged robot distance record.

8.2 Discussion and conclusion

Our overall design and control approach can be summarized as this:

- Design and build a modelable robot. This precludes flat feet, which can have indeterminate collisions, and high-gain control, which can excite poorly characterized vibration modes. Similarly, friction and play need to be minimized, as both are hard to model.
- Make and verify a high-fidelity simulation. This requires that all robot parts be well-characterized, particularly the motors and transmissions, and checking a simulated walking cycle with a real walking cycle.
- Make a fine-grid trajectory optimization (in our case, minimizing TCOT). This optimization includes some events necessitated by discontinuities in the model description (when contacts are made or broken).
- Pull out features of the fine-grid optimization to define a coarse-grid parameterization of the trajectories. The discretization of the coarse-grid description defines new events and new logical states. Then, offline, use this coarse-grid description for a new trajectory optimization. Tune the description to minimize complexity and maximize closeness of the objective to the fine-grid objective.
- Define a plausible reflex-based (discrete) controller with a manageably small number of free parameters. These measure the dynamical state at logical state transitions and use the values to adjust the parameters in the motor program in

the next logical state. Tune the free parameters of the feedback, in simulation, to maximize simplicity and minimize gains and errors as desired, using, e.g., DLQR.

- Implement the controller on the robot, comparing simulation with experiment so as to check for physical machine, simulation or optimization problems.

In this list, there is no part that we believe is novel. The benefit of this approach to discrete (reflex) control, as summarized above is, that it allows:

- Low gains are possible because extended time is available to make corrections. High gains demand higher machine stiffness to prevent excitation of control oscillations or higher vibration modes.
- Use of a manageably low number of parameters in the trajectory and stabilizing control, with a small resultant energy penalty (compared to the best physically obtainable optimal-energy gait).
- Ability to be progressively refined so as to better approximate full state feedback control.

While we made a machine that was reliable for its purpose, the biggest defect in the approach, as implemented so far, is that it has no means of systematically maximizing the sizes of allowable disturbances and modeling errors.

Three uses of events. Discrete events have been used for three purposes in the controller design. First, the fine-grid optimization needs to change its form when the governing equations change form. This occurs when contacts are made or broken. Second, the coarse-grid optimization introduces new logical state transitions (at the grid points) and new logical states (between grid points). Finally, the discrete feedback uses the

same transitions (same events) as the coarse-grid optimization does, and then uses the dynamical state at that time to adjust the parameters used during the time span of the next logical state.

Four kinds of feedback. We think of the primary stabilizing feedback as the adjustment to parameters at state transitions. However, in reality there are three other forms of feedback. First, having sensor-based state transitions of any kind is a form of feedback. Second, within each logical state the motors are run by simple continuous-time (approximately, actually a 2 kHz digital control) compliant controllers. Third, each motor has an inner feedback loop, constantly adjusting the PWM signal to impose a specified motor current.

Extensions and improvements. Ways to make the robot better include using more efficient electronics and adding knees so that energy can be saved using the natural dynamics to provide foot clearance. Although this machine was designed to walk well, it should also be capable of some other simple behaviors (e.g., starting, stopping, walking backwards, balancing on two feet) and higher (meta) states for different behaviors (to switch between, say, walking forwards and backwards). While we had relative success with brushed DC motors, a major modeling issue was the brush contact resistance; the issue would be eliminated by using brushless motors. This would also improve motor efficiency.

Most importantly, the control approach also can be further developed. There are several steps in our controller design that depended on human insight. Some of these could be automated. These include the selection of the coarse-grid parameterization and the architecture of the discrete stabilizing controller. Because the overall control architecture is perhaps more appropriate to motor control than, say, a more general neu-

ral network, it may thus be more manageable (i.e., require fewer free parameters for a given quality of control) for optimization in simulation or for offline or online learning. Though the system we have is reliable in a limited context, we have yet to determine how robust it can be made to disturbances and terrain variation.

APPENDIX A

NOTATION AND EQUATIONS OF MOTION

In this chapter we describe our notation and followed by derivation of the equations of motion.

A.1 Notation

Variables used in this appendix are listed here, some with respect to the side-view robot schematic figure A.1. The main internal degrees of freedom are the hinges at the ankles (A_1, A_2) and hip (H). Because the ankle drive cables are elastic, the motors at A_1^* and A_2^* add two additional internal degrees of freedom.

Robot parameters

Symbol	Value	Parameter description
ℓ	0.96 m	Leg length.
r	0.2 m	Foot radius.
d	0.11 m	Ankle eccentricity.
w	0	Fore-aft distance of COM.
c	0.15 m	Distance of COM from hip along the leg.
k_h	7.6 N m/rad	Hip Spring constant.
k_s	14 N m/rad	Ankle Spring constant.
J_ℓ	0.45 kg m^2	Inertia of legs about COM.
J_{hip}	0.55 kg m^2	Inertia of legs about hip hinge.
m	4.96 kg	Mass of a leg.

M_{tot}	9.91 kg	Total robot mass.
g	9.81m/s^2	Gravitational constant.
γ	0	Ground slope (positive value is downhill).
C_{1FW}	0.05 N s/m	Coefficient of viscous friction between ground and stance leg.
C_{2FW}	0.05 N s/m	Coefficient of viscous friction between ground and trailing leg... in double stance.

Motor parameters

Symbol	**Value**	**Parameter description**
G_H	66	Hip gear ratio.
G_A	34	Ankle gear ratio.
K	0.018 N m/A	Motor torque constant.
R	1.3 Ω	Motor terminal resistance.
V_c	0.7 V	Contact voltage of the brush-commutator interface.
J_m	0.002 kg m^2	Motor inertia.
μ_H	0.1	Coefficient of current dependent constant friction: hip motor.
μ_A	0.1	Coefficient of current dependent constant friction: ankle motor.
C_{H1}	0.01 N s/m	Coefficient of viscous friction: hip motor.
C_{H0}	0.1 N	Coefficient of constant friction: hip motor.
C_{A1}	0.01 N s/m	Coefficient of viscous friction: ankle motor.
C_{A0}	0.1 N	Coefficient of constant friction: ankle motor.
P_{fixed}	5.15 W	Power used by microprocessors, sensors and motor controller.

Other variables

Symbol	Variable description
t	time.
q_1, r_1	absolute angle made of stance foot wrt. vertical after and before heelstrike.
q_2, r_2	relative angle between stance foot and stance leg after and before heelstrike.
q_3, r_3	relative angle between legs; also hip angle after and before heelstrike.
q_4, r_4	relative angle between swing foot and swing leg after and before heelstrike.
q_{2m}, q_{4m}	motor angles at points $A_1^\star$ and $A_2^\star$ after heel-strike respectively.
r_{2m}, r_{4m}	motor angles at points $A_1^\star$ and $A_2^\star$ before heel-strike respectively.
x, y	world reference frame, x in walking direction and y is against gravity.
x_h, y_h	x and y co-ordinate of the hip joint respectively.
I_i	Motor current. $i = 2, 3, 4$ at the points $A_1^\star$, H and $A_2^\star$ respectively.
d_{step}	Step length.
t_{step}	Step time.
v_{step}	Step velocity.
M	Mass at the hip (for benchmarks in appendices B and D).
E	Energy.
P	Power.
N_i	Number of grid points, $i = ss$ (single stance) or $i = ds$ (double stance).
χ	Robot state vector and includes angles and angular rates.
T_{1FW}	Torque between ground and stance foot. ($T_{1FW} = -C_{1FW}\dot{q}_1$).
T_{2FW}	Torque between ground and trailing stance foot in double stance...
	($T_{2FW} = -C_{2FW}(\dot{q}_1 + \dot{q}_2 - \dot{q}_3 - \dot{q}_4)$).
T_3	Hip Motor Output Torque ($= G_H K I_3 - T_{fH}(I_3, \dot{q}_3)$)
$T_{fH}(I_3, \dot{q}_3)$	Hip Motor Friction Torque ($= \mu_H \text{sgn}(\dot{q}_3) G_H K \lvert I_3 \rvert + C_{H1}\dot{q}_3 + C_{H0}\text{sgn}(\dot{q}_3)$).
$T_{fA}(I_i, \dot{q}_{im})$	Ankle Motor Friction Torque ...

	$(= \mu_A \text{sgn}(\dot{q}_{im}) G_A K \lvert I_i \rvert + C_{A1}\dot{q}_{im} + C_{A0}\text{sgn}(\dot{q}_{im}))$ where $i = 2, 4$.
T_{2S}	Ankle Spring Torque ($T_{2S} = k_s(q_{2m} - q_2)$).
T_{3S}	Hip Spring Torque ($T_{3S} = k_h q_3$).
T_{4S}	Ankle Spring Torque ($T_{4S} = k_s(q_{4m} - q_4)$).
F_{2S}, F'_{2S}	Tensional force in the ankle cables at joint associated with dof. q_2.
F_{4S}, F'_{4S}	Tensional force in the ankle cables at joint associated with dof. q_4.
H_i, V_i	Horizontal and vertical reaction forces respectively at joint i.
$H_i^\star$, $V_i^\star$	Horizontal and vertical impulse respectively at joint i.
$\vec{g}$	gravity vector $(= g\sin(\gamma)\hat{\imath} - g\cos(\gamma)\hat{\jmath})$.
$\vec{\omega}_1$	Absolute angular velocity of G_1 after heelstrike ($\omega_1 = \dot{q}_1 + \dot{q}_2$).
$\vec{\omega}_2$	Absolute angular velocity of G_2 after heelstrike ($\omega_2 = \dot{q}_1 + \dot{q}_2 - \dot{q}_3$).
$\vec{\omega'}_1$	Absolute angular velocity of G_1 before heelstrike ($\omega'_1 = \dot{r}_1 + \dot{r}_2$).
$\vec{\omega'}_2$	Absolute angular velocity of G_2 before heelstrike ($\omega'_2 = \dot{r}_1 + \dot{r}_2 - \dot{r}_3$).
$\vec{\alpha}_1$	Absolute angular acceleration of G_1, the stance leg ($\alpha_1 = \ddot{q}_1 + \ddot{q}_2$).
$\vec{\alpha}_2$	Absolute angular acceleration of G_2, the swing leg ($\alpha_2 = \ddot{q}_1 + \ddot{q}_2 - \ddot{q}_3$).
$\vec{v}_H$	Velocity of point H after heelstrike, $(= \dot{x}_h\hat{\imath} + \dot{y}_h\hat{\jmath})$.
$\vec{v}_{H'}$	Velocity of point H before heelstrike, $(= \dot{x}_{h'}\hat{\imath} + \dot{y}_{h'}\hat{\jmath})$.
$\vec{a}_H$	Acceleration of point H $(= \ddot{x}_h\hat{\imath} + \ddot{y}_h\hat{\jmath})$.
$\vec{v}_{G_1}$	Velocity of point G_1 after heelstrike $(= \vec{v}_H + \vec{\omega}_1 \times \vec{r}_{G_1/H})$.
$\vec{v}_{G'_1}$	Velocity of point G_1 before heelstrike $(= \vec{v}_{H'} + \vec{\omega}'_1 \times \vec{r}_{G'_1/H'})$.
$\vec{a}_{G_1}$	Acceleration of point G_1 $(= \vec{a}_H - \omega_1^2\ \vec{r}_{G_1/H} + \vec{\alpha}_1 \times \vec{r}_{G_1/H})$.
$\vec{a}_{G_2}$	Acceleration of point G_2 $(= \vec{a}_H - \omega_2^2\ \vec{r}_{G_2/H} + \vec{\alpha}_2 \times \vec{r}_{G_2/H})$.
$\vec{v}_{G_2}$	Velocity of point G_2 after heelstrike $(= \vec{v}_H + \vec{\omega}_2 \times \vec{r}_{G_2/H})$.
$\vec{v}_{G'_2}$	Velocity of point G_2 before heelstrike $(= \vec{v}_{H'} + \vec{\omega}'_2 \times \vec{r}_{G'_2/H'})$.
$\vec{r}_{H/P_1}$	Position vector from point P_1 to point H and so on.
$\vec{P}$	Force on the trailing foot from ground in double stance $(= P_x\hat{\imath} + P_y\hat{\jmath})$.

$\vec{P^\star}$	Impulse on the trailing foot from ground in double stance ($= P^\star_x \hat{\imath} + P^\star_y \hat{\jmath}$).
$\vec{H}_{/P_1}$	Angular momentum about point P_1 and so on.
$\dot{\vec{H}}_{/P_1}$	Rate of change of angular momentum about point P_1 and so on.
$\vec{M}_{/P_1}$	External angular moment about point P_1 and so on.
$\lvert . \rvert$	Absolute value function ($\lvert x \rvert = x$ for $x > 0$ and $\lvert x \rvert < -x$ for $x \leq 0$).
$[.]^+$	Ramp function ($[x]^+ = x$ for $x > 0$ and $[x]^+ = 0$ for $x \leq 0$).
sgn(.)	Signum function ($\mathrm{sgn}(x) = x/\lvert x \rvert$).

A.2 Equations of motion

The robot model is described in chapter 3, section 3.2 and shown in figure 3.1 therein. The dimensions are shown in figure A.1. Reference frames and angles are shown in figure A.2.

Coordinates. During double stance we can write the co-ordinates of the hip x_h, y_h, using the fixed (Newtonian) coordinate system xy in two ways (see figure A.2): 1) using the path $\mathrm{OP_1A_1H}$, and 2) using the path $\mathrm{OP_2A_2H}$. We thus have two kinematic restrictions (constraints) on the joint angles (q_2, q_3 and q_4), the absolute angle of the

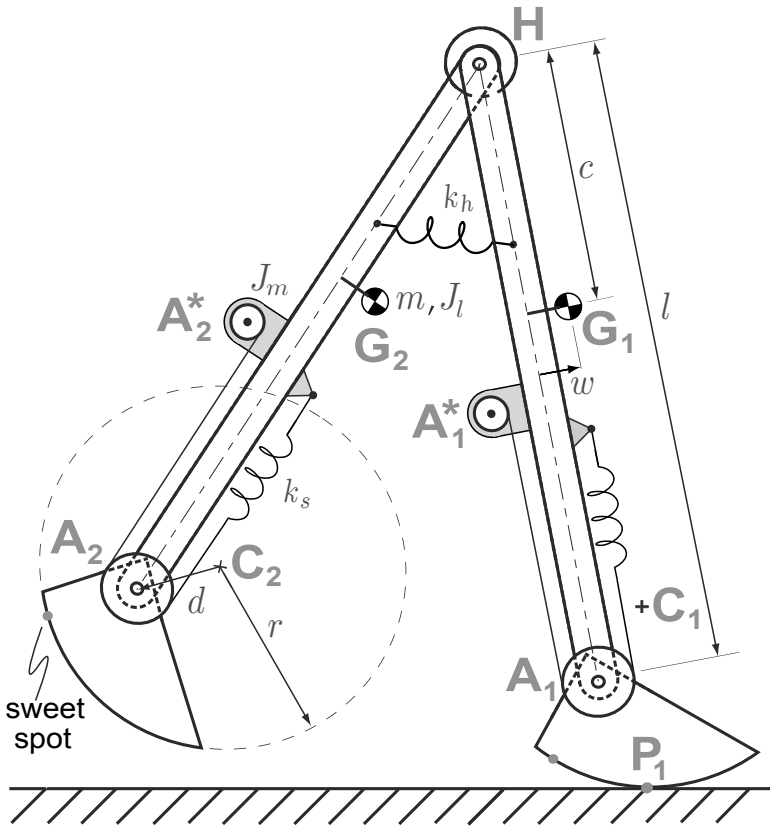

Figure A.1: **Robot dimensions and feet geometry.** The feet bottoms are roughly circular arcs with radius r. The ankle joints A_1 and A_2 are offset from the center of circle by the distance d. As dictated by the geometry of circles the contact points P_1 and P_2 are always directly below the center of the circles C_1 and C_2, respectively, in level-ground walking. There is one foot configuration in which the ankle joint lies on the line joining the center of the circle and the contact point. For vertical ground forces this is a natural equilibrium position for the feet; it takes no ankle torque to hold the foot in this position. The contact point is then that part of the foot circular arc that is closest to the ankle. We call this point on the foot the 'sweet-spot'. The ankle motors are connected to the ankle joints via cables that we approximate as linear springs. The ankle motors (A_1^*, A_2^*) are actually nearly coincident with the hip H, but are separated in this diagram for clarity.

