Legged Locomotion over Irregular Terrain using the Control Basis Approach

A Master Project Presented

by

Willard Schroeder MacDonald

Submitted to the Department of Computer Science of the University of Massachusetts Amherst in partial fulfillment of the requirements for the degree of

Master of Science

May 1996

Department of Computer Science

Legged Locomotion over Irregular Terrain using the Control Basis Approach

A Master Project Presented

by

Willard Schroeder MacDonald

Approved as to style and content by:

____

Roderic A. Grupen, Member

____

Robin J. Popplestone, Member

Acknowledgements

Manfred Huber designed all of the control composition for walking on flat terrain that is described in chapter 3. Thank you to Professor Rod Grupen and the other members of the Laboratory for Perceptual Robotics who were always very supportive and willing to take time to help me. Thank you to Mom, Dad, Nate, and Bob for being great family. Thank you to Pierre Tehrany and Bill Rohan for good times...And thanks be to HippoHumper.

Abstract

Legged Locomotion over Irregular Terrain using the Control Basis Approach

May 1996

Willard Schroeder MacDonald, BS EE, University of Massachusetts at Amherst

M.S., University of Massachusetts Amherst

Directed by: Professor Roderic A. Grupen

Robust legged locomotion over natural irregular terrain has yet to be achieved by an autonomous robot. This thesis will present the *control basis* approach as a viable solution to generating walking gaits on-line for irregular terrain.

A control basis is a set of reusable feedback control laws that are task and platform independent. Different walking tasks are achieved by varying the composition functions over the same basis controllers, rather than by geometric planning of leg placements or the design of new task-specific behaviors.

Hardware modifications to *Thing*, a four legged walking robot, are described which allow it to map local terrain features. A control basis and a control composition for walking on irregular terrain is proposed and tested on unknown piece-wise planar terrain.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS . . . iv

ABSTRACT . . . v

LIST OF FIGURES . . . viii

Chapter

1. INTRODUCTION . . . 1
    - 1.1 Terminology . . . 1
        - 1.1.1 Terminology for Periodic Gaits . . . 4
    - 1.2 Gaits . . . 5
        - 1.2.1 Wave Gaits . . . 5
        - 1.2.2 Discontinuous Gaits . . . 6
    - 1.3 Geometric Techniques for Gait Generation . . . 8
        - 1.3.1 Geometric Techniques for Irregular Terrain . . . 8
        - 1.3.2 Representing Terrain . . . 9
        - 1.3.3 Geometric Gaits with Reactive Components . . . 9
        - 1.3.4 Results and Contributions of the Geometric Paradigm . . . 11
    - 1.4 Dynamics . . . 12
    - 1.5 Behavior-Based Control . . . 12
    - 1.6 The Control Basis Approach . . . 15
2. EXPERIMENTAL HARDWARE . . . 16
    - 2.1 Thing . . . 16
    - 2.2 Infrared Proximity Sensors . . . 17
    - 2.3 Force Sensors . . . 18
        - 2.3.1 Hardware . . . 19
        - 2.3.2 Interpreting Control Signals . . . 20

2.3.3 Modeling Spring Torques 22
2.4 Maximizing Workspace 24
3. A Control Basis For Multilegged Walking 26
3.1 The Control Basis Approach 26
3.2 A Control Basis for Walking Tasks 28
3.2.1 Control Gaits for Quadruped Walking 29
3.3 Results 34
4. Terrain Mapping 38
4.1 Disturbance Suppression 38
4.2 Mapping Facets with Contact Normals 40
4.3 Mapping Facets with Linear Regression 41
4.3.1 Results 44
5. Walking on Irregular Terrain 49
5.1 Control Composition for Irregular Terrain 49
5.1.1 Terrain Probing 50
5.1.2 Z-Posture Control 50
5.1.3 Foot Drag 51
5.1.4 Alternative State Transitions 55
5.2 Walking Experiment on Irregular Terrain 55
6. Conclusion 58
6.1 Future Work 59
Bibliography 60

List of Figures

Figure Page

1.1 Longitudinal Stability Margin for a Three Legged Stance. 3

1.2 Calculating the Energy Stability Margin. 4

1.3 Gait Diagram for a Six Legged Wave Gait with Duty Factor = 5/6. 6

1.4 Stability Margin Versus β for Discontinuous and Wave Gaits. . . . 7

1.5 Velocity Versus β for a Given Step Height and Stroke for Discontinuous and Wave Gaits. 7

1.6 Discrete Representation of Illegal Foot-Holds. 10

1.7 Simple Braitenberg Vehicles. 13

2.1 *Thing* a Quadruped Walking Robot. 16

2.2 Thing's Leg Design 17

2.3 Thing's Infrared Sensors and Their Range. 18

2.4 Sensing Forces on the Foot. 19

2.5 The Servo Control Loop used on Thing. 20

2.6 Signal Taps on Futaba Servos (bottom covers removed from servos). 20

2.7 Apparatus for Calibrating Servo Control Signals. 21

2.8 Torque and Angular Error Versus Control Signal Pulse-Width for Futaba S9601 and S9201 Servos. 22

2.9 Experimental Torsion Spring Characteristics. 23

2.10 Workspace of Joints 1 and 2 in Configuration-Space. 24

2.11 Cartesian Workspace. 25

3.1 Control Composition 27

3.2 Control Gait for Rotating in Place 30

3.3 Control Gait for Forward Walking 32

3.4 Stability, Forward Progress, and Forces During A Walking Experiment on Flat Terrain. 33

3.5 Gait Change Policy for Example Task 34

3.6 Walking Task in an Unknown Environment. Start and goal on the internal map (left) and actual robot (right) are indicated by initially shaded grid cells and crosses on the floor, respectively. The path of the robot is marked by black dots corresponding to the location of the center of mass. 36

4.1 Active Compliance at the Feet. 39

4.2 Sensing Surface Normals at the Foot. 41

4.3 Least-Square Approximation of Plane Minimizes $\sum_{i=0}^{n} e_i^2$. 42

4.4 Probing Pattern Looking Down $\hat{z}$ (top of leg shown). 44

4.5 Actual Flat Level Terrain (top) and the Resulting Mapped Terrain (bottom). 45

4.6 Actual Step with Flat Surfaces (top) and the Resulting Mapped Terrain (bottom). 45

4.7 Actual Ramp Terrain (top) and the Resulting Mapped Terrain (bottom). 46

4.8 Mapping Terrain Using a Finite History Least-Square Linear Regression. 46

4.9 Hough Transform on Sample Probe Data (ρ is plotted horizontally and ϕ is plotted vertically). 48

4.10 Determining the Spherical Parameters of a Facet's Normal. 48

5.1 Effect of Neglecting Body Posture in the Z-Direction 51

5.2 Body Posture During Flat and Irregular Terrain Walking. 53

5.3 (A) Joint Limits Preventing Stable Stance. (B) Dragging Leg 0 Allows Stable Stance. 54

5.4 Control Gait for Walking on Irregular Terrain. Controller instantiations that are new or modified from the flat terrain gait are highlighted. 56

5.5 A Demonstration of Walking over Irregular Terrain. Steps (white) are $1cm$ high. 57

Chapter 1

Introduction

The most general goal for research on walking robots is to develop an autonomous machine capable of traversing irregular terrain in a robust and timely manner. The literature has identified a number of issues important to walking robots, including stability, versatility of control, computational complexity of gait planning, efficiency, and sensor data interpretation. Proposed solutions fall into numerous categories, including leg design [18, 4], geometric planning [25, 30, 14, 22], behavior-based control [9, 29, 11], hybrid geometric/reactive control [18, 27, 28, 33], dynamic control [17, 32, 34, 29], and force-based control [1, 10, 35]. These are loose categories and are not mutually exclusive. The more recent approaches usually span a number of (traditionally) separate approaches.

This chapter will provide an introduction to the terminology used in the field of walking robots and will outline the most popular gaits and techniques for gait generation. Chapter 2 will describe the experimental platform and the design of sensors necessary for traversing irregular terrain. Chapter 3 will introduce the *control basis approach* to walking and will present a control composition for walking on flat terrain. Chapter 4 will discuss techniques for sensing and mapping terrain and will demonstrate one that represents local regions with planar facets. Chapter 5 will discuss changes to the control composition described in chapter 3 that allow the robot to successfully traverse irregular terrain and will describe an experiment that demonstrates the approach.

1.1 Terminology

The following is a list of common terminology used when talking about walking robots:

Gait:

A gait is a sequence of foot placements that collectively provide stability and locomotion to a walking robot.

Statically Stable:

A gait is statically stable if at any given time during the execution of the gait, the robot is stable. This means that if one were to stop the robot in mid stride, it would not topple over.

Periodic Gait:

In a periodic gait, a sequence of leg placements is established and then repeated at regular time intervals.

Continuous Gait:

The body of a robot using a continuous gait is moving at a constant speed with respect to the surface it is walking on. If it is using a *discontinuous* gait, the body may only be progressing forward during certain parts of the gait sequence.

Regular Gait:

In a regular gait, every leg has the same duty factor (*i.e.* every leg is on the ground for the same amount of time).

Symmetric Gait:

A gait is said to be symmetric if the motion of any right-left pair of legs is exactly half a cycle out of phase.

Crab Gait:

A crab gait is a general term for any gait whose direction of locomotion is different from, or equal to, the longitudinal axis of the vehicle's body. The angle between the direction of motion and the longitudinal axis of the body is called the *crab angle*.

On-line/Off-line:

A gait is said to be computed off-line if movements are determined before the robot is actually run. In this case, no sensor data is used to form the gait. A gait is said to be computed *on-line* if the robot uses sensor data to compute foot holds while it is running.

Dead-lock:

Deadlock is a condition in which no forward progress is possible due some conjunction of constraints yielding no feasible walking strategy.

Polygon of Support:

The convex hull of the supporting feet.

Longitudinal stability margin (S):

The minimum distance between the front or back boundary of the polygon of support and the projection of the center-of-mass onto the polygon. This is shown for a three legged stance in Figure 1.1.

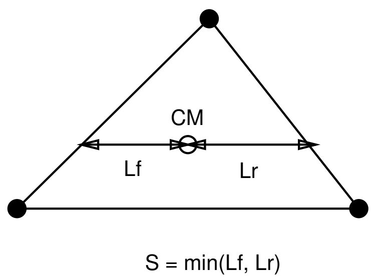

Figure 1.1. Longitudinal Stability Margin for a Three Legged Stance.

Energy Stability Margin (ESM):

The minimum energy required to tip over the walker [26]. The ESM is found by first calculating the potential energy to tip the center of mass over each pair of adjacent legs of the support polygon. This energy is mgh where m is the mass of the walker, g is gravity, and h is the height that must be reached to cause incipient tip-over as shown in Figure 1.2 [27].

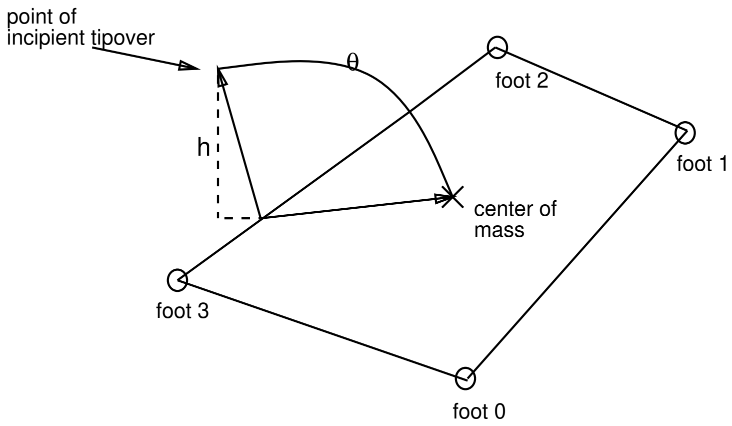

Figure 1.2. Calculating the Energy Stability Margin.

1.1.1 Terminology for Periodic Gaits

For smooth terrain, one can often formulate and prove theorems about the kinematics, stability, speed, or power consumption of a periodic gait, allowing for analytical comparisons between gaits. Formal definitions and measurable parameters have been established that create a framework for such analysis [30, 25]. Some of these parameters are defined as follows:

Duty factor (β_i):

β_i is the time fraction of a cycle in which leg i is in the support phase (supporting part of the robot's weight).

$$\beta_i = \frac{time\ of\ support\ phase\ of\ leg\ i}{cycle\ time\ of\ leg\ i}$$

Stride (λ):

λ is the distance the center-of-mass translates during one complete locomotion cycle.

Leg stroke (R):

R is the distance through which the foot is translated relative to the body during the support phase.

Stroke pitch (P):

P is the distance between the centers of strokes of the adjacent legs on one side.

Support Phase:

The support phase of a leg is the portion of a gait during which the leg is on the ground.

Transfer Phase:

The transfer phase of a leg is the portion of a gait during which the leg is in the air.

1.2 Gaits

Most of the literature has focused on generating statically-stable periodic gaits off-line, however there are many different forms that gaits come in and almost as many approaches to the generation of gaits. The following sections describe some of the most common gaits and then general techniques for generating gaits.

1.2.1 Wave Gaits

One of the most common kinds of gaits, in both robotics and in slow moving animals [30], is the *wave gait*. In this gait, the sequence of leg placements progresses with each successive leg being the next one spatially from back to front (forward wave gait) or from front to back (backward wave gait). They are called "wave gaits" because the sequential lifting of legs flows longitudinally along the length of the robot like a wave. Wave gaits are generally *continuous* and *periodic*.

The wave gait provides optimum stability among the class of regular symmetric gaits [31]. This is believed to be one of the reasons that animals use them. A wave gait is shown graphically in Figure 1.3 [30]. The dark line represents the time the leg is in the support phase. This gait is clearly a static one, since only one leg is up at any given time. As β is increased, the time that the leg is up decreases. So assuming the *transfer rate* (speed of leg swinging forward while up) of the leg is held constant, the steps will become shorter, and for β close to 1, the robot will be moving its legs in short quick steps, but not making any forward progress. On the other hand, if β is decreased, the transfer phase of the leg increases to the point where more than one leg is up at a given time. This gait would be faster than one of higher β, but would have a lower stability margin. (For an analytical discussion of stability with respect to β, λ, R and P for four and six legged robots see [30].)

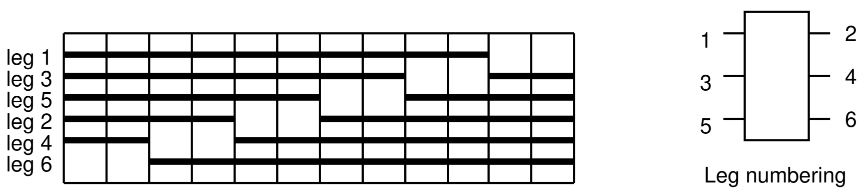

Figure 1.3. Gait Diagram for a Six Legged Wave Gait with Duty Factor = 5/6.

1.2.2 Discontinuous Gaits

To traverse irregular terrain in a more robust way, discontinuous gaits may be used. Discontinuous gaits only propel the body forward when all feet are stabily on the ground. On extreme terrain, discontinuous gaits are preferred over continuous gaits (such as wave gaits) because of their characteristically higher longitudinal stability margin.

Wave gaits can achieve higher speeds than discontinuous gaits, but only by using small duty factors, which means lower stability margins [14]. Figure 1.4

compares the stability margin versus β for discontinuous and wave gaits. Figure 1.5 compares the velocity versus β at a given step height and stroke for discontinuous and wave gaits.

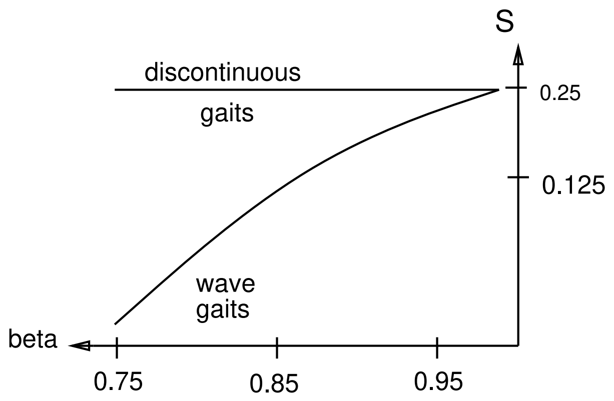

Figure 1.4. Stability Margin Versus β for Discontinuous and Wave Gaits.

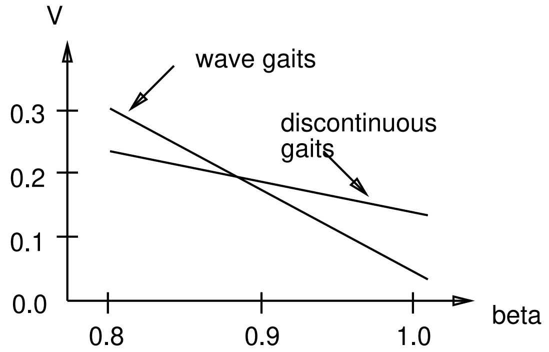

Figure 1.5. Velocity Versus β for a Given Step Height and Stroke for Discontinuous and Wave Gaits.

De Santos and Jimenz [14] use a robot model with a discretized configuration (*e.g.* a leg is forward, back, up, *etc.*) to facilitate the above analysis and for developing discontinuous crab and turning gaits. They argue for *periodic* discontinuous gaits so that gait segments can be pieced together for extended locomotion. These techniques for gait generation suffer from model uncertainties in the the same way as geometric gaits (see section 1.3). It should be noted, however, that not all discontinuous gaits are generated geometrically.

Often, one is interested in the *energy efficiency* of a gait, since an autonomous robot must carry its own energy supply. Discontinuous gaits have been argued to have relatively low power consumption, since they require only a few actuators to be moving at any given time (*e.g.* a leg is lowered while the body remains still). However, this saving must be compared with the energy lost in repeatedly accelerating and decelerating the body.

1.3 Geometric Techniques for Gait Generation

The term *geometric*, when referring to robot control and gait planning, is used to mean controlling a robot based on fixed relationships between models of the robot and models of the world in which the robot operates. The models characterize the instantaneous state of the world (*e.g.* texture, rigidity, position, *etc.*). Often the effects of uncertainty and imprecision are neglected, and models are assumed to be complete. When the models do not match the actual robot and world, the pre-determined relationships can break down.