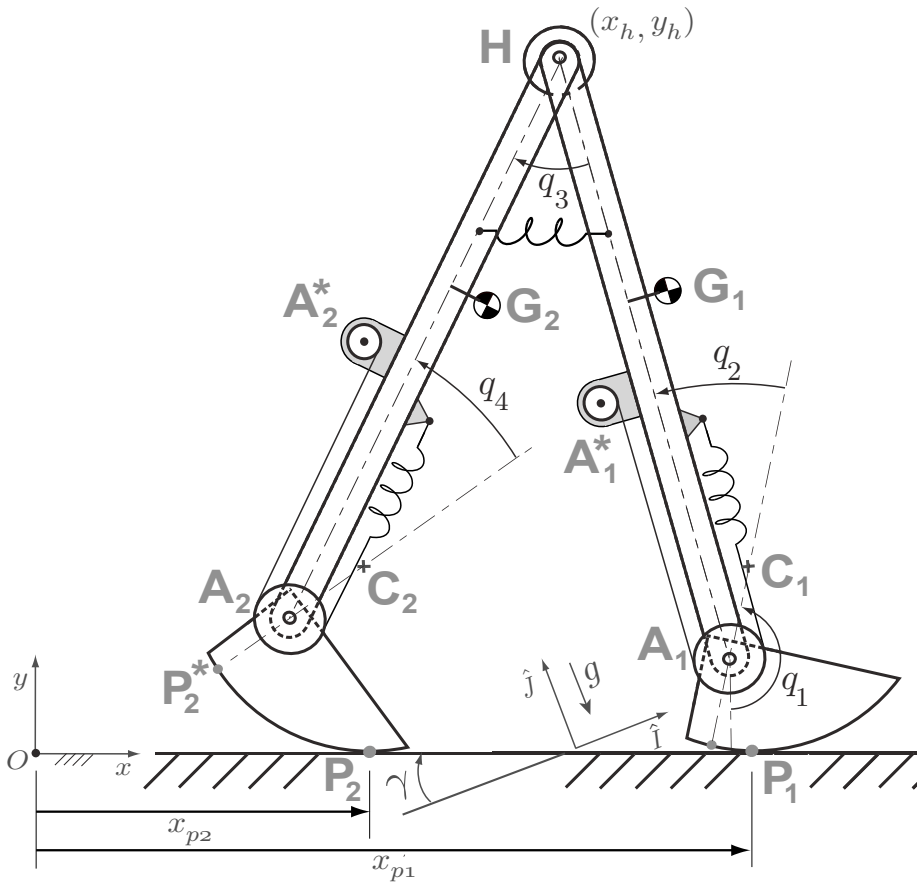

Figure A.2: **Robot reference frames and degrees of freedom used in the derivation of the equations of motion.** The absolute angle made by the lead foot on the ground with the vertical is q_1. Joint angles are q_2, q_3 and q_4. Hip motor angle is the same as hip joint angle q_3. Ankle motor angles associated with the joint $A_1^\star$ is q_{2m} and with joint $A_2^\star$ is q_{4m}.

lead foot q_1 and step length $(x_{p2} - x_{p1})$ during double stance.

$$x_h = x_h$$

$$\begin{aligned} \implies x_{p1} + l\sin(q_1+q_2) - d\sin(q_1) - rq_1 = x_{p2} - l\sin(q_3 - q_1 - q_2)\ldots \\ - d\sin(q_1+q_2-q_3-q_4)\ldots \\ - r(q_1+q_2-q_3-q_4) \end{aligned} \tag{A.1}$$

$$y_h = y_h$$

$$\begin{aligned} \implies r - l\cos(q_1+q_2) + d\cos(q_1) = r - l\cos(q_3 - q_1 - q_2)\ldots \\ + d\cos(q_1+q_2-q_3-q_4) \end{aligned} \tag{A.2}$$

A.2.1 Equations of motion during single and double stance

The governing differential equations are found using the free body diagrams (FBDs) shown in figures A.3 and A.4, respectively, and using angular momentum balance about judiciously chosen points that eliminate (at least some of) the constraint forces.

Double stance. In double stance the robot has 2 kinematic degrees of freedom and two motor degrees of freedom. Releasing one foot from the ground and then constraining it, we think of this rather as 4 kinematic degrees of freedom, 2 motor degrees of freedom and 2 kinematic constraint equations. From figure A.3 we use the systems defined by the free body diagrams (a) – (d). We then use angular momentum balance about the points P_1, A_1, H and A_2, respectively, to generate 4 equations. The motor degrees of freedom are described with the motor equations (equations A.4 and A.5 below). Two additional constraint equations for double stance are obtained by taking the second derivatives of equations A.1 and A.2 to obtain equations A.6 and A.7,

$$\dot{\overrightarrow{H}}_{/i} = \overrightarrow{M}_{/i} \qquad \text{where } i = P_1, A_1, H, A_2 \tag{A.3}$$

$$T_{2S} = G_A(KI_2 - G_A J_m \ddot{q}_{2m}) - T_{fA}(I_2, \dot{q}_{2m}) \tag{A.4}$$

$$T_{4S} = G_A(KI_4 - G_A J_m \ddot{q}_{4m}) - T_{fA}(I_4, \dot{q}_{4m}) \tag{A.5}$$

$$\ddot{x}_h = \ddot{x}_h \tag{A.6}$$

$$\ddot{y}_h = \ddot{y}_h \tag{A.7}$$

$\overrightarrow{M}_{/i}$ and $\dot{\overrightarrow{H}}_{/i}$ are the sum of external torques and rate of change of angular momentum about the point i. These expressions are shown expanded in section A.2.3. Note, from the main text the spring torque at joint 2 is, $T_{2S} = F_{2S} r_p = k r_p^2 (q_{2m} - q_2) = k_s(q_{2m} - q_2)$ and spring torque at joint 4 is $T_{4S} = F_{4S} r_p = k r_p^2 (q_{4m} - q_4) = k_s(q_{4m} - q_4)$. F_{2S} and F_{4S} are the spring forces in the springs at joint 2 and 4 respectively, r_p is the radius of the

ankle pulley and k is the linear spring constant of the cable joining ankle to the ankle motor. Please see section A.1 for definitions of terms.

Altogether we have 6 differential equations of motion and two differential equations from differentiating the closed-linkage geometric constraint (equations A.3-A.7). From these we can solve, at any instant in time, for the angular accelerations of each robot part, both ankle motors, and the ground contact force at one foot.

Single stance. The swing-foot is airborne for the single stance phase. Because we neglect the masses of the feet, the swing foot and swing motor have the same motions and the swing foot does not have independent motion. This eliminates one degree of freedom from the single stance phase. Thus, in the single stance phase we have five degrees of freedom (stance ankle angle, hip angle, two motor angles and the stance foot angle). Altogether we then have three kinematic degrees of freedom, and two motor degrees of freedom.

Using figure A.4 we use the systems shown in the free body diagrams (a)–(c). With these we use angular momentum balance about the points P_1, H and A_1 respectively, to generate 3 equations of motion. Two more equations come from the two ankle motor equations.

Thus, we have the following equations

$$\dot{\overrightarrow{H}}_{/i} = \overrightarrow{M}_{/i} \qquad \text{where } i = P_1, A_1, H \tag{A.8}$$

$$T_{2S} = G_A(KI_2 - G_A J_m \ddot{q}_{2m}) - T_{fA}(I_2, \dot{q}_{2m}) \tag{A.9}$$

$$0 = G_A(KI_4 - G_A J_m \ddot{q}_{4m}) - T_{fA}(I_4, \dot{q}_{4m}) \tag{A.10}$$

$\overrightarrow{M}_{/i}$ and $\dot{\overrightarrow{H}}_{/i}$ are the sums of the external torques and the rate of change of angular momentum about the point i. In section A.2.3, we give the detailed expansions of these

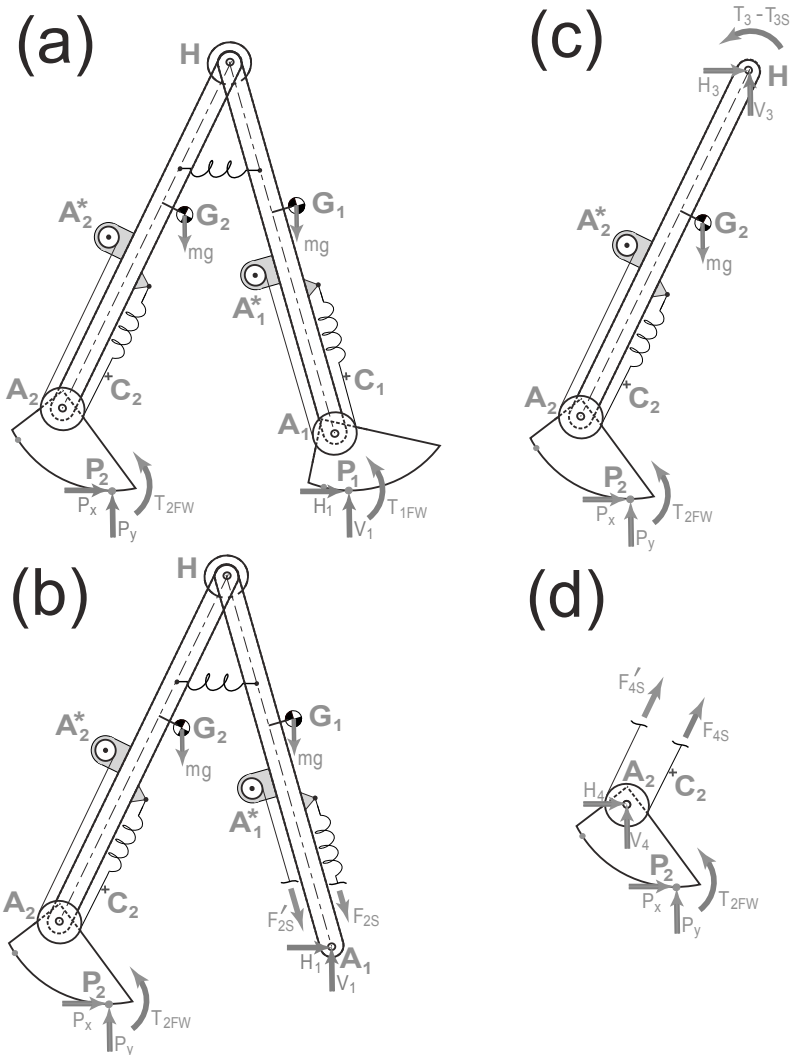

Figure A.3: **Free Body Diagrams (FBD) to derive equations for double stance.** We have four free body diagrams. The arrows indicate all of the non-neglected forces and torques acting on each of the four systems.

expressions. The equations above make up 5 differential equations for the angles of three body parts (all but the swing foot) and the two motors.

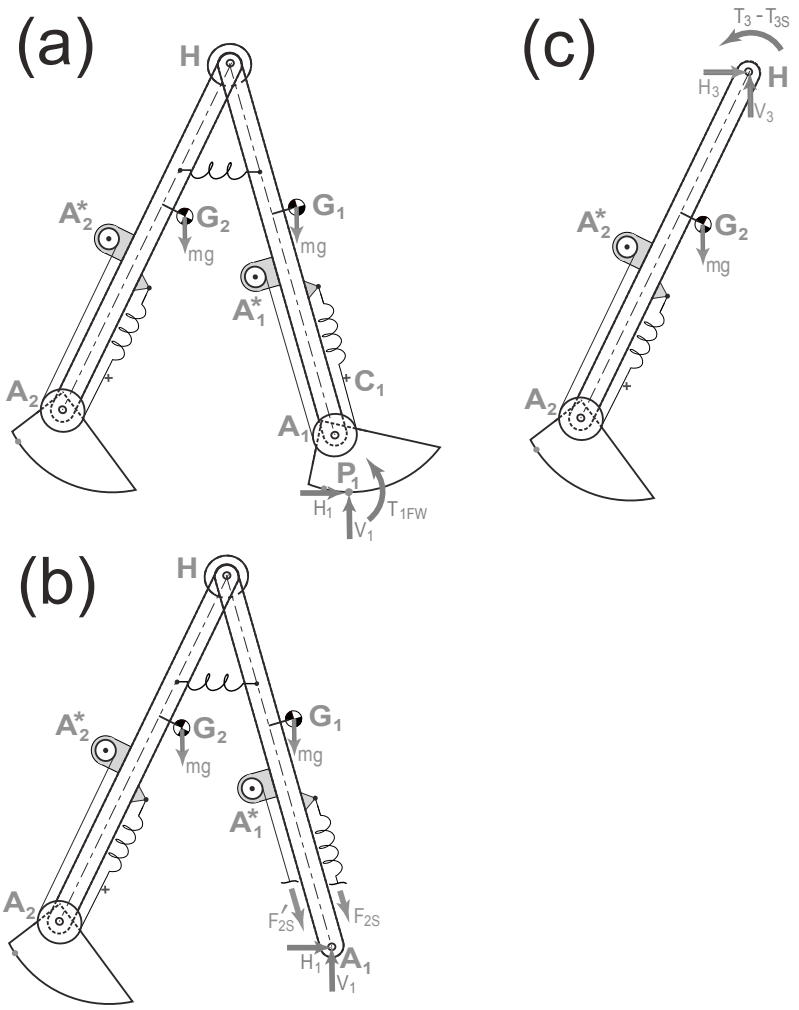

Figure A.4: **Free Body Diagrams (FBD) to derive equation for single stance.** We have 3 free body diagrams. Because the feet are massless there is no information in drawing a FBD of the swing foot.

Double stance equations in matrix form. Equations A.3 to A.7 can be re-arranged to get the following equation for double stance,

$$\mathbf{A}_{ds}\mathbf{X}_{ds} = \mathbf{b}_{ds} \tag{A.11}$$

where the unknown is the 8×1 vector, $\mathbf{X}_{ds} = [\ddot{q}_1\ ,\ \ddot{q}_2\ ,\ \ddot{q}_{2m}\ ,\ \ddot{q}_3\ ,\ \ddot{q}_4\ ,\ \ddot{q}_{4m}\ ,\ P_x\ ,\ P_y]'$. At a given dynamical state (given angles and rates) the 8×8 matrix $\mathbf{A}_{ds}$ and the 8×1

vector $\mathbf{b}_{ds}$ are known.

Extracting the elements of $\mathbf{A}_{ds}$ and $\mathbf{b}_{ds}$. There are various ways to find the elements of $\mathbf{A}_{ds}$ and $\mathbf{b}_{ds}$. Here is our somewhat clumsy method. We use symbolic algebra to evaluate the eight equations given in equation A.3 to equation A.7. Our next goal is to compute symbolic values of the individual elements of the matrices in $\mathbf{A}_{ds}$ and vector $\mathbf{b}_{ds}$. The first element of $\mathbf{b}_{ds}$ i.e. b_1 is obtained by setting $\mathbf{X}_{ds}$ = [0 , 0 , 0 , 0 , 0 , 0 , 0 , 0] and evaluating the first equation in A.3. Similarly, we can calculate the other elements of $\mathbf{b}_{ds}$. To get the first row and first column element of $\mathbf{A}_{ds}$ i.e. A_{11}, we evaluate first equation in A.3 by putting $\mathbf{X}_{ds}$ = [1 , 0 , 0 , 0 , 0 , 0 , 0 , 0] and from this value subtract out b_1. Following a similar procedure it is possible to get every element of the matrix $\mathbf{A}_{ds}$. As pointed out by Manoj Srinivasan (private communication) the MATLAB symbolic command JACOBIAN could have simplified this extraction.

Single stance equations in matrix form. Equations A.8 to A.10 can be re-arranged to get the following equation for single stance.

$$\mathbf{A}_{ss}\mathbf{X}_{ss} = \mathbf{b}_{ss} \tag{A.12}$$

where the unknown is the 6×1 vector, $\mathbf{X}_{ss} = [\ddot{q}_1 , \ddot{q}_2 , \ddot{q}_{2m} , \ddot{q}_3 , \ddot{q}_4 , \ddot{q}_{4m}]'$, while the 6×6 matrix $\mathbf{A}_{ss}$ and the 6×1 vector $\mathbf{b}_{ss}$ are known and can be found in a similar way as found for the double stance equations.

A.2.2 Collisional heel-strike equations

Here we consider the jump (the discontinuity) in angular rates when the swing foot collides with the ground at heel-strike. We consider first the case when this is a transition

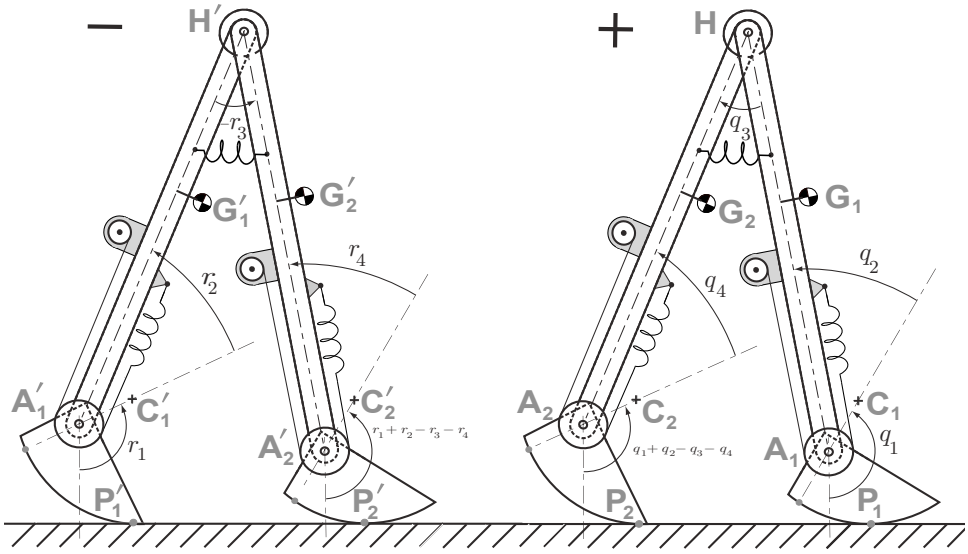

Figure A.5: **Angle swap for heel-strike derivations.** The instant just before heel-strike is denoted by $^-$ and the instant just after heel-strike is denoted by $^+$. We swap the names of the legs during heel-strike as shown. To simplify notation the angles are named r_i before collision and q_i after collision., where i is the joint number.

from a single-stance phase to a double-stance phase.