Geometric techniques rely heavily on off-line calculations to set up the models. Commonly, off-line gaits are also periodic gaits, since these are easier to compute and analyze; however, periodic gaits clearly have a disadvantage in terms of flexibility, since in general the terrain being traversed is *not* periodic.

1.3.1 Geometric Techniques for Irregular Terrain

Walking over non-planar terrain poses complications for geometric gaits. Discontinuous and continuous *follow-the-leader* gaits are proposed by Song and Waldron [30] for navigating terrain sparsely scattered with forbidden regions (regions where foot placement is not allowed or not possible). The basic nature of the follow-the-leader gait is that each successive foot on a side is placed in the foot print of the one before it. This technique simplifies gait planning since only the front two feet need to be placed. The consequences of the initial foot-holds chosen, however, put great constraints on where they can be placed. Choosing these foot-holds is difficult, and Song and Waldron propose human direction.

A set of gaits can be defined where each is designed for a particular situation or type of terrain. Which gait to use depends on terrain type, desired speed, stability, *etc.*. Song and Waldron [30] propose specific sequences of motions for specific types of obstacles encountered such as ditches, walls, and slopes.

Modifications to existing gaits can be employed as well. For example, Adachi *et al* [1] adjust the leg transfer timing of a wave gait based on the position of the robot's center-of-mass and the terrain inclination.

1.3.2 Representing Terrain

The way in which terrain is represented internally can influence the performance of the robot. The most common technique is to discretize the terrain. Illegal foot-holds can be represented as on/off cells in a discrete two dimensional grid, as shown in Figure 1.6 [25]. A foot-hold is illegal if there is an obstacle, if it is not valid kinematically, or if placement in that spot would result in a loss of stability. The selection of foot placements requires an exhaustive search over every possible foothold.

Jeong *et al* [22] derive the necessary conditions for a walking robot to maintain its stepping sequence in the presence of forbidden regions in the terrain. Once the allowable modifications to a pre-planned gait have been determined, a method for making the required modifications must be developed.

1.3.3 Geometric Gaits with Reactive Components

Another way to deal with uncertainties in sensing and predicting irregular terrain, is to augment the underlying geometric control with *reactive behaviors.*

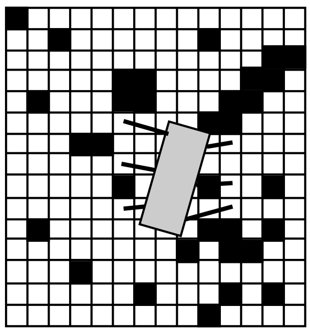

Figure 1.6. Discrete Representation of Illegal Foot-Holds.

Hirose [18] proposed a “reflex-motion regulation system” that attempts to make disturbances caused by irregularities in the terrain transparent to the fixed-gait controller. The reflexive controller monitors and adjusts body height, body posture, weight distribution, and stepping behavior (by lowering a leg until ground is reached in the put-down phase and stepping higher when an obstacle is sensed in the swing phase).

In a similar manner Nagy *et al* [27] use a hierarchical controller that combines a geometric path planner with a “nominal controller” and a “reactive controller” to negotiate irregular terrain. The nominal controller adjusts *altitude* (height of robot body), *attitude* (inclination of robot body), *propulsion*, *stepping*, and *force distribution*. They propose an alternative method to the common approach to attitude control that avoids foot slippage and build-up of internal link forces. A method of predicting vertical force redistribution for use in characterizing the terrain is also described. They propose an augmentation to the energy stability margin (ESM) called *compliant energy stability margin* (CESM) that takes into account the compliance of the terrain; a foot with more weight on it will tend to deform the terrain more, causing the center-of-mass to shift toward the more heavily weighted leg.

Their reactive controller attempts to correct for unforeseen real-time changes in the state of the robot due to foot slippage or any event that threatens the static stability of the robot. A *reactive leveling* controller monitors robot stability (through a tilt sensor) and tries to correct for sudden changes. This controller does not affect walking under normal circumstances (*e.g.* when there is no foot slippage).

1.3.4 Results and Contributions of the Geometric Paradigm

An example of a geometrically controlled robot is the Adaptive Suspension Vehicle [30], a large six-legged robot that was built to operate in a variety of modes from fully driver-operated to fully automated. The designers were successful in building a human controlled robot capable of traversing irregular terrain. The amount of human intervention, however, makes their results applicable only where such human resources can be used. When the robot was running in more autonomous modes, results were less successful.

In general, with geometric paradigms, the problem of legged locomotion becomes a problem of providing gaits (or leg motions) for every possible situation and a planner capable of scheduling the different gaits as terrain changes.

Adding reactive components to the gaits provides a much needed flexibility to the otherwise rigid geometric approach. In some cases, however, if the reactive behavior is not carefully designed and incorporated into the geometric controller, it can lead to unpredictable behavior and interfere with the geometric planner. It can become difficult to guarantee that deadlock can be avoided since it is hard to predict what terrain elements will trigger the reactive behaviors.

In general, geometric approaches work well in smooth, not-too-cluttered, and otherwise “friendly” environments. Unfortunately, they fail to address successfully the uncertainties found in real irregular terrain. It is unrealistic to hope to be able to classify all possible types of terrain and design a stepping sequence that will suitably traverse them. Moreover, it is impractical to assume that the environment can be sensed to the degree of accuracy needed in such a scheme. Computing foot-holds on-line based on geometric information often becomes intractable, and suffers from the same sensor uncertainties as off-line planners.

1.4 Dynamics

The work discussed in this document has been conducted on quasi-static platforms using statically stable gaits. The class of static gaits is a subset of the set of dynamically feasible gaits. As such, dynamic gaits improve the flexibility of walking algorithms, especially those designed for irregular terrain. New types of gaits can be considered since a minimum of three legs on the ground at all times is no longer required. Higher speeds, and more difficult terrain can be traversed with dynamic gaits. Dynamically stable robots have been built that hop, gallop and jump, for example, Raibert [29] developed a hopping robot that acts like a pogo stick.

Yoneda *et al* [19] developed a "dynamic and static fusion controller" that can transition between static and dynamic gaits. By decreasing the duty factor, a periodic gait will naturally and smoothly change from a crawl to a trot to a gallop. The controller developed minimizes jerk by using sway to eliminate discontinuities in acceleration of the robot [34].

1.5 Behavior-Based Control

To address the problem of fitting pre-determined gaits to uncertain irregular terrain, and more generally controlling a robot with no, or minimal, modeling, behavior-based controllers were developed. Behavior-based control works on the premise that intelligent behavior can emerge from the interaction of very low-level reflexive behaviors. This notion was expressed most clearly by Braitenberg [7]—one of the first to publish on the subject—who proposed building little vehicles with simple electro-mechanical circuits that would cause the vehicle to exhibit behaviors such as "coward" and "aggressive" as in Figure 1.7. Light sensors are connected through positive or negative logic to drive motors. Braitenberg incrementally augments these basic vehicles with additional simple circuits and allows them to interact with one another, until they get to the point where their behavior is very complicated and difficult to describe in terms of the underlying circuits and individual behaviors.

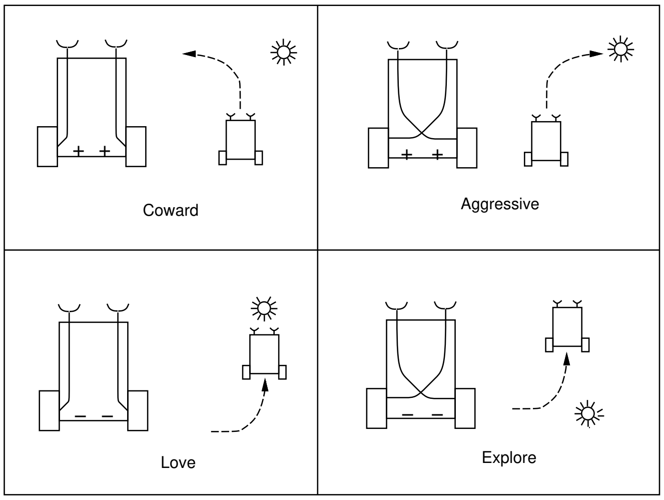

Figure 1.7. Simple Braitenberg Vehicles.

Brooks [8] used this same paradigm in what he calls the *subsumption architecture*. The idea is to control a robot indirectly through the interaction of multiple *layers* of behavior running simultaneously instead of (the more traditional approach of) serial processing where the sensor inputs are analyzed, a plan is constructed and then the actuators execute the plan.

The subsumption architecture uses "augmented finite state machines" (AFSMs) connected in a network. A layer in this network consists of multiple AFSMs whose operation can be suppressed by higher layers. Any set of lower layers form a complete operational control system.

Brooks demonstrated this approach on numerous mobile robots, including a six legged walker called Ghengis [2]. With this robot it was shown that walking can emerge from a network of simple reflexes with little central control [9]. The behavioral layers used to control the robot are: Stand up, Simple walk, Force balancing, Leg lifting, Whiskers, Pitch stabilization, Prowling, and Steered prowling. Each of these layers of competence is built from a number of AFSMs—often with the machines controlling only a single leg.