Figure A.5 shows the robot an instant before heel-strike, denoted by $^-$ and an instant after heel-strike, denoted by $^+$. We are interested in finding the angles after heelstrike, i.e., $[\mathbf{q}] = [q_1\ ,\ q_2\ ,\ q_{2m}\ ,\ q_3\ ,\ q_4\ ,\ q_{4m}]'$ and angular velocities after heel-strike, i.e., $[\dot{\mathbf{q}}^+] = [\dot{q}_1\ ,\ \dot{q}_2\ ,\ \dot{q}_{2m}\ ,\ \dot{q}_3\ ,\ \dot{q}_4\ ,\ \dot{q}_{4m}]'$.

The angles after heel-strike are found by using figure A.5 and swapping the angles to generate equation A.13. The velocities of joints after heel-strike are found by conservation of angular momentum. Doing conservation of angular momentum about appropriate points as shown in figure A.6 (a)–(d) we get equations A.14. We assume that the motors (buffered by the ankle spring) do not participate in the heel-strike. We swap the motor velocities to get equations A.15. Finally, two additional constraint equations

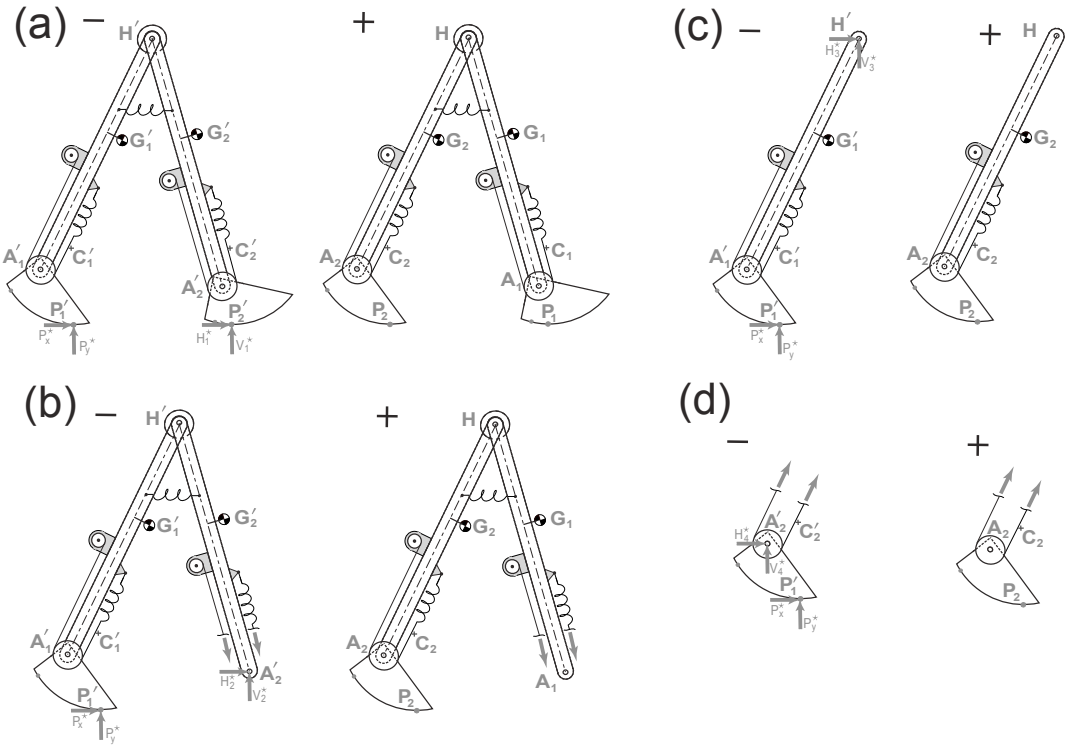

Figure A.6: **Free Body Diagrams (FBDs) for heel-strike discontinuity.** The case shown is for the transition to a double-stance phase; there are impulses at both feet. The instant just before and after heel-strike is indicated by $^-$ and $^+$ respectively. The collisional forces are shown in the $^-$ configuration. We have four subsystems corresponding to the four body parts. The ankle motors are buffered by the ankle springs and do not participate in the collision; so the ankle motors and and rates simply exchange values (i.e., keep their values and exchange their names).

are generated by taking the first derivatives of equations A.1 and A.2 to give equation A.16.

$$\begin{aligned} q_1 &= r_1 + r_2 - r_3 - r_4 & q_2 &= r_4 \\ q_{2m} &= r_{4m} & q_3 &= -r_3 \\ q_4 &= r_2 & q_{4m} &= r_{2m} \end{aligned} \tag{A.13}$$

$$\begin{aligned} \overrightarrow{H}^{+}_{/P_1} &= \overrightarrow{H}^{-}_{/P'_2} + \vec{r}_{P'_1/P'_2} \times \vec{P^\star} & \overrightarrow{H}^{+}_{/A_1} &= \overrightarrow{H}^{-}_{/A'_2} + \vec{r}_{P'_1/A'_2} \times \vec{P^\star} \\ \overrightarrow{H}^{+}_{/H} &= \overrightarrow{H}^{-}_{/H'} + \vec{r}_{P'_1/H} \times \vec{P^\star} & \overrightarrow{H}^{+}_{/A_2} &= \overrightarrow{H}^{-}_{/A'_1} + \vec{r}_{P'_1/A'_1} \times \vec{P^\star} \end{aligned} \tag{A.14}$$

$$\dot{q}_{2m} = \dot{r}_{4m} \qquad \dot{q}_{4m} = \dot{r}_{2m} \tag{A.15}$$

$$\dot{x}_h = \dot{x}_h \qquad \dot{y}_h = \dot{y}_h \tag{A.16}$$

where $\overrightarrow{H}^{-}_{/i}$ and $\overrightarrow{H}^{+}_{/i}$ are the angular momentum about the point i before and after heel-strke. In section A.2.3, we give the detailed expansions of these expressions.

The equations A.14 to A.16 can be re-arranged to give the following equation for heel-strike phase,

$$\mathbf{A}_{hs}\mathbf{X}_{hs} = \mathbf{b}_{hs} \tag{A.17}$$

where the unknown is the 8×1 vector, $\mathbf{X}_{hs} = [\dot{q}_1\ ,\ \dot{q}_2\ ,\ \dot{q}_{2m}\ ,\ \dot{q}_3\ ,\ \dot{q}_4\ ,\ \dot{q}_{4m}\ ,\ P^\star_x\ ,\ P^\star_y]'$, while the 8×8 matrix $\mathbf{A}_{hs}$ and the 8×1 vector $\mathbf{b}_{hs}$ are known and can be found in a similar way as found for the double stance equations.

Single-stance to single-stance collisional transition. The equations presented next are for gait sequences that do not have any double stance; i.e., the gait sequence is single stance → heel-strike → single stance. This sequence is not used on ranger, but is used as a special case to benchmark the equations of motion (appendix B) and optimal trajectory (appendix D).

The collisional jump equations for the single stance to single stance transition can be derived by similar means as used for the single stance to double stance transition as described before. We use the same figure A.5 for this derivation, but 1) leave off the kinematic constraint that the former stance foot maintain contact; 2) leave off the related constraint impulses (set the impulses on the former stance foot to zero). The

jump equations are:

$$q_1 = r_1 + r_2 - r_3 - r_4 \qquad q_3 = -r_3 \tag{A.18}$$

$$q_{2m} = r_{4m} \qquad q_2 = r_4 \tag{A.19}$$

$$q_4 = r_2 \qquad q_{4m} = r_{2m} \tag{A.20}$$

$$\overrightarrow{H}^{+}_{/P_1} = \overrightarrow{H}^{-}_{/P_2'} \tag{A.21}$$

$$\overrightarrow{H}^{+}_{/A_1} = \overrightarrow{H}^{-}_{/A_2'} \tag{A.22}$$

$$\overrightarrow{H}^{+}_{/H} = \overrightarrow{H}^{-}_{/H'} \tag{A.23}$$

$$\dot{q}_4 = \dot{r}_{2m} \qquad \dot{q}_{2m} = \dot{r}_{4m} \qquad \dot{q}_{4m} = \dot{r}_{2m} \tag{A.24}$$

$\overrightarrow{H}^{-}_{/i}$ and $\overrightarrow{H}^{+}_{/i}$ are the angular momentum about the point i before and after heel-strke. Formulas for these follow.

A.2.3 Full expansion of terms in the single stance, double stance and heel-strike equations

The external moment $\overrightarrow{M}_{/i}$ and rate of change of angular momentum $\dot{\overrightarrow{H}}_{/i}$ about different points in the above equations are given next. We define $\delta_{DS} = 1$ in double stance and $\delta_{DS} = 0$ in single stance in the equations below.

$$\overrightarrow{M}_{/P_1} = \vec{r}_{G_1/P_1} \times m\,\vec{g} + \vec{r}_{G_2/P_1} \times m\,\vec{g} + \vec{r}_{P_2/P_1} \times \delta_{DS}\,\vec{P} + (T_{1FW} + \delta_{DS} T_{2FW})\hat{k}$$

$$\dot{\overrightarrow{H}}_{/P_1} = \vec{r}_{G_1/P_1} \times m\,\vec{a}_{G_1} + J_\ell\,\vec{\alpha}_1 + \vec{r}_{G_2/P_1} \times m\,\vec{a}_{G_2} + J_\ell\,\vec{\alpha}_2$$

$$\overrightarrow{M}_{/A_1} = \vec{r}_{G_1/A_1} \times m\,\vec{g} + \vec{r}_{G_2/A_1} \times m\,\vec{g} + \vec{r}_{P_2/A_1} \times \delta_{DS}\,\vec{P} + (T_{2S} + \delta_{DS} T_{2FW})\hat{k}$$

$$\dot{\overrightarrow{H}}_{/A_1} = \vec{r}_{G_1/A_1} \times m\,\vec{a}_{G_1} + J_\ell\,\vec{\alpha}_1 + \vec{r}_{G_2/A_1} \times m\,\vec{a}_{G_2} + J_\ell\,\vec{\alpha}_2$$

$$\overrightarrow{M}_{/H} = \vec{r}_{G_2/H} \times m\,\vec{g} + \vec{r}_{P_2/H} \times \delta_{DS}\,\vec{P} + (T_3 - T_{3S} + \delta_{DS} T_{2FW})\hat{k}$$

$$\dot{\overrightarrow{H}}_{/H} = \vec{r}_{G_2/H} \times m\,\vec{a}_{G_2} + J_\ell\,\vec{\alpha}_2$$

$$\overrightarrow{M}_{/A_2} = \vec{r}_{P_2/A_2} \times \vec{P} + (T_{4S} + T_{2FW})\hat{k} \qquad \text{Only in double stance}$$

$$\dot{\overrightarrow{H}}_{/A_2} = 0 \qquad \text{Only in double stance}$$

The angular momentum $\overrightarrow{H}_{/i}$ about different points in the heel-strike equations are given next.

$$\overrightarrow{H}^{+}_{/P_1} = \vec{r}_{G_1/P_1} \times m\,\vec{v}_{G_1} + J_\ell\,\vec{\omega}_1 + \vec{r}_{G_2/P_1} \times m\,\vec{v}_{G_2} + J_\ell\,\vec{\omega}_2$$

$$\overrightarrow{H}^{-}_{/P'_2} = \vec{r}_{G'_1/P'_2} \times m\,\vec{v}_{G'_2} + J_\ell\,\vec{\omega}'_1 + \vec{r}_{G'_2/P'_2} \times m\,\vec{v}_{G'_2} + J_\ell\,\vec{\omega}'_2$$

$$\overrightarrow{H}^{+}_{/A_1} = \vec{r}_{G_1/A_1} \times m\,\vec{v}_{A_1} + J_\ell\,\vec{\omega}_1 + \vec{r}_{G_2/A_1} \times m\,\vec{v}_{G_2} + J_\ell\,\vec{\omega}_2$$

$$\overrightarrow{H}^{-}_{/A'_2} = \vec{r}_{G'_1/A'_2} \times m\,\vec{v}_{G'_2} + J_\ell\,\vec{\omega}'_1 + \vec{r}_{G'_2/A'_2} \times m\,\vec{v}_{G'_2} + J_\ell\,\vec{\omega}'_2$$

$$\overrightarrow{H}^{+}_{/H} = \vec{r}_{G_2/H} \times m\,\vec{v}_{G_2} + J_\ell\,\vec{\omega}_2$$

$$\overrightarrow{H}^{-}_{/H'} = \vec{r}_{G'_1/H'} \times m\,\vec{v}_{G'_1} + J_\ell\,\vec{\omega}'_1$$

$$\overrightarrow{H}^{+}_{/A_2} = \overrightarrow{H}^{-}_{/A'_1} = \vec{0}$$

APPENDIX B

BENCHMARK TESTS OF THE EQUATIONS OF MOTION

To validate the equations of motion we consider various special cases about which much is known. In particular, we reduced the equations of motion of Ranger to that of a passive walker, introduced a ramp, found stable limit cycles and compared our results with previously published results. Two special passive cases were considered: 1) a rimless wheel [22, 71], and the simplest walker [41, 42]. Without such checks how are we to trust our equations?

B.1 Recipe for analyzing passive dynamic walkers

A recipe for analyzing passive dynamic walkers has been presented in detail in Garcia's PhD thesis (see chapter 2 in [42]) and in Coleman's PhD thesis (see chapter 1 in [22]). Here is a summary.

1. Create a complete mechanical model of the walker. This includes defining model assumptions, defining model parameters, and deriving the equations of motion of the walker. The equations of motion generally involve smooth portions of the walk like single stance phase and discontinuous portions like heel-strike.
2. Define a Poincare map (McGeer's stride function) that maps the state of the system from one step to the next. Here we used the map f from the state of the system q_n just after one heel-strike to the state q_{n+1} just after the next heel-strike. We use numerical root finding (e.g. Newton-Raphson method or bisection method) to find fixed points (roots of the function $f(q) - q$).
3. Find the Jacobian of the stride function at this root using numerical differentiation. If the magnitude of the biggest eigenvalue of the Jacobian is smaller than one than

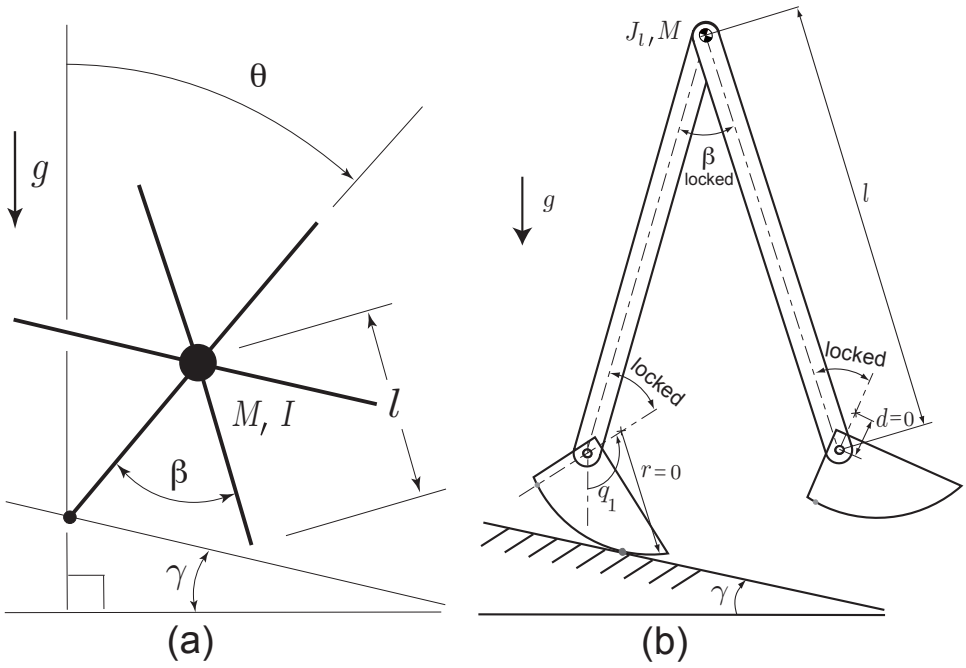

Figure B.1: **2-D rimless wheel.** (a) 2-D rimless wheel analyzed by Coleman. (figure source: Coleman's PhD thesis [22]) (b) Ranger model simplified to a rimless wheel; The hip angle and ankle angles are locked and the centers of mass of the leg are put at the hip.

the system is stable, otherwise it is not.