The "Simple Walk" behavior, for example, uses three AFSMs that are repeated for each leg and know nothing of the other legs, and one global AFSM that coordinates the legs. The *leg down* machine notices whenever its leg is not in the down position and puts the leg down. The *alpha balance* machine monitors the forward swing of all six legs and controls the sum of swing angles so that when one leg happens to be moving forward, the other five move slightly backward. The *alpha advance* machine moves a leg forward when it is up (inhibiting the global *alpha balance* machine). The *up-leg trigger* machine lifts a leg (momentarily inhibiting the leg down machine).

The asynchronous combination of these behaviors can produce walking-type behavior: The *up-leg trigger* lifts its associated leg, which triggers *alpha advance* to swing the leg forward, causing *alpha balance* to move all the other legs back slightly (moving the robot forward). Then *leg down* puts the leg back on the ground. By supplying a global temporal pattern of triggers to the *up-leg triggers*, the robot can be made to perform a fixed walking gait.

This approach has had significant implications in the debate over central versus peripheral control of motion [6]. Further work, however, has shown that, although on the surface it appears that this approach is easy to scale up to more complex tasks (simply add new layers on top of the existing ones), it is not so easy. As the complexity of the task increases, so does the designer's task of fine tuning the behaviors to produce the desired result. The controller design becomes an *ad hoc* process, and as shown by Celaya and Porta [11], it may be necessary to completely re-design an architecture when a task is scaled up to more difficult goals such as walking over irregular terrain.

1.6 The Control Basis Approach

This thesis will present an on-line approach to the generation of gaits for irregular terrain that uses a basis of reusable feedback control laws. Chapter 3 will present a control policy for flat terrain, and chapter 5 will modify this policy for irregular terrain.

Chapter 2

Experimental Hardware

2.1 Thing

The experimental platform used in the work reported in this thesis is a four-legged 12 degree-of-freedom walking robot named *Thing* that was designed for autonomous locomotion over irregular terrain [24]. The robot is shown in Figure 2.1. It stands $0.25m$ tall and weighs approximately $1.5kg$. Thing is equipped with five 8-bit microprocessors connected in a serial-star configuration. It can operate autonomously with its own processors and NiCad batteries or can be linked via a serial tether to a workstation.

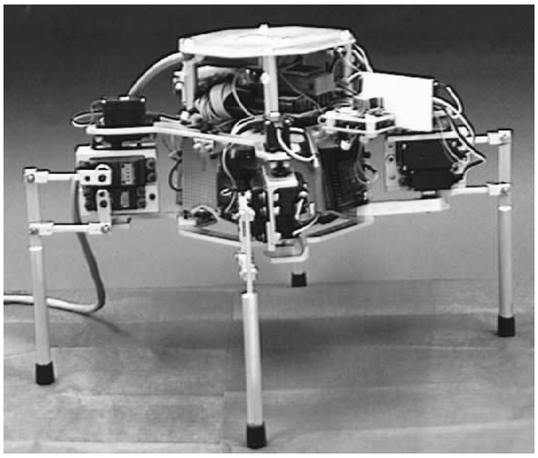

Figure 2.1. *Thing* a Quadruped Walking Robot.

The legs are four-bar linkage mechanisms with fork-and-shaft joints actuated by PWM controlled servos, as shown in Figure 2.2. Joint 0 rotates about the *z-axis* and provides the swing motion of the leg, while Joint 1 rotates about the *y-axis* (into the page), supporting the weight of the robot. A passive torsion spring is utilized to counteract a portion of this weight. Joint 2 also rotates about the *y-axis* and drives the "knee" motion of the leg. The legs are made of milled aluminum and were designed for low cost, ease of construction, high efficiency, and maximum usable workspace.

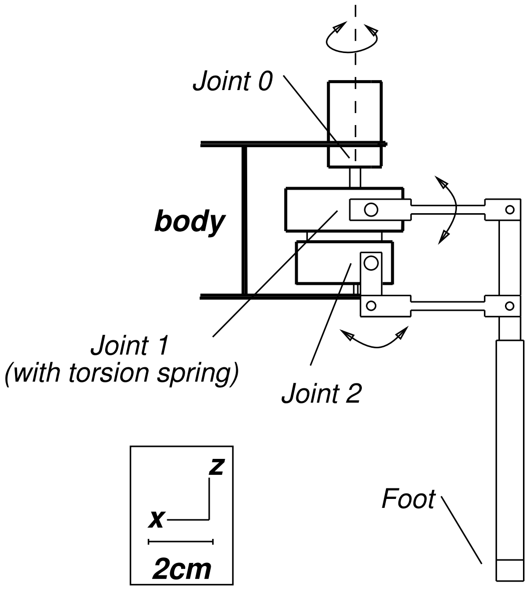

Figure 2.2. Thing's Leg Design

Odometry in the robot is maintained by accruing relative translations and rotations of the center-of-mass of the robot with respect to a known initial configuration.

The sensory suite on Thing consists of infrared sensors for sensing large obstacles in the terrain that must be walked around and touch sensing on the feet that allow the robot to map local terrain and choose foot holds.

2.2 Infrared Proximity Sensors

Two infrared (IR) proximity sensors provide a binary signal indicating the presence of large obstacles within a range of $0.2m$. The sensor configuration allows the robot to determine the existence of obstacles within three sectors in front of the robot, as shown in Figure 2.3.

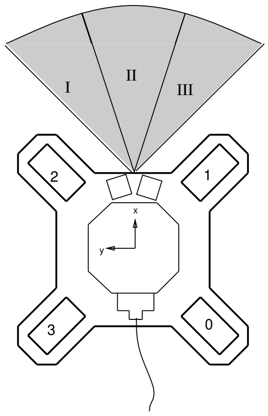

Figure 2.3. Thing's Infrared Sensors and Their Range.

The infrared light is a modulated $40KHz$ signal that allows the receiver to filter out noise from fluorescent and natural light. Further statistical filtering is provided by averaging sensor readings over multiple trials. The threshold for a positive signal is software controllable and provides a means of adjusting the approximate range of the sensors.

2.3 Force Sensors

Navigation over irregular un-mapped surfaces requires the ability to sense the terrain. Force sensors are often used to measure aspects of the terrain [10, 3, 35, 11, 18] and of the state of the machine [27, 1]. This section will describe the development of sensors for Thing that allow it to do this.

A common method of sensing forces is through sensors on the legs and feet. These are often in the form of torque sensors at the actuating joints whose data can then be translated into forces on the foot (or in some cases, contact localizations elsewhere on the manipulator [20]). Force sensors are also found in the form of strain gauges that measure deflection (bending) of an otherwise rigid link in the manipulator due to forces and torques acting at either end of the link. These are often on the feet of the robot [27, 10, 3].

Thing uses torque sensors at the actuating joints to determine forces on the feet, as shown in Figure 2.4. The translation is through the inverse transposed Jacobian:

$$\vec{f} = (J^T)^{-1}\vec{\tau}$$

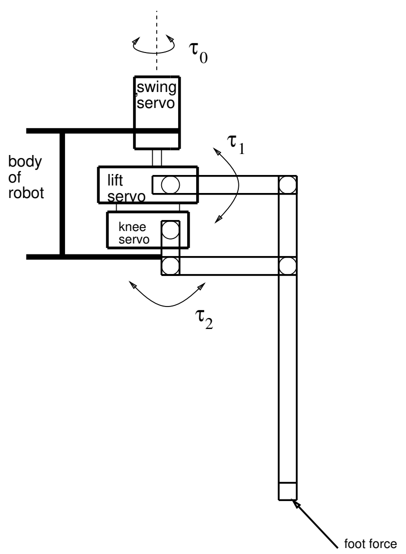

Figure 2.4. Sensing Forces on the Foot.

2.3.1 Hardware

Hardware development was required for Thing to be able to acquire torque data. The robot uses PWM (pulse-width-modulated) reference signals to position-control the joints [24]. Each Futaba servo converts its reference signal into current to the motor based on the angular error read with a potentiometer connected to the output shaft, as shown in Figure 2.5. The command from the controller is broken into two different PWM signals: one for clockwise (CW) control and the other for counter-clockwise (CCW) control. These signals are easily tapped from the locations on the Futaba circuit boards shown in Figure 2.6.

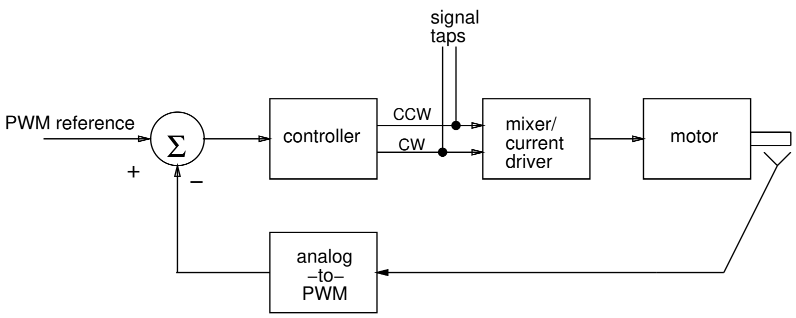

Figure 2.5. The Servo Control Loop used on Thing.