B.2 Reduction to 2-D rimless wheel

Figure B.1a shows the 2-D rimless wheel analyzed by Coleman. The mass at the center is M, legs have inertia I_ℓ, legs length is ℓ and number of spokes is n. The inter-spoke angle β is calculated from the number of spokes and is given by $\beta = 2\pi/n$. The ramp slope is γ. Coleman found that the roots and Jacobian of the stride function depend on the number of spokes n, the slope γ and the non-dimensional term $\lambda^2 = M\ell^2/(I_\ell + M\ell^2)$. Coleman reports analytical results for $n = 6$, $\gamma = 0.2$ and $\lambda^2 = 2/3$ [23]. We will use these parameters for the benchmark.

Ranger model reduced to a rimless wheel. To derive the equations of motion we proceed as follows. First, we assume the gait sequence; single stance phase followed by heel-strike phase followed by single stance phase and so on. Next, we draw the free body diagram for Ranger model shown in figure B.1b. Finally, we use angular momentum balance about the foot contact points to derive the equations of motion.

Alternately, the equations of motion can be obtained from Ranger's equations presented earlier. We ignore the equation for angular momentum balance of the swing leg about the hip. And in all other equations we set the hip angle to be constant (no acceleration, no discontinuity in velocity at the collision). The equation of single stance are obtained from equation A.8 with $i = P_1$. The equation for heel-strike is obtained from equations A.18 and A.21.

Single stance (continuous): $\dot{\overrightarrow{H}}_{/P_1} = \overrightarrow{M}_{/P_1}$

Heel-strike (instantaneous): $q_1 = r_1 - r_3 \qquad q_3 = -r_3$

$\overrightarrow{H}^{+}_{/P_1} = \overrightarrow{H}^{-}_{/P_2'}$

Table B.1 gives the parameters of the 2-D rimless wheel Ranger model.

Comparison of fixed points. We analyzed the simplified Ranger model with the parameters given in table B.1 and using the recipe presented in section B.1.

The fixed points based on analytical solutions is given in [23] and is $(\theta_1^*, \dot{\theta}_1^*) = (\pi/n, -0.4603411266094583)$. Using an adaptive step integrator (Runge-Kutta 45) with integration tolerance of 10^{-13} and a similar root finder accuracy, we calculated the fixed points as $(q_1^*, \dot{q}_1^*) = (0.523598775598278, -0.460341126609482)$ which is accurate to the 13th decimal place.

(a) Rimless wheel

Parameter	Value
ℓ	1 m
r	0
d	0
w	0
c	0
k_h	Not in equations
k_s	Not in equations
J_ℓ	0.25 kg m^2
m	0
M	1 kg
g	1 m/s^2
γ	0.2
C_{1FW}	0
C_{2FW}	Not in equations

(b) Simplest walker

Parameter	Value
ℓ	1 m
r	0
d	0
w	0
c	1 m
k_h	Not in equations
k_s	Not in equations
J_ℓ	0
m	1 kg
M	10^6 kg
g	1 m/s^2
γ	0.009
C_{1FW}	0
C_{2FW}	Not in equations

Table B.1: **Reduction of Ranger to simpler cases.** (a) Values of Ranger parameters for model reduction to a 2-D rimless wheel. (b) Ranger parameters for model reduction to simplest walker.

Comparison of eigenvalues. The biggest eigenvalue based on analytical solution [23] is $\sigma = 4/9 = 0.44444444444444\bar{4}$. Using fixed point found earlier and using perturbation of 10^{-5}, we calculated the Jacobian of the linearized map using central difference. We computed the biggest eigenvalues as $\sigma = 0.444444444411274$. Our Ranger-based eigen-value is accurate to 10th decimal place.

B.3 Reduction to 2-D simplest walker

A more stringent comparison is with 'The Simplest Walker' which has a non-locked swing leg.

Figure B.2 (a) shows the 2-D simplest walker analyzed by Garcia [41]. The simplest walker has a point-mass M at the hip. The legs of length ℓ are nearly massless, but

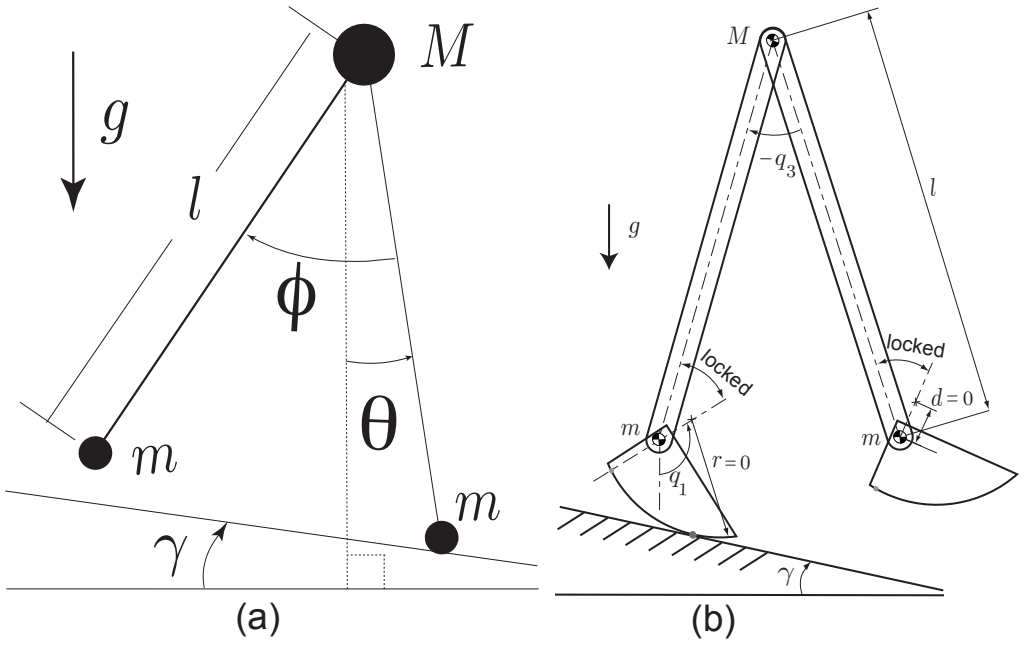

Figure B.2: **2-D simplest walker.** (a) Simplest walker analyzed by Garcia (figure source: Garcia's PhD thesis [42]) (b) Ranger model simplified to the simplest walker

there is a point-mass $m \ll M$ at the end of the swing leg. Garcia considered the case, $m/M = 0$. He found that the non-dimensional equations of motion are functions of a single parameter; the slope of the ramp γ. Garcia [41] does not report benchmark results for the simplest walker. So, using the equations derived by him and for a slope of $\gamma = 0.009$, we computed the fixed points and eigenvalues. These values are reported here and are used as benchmarks.

Ranger model reduced to the simplest walker. To derive the equations of motion we proceed as follows. First, we assume the gait sequence; single stance phase followed by heel-strike phase followed by single stance phase and so on. Next, we draw the free body diagram for Ranger model shown in figure B.2b. Finally, we use angular momentum balance about appropriate points to derive equations of motion.

Alternately, the equations of motion can be obtained from Ranger's equations of motion presented earlier. The equation of single stance are obtained from A.8 with $i = P_1, H$. The equation for heel-strike is obtained from equations A.18, A.21 and A.23 and A.21.

Single stance (continuous):	$\dot{\overrightarrow{H}}_{/P_1} = \overrightarrow{M}_{/P_1}$	$\dot{\overrightarrow{H}}_{/H} = \overrightarrow{M}_{/H}$
Heel-strike (instantaneous):	$q_1 = r_1 - r_3$	$q_3 = -r_3$
	$\overrightarrow{H}^{+}_{/P_1} = \overrightarrow{H}^{-}_{/P_2'}$	$\overrightarrow{H}^{+}_{/H} = \overrightarrow{H}^{-}_{/H'}$

Table B.1 gives the parameters for the reduced Ranger model to match up with Garcia's simplest walker model.

Comparison of Ranger's reduction to the simplest walker. We analyzed the simplified Ranger model using the parameters given in table B.1 and using the recipe presented in section B.1.

Comparison of fixed points. First using MATLAB's ODE45 (mixed 4th and 5th order Runge Kutta algorithm) with integration and root finder tolerance of 10^{-13}, and with Garcia's equations of motion [42], we calculated a fixed point, $(\theta^*,\dot{\theta}^*,\phi^*,\dot{\phi}^*)$ = (0.200310900544287, −0.199832473004977, 0.400621801088574, −0.015822999948318).

Next, using the same tolerances for integration and root finder, but with Ranger's reduced equations of motion, we re-calculated the fixed points. We found the fixed point to be, $(q_1^*,\dot{q}_1^*,q_3^*,\dot{q}_3^*)$ = (0.200310750572992, −0.199832546623645, 0.400621501145995, −0.015822982402157). The fixed points differ in the 6th decimal place. This is consistent with the $m/M = 10^{-6} \neq 0$ that we used.

Comparison of eigenvalues. Using the fixed point from Garcia's equations of motion and using a perturbation of 10^{-5}, we calculated the Jacobian of the linearized map using central difference. The non-zero eigenvalues of the linearized map were found to be, $\sigma_1 = -0.190099639087901 + 0.557599274928213i$ and $\sigma_2 = -0.190099639087901 - 0.557599274928213i$.

Using Ranger's reduced equations of motion and the above method to calculate the Jacobian, we got the following non-zero eigenvalues, $\sigma_1 = -0.190106213101483 + 0.557586570259502i$ and $\sigma_2 = -0.190106213101483 - 0.557586570259502i$. The eigenvalues agree to 3 decimals.

APPENDIX C

SMOOTHINGS FOR SIMULATIONS AND OPTIMIZATIONS

Because of the reversals of contact voltage at current reversals and of friction force at velocity reversals various terms in the simulations and optimizations are not smooth.

There is some subtlety in the reason for the lack of smoothness in the differential equations to survive to the optimization objective function. If the discontinuities were simply crossed they would not lead to discontinuities in the objective function. But because the optimal trajectories tend to sit on the discontinuities for extended times, not just passing through them, they do survive to the optimizations.

To eliminate the related numerical issues, especially problems with convergence of the optimization software, we smooth such discontinuities.

The smoothings are also needed in our benchmark optimizations based on mechanical power, which, assuming no regeneration, has a discontinuity at $x = 0$.

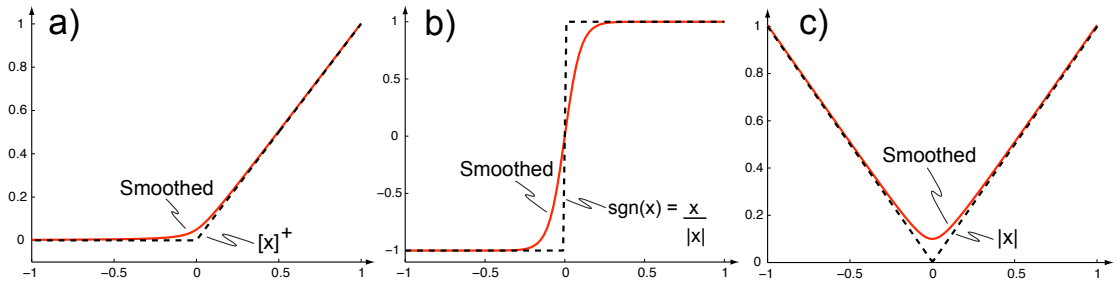

Figure C.1: **Smoothings for discontinuous functions.** The smoothings for (a) the unit ramp function, (b) the step function and (c) the absolute value function are shown. Here dotted black lines represents the true function while solid red lines represents our approximation.

Smoothing for $[x]^+$. Steps (Heaviside functions) in voltage and force only lead to ramps (integral of Heaviside function) in the electrical power (which is integrated to obtain our objective function). So our main concern is with smoothing ramp functions.

The unit ramp function $[x]^+$ is zero for negative x and simply x for positive x:

$$[x]^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{C.1}$$

The function has a kink at $x = 0$. A smooth approximation to $[x]^+$ is given in by [96, 102].

$$[x]^+ \approx \frac{x + \sqrt{x^2 + \epsilon_1^2}}{2} \tag{C.2}$$

As a rule we use $\epsilon_1 = 0.01$. Figure C.1a compares the smooth approximation with the actual function.

Smoothing for sgn(x). When needed we also smooth the signum function $\text{sgn}(x)$:

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{C.3}$$

The function is discontinuous at $x = 0$. A smooth, continuous approximation of $\text{sgn}(x)$ is in [30].

$$\text{sgn}(x) \approx \tanh\left(\frac{x}{\epsilon_2}\right) \tag{C.4}$$

We generally use $\epsilon_2 = 0.01$. Figure C.1b compares the smooth approximation with the actual function.

Smoothing for $|x|$. The absolute value function can be defined as:

$$|x| = \begin{cases} x & \text{if } x > 0 \\ -x & \text{if } x \leq 0 \end{cases} \tag{C.5}$$

This function also has a kink at $x = 0$. A smooth approximation to $|x|$ is given in equation by [96, 102] as

$$|x| \approx \sqrt{x^2 + \epsilon_3^2} \tag{C.6}$$

Again we generally us $\epsilon_3 = 0.01$. Figure C.1c compares the smooth approximation with the actual function.

APPENDIX D

BENCHMARKS FOR OPTIMAL TRAJECTORY CONTROL

The trajectories used on Ranger are based on optimizations. To check the veracity of our optimization calculations we optimize situations in which we already have some understanding of the optima and which also can be made to conform to the overall Ranger governing equations and framework, namely:

1. passive dynamic walking [70], and
2. point-mass energy-optimal walking [98].

These tests are explained below.

D.1 Discovering passive dynamic walking

McGeer showed that a 2D 2-legged robot with suitable mass distribution can walk down a shallow slope with no actuation [70]. For the first validation of the Ranger simulation we keep (and optimize) the hip motor (but neglect friction). To keep this check simple we locked the ankles. We know from previous passive-dynamics research that on a small slope this model has periodic solutions with zero hip torque and thus, with a simplified motor model, zero hip current. We also know that for the motor model the minimum conceivable energy use is zero, with zero current and thus zero torque at all times. Thus the optimal control solution should be one with zero hip current for all time, namely passive-dynamic walking. Here is how we checked for the optimization's ability to make this discovery.

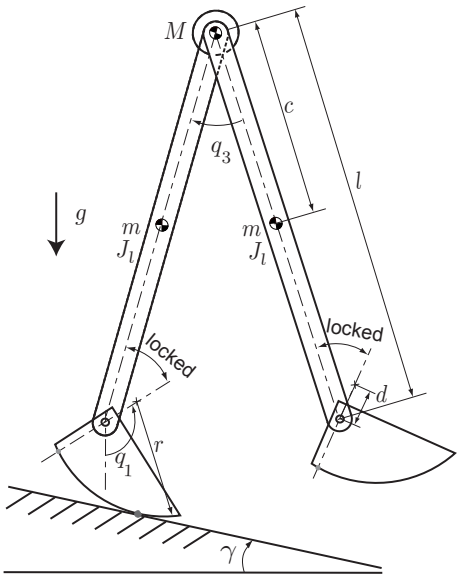

Figure D.1: **Passive dynamic walking validation.** Ranger with ankles locked on a shallow slope. The hip motor can power walking.

Model used for 'discovery' of passive dynamic walking. The robot model is shown in figure D.1. The equations of motion are simplified from Ranger's governing equations. The equation of single stance is obtained from A.8 by putting $i = P_1, H$. The equation for heel-strike is obtained from equations A.18, A.21 and A.23. Note that we *a priori* preclude a double-stance phase.

For this optimization we use a simplified (i.e., frictionless) motor model

$$\text{Power model:} \qquad P = I^2R + G_H K I \omega$$

$$\text{Torque model:} \qquad T = G_H K I$$

where the hip power is P, the hip torque is T, hip motor current is I, hip motor speed ω, motor resistance is R, motor constant is K, and hip motor gear ratio is G_H. Note that this model is torque-free when the electrical power (and current) is zero.