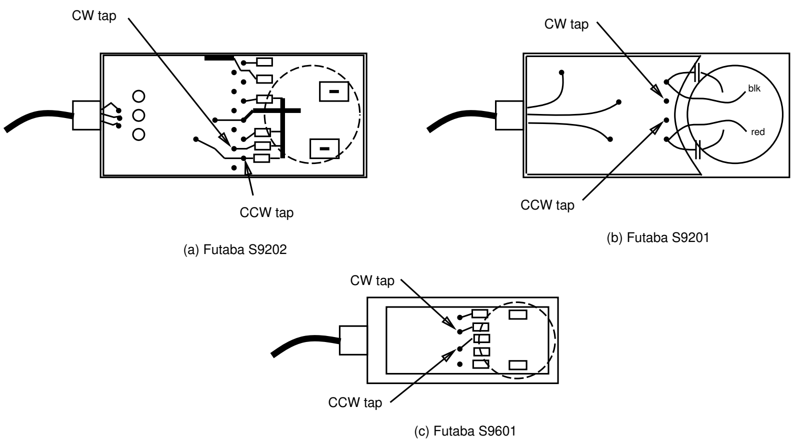

Figure 2.6. Signal Taps on Futaba Servos (bottom covers removed from servos).

The MC68HC811 leg processors have three inputs capable of measuring pulse width. Therefore, in order to handle the three leg servos (with two signals per servo), the signals are multiplexed.

2.3.2 Interpreting Control Signals

The internal control signals taken from the Futaba servos are not directly proportional to the joint error or torque. Moreover, since the transfer functions of the various blocks in Figure 2.5 are not available from the manufacturer, a closed-form relationship can not be calculated. Instead, the relationships were found empirically.

Data was collected using the apparatus shown in Figure 2.7. Static tests were run in which successive reference θ commands were given to the servo, and the control signals (from the servo taps) and the resulting forces (from a calibrated Lord six-axis force sensor) were recorded.

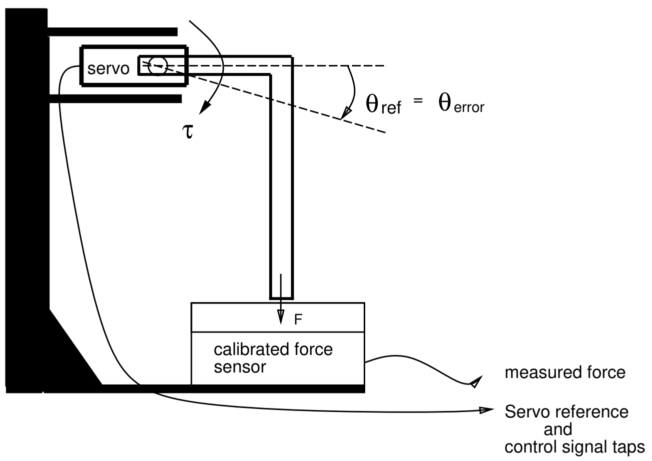

Figure 2.7. Apparatus for Calibrating Servo Control Signals.

Second-order polynomial equations were fit to the data for (positive and negative) torque and angular error for each type if servo on the robot, as shown in Figure 2.8. In the top two plots, the angular errors versus the control signal pulse-widths were identical for the two servo types used by the robot (Futaba S9201 and S9601). The resulting curve that was fit is:

$$\theta_{err} = \left\{ \begin{array}{ll} 3.12 * 10^{-10} w^2 - 6.21 * 10^{-7} w + 0.0267 & w \leq 0 \\ -4.71 * 10^{-10} w^2 + 5.14 * 10^{-6} w - 0.0230 & w > 0 \end{array} \right\}$$

Where w is the measured pulse width.

In the bottom two plots the torques versus control pulse widths for the Futaba S9201 were higher than those for the S9601. This was to be expected, since the S9201 is larger and has a higher torque rating. The resulting curve that was fit to the S9201 data is:

$$\tau_a = \left\{ \begin{array}{ll} -4.46 * 10^{-10} w_a^2 - 3.19 * 10^{-5} w_a - 0.00696 & w_a \leq 0 \\ 3.82 * 10^{-10} w_a^2 - 3.20 * 10^{-5} w_a + 0.0106 & w_a > 0 \end{array} \right\}$$

And the resulting curve that was fit to the S9601 data is:

$$\tau_b = \left\{ \begin{array}{ll} -4.09 * 10^{-10} w_b^2 - 2.61 * 10^{-5} w_b - 0.00797 & w_b \leq 0 \\ -2.62 * 10^{-10} w_b^2 - 1.75 * 10^{-5} w_b - 0.0125 & w_b > 0 \end{array} \right\}$$

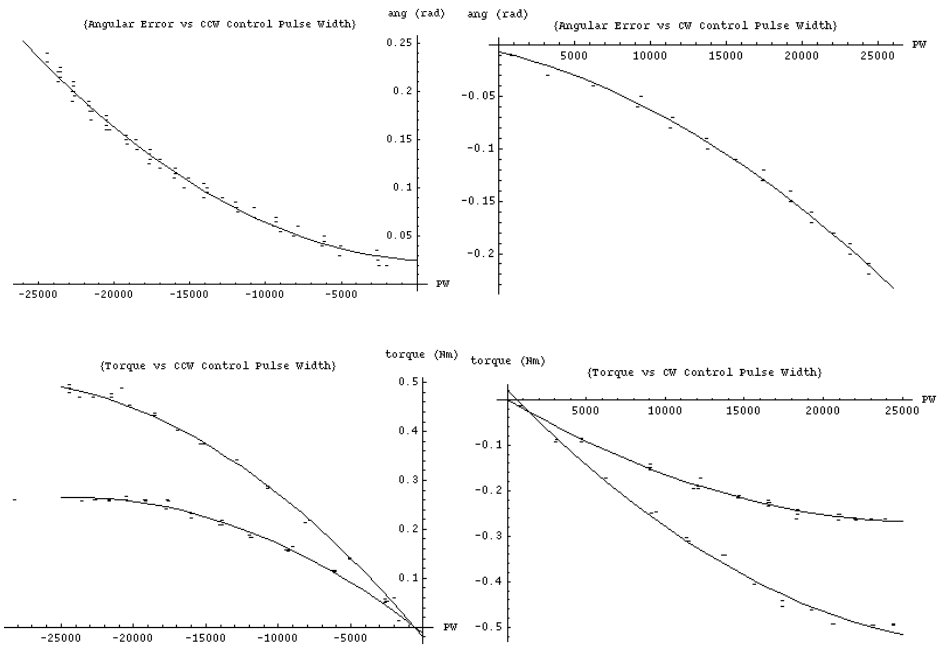

Figure 2.8. Torque and Angular Error Versus Control Signal Pulse-Width for Futaba S9601 and S9201 Servos.

2.3.3 Modeling Spring Torques

Thing has passive torsion springs on joint 1 of each leg. The torque added by these springs varies with angular position of the joint and must be accounted for when measuring torque at the joint. The springs on each leg were characterized individually. Figure 2.9 shows experimental plots of measured torque versus joint angle.

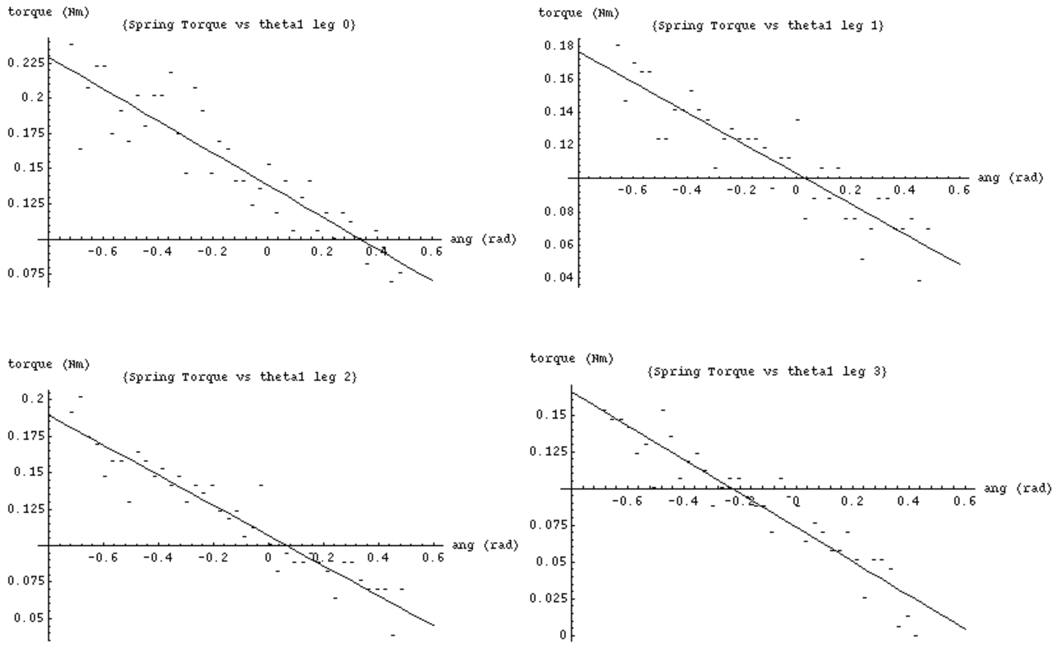

Figure 2.9. Experimental Torsion Spring Characteristics.