Parameter	Value
ℓ	0.96 m
r	0.2 m
d	0.11 m
k_h	0
k_s	Not in equation
J_ℓ	0.24 kg m^2
m	2 kg
M	4.5 kg
w	0 m
c	0.3 m
g	10 m/s^2
γ	0.005
C_{1FW}	0
C_{2FW}	Not in equation

Table D.1: **Ranger parameter values for discovering passive dynamic walking.**

Parameter	Value
G_A	Not in equation
G_H	66
K	0.017 N m/A
R	1.3 Ω
V_c	0
J_m	Not in equation
μ_H	0
μ_A	0
C_{H1}	0
C_{H0}	0
C_{A1}	0
C_{A0}	0
P_{fixed}	Not in equation

Table D.2: **Motor parameter values for discovering passive dynamic walking.** Symbols A and H denote ankle and hip respectively.

Optimization problem for 'discovery' of passive dynamic walking. We seek a control strategy (hip motor current as a function of time) and initial conditions, that mini-

mize the cost

$$\text{COT} \;=\; \frac{\text{Energy used per step}}{\text{Weight} \times \text{Step length}} \;=\; \int_{t=0}^{t=t_{\text{step}}} \frac{|P|dt}{M_{\text{tot}}\, g\, d_{\text{step}}} \tag{D.1}$$

subject to the following constraints: periodicity, namely that the state vector at the beginning of single stance should be equal to the state vector just after heel-strike; and the vertical reaction forces on the grounded legs should be positive at all times.

The absolute value of hip power P in the COT expression above is not smooth, so we smooth it in such a way that our gradient-based optimization software will perform better (see appendix C).

Parameterization of the optimal trajectory. The optimization parameters are:

- The state at beginning of single stance ($\chi^i_{ss}(t=0)$);
  $\chi^i_{ss}(t=0) \;=\; [q_1\,,\; q_3,\; \dot{q}_1\,,\; \dot{q}_3]_{ss}(t=0)$.
- Step time (t_{step})
- The currents in the hip motor in single stance ($I(t)$). We assume piecewise linear-in-time currents. In single stance, we divide time into N intervals; $t_0, t_1, \ldots, t_N$. Here $t_{i+1} - t_i \;=\; 1/N,\; i \;=\; 0, 1, ..., N,\; t_0 \;=\; 0$ and $t_N \;=\; t_{\text{step}}$. This means we have $2(N+1)$ unknowns for currents in single stance; $I(t=t_0), I(t=t_1), \ldots, I(t=t_N)$.

Constraints. The optimization proceeds subject to various constraints on the optimization parameters and things calculated from those parameters:

- The periodicity constraints of the state at the beginning of single stance should match the state just after heel-strike; $\chi^i_{ss}(t=0) = \chi^+_{hs}(t=t_{\text{step}})$.

- The transition from single stance to heel-strike takes place when the swinging leg's foot hits the ground at time $t = t_{\text{step}}$, as expressed in equation A.2 .
- Vertical ground reaction force for the foot on the ground in single stance should be positive and this is enforced at the grid points. This gives $N + 1$ inequality constraints.

Method of optimization. With the initial conditions in single stance $\chi^i_{ss}(t = 0)$, we integrate equations of motion for single stance till we reach the end of single stance at time $t = t_{\text{step}}$. The torques for use in the single stance equations are obtained from the motor's torque equation using the assumed piecewise linear currents in single stance. Integration is from grid point to grid point so there are no discontinuities during an integration step.

After the integration reaches the end of single stance, we apply the heel-strike condition to get the state after heel-strike $\chi^+_{hs}(t = t_{\text{step}})$.

For each integration of a full step the total energy cost is evaluated. The optimization software varies the various optimization parameters attempting to minimize this cost (equation D.1), while respecting the various constraints described above. We used the sequential quadratic programming package SNOPT [43] to solve the optimization problem.

Results

Passive walking. First, without any optimization, we looked for passive solutions for the set of Ranger's parameters assumed here using the root-finding procedure outlined in section B.1. We found two such passive solutions. The

first solution had a step time of $t_{\text{step}} = 0.79329202031544$, and fixed points of $(q_1^*, \dot{q}_1^*, q_3^*, \dot{q}_3^*) = (2.99716718892289, -0.54617378605291, -0.28885092932654, -0.53131730110868)$. The second solution had a step time of $t_{\text{step}} = 0.93624752862330$, and fixed points of $(q_1^*, \dot{q}_1^*, q_3^*, \dot{q}_3^*) = (2.96417643512919, -0.65228639005082, -0.35483243689880, -0.02749381544188)$. Generally people find two periodic solutions for the passive dynamic walkers [42, 70] for a given slope.

Passive walking is discovered. Next, we ran the trajectory optimization starting with initial guesses far from the passive solutions. Each time the optimization converged to one of the above two solutions and with zero current for all time. Thus the trajectory optimization successfully discovered passive-dynamic walking.

D.2 Discovering optimal level-ground walking of a point-mass model

Srinivasan, Ruina [98] and Srinivasan [96] present a point-mass legged locomotion model. Using energy-optimal trajectory control, they show that the model chooses to walk at low speeds, run at fast speeds, and at intermediate speeds discovers a new kind of walk, which they call the pendular-run. For this benchmark we are only interested in the walking solution. In particular, the optimal strategy for walking is an impulsive push-off just before heel-strike followed by a stance phase consisting of motion as a simple inverted pendulum. For this second validation, we approximate the point-mass model by putting most of Ranger's mass on the hip, and making the legs light. We leave the foot eccentricity non-zero but make the foot zero radius. In effect this makes the leg extensible, running between the hip and the infinitesimal foot (which is not inline

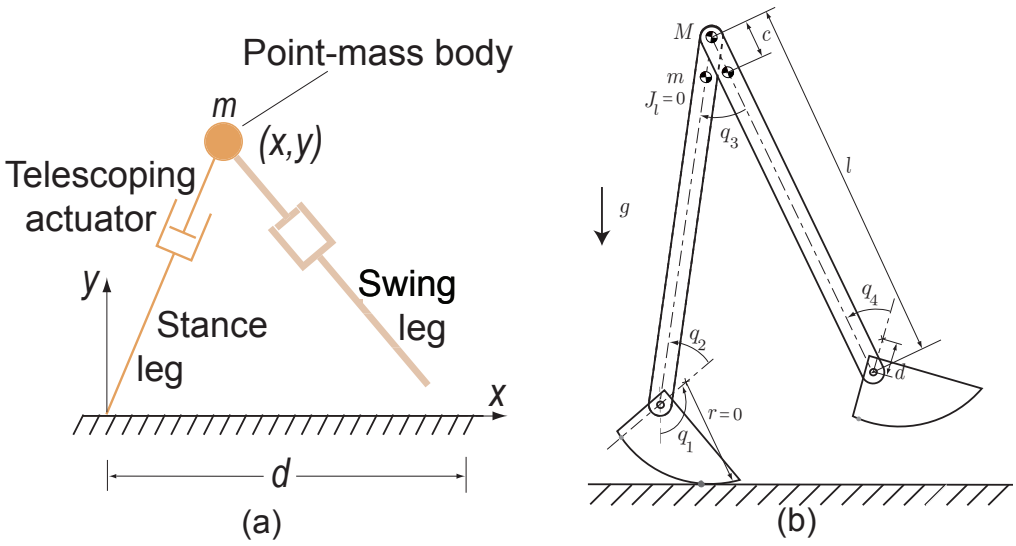

Figure D.2: **Point-mass level walking validation.** (a) Point-mass model analyzed by Srinivasan (figure source: Srinivasan's PhD thesis [96]) (b) Ranger model simplified to a point-mass model by various special parameter values. In particular the foot radius r is set to zero, making the foot a point at a distance d from the ankle.

with the leg). We then see if the optimization discovers optimum level walking with an impulsive push-off just before heel-strike.

Model for comparison with point-mass locomotion. The model is shown in figure D.2b. We obtain the equations of motion from Ranger's equations of motion. The equations for single stance are obtained from A.8 with $i = P_1, A_1, H$. The equations for

Parameter	Value
ℓ	1 m
r	0
d	0.05 m
k_h	0
k_s	Not in equation
J_ℓ	Not in equation
m	0.01 kg
M	1 kg
w	0
c	0.5 m
g	1 m/s^2
γ	0
C_{1FW}	0
C_{2FW}	Not in equation

Table D.3: **Ranger parameter values used to discover energy-optimal level walking.**

heel-strike are obtained from equations A.18 to A.23.

Single stance (continuous): $\dot{\overrightarrow{H}}_{/i} = \overrightarrow{M}_{/i}$ where $i = P_1, A_1, H$

Heel-strike (instantaneous):

$$q_1 = r_1 - r_3 \qquad q_3 = -r_3$$

$$q_2 = r_4 \qquad q_4 = r_2$$

$$\overrightarrow{H}^{+}_{/P_1} = \overrightarrow{H}^{-}_{/P'_2}$$

$$\overrightarrow{H}^{+}_{/A_1} = \overrightarrow{H}^{-}_{/A'_2}$$

$$\overrightarrow{H}^{+}_{/H} = \overrightarrow{H}^{-}_{/H'}$$

Optimal trajectory control problem for comparison with the point-mass model. We assume a walking sequence given by these phases: single stance, heel-strike, single stance, repeat. We seek a control strategy that includes finding initial conditions in single stance, and torque in the hip and stance ankle motor as a function of time, that minimize

the cost given by,

$$\text{COT} = \frac{\text{Energy used per step}}{\text{Weight} \times \text{Step length}} \tag{D.2}$$

We use the following constraints: periodicity requires that the state vector at the beginning of single stance should be equal to the state vector just after heel-strike; step length and step velocity are specified; and the vertical reaction forces on the grounded legs should be positive at all times. Because the solution we are trying to discover has infinite forces, the numerics are helped by constraining hip and ankle motor torques to be within given bounds. Convergence to the singular solution is inferred by the torques always using the bounds, no matter how high (as explained in [98]).

Numerical formulation of point-mass optimal trajectory problem

Cost. The cost is the energy metric cost of transport and is defined as,

$$\text{COT} = \int_{t=0}^{t=t_{\text{step}}} \frac{\{|T_2 u_2| + |T_3 u_3|\} dt}{M_{\text{tot}}\, g\, d_{\text{step}}} \tag{D.3}$$

The ankle and hip torques are T_2 and T_3 respectively. The ankle and hip speeds are u_2 and u_3 respectively. The absolute value is not a smooth function; it has a kink at 0. So we smooth this function as shown in appendix C. The numerator in equation D.3 is the mechanical power at the joints, counting negative and positive power equally.

Parameters. The optimization parameters are:

- State at beginning of single stance ($\chi^i_{ss}(t = 0)$);
  $\chi^i_{ss}(t = 0) \;=\; [q_1\,,\; q_2\,, q_3,\; \dot{q}_1\,,\; \dot{q}_2\,,\; \dot{q}_3]_{ss}(t = 0)$.
- Note that we ignore foot scuffing altogether and that we do not model the motion of the foot attached to the swinging leg's foot. However, we do need to specify

the touchdown angle of the swinging leg's foot (q_4^{hs}) at heel-strike and this is an optimization parameter.

- Step time (t_{step})
- Torque in the motors in single stance ($T_2(t)$, $T_3(t)$). We assume piecewise linear torques. In single stance, we divide time into N intervals; $t_0, t_1, \ldots, t_N$. Here $t_{i+1} - t_i = 1/N$, $i = 0, 1, ..., N$, $t_0 = 0$ and $t_N = t_{\text{step}}$. This means we have $2(N + 1)$ unknowns for torques in single stance; $T_j(t = t_0), T_j(t = t_1), \ldots, T_j(t = t_N)$, where $j = 2, 3$.

Constraints. The optimization constraints are:

- Periodicity constraints: the state at the beginning of single stance should match the state just after heel-strike; $\chi_{ss}^{i}(t = 0) = \chi_{hs}^{+}(t = t_{\text{step}})$.
- Step velocity is specified, $v_{\text{step}} = V$.
- Step length is specified, $d_{\text{step}} = D$.
- Transition from single stance to heel-strike takes place when the swinging leg's foot hits the ground at time $t = t_{\text{step}}$. This constraint is given by equation A.2.
- Vertical ground reaction force for the foot on the ground in single stance should be positive. This is enforced at the grid points. This gives $N+1$ inequality constraints.
- Torques in the hip and ankle motors have to be within the actuator limits These are enforced at grid points and correspond to $2(N + 1)$ inequality constraints.
- There are singularities in the governing equations and in the behavior of the optimal solution near q_2 or q_4 equal zero or π. To avoid this the ankle angle was constrained to be $\pi/2$ at heel strike, i.e., $q_4^{hs} = \pi/2$.

Method of optimization. With the initial conditions in single stance $\chi^i_{ss}(t = 0)$, we integrate equations of motion for single stance till we reach the end of single stance at time $t = t_{\text{step}}$. After we reach the end of single stance, we apply the heel-strike condition to get the state after heel-strike $\chi^+_{hs}(t = t_{\text{step}})$. The optimization software varies the various optimization parameters listed above, in an attempt to minimize the cost given in equation D.3, while respecting the various constraints described earlier. Again, we used the sequential quadratic programming package SNOPT [43] which we use later for our controlled-robot optimization.

Results

We present results obtained for the step velocity of $V = 0.4$ and step length of $D = 0.4$. Srinivasan [96] pp. 24-25 gives formulas for calculating the COT at various step length and step velocity combinations. Using his formulas, we calculated the ideal point-mass COT to be 0.009882649139799 at $V = D = 0.4$.

Using Ranger's reduced point-mass model and using a grid size $N = 12$, we calculated the COT to be 0.010113205021986. The error between our result and analytical calculations is about 2%. This error is consistent with numerical optimization results presented by Srinivasan [96] on pp. 66 Table 3.1, also taking account that this model, as opposed to the point-mass comparison, has a small, but non-zero leg swing cost.

Figure D.3 shows the trajectories for the ankle joints position and velocity, the actuator torques and the mechanical power versus time. The hip motion is low power throughout, due to the light legs. Almost all the energy for walking goes to the ankles to generate the push-off. By increasing the grid size N, we found that that the push-off becomes more pre-emptive, decreasing its duration and increasing the peak. This results

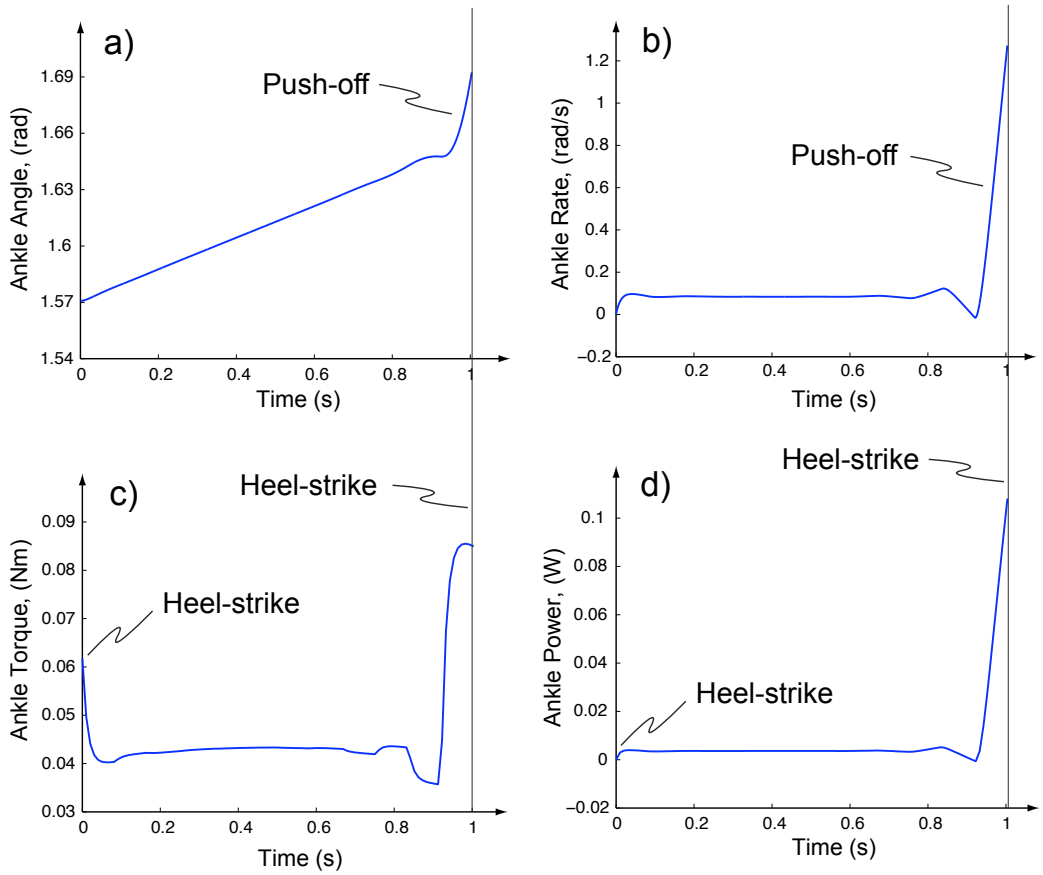

Figure D.3: **Ankle trajectory and controls for point-mass model limit of the Ranger model.** The results of the optimization of the Ranger model, with parameters chosen to mimic a point-mass model, are shown. (a) The ankle angle shows sudden lengthening at push-off; (b) The ankle rate, being near constant for the small-angle inverted-pendulum phase; (c) The ankle torque, which has no cost in this model when the ankle rate is zero, the optimizations seeming attempt to discover an impulse is shown by the spike at the right; and (d) The ankle power, which is effectively zero but for a sudden, seemingly-attempting-to-be-singular rise at push off.

suggests that the push-off tends to an impulse as the grid is made to grow infinitely big.

These results are in agreement with those found by Srinivasan and Ruina [98].

APPENDIX E

TRAJECTORY OPTIMIZATION OF ROBOT: FINE-GRID

We use the same optimization software and same general approach described in the appendix D to optimize energy use based on the full robot and motor models. As described in chapter 5, we first do a fine-grid optimization, described in here, and then a coarse-grid optimization (appendix F) for use on the robot.

The trajectory optimization for the full robot is infinite dimensional. We resort to a discrete approximation of the problem but with a grid that is fine enough so that, within our model and measurement accuracy, we have achieved an optimum.

E.1 Foot-flip energy optimization

Cost. We are interested in minimizing the energy for foot-flip and given by,

$$\mathrm{E}_{\text{foot-flip}} = 2 \int_{t=t_0=0}^{t=t_f=0.25s} [P_{\text{foot-flip}}]^+ \, dt \tag{E.1}$$

$P_{\text{foot-flip}}$ is the power in the motor that turns the foot attached to the swing leg in single stance. The factor 2 accounts for flip up and flip down. We assume that our batteries cannot regenerate and to accommodate this fact we define the function $[P]^+$ as follows: $[P]^+ = P$, if $P \geq 0$ and $[P]^+ = 0$, if $P < 0$. The function [P]$^+$ is non smooth at $P = 0$. We smooth the function using equation C.2 as shown in appendix C. Also, in the equation above we have heuristically assigned the flip up time to be 0.25 s. This implies that foot flip up and flip down takes a total time of 0.5 s, with additional swing time in the flip up configuration. Consequently, the single stance time has to be at least 0.5 s to prevent foot scuffing.

Parameters. The optimization parameters are the motor current as a function of time. We parameterize the current as piecewise linear functions of time. We divide the time into N^f intervals i.e. $t_0 = 0, t_1, \ldots, t_N^f = t_f$, where $t_{i+1} - t_i = 1/N^f$ and $i = 0, 1, ..., N^f$. Thus we have $(N^f + 1)$ unknowns for the current, $I_4^f(t = t_0), I_4^f(t = t_1), \ldots, I_4^f(t = t_{N^f} = t_f)$.

Constraints. The optimization constraints are as follows:

$$q_{4m}(t = 0) = 0 \tag{E.2}$$

$$q_{4m}(t = t_f) = 1.7 \text{ rad} \tag{E.3}$$

$$\dot{q}_{4m}(t = 0) = 0 \tag{E.4}$$

$$\dot{q}_{4m}(t = t_f) = 0 \tag{E.5}$$

$$-8 \text{ A} \leq I_4^f(t_i) \leq 8 \text{ A} \text{ where } i = 0, 1, ..., N^f \tag{E.6}$$

where q_{4m} and $\dot{q}_{4m}$ are the motor position and motor velocity respectively. The 1.7 rad in equation E.3 is the angle by which we want the swinging leg's feet to turn in order to clear the worst case obstacle. The actuator current limit of 8 A is specified in constraint equation E.6.

Methods. With the initial conditions for the motor angle and motor velocity given in E.2 and E.4, we integrate the equation of motion (see equation A.10) of the motor attached to the swinging leg's foot. The output motor torque for use in the equation of motion is calculated using the piecewise linear current profile. The optimization software varies the current parameters at the grid points, while respecting the current bounds in equation E.6. The goal of the optimization is to meet the terminal time constraints given in equations E.3 and equation E.5 and also minimizing the cost $E_{\text{foot-flip}}$. Again we used SNOPT [43].

E.2 'Walk' COT optimization

Cost. We are interested in minimizing the Total Cost of Transport, one term of which is given by

$$\begin{aligned} \mathrm{COT}_{\mathrm{walk}} &= \frac{\mathrm{E}_{\mathrm{walk}}}{M_{\mathrm{tot}}\, g\, d_{\mathrm{step}}} \\ &= \frac{\int_{t=0}^{t=t_{\mathrm{step}}} \sum [\mathrm{P}_{\mathrm{walk}}]^{+}\, \mathrm{dt}}{M_{\mathrm{tot}}\, g\, d_{\mathrm{step}}} \end{aligned} \tag{E.7}$$

P_{walk} is the power used by the motors but excluding the swing leg's ankle motor in single stance (whose power has already been accounted for in the foot-flip optimization in section E.1). We assume that our electronics and batteries cannot regenerate; to accommodate this we define the function $[P]^{+}$ as follows: $[P]^{+} = P$, if $P \geq 0$ and $[P]^{+} = 0$, if $P < 0$. The function $[\mathrm{P}]^{+}$ is non smooth at $P = 0$. We smooth the function using equation C.2 as shown in appendix C.

Parameters. The optimization parameters are as follows:

- State at beginning of single stance ($\chi^{i}_{ss}(t = 0)$) (see figure E.1a).

  $\chi^{i}_{ss}(t = 0) = [q_1\ ,\ q_2\ ,\ q_{2m}\ ,\ q_3,\ \dot{q}_1\ ,\ \dot{q}_2\ ,\ \dot{q}_{2m}\ ,\ \dot{q}_3]_{ss}(t = 0)$. We have skipped the variables $q_4, q_{4m}, \dot{q}_4, \dot{q}_{4m}$ as the swinging leg's foot is decoupled from the rest of the robot in single stance.

- State at the instant before heel-strike ($\chi^{-}_{hs}(t = \alpha t_{\mathrm{step}})$) (see figure E.1a).

  $\chi^{-}_{hs}(t = \alpha t_{\mathrm{step}}) = [q_1\ ,\ q_2\ ,\ q_{2m}\ ,\ q_3\ ,\ q_4\ ,\ q_{4m}\ ,\ \dot{q}_1\ ,\ \dot{q}_2\ ,\ \dot{q}_{2m}\ ,\ \dot{q}_3\ ,\ \dot{q}_4\ ,\ \dot{q}_{4m}]^{-}_{hs}(t = \alpha t_{\mathrm{step}})$. As the feet are taken to be massless, we assume that the swinging leg's foot motor angle coincides with the swinging leg's ankle angle just before heel-strike. Thus we have $q_4 = q_{4m}$ and $\dot{q}_4 = \dot{q}_{4m}$. Further, we assume that the motor attached

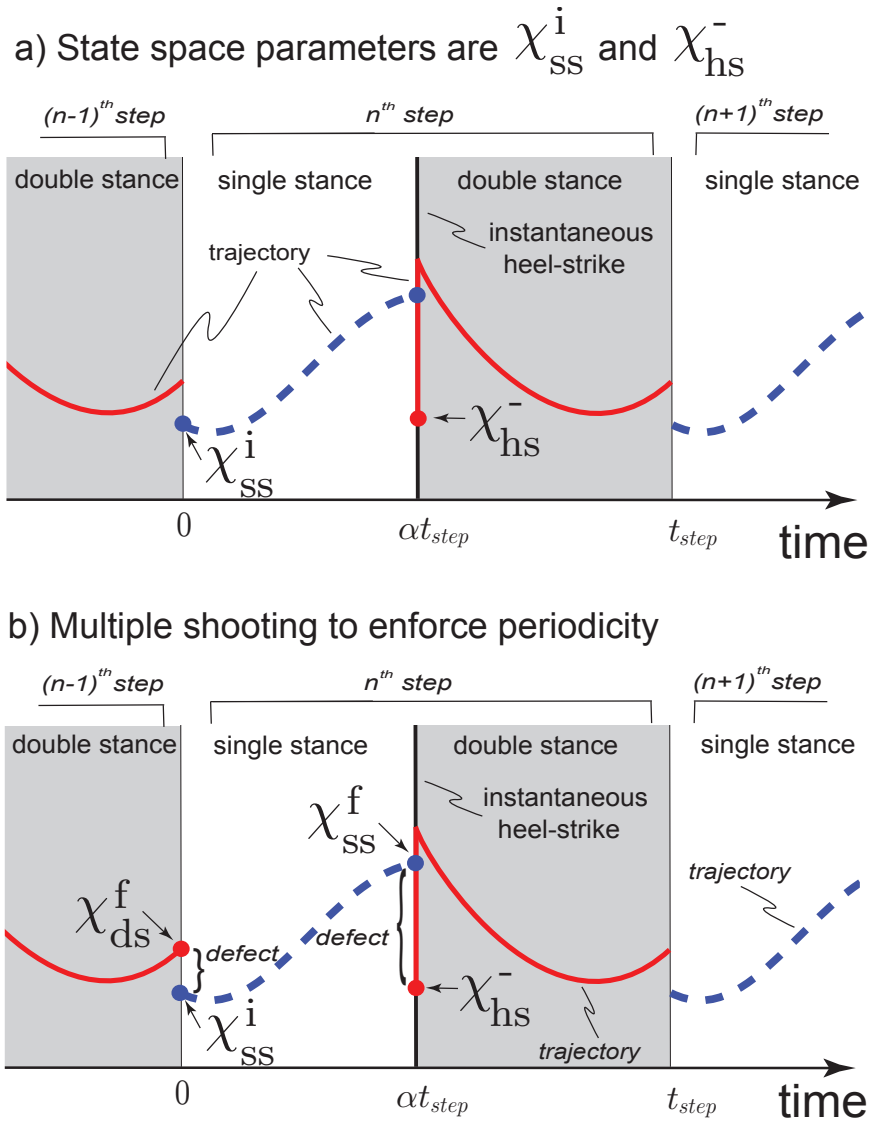

Figure E.1: **Multiple shooting method to solve optimal trajectory control problem.** (a) Initial condition at the beginning of single stance phase χ^{i}_{ss} and just before heelstrike χ^{-}_{hs} are the state space optimization parameters. The step time is t_{step} and fraction of time spent in single stance is α. (b) To enforce periodicity we equate the state in the beginning of single stance to that at the end of the double stance, i.e. $\chi^{i}_{ss}(t = 0) = \chi^{f}_{ds}(t = t_{\text{step}})$ and the state at the instant before heel-strike to that at the end of single stance $\chi^{-}_{hs}(t = \alpha t_{\text{step}}) = \chi^{f}_{ss}(t = \alpha t_{\text{step}})$. That is the optimization drives the defects, shown as discontinuities between red solid line and blue dashed curves in the figure, to zero.

to the swinging leg's foot and swinging leg's ankle are at rest just before heel-strike. Thus $\dot{q}_4 = \dot{q}_{4m} = 0$. The initial conditions before heel-strike simplifies to, $\chi^-_{hs}(t = \alpha t_{\text{step}}) = [q_1\ ,\ q_2\ ,\ q_{2m}\ ,\ q_3\ ,\ q_4\ ,\ q_4\ ,\ \dot{q}_1\ ,\ \dot{q}_2\ ,\ \dot{q}_{2m}\ ,\ \dot{q}_3\ ,\ 0\ ,\ 0]^-_{hs}(t = \alpha t_{\text{step}})$.

- Step time (t_{step})
- Percentage of time spent in single stance (α)
- Current in the motors in double stance ($I_2^{ds}(t)$, $I_3^{ds}(t)$, $I_4^{ds}(t)$). We assume piecewise linear currents. In double stance, we divide time into N^d intervals; $t_0^d, t_1^d, \ldots, t_{N^d}^d$. Here $t_{i+1}^d - t_i^d = 1/N^d$, $i = 0, 1, ..., N^d$, $t_0^d = \alpha t_{\text{step}}$ and $t_{N^d}^d = t_{\text{step}}$. This means we have $3(N^d + 1)$ unknowns for currents in double stance; $I_j^d(t = t_0), I_j^d(t = t_1), \ldots, I_j^d(t = t_N^d)$, where $j = 2, 3, 4$.
- Current in the motors in single stance ($I_2^{ss}(t)$, $I_3^{ss}(t)$). We assume piecewise linear currents. In single stance, we divide time into N^s intervals; $t_0^s, t_1^s, \ldots, t_{N^s}^s$. Here $t_{i+1}^s - t_i^s = 1/N^s$, $i = 0, 1, ..., N^s$, $t_0^s = 0$ and $t_{N^s}^s = \alpha t_{\text{step}}$. This means we have $2(N^s+1)$ unknowns for currents in single stance; $I_j^s(t = t_0^s), I_j^s(t = t_1^s), \ldots, I_j^s(t = t_{N^s}^s)$, where $j = 2, 3$.

Constraints. The optimization constraints are as follows:

- Periodicity constraints; $\chi^i_{ss}(t = 0) = \chi^f_{ds}(t = t_{\text{step}})$ (see figure E.1b).
- Periodicity constraints; $\chi^-_{hs}(t = \alpha t_{\text{step}}) = \chi^f_{ss}(t = \alpha t_{\text{step}})$ (see figure E.1b).
- Transition from single stance to double stance takes place when the swinging leg's foot hits the ground at time $t = \alpha t_{\text{step}}$. This constraint is given by $y_h = y_h$ in equation A.2.
- Transition from double stance to single stance takes place when the vertical reaction force on trailing foot goes to zero at time $t = t_{\text{step}}$.

- The current at the motors is continuous in the transition from double stance to single stance; ($I_j^d(t = t_{\text{step}}) = I_j^s(t = 0)$ where $j = 2, 3$).
- Step velocity is specified, $v_{\text{step}} = V$.
- Step length is specified, $d_{\text{step}} = D$.
- To prevent scuffing at the beginning of single stance phase we impose that the absolute velocity of the hip at the beginning of single stance should not be in the walking direction.
- Vertical ground reaction force for the foot on the ground in single stance should be positive and this is enforced at the grid points. This gives $N^s + 1$ inequality constraints.
- Vertical ground reaction force on both the feet in double stance should be positive, except on the trailing leg at the double stance to single stance transition when it should be zero. These constraints are enforced at the grid points. This gives us $2N^d + 1$ inequality constraints.
- Currents in the motors have to be within the actuator limits ($-8\text{A} \leq I_j^i \leq 8\text{A}$, $i = \text{ss}, \text{ds}$; for ds $j = 2, 3, 4$ and for ss $j = 2, 3$). These are enforced at grid points and correspond to $3(N^d + 1) + 2(N^s + 1)$ inequality constraints.

Methods. With the initial conditions in single stance $\chi_{ss}^i(t = 0)$, we integrate equations of motion for single stance till we reach the end of single stance at time $t = \alpha t_{\text{step}}$. The torques for the single stance equations are obtained using the assumed piecewise linear currents in single stance. Similarly, we take the initial conditions just before heel-strike $\chi_{hs}^-(t = \alpha t_{\text{step}})$ and first apply the heel-strike condition to get the initial condition at the beginning of double stance. With the initial conditions in double stance, we integrate equations of motion for double stance till we reach the end of double stance

at time $t = t_{\text{step}}$. The torques for use in the equations of motion are obtained using the assumed piecewise linear currents in double stance. The optimization software varies the various optimization parameters listed above, in an attempt to minimize the cost E.7, while respecting the various constraints described earlier. We used sequential quadratic programming package SNOPT [43], to solve the optimization problem. The results of this optimization are shown and described in chapter 5.

APPENDIX F

TRAJECTORY OPTIMIZATION: COARSE-GRID

To simplify the control implemented by the main brain (ARM 9), the motor current trajectories are defined with far few parameters than are used in the fine-grid optimization. Rather than try to approximate the fine-grid trajectories with the coarse-grid trajectories, a new optimization is done with the coarse-grid description. This is described in chapter 5, section 5.3 with some more details provided here.

F.1 Problem formulation.

We simplify the problem by separating the foot flip-up and flip-down (during swing) optimization from the optimization of other costs. This is a reasonable approximation because the feet have such low mass that their motions during swing are negligibly coupled to the other motion of the robot.

We carry out two sequential optimizations as follows.

1. We approximate the current profile needed to do the foot-flip optimization. This optimization gives us the energy needed to do the foot-flip using the approximate control representation ($E^{const}_{coarse\text{-}grid\text{-}foot\text{-}flip}$).
2. Using the energy for flip-foot optimization ($E^{const}_{coarse\text{-}grid\text{-}foot\text{-}flip}$) and using an approximation to the converged current profile for the walk optimization presented in E.2 we minimize the total cost of transport (TCOT).

We elaborate on the details of these optimization in the next two sections.

F.2 Foot-flip energy optimization

Cost, constraints and methods. These are the same as those presented in E.1. The result of this optimization will give the energy needed for doing the foot-flip using the approximate representation $\mathrm{E}^{\text{const}}_{\text{coarse-grid-foot-flip}}$.