The resulting correcting equation for torques measured in joint 1 is:

$$\vec{\tau} = \vec{\tau_m} + \vec{s^T}\vec{\theta} + \vec{b}$$

where $\vec{\tau}$ is the actual torque at joint 1 on each of the four legs: $\vec{\tau} = [\tau_0 \ \tau_1 \ \tau_2 \ \tau_3]^T$, and $\vec{s}$ and $\vec{b}$ encapsulate the empirical lines fit to the data of Figure 2.9 for all four legs. $\vec{s}$ and $\vec{b}$ were found to be:

$$\vec{s} = \begin{bmatrix} -0.114 \\ -0.0917 \\ -0.103 \\ -0.115 \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} 0.139 \\ 0.103 \\ 0.107 \\ 0.0740 \end{bmatrix}$$

It was found that when taking torque readings, friction in the joints caused erroneous readings. A technique was discovered that avoided this problem: move the joint to approximately 0.15 *rad* past the desired position, then back 0.15 *rad* and take the reading.

2.4 Maximizing Workspace

In order to handle irregular terrain, the largest workspace possible is needed. Previously, absolute joint limits were used when controlling the legs. These limits, however, waste joint space, since Thing actually has *relative* joint limits—the limits of joints 1 and 2 are kinematically coupled.

The relative workspace was determined as shown in Figure 2.10. The additional workspace allowed Thing to handle more difficult terrain than if it only used the absolute joints limits. Note, joint 0 is orthogonal to joints 1 and 2 and, therefore, has only absolute limits.

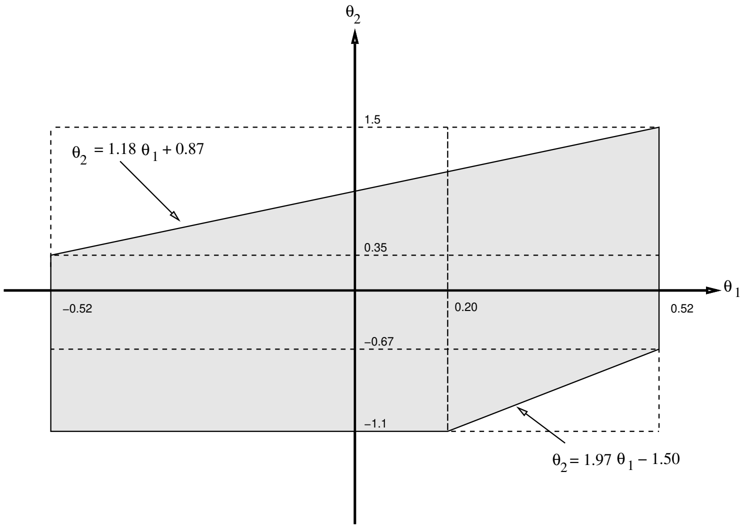

Figure 2.10. Workspace of Joints 1 and 2 in Configuration-Space.

Figure 2.11 shows Thing's entire Cartesian workspace. Each panel shows the reachable xy position for a given z height. Below $z = -0.20m$ the leg starts to run into joint limits.

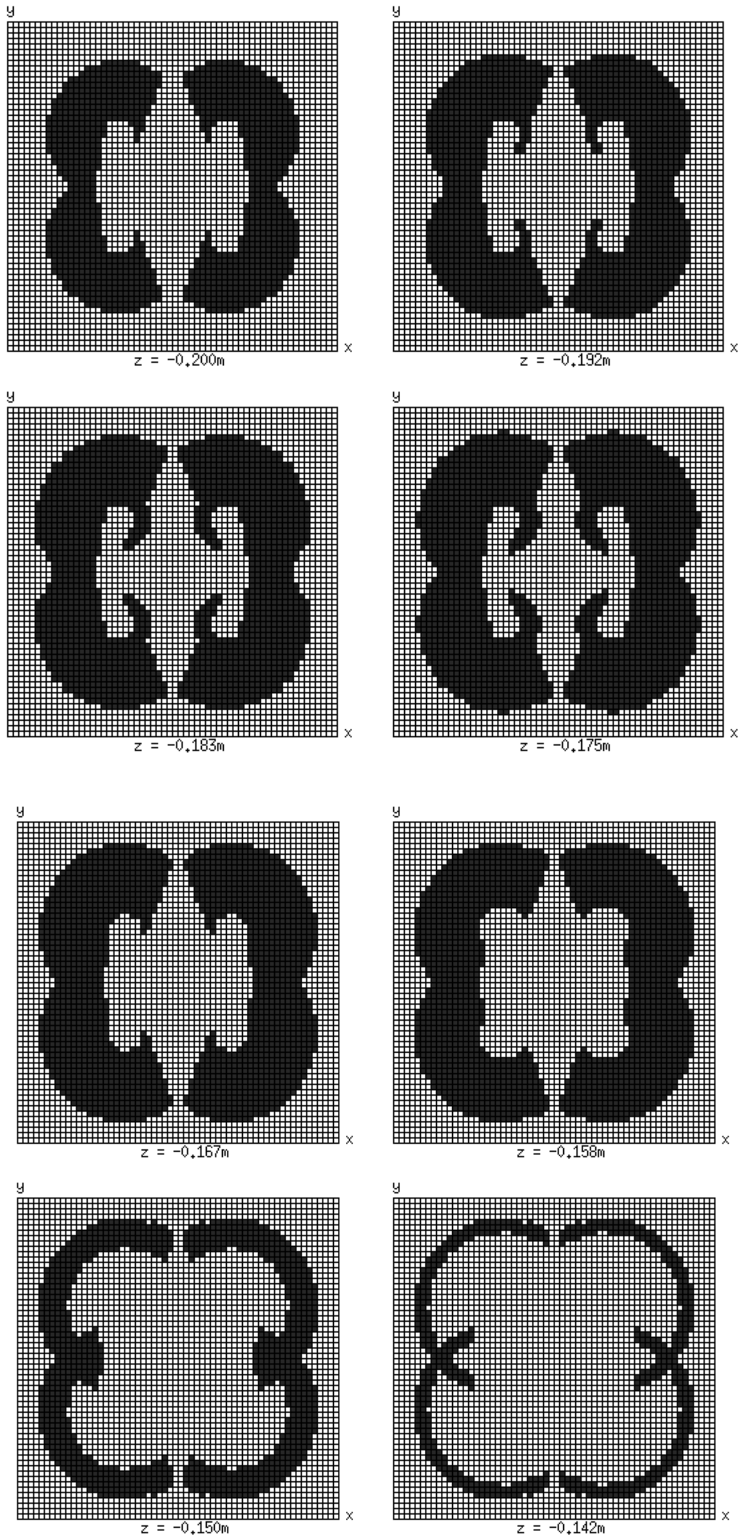

Figure 2.11. Cartesian Workspace.

Chapter 3

A Control Basis For Multilegged Walking

3.1 The Control Basis Approach

The word *basis* comes from elementary linear algebra and is defined as a linearly independent spanning set for some subspace of R^n. This means that given some basis, any vector in the subspace spanned by the basis can be generated from a linear combination of the basis vectors. A *control basis* [16] is an abstraction of this vector basis in which control functions are constructed from a set of basis controllers. A *composition policy* activates subsets of these controllers in order to achieve reference states while satisfying operational constraints.

Each controller in the basis has stability and convergence properties over a range of possible plants. We assume that each controller will achieve a goal state. Cases where a goal is not reached may cause a transition to a new set of active controllers that *do* converge. Figure 3.1 shows a schematic of the control composition. A composition policy determines which controllers in the basis will be active. Multiple controllers can run simultaneously if they are orthogonal. Non-orthogonal controllers can be run simultaneously if one is designated the *dominant* controller and the others *subordinate* controllers. The subordinate controllers can operate only in the *null space* of the dominant one. The null space of a controller (another abstraction from linear algebra) is a set of states of the system in which the controller remains at equilibrium. A subordinate controller can only introduce actions which do not interfere with the equilibrium of the dominant controller.

Each controller is assigned a set of input resources and a set of output resources. These resources could be sensors or robot degrees of freedom. They may

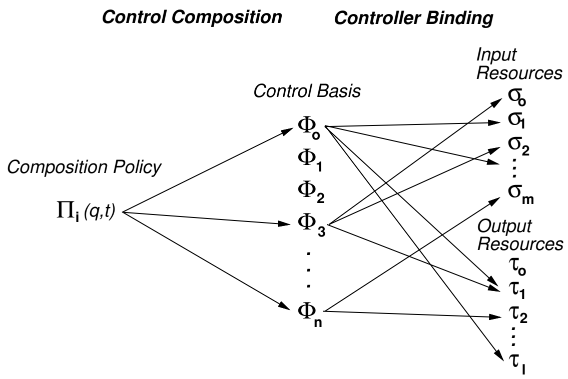

Figure 3.1. Control Composition

also be abstractions of these, such as the position of an end effector (this position would be derived from the actual angles sensed in the joints).