Parameters. We want the foot on the swinging leg to get to a prescribed position for foot clearance. Hence we chose to do a proportional derivative control on motor position for the foot-flip. Thus, we have three parameters; proportional gain, derivative gain, and reference angle.

F.3 Total Cost of Transport minimization

Cost. We are interested in minimizing the total cost of transport (COT) and defined as,

$$\begin{aligned}
\text{TCOT} &= \frac{\text{Energy used per step}}{\text{Weight} \times \text{Step length}} \\
&= \frac{\mathrm{P}_{\text{fixed}}\mathrm{t}_{\text{step}} + \mathrm{E}^{\text{const}}_{\text{coarse-grid-foot-flip}} + \mathrm{E}_{\text{walk}}}{M_{\text{tot}}\, g\, d_{\text{step}}} \\
&= \frac{\mathrm{P}_{\text{fixed}}\mathrm{t}_{\text{step}} + \mathrm{E}^{\text{const}}_{\text{coarse-grid-foot-flip}} + \int_{t=0}^{t=t_{\text{step}}} \sum [\mathrm{P}_{\text{walk}}]^{+}\, \mathrm{dt}}{M_{\text{tot}}\, g\, d_{\text{step}}}
\end{aligned}$$

Here $\mathrm{P}_{\text{fixed}}$ is the power need to run the sensors, microprocessors, and motor controllers and is fixed at 5.15 W. $\mathrm{E}^{\text{const}}_{\text{coarse-grid-foot-flip}}$ is the energy needed to flip the foot on the swing leg to do ground clearance, is a constant per step, and is obtained from foot-flip optimization presented in F.2. E_{walk} is the energy needed to take a step. This is calculated by taking time integral of the sum of power of motors involved in taking a step.

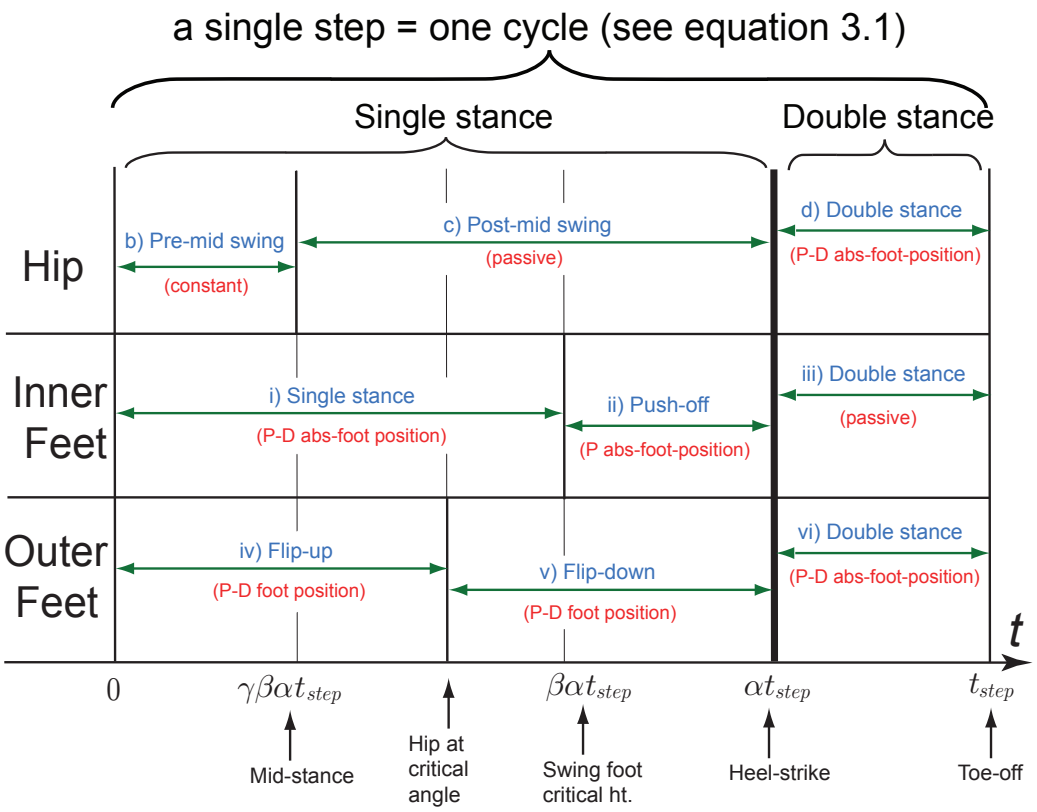

Figure F.1: **Finite state machine used to do the coarse-grid optimization.** We show the state machine for only one step. A step is divided into single stance and double stance using the parameter α ($0 < \alpha < 1$). α denotes the fraction of step time spent in single stance. Further, the single stance phase is divided into three phases using the parameters β and γ ($0 < \beta, \gamma < 1$). Each state is given a name, e.g., pre-mid swing, push-off. The control action associated with the state is shown in parenthesis. State transitions conditions are written on the time axis. Note that the foot-flip is decoupled and the necessary optimization is done separately.

Parameters. The optimization parameters are as follows:

- State at beginning of single stance ($\chi^i_{ss}(t = 0)$) (see figure E.1a).

  $\chi^i_{ss}(t = 0) = [q_1 , q_2 , q_{2m} , q_3, \dot{q}_1 , \dot{q}_2 , \dot{q}_{2m} , \dot{q}_3]_{ss}(t = 0)$

- State at the instant before heel-strike ($\chi^-_{hs}(t = \alpha t_{\text{step}})$) (see figure E.1a).

  $\chi^-_{hs}(t = \alpha t_{\text{step}}) = [q_1 , q_2 , q_{2m} , q_3 , q_4 , q_4 , \dot{q}_1 , \dot{q}_2 , \dot{q}_{2m} , \dot{q}_3 , 0 , 0]^-_{hs}(t = \alpha t_{\text{step}})$

- Step time (t_{step})
- Fraction of time spent in different phases; α, β, γ; $0 < \alpha, \beta, \gamma < 1$ (see figure F.1).
- We divide the walking cycle into a finite state machine as shown in figure F.1; one for the hip and one for each foot. Each state machine is divided into a state and given a name (e.g. pre-mid swing, push-off). Each state has a suitable current parameterization and is shown in the parenthesis. Transitions for states are shown along the time axis.

  Hip state machine (top panel): is divided into three states; a pre-mid swing state, a post-mid state, and a double stance state. In the pre-mid swing state we apply a constant current, while the post-mid swing and double stance states are both passive. We start with pre-mid swing state. When the robot's stance leg is vertical (in the single stance phase), the pre-mid swing to post-mid swing transition occurs. On heel-strike, we get the post-mid swing to double stance state transition. Finally, the double stance to pre-mid swing transition leading to the next step occurs when the vertical reaction force on the trailing leg goes to zero (also called the toe-off).

  Foot state machine (middle panel): is the state machine for the foot which stays on the ground for the full step. This foot has three states; single stance, push-off, and double stance. For single stance we do a Proportional-Derivative (P-D) control on the absolute foot angle. For the push-off we do a Proportional (P) control on the absolute foot angle. The double stance state is passive. We start with the state single stance. When the swinging leg's ankle reaches a critical height the single stance to push-off transition occurs. On heel-strike, the push-off to double stance transition occurs. Finally, when the vertical reaction force on the trailing leg goes to zero i.e. on toe-off, we get the double stance to flip-up transition leading to the next step.

Foot state machine (bottom panel): is the state machine for the foot that is in the air during single stance and is on the ground during double stance. This foot has three states; flip-up, flip-down, and double stance. For double stance we do a P-D control on absolute foot angle. In the flip-up and flip-down we do a P-D control on the foot angle. Since the foot is massless, we decoupled the motion of the swinging leg's foot from the rest of the robot simulation. We did a separate optimization for the flip-up and flip-down states and is presented in section F.2. We start with flip up position for this foot at time t=0. We transition from flip-up to flip-down state when the hip reaches a certain critical angle. On heel-strike, we transition from flip down to double stance state. The double stance to single stance transition occurs when the vertical reaction force on trailing legs goes to zero (also called toe-off) leading to the next step.

Constraints. The constraints for this optimization are same as those presented for the optimal control problem presented in section E.2 but with a few exceptions. We do not have the step length and step velocity constraint. Instead, we bound the hip angle at heel-strike to have a maximum value of 0.4 rad (to prevent robot wear and tear associated with big step lengths). We do not have a grid to enforce the vertical ground reaction force and current constraints. So, we create a grid of size N^{s1} for time interval $0 \leq t \leq \gamma\beta\alpha t_{\text{step}}$, N^{s2} for time interval $\gamma\beta\alpha t_{\text{step}} \leq t \leq \beta\alpha t_{\text{step}}$, and N^{s3} for time interval $\beta\alpha t_{\text{step}} \leq t \leq \alpha t_{\text{step}}$, all these are in single stance phase. We also create a grid of size N^{d} for the time interval $\alpha t_{\text{step}} \leq t \leq t_{\text{step}}$ in double stance phase. Now we enforce the reaction force and current constraints on this grid in a similar way as that presented in section E.2.

Methods. Same as those presented in section E.2.

APPENDIX G

FINITE STATE MACHINE FOR HIGH LEVEL WALK CONTROL

We use a concurrent, hierarchical finite state machine to code our combined coarse-grid and reflex based discrete controller on the robot. These are shown below.

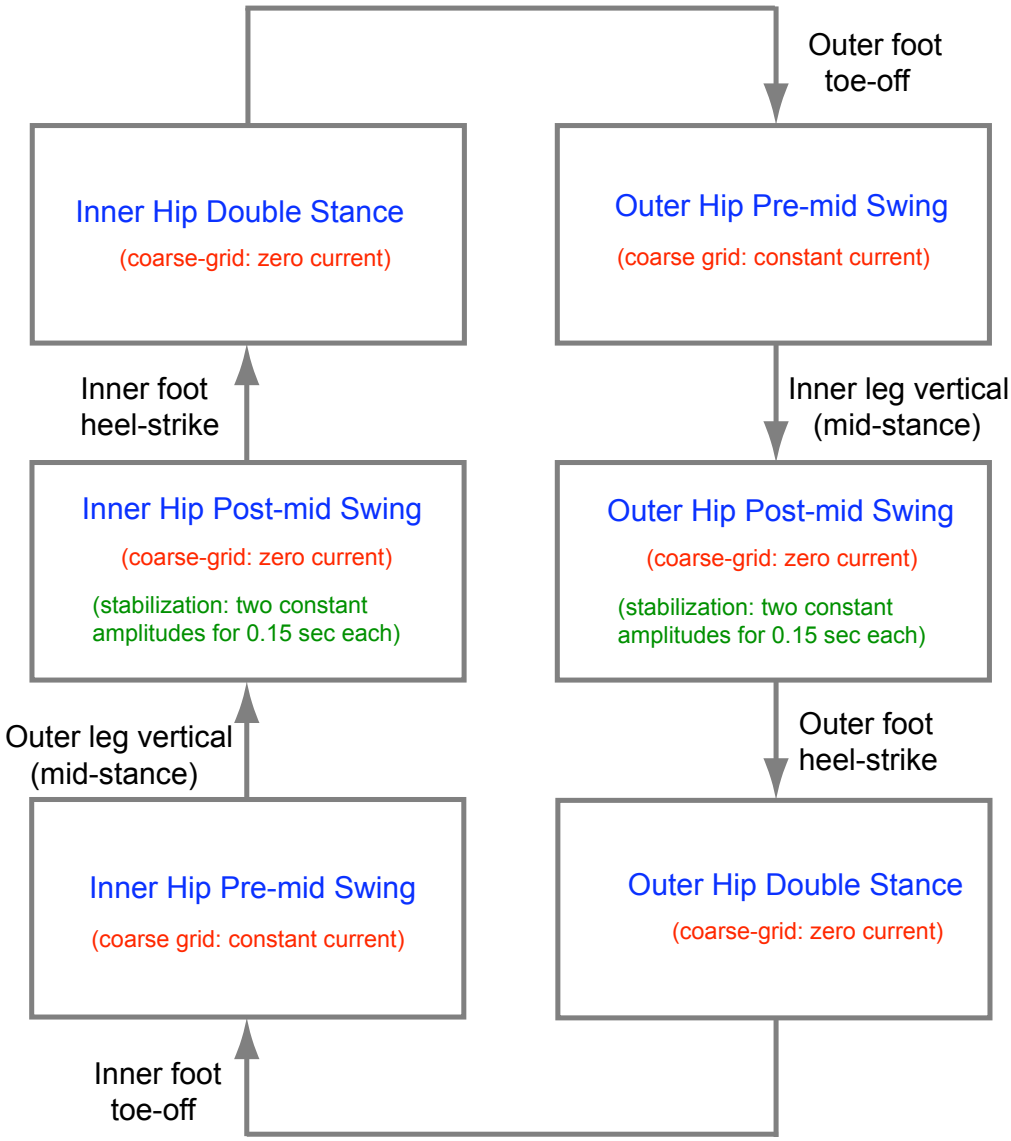

Figure G.1: **Hip finite state machine.**

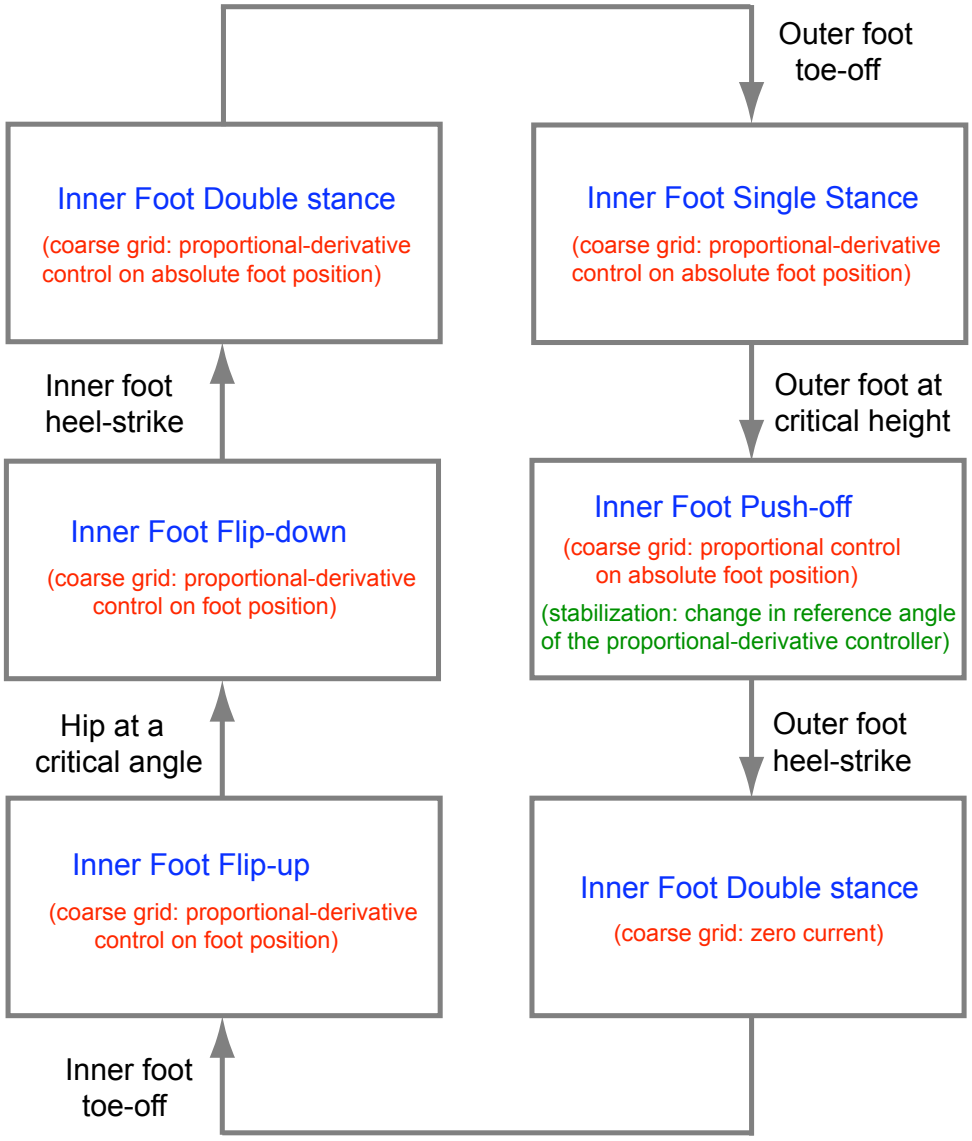

Figure G.2: **Inner foot finite state machine.** The outer foot finite state machine is similar.

BIBLIOGRAPHY

[1] P.J. Antsaklis and A.N. Michel. *Linear systems*. Birkhauser, 2006.

[2] C.G. Atkeson and J. Morimoto. Nonparametric representation of policies and value functions: A trajectory-based approach. *Advances in neural information processing systems*, pages 1643–1650, 2003.

[3] E. Atzler and R. Herbst. Arbeitsphysiologische studien. *Pflügers Archive European Journal of Physiology*, 215(1):291–328, 1927.

[4] C. Azevedo, P. Poignet, and B. Espiau. Moving horizon control for biped robots without reference trajectory. In *Proc. of the International Conference on Robotics and Automation, Washington DC, USA*, volume 3, pages 2762–2767, 2002.

[5] D.P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 1995.

[6] G. Bessonnet, S. Chessé, and P. Sardain. Optimal gait synthesis of a seven-link planar biped. *The International journal of robotics research*, 23(10-11):1059–1073, 2004.

[7] G. Bessonnet, P. Seguin, and P. Sardain. A parametric optimization approach to walking pattern synthesis. *The International Journal of Robotics Research*, 24(7):523, 2005.

[8] P. A. Bhounsule, J. Cortell, B. Hendriksen, J.G.D. Karssen, C. Paul, and A. Ruina. A robot that walked 65 km on a single charge: energy-effective and reliable locomotion using trajectory optimization and stabilization from reflexes. *Submitted to the International Journal of Robotics Research*, 2012.

[9] Bio-Robotics and Locomotion Laboratory. Cornell ranger 2006 [available online]. http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/ranger2006.php, April 2012.

[10] Bio-Robotics and Locomotion Laboratory. Cornell ranger 2008 [available online]. http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/ranger2008.php, April 2012.

[11] Bio-Robotics and Locomotion Laboratory. Cornell ranger 2010 [available online]. http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/Ranger2010/, April 2012.

[12] Biorobotics and Locomotion Laboratory. Cornell ranger 2011, 4-legged bipedal robot. [available online]. http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/Ranger2011/, April 2012.

[13] A.C. Bobbert. Energy expenditure in level and grade walking. *Journal of Applied Physiology*, 15(6):1015–1021, 1960.

[14] G. Cabodevila and G. Abba. Quasi optimal gait for a biped robot using genetic algorithm. In *Proc. of the IEEE International Conference on Systems, Man and Cybernetics, Computational Cybernetics and Simulation, Orlando, Florida, USA*, volume 4, pages 3960–3965, 1997.

[15] J. Camp. Powered 'passive' dynamic walking. Master's thesis, Cornell University, 1997.

[16] T.H. Chang and Y. Hurmuzlu. Sliding control without reaching phase and its application to bipedal locomotion. *ASME Journal of Dynamic Systems, Measurement, and Control.*, 105:447–455, 1994.

[17] P.H. Channon, S.H. Hopkins, and D.T. Pham. Simulation and optimization of gait for a bipedal robot. *Mathematical and Computer Modelling*, 14:463–467, 1990.

[18] P.H. Channon, S.H. Hopkins, and D.T. Pham. A variational approach to the optimization of gait for a bipedal robot. *Proceedings of the Institution of Mechanical Engineers*, 210(23):177–186, 1996.

[19] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, and J. W. Grizzle. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, 2003.

[20] C. Chevallereau, Y. Aoustin, and A. Formal'sky. Optimal walking trajectories for a biped. In *Proc. of IEEE Robot Motion and Control, Kiekrz, Poland*, pages 171–176, 1999.

[21] C. Chevallereau, J.W. Grizzle, and C.L. Shih. Asymptotically stable walking of a five-link underactuated 3-d bipedal robot. *IEEE Transactions on Robotics*, 25(1):37–50, 2009.

[22] M.J. Coleman. *A stability study of a three-dimensional passive-dynamic model of human gait*. PhD thesis, Cornell University, 1998.

[23] M.J. Coleman. Numerical accuracy case studies of two systems with intermittent dynamics a 2d rimless spoked wheel and a 3d passive dynamic model of walking. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Application and Algorithms*, 16:59–87, 2009.

[24] M.J. Coleman, M. Garcia, K. Mombaur, and A. Ruina. Prediction of stable walking for a toy that cannot stand. *Physical Review E*, 64(2):022901, 2001.

[25] M.J. Coleman and A. Ruina. An uncontrolled walking toy that cannot stand still. *Physical Review Letters*, 80(16):3658–3661, 1998.

[26] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, 2005.

[27] S.H. Collins and A. Ruina. A bipedal walking robot with efficient and human-like gait. In *Proceeding of 2005 International Conference on Robotics and Automation, Barcelona, Spain*, 2005.

[28] S.H. Collins, M. Wisse, and A. Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607, 2001.

[29] M.R. Dimitrijevic, Y. Gerasimenko, and M.M. Pinter. Evidence for a spinal central pattern generator in humans. *Annals of the New York Academy of Sciences*, 860(1):360–376, 1998.

[30] C. Duan and R. Singh. Dynamics of a 3dof torsional system with a dry friction controlled path. *Journal of sound and vibration*, 289(4-5):657–688, 2006.

[31] P.E. Dupont. The effect of friction on the forward dynamics problem. *The International Journal of Robotics Research*, 12(2):1442–1447, 1993.

[32] Boston Dynamics. Petman - bigdog gets a big brother. [available online]. `http://www.bostondynamics.com/robot_petman.html`, April 2012.

[33] H. Elftman. Biomechanics of muscle with particular application to studies of gait. *Journal of bone and joint surgery (american volume)*, 48(2):363–377, 1966.

[34] J. Englsberger, C. Ott, M.A. Roa, A. Albu-Schaffer, and G. Hirzinger. Bipedal walking control based on capture point dynamics. In *International Conference on Intelligent Robots and Systems (IROS), 2011*, pages 4420–4427. IEEE, 2011.

[35] L.L. Flynn, R. Jafari, and R. Mukherjee. Active synthetic-wheel biped with torso. *IEEE Transactions on Robotics*, 26(5):816–826, 2010.

[36] A Formalsky. Motion of anthropomorphic mechanism under impulsive control. *Proceedings of Institute of Mechanics, Lomonosov Moscow State University*, pages 17–34, 1978.

[37] J. Furusho and M. Masubuchi. Control of a dynamical biped locomotion system for steady walking. *Journal of Dynamic Systems, Measurement, and Control*, 108:111–118, 1986.

[38] J. Furusho and A. Sano. Sensor-based control of a nine-link biped. *The International Journal of Robotics Research*, 9(2):83–98, 1990.

[39] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: Theory and practice–a survey. *Automatica*, 25(3):335–348, 1989.

[40] M. Garcia, A. Chatterjee, and A. Ruina. Efficiency, speed, and scaling of two-dimensional passive-dynamic walking. *Dynamics and Stability of Systems*, 15(2):75–99, 2000.

[41] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME J. of Biomech. Eng.*, 120:281–288, 1998.

[42] M.S. Garcia. *Stability, scaling, and chaos in passive-dynamic gait models*. PhD thesis, Cornell University, 1999.

[43] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[44] A. Goswami, B. Thuilot, and B. Espiau. A study of the passive gait of a compass-like biped robot. *The International Journal of Robotics Research*, 17(12):1282, 1998.

[45] J. W. Grizzle. Remarks on event-based stabilization of periodic orbits in systems with impulse effects. In *International Symposium on Communications, Control and Signal Processing, Marrakech, Morocco*, 2006.

[46] J.W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 46(1):51–64, 2001.

[47] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *International Conference on Robotics and Automation, Leuven, Belgium*, pages 1321–1326, 1998.

[48] D.G.E. Hobbelen and M. Wisse. A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *IEEE Transactions on robotics*, 23(6):1213–1224, 2007.

[49] D.G.E. Hobbelen and M. Wisse. Limit cycle walking. *Humanoid Robots, Human-like Machines, book chapter 14*, 2007.

[50] E. Holm. Contribution to the Theory of the Contact between a Carbon Brush and a Copper Collector Ring. *Journal of Applied Physics*, 28:1171–1176, 1957.

[51] Y. Ikemata, A. Sano, and H. Fujimoto. A physical principle of gait generation and its stabilization derived from mechanism of fixed point. In *International Conference on Robotics and Automation, Orlando, Florida, USA*, pages 836–841, 2006.

[52] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *International Conference on Robotics and Automation, Taipei, Taiwan*, pages 1620–1626, 2003.

[53] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. A realtime pattern generator for biped walking. In *Proceedings of the International Conference on Robotics and Automation, Washington DC, USA*, pages 31–37, 2002.

[54] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *International Conference on Intelligent Robots and Systems, Hawaii, USA*, pages 239–246, 2001.

[55] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain-theory and basic experiment. In *Proc. of the IEEE International Conference on Robotics and Automation, Sacramento, California, USA*, pages 741–746, 1991.

[56] S. Kajita and K. Tani. Experimental study of biped dynamic walking. *IEEE Control Systems Magazine*, 16(1):13–19, 1996.

[57] S. Kajita, T. Yamaura, and A. Kobayashi. Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Transactions on Robotics and Automation*, 8(4):431–438, 1992.

[58] J.G.D. Karssen. Design and construction of the cornell ranger, a world record distance walking robot., 2007.

[59] T. Kato, A. Takanishi, H. Jishikawa, and I. Kato. The realization of the quasi-dynamic walking by the biped walking machine. In *Fourth Symposium on The-*

ory and Practice of Walking Robots, pages 341–351. Warsaw: Polish Scientific Publishers, 1983.

[60] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J.E. Pratt. Capturability-based analysis and control of legged locomotion. part 1: Theory and application to three simple gait models. *To appear in International Journal of Robotics Research*, 2012.

[61] A.D. Kuo. A simple model of bipedal walking predicts the preferred speed–step length relationship. *Journal of biomechanical engineering*, 123:264–269, 2001.

[62] A.D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124:113–120, 2002.

[63] M.L. Latash, J.P. Scholz, and G. Schöner. Motor control strategies revealed in the structure of motor variability. *Exercise and sport sciences reviews*, 30(1):26–31, 2002.

[64] H.O. Lim and A. Takanishi. Waseda biped humanoid robots realizing human-like motion. In *International Workshop on Advanced Motion Control*, pages 525–530, 2000.

[65] I.R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[66] T. Mandersloot, M. Wisse, and C.G. Atkeson. Controlling velocity in bipedal walking: A dynamic programming approach. In *Humanoids 2006, Genova, Italy*, pages 124–130, 2006.

[67] P. Manoonpong, T. Geng, T. Kulvicius, B. Porr, and F. Wörgötter. Adaptive, fast

walking in a biped robot under neuronal control and learning. *Computational Biology*, 3(7):1305–1320, 2007.

[68] P. Manoonpong, T. Geng, B. Porr, and F. Wörgötter. The runbot architecture for adaptive, fast, dynamic walking. In *IEEE International Symposium on Circuits and Systems, New Orleans, USA*, pages 1181–1184, 2007.

[69] R. Margaria. Positive and negative work performances and their efficiencies in human locomotion. *European Journal of Applied Physiology and Occupational Physiology*, 25(4):339–351, 1968.

[70] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62, 1990.

[71] T. McGeer. Passive dynamic biped catalogue. In *Proc. of 2nd International Symposium on Experimental Robotics*, pages 465–490, 1991.

[72] K. Mitobe, N. Mori, K. Aida, and Y. Nasu. Nonlinear feedback control of a biped walking robot. In *International Conference on Robotics and Automation, Nagoya, Aichi, Japan*, volume 3, pages 2865–2870, 1995.

[73] H. Miura and I. Shimoyama. Dynamic walk of a biped. *The International Journal of Robotics Research*, 3(2):60, 1984.

[74] K. Mombaur, H. Georg Bock, J. Schlöder, and R. Longman. Stable walking and running robots without feedback. *Climbing and Walking Robots*, pages 725–735, 2005.

[75] K.D. Mombaur, H.G. Bock, J.P. Schloder, and R.W. Longman. Human-like actuated walking that is asymptotically stable without feedback. In *International Conference on Robotics and Automation, Seoul, Korea*, pages 4128–4133, 2001.

[76] K.D. Mombaur, R.W. Longman, H.G. Bock, and J.P. Schlöder. Open-loop stable running. *Robotica*, 23(01):21–33, 2005.

[77] K. Ogata. *Discrete-Time Control Systems*. Prentice Hall, 1995.

[78] J.D. Ortega and C.T. Farley. Minimizing center of mass vertical movement increases metabolic cost in walking. *Journal of Applied Physiology*, 99(6):2099–2107, 2005.

[79] D. Owaki, M. Koyama, S. Yamaguchi, S. Kubo, and A. Ishiguro. A two-dimensional passive dynamic running biped with knees. In *International Conference on Robotics and Automation, Anchorage, Alaska, USA*, pages 5237–5242, 2010.

[80] J.H. Park and M. Choi. Generation of an optimal gait trajectory for biped robots using a genetic algorithm. *JSME International Journal*, 47(2):715–721, 2004.

[81] J.H. Park and K.D. Kim. Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *International Conference on Robotics and Automation, Leuven, Belgium*, pages 3528–3533, 1998.

[82] C. Paul. Sensorimotor control of biped locomotion. *Adaptive Behavior*, 13(1):67–80, 2005.

[83] J.E. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *To appear in International Journal of Robotics Research*, 2012.

[84] J.E. Pratt and R. Tedrake. Velocity-based stability margins for fast bipedal walking. *Fast Motions in Biomechanics and Robotics*, pages 299–324, 2006.

[85] V. Radhakrishnan. Locomotion: dealing with friction. *Proceedings of the National Academy of Sciences*, 95(10):5448–5455, 1998.

[86] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, et al. Bigdog, the rough-terrain quadruped robot. *Proceedings of the 17th International Federation of Automatic Control*, 2008.

[87] M. Raibert, S. Tzafestas, and C. Tzafestas. Comparative simulation study of three control techniques applied to a biped robot. In *International Conference on Systems, Man and Cybernetics, Le Touquet, France*, pages 494–502, 1993.

[88] M.H. Raibert et al. *Legged robots that balance*. MIT press Cambridge, MA, 1986.

[89] L. Righetti and A.J. Ijspeert. Programmable central pattern generators: an application to biped locomotion control. In *Proc. of IEEE International Conference on Robotics and Automation, Orlando, Florida, USA*, pages 1585–1590, 2006.

[90] L. Roussel and A. Goswami. Generation of energy optimal complete gait cycles for biped robots. In *Proceedings of IEEE International Conference on Robotics and Automation, Leuven, Belgium*, 1998.

[91] A. Ruina. Nonholonomic stability aspects of piecewise holonomic systems. *Reports on mathematical physics*, 42(1-2):91–100, 1998.

[92] A. Ruina, J.E.A. 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*, 237(2):170–192, 2005.

[93] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent asimo: System overview and integration. In *Proc. of*

International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, volume 3, pages 2478–2483, 2002.

[94] A. Seyfarth, H. Geyer, and H. Herr. Swing-leg retraction: a simple control model for stable running. *Journal of Experimental Biology*, 206(15):2547–2555, 2003.

[95] J.H. Solomon, M. Wisse, and M.J.Z. Hartmann. Fully interconnected, linear control for limit cycle walking. *Adaptive Behavior*, 18(6):492–506, 2010.

[96] M. Srinivasan. *Why walk and run: energetic costs and energetic optimality in simple mechanics-based models of a bipedal animal*. PhD thesis, Cornell University, 2006.

[97] M. Srinivasan. Fifteen observations on the structure of energy-minimizing gaits in many simple biped models. *Journal of The Royal Society Interface*, 8(54):74–99, 2011.

[98] M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75, 2005.

[99] S.H. Strogatz. *Nonlinear dynamics and chaos*. Addison-Wesley Reading, 1994.

[100] G. Taga. A model of the neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics*, 73(2):97–111, 1995.

[101] G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65(3):147–159, 1991.

[102] K. Taji and M. Miyamoto. A globally convergent smoothing newton method for nonsmooth equations and its application to complementarity problems. *Computational Optimization and Applications*, 22:81–101, 2002.

[103] R. Tedrake, I.R. Manchester, M. Tobenkin, and J.W. Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038, 2010.

[104] E. Todorov and M.I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.

[105] M. Vukobratovic and B. Borovac. Zero-moment point-thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[106] M. Vukobratovic and D. Juricic. Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, 16(1):1–6, 1969.

[107] E.R. Westervelt. *Toward a coherent framework for the control of planar biped locomotion*. PhD thesis, University of Michigan, 2003.

[108] E.R. Westervelt and J.W. Grizzle. Design of asymptotically stable walking for a 5-link planar biped walker via optimization. In *Proceeding of 2005 International Conference on Robotics and Automation, Washington DC, USA*, 2002.

[109] E.C. Whitman and C.G. Atkeson. Control of a walking biped using a combination of simple policies. In *Humanoid 2009. Paris, France*, pages 520–527, 2009.

[110] M. Wisse, C.G. Atkeson, and D.K. Kloimwieder. Swing leg retraction helps biped walking stability. In *Proceedings of 2005 IEEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan*, 2005.