This distributed control approach constructs behavior on-line, and the basis controllers are task and platform independent. This means that the *same* basis controllers can be used to perform many different tasks on many different robot platforms by simply changing the composition scheme.

On the surface, the control basis approach resembles behavior-based techniques in that behavior is derived on-line from the interaction of independent elemental controllers. In fact, there are profound differences between the two paradigms, the most significant being that the controllers that make up a control basis are provably convergent—they are guaranteed to reach equilibrium within the capabilities of the sensors and scope of the environment in which they are designed to operate. Also, they perform more general subtasks like *kinematic optimization* as opposed to the more task-specific behaviors of the subsumption approach, such as *lift leg*. This means that the basis controllers are task and platform independent. Control composition over a basis of controllers is a formal process with strict guidelines that guarantee the correct interaction of the controllers running at any given time. The design of a task-specific controller can be conducted at the compositional level, since the basis will contain all of the capabilities of the system.

The control basis approach has been demonstrated by Grupen *et al* [16] who originally built a basis of controllers for grasping and manipulation tasks with a four fingered hand. The *same* control basis—with a different composition scheme—was used by Huber *et al* [21] to perform walking tasks on Thing, the four legged walker introduced in chapter 2. This control basis for walking is described in the next section.

3.2 A Control Basis for Walking Tasks

Locomotion through an unknown, obstacle field over flat terrain was achieved using a basis of three controllers [21]:

Φ_0: Configuration space motion control.

Harmonic function path controllers [13] are used to generate robust, reactive, and collision-free motion through the configuration space of the robot. In this case, the configuration space is the x, y, Φ (position and orientation) of the center-of-mass of the robot. This controller will try to move the robot from the start to the goal without running into large obstacles in the path.

Φ_1: Contact configuration control.

A contact configuration controller [12] is employed to move foot contacts based on the local geometry of the environment in order to minimize residual forces and moments acting on the center-of-mass of the platform and, thus, to stabilize the robot.

Φ_2: Kinematic conditioning.

The kinematic conditioning controller optimizes the posture of the articulated structure while it is engaged in an interaction with the world. It tries to maintain leg postures in which the leg has maximum workspace (*i.e.* the actuators are positioned optimally between their limits).

Each instance of a controller takes the form $\Phi_i{}^{\sigma}_{\underline{\tau}}$, where Φ_i is an element of the control basis, and superscript σ and subscript τ denote the sets of input and output resources, respectively, which are bound to this instance of controller i. These resources are combinations of the position of legs $0, 1, 2$, and 3, and of the position and orientation (x, y, φ) of the center-of-mass.

Task level control is achieved by composing these control elements in a context-dependent way. Figures 3.2 and 3.3 present examples of the type of finite state supervisors employed by this approach. The control policy at each point in time is shown inside each state as a set of asynchronous, ";" (*i.e.* running in parallel), and/or hierarchical, "$\triangleleft$" (*e.g.* $\Phi_i \triangleleft \Phi_j$ means controller i is running in the null space of controller j), activations of controllers, which are bound to specific system resources (shown in the top right corner of Figures 3.2 and 3.3). The convergence event of a particular controller can be thought of as the assertion of a condition predicate: $convergence\ of\ \phi_1 \Rightarrow p_1$. The transition to a new state occurs when all of the predicates for the controllers in the current state have been asserted.

Tasks are, therefore, not defined in terms of geometric foot placements, but rather as stability, kinematic conditioning, and navigational objectives. Body stability and forward progress are established explicitly through the activation of corresponding controllers. Throughout execution, the safety of the composite system can thus be guaranteed within the controller capabilities and the sensory and motor limitations of the robot.

3.2.1 Control Gaits for Quadruped Walking

In order to allow the system to perform general walking, two major tasks were implemented in the form of control gaits, namely pure rotation and forward translation. The finite state control policy for the rotation task is shown in Figure 3.2.

Starting from an initial four-legged stance, this gait first shifts the center of mass (x, y) of the robot to form a stable stance on legs 0, 1, and 2, while

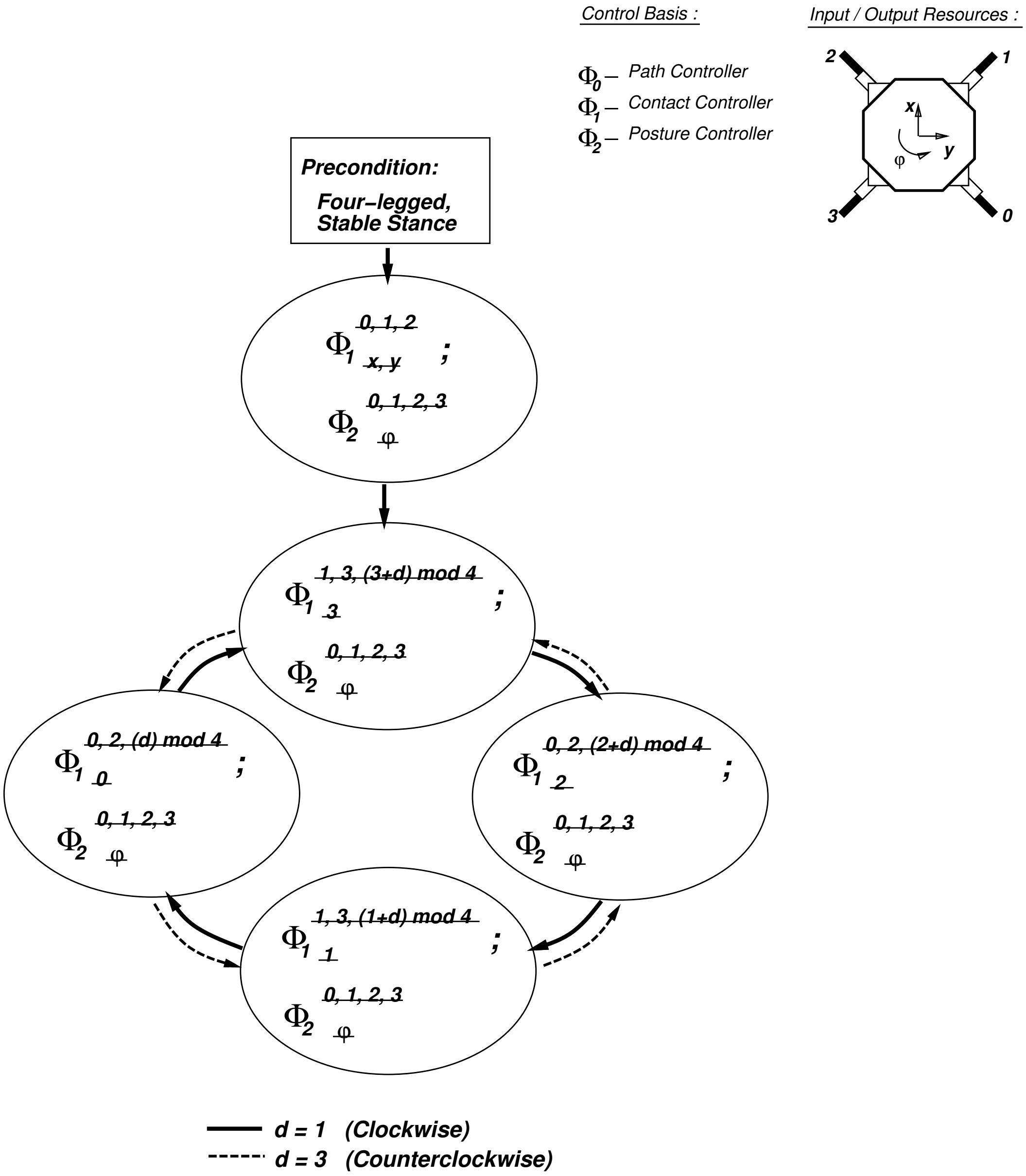

Figure 3.2. Control Gait for Rotating in Place

the posture controller asynchronously adjusts the orientation, φ, of the body to kinematically condition all 4 legs. After convergence of the controllers in this initial control state, the control policy lets the robot system cycle through a succession of stable, three-legged stances, moving one leg at a time, while achieving the rotation through kinematic conditioning around the z-axis.

In order to allow spatial progress of the system, a second gait for general forward walking was implemented. Exhibiting a similar periodic structure to the rotation gait, the control policy for this gait is shown in Figure 3.3. The relative positions of the feet are shown next to each state to help clarify the gait. The current polygon of support is indicated by a solid line, and the *next* stable stance that the contact controller must establish is shown by a dashed line.

In this gait the robot maintains its stability by cycling through a sequence of three and four-legged stances, moving the individual legs in the sequence $3, 2, 0, 1$. At the same time, the posture controller attempts to optimize the kinematic configuration of the legs subject to the stability constraints imposed by the contact controller. As an example, $\left(\Phi_2{}^{3}_{\underline{3}} \lhd \Phi_1{}^{0,1,3}_{\underline{3}}\right)$ activates the contact controller, Φ_1, in order to establish a stable three-legged stance on legs $0, 1,$ and 3 by moving leg 3. Limited by the "subject to" constraint, the posture controller, Φ_2, optimizes the posture of leg 3 within the range of stable stances.

In addition to stabilizing the body, this gait also includes two states, (3) and (7), that activate instances of the path controller, $\Phi_0{}^{x,y}_{\underline{x,y,\varphi}}$ subject to the stability constraints of the existing three-legged stance. This allows the gait to adjust the body location and orientation to conform with the direction of the desired path through and around obstacles.

The gait diagram, stability, differential forward progress ($\frac{dx}{dt}$), and forces on leg 0 during a segment of a straight walking is shown in Figure 3.4. This data was collected after a slight disturbance in the gait to show the aperiodic, non-regular nature of the controller.

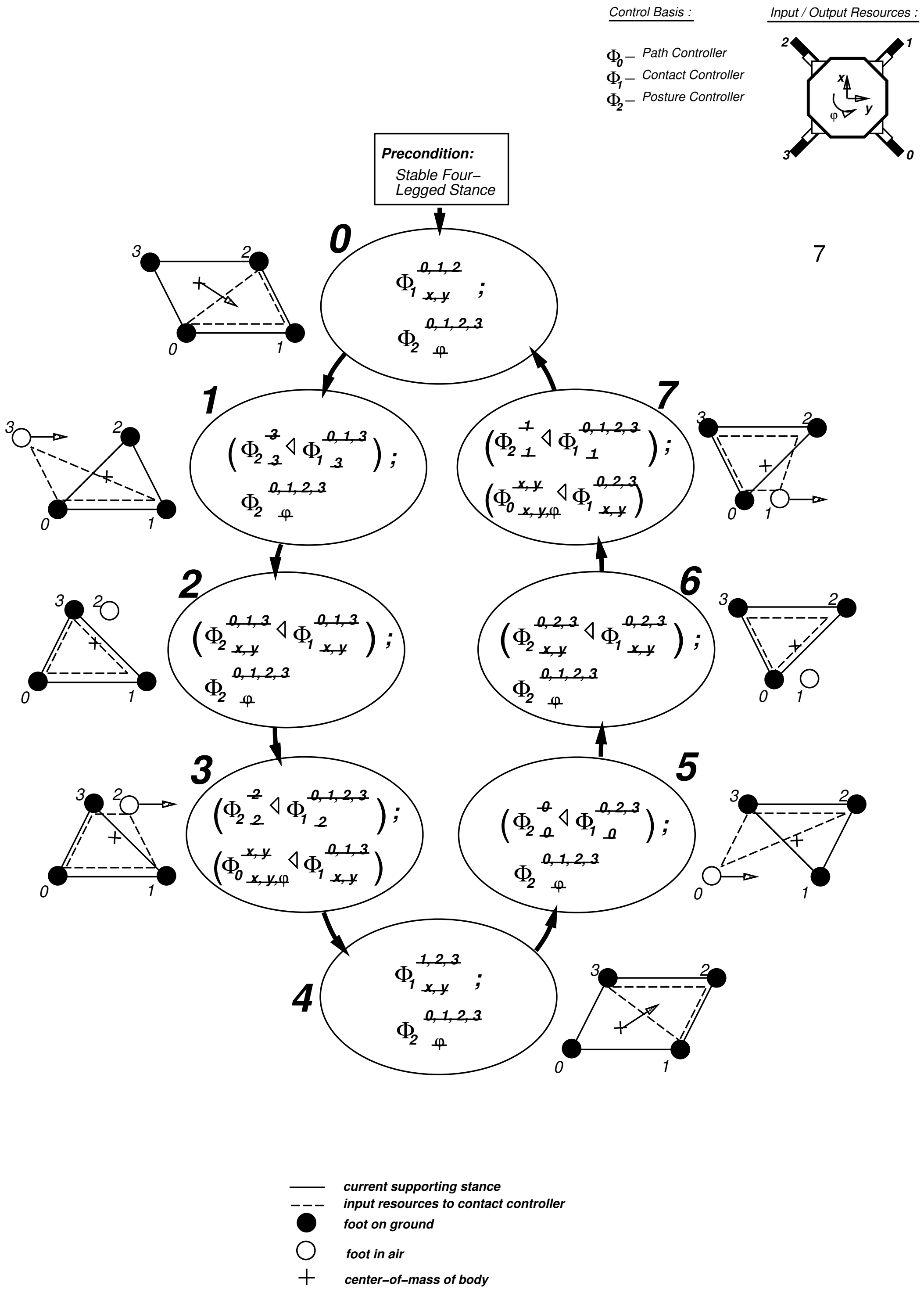

Figure 3.3. Control Gait for Forward Walking

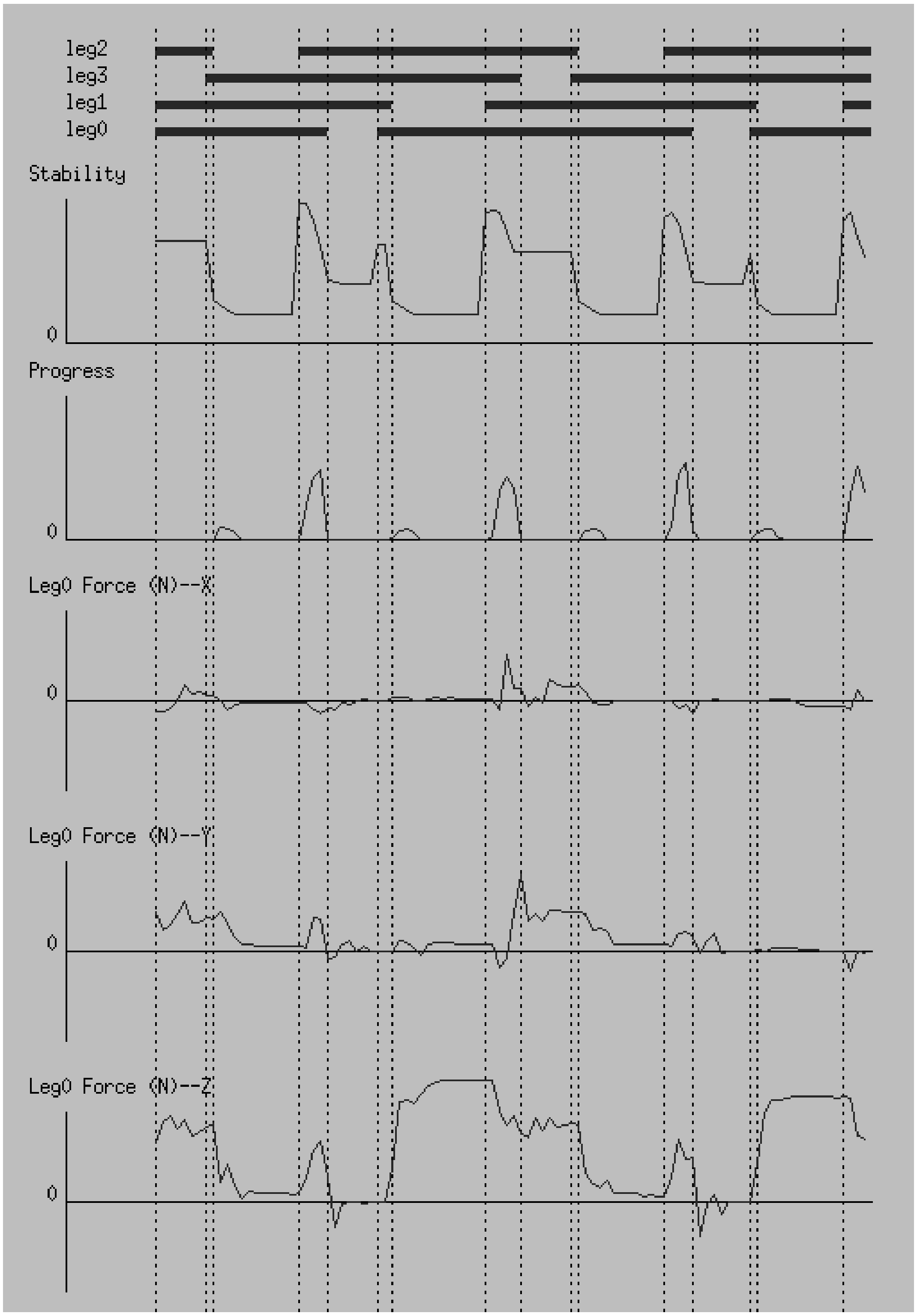

Figure 3.4. Stability, Forward Progress, and Forces During A Walking Experiment on Flat Terrain.

In contrast to geometric gait planning, where a different set of repetitive foot patterns is used to adjust to different walking directions, the control gaits used here continuously recompute foot placements based on the contingencies of the task and local sensory information. This can lead to significantly more flexible and reactive performance. The walking gait, for example, is capable of walking not only in a straight line but also at limited angles without activation of the rotate gait. Moreover, it can execute path curvatures with instantaneous radii larger than approximately $0.4m$. In geometric terms, this strategy encompasses aperiodic versions of discontinuous walking, as well as crab and turning gaits [14], augmented by a reactive control component. In addition to this reactivity and the simplified gait switching, this control gait is also still largely device independent and thus generalizes well to other four-legged walking robots.

3.3 Results

To demonstrate the capabilities of the composite controllers described above, the two control gaits were integrated using the switching policy shown in Figure 3.5, where $\Delta\varphi$ is the difference between the current and the desired heading given by the path controller [21].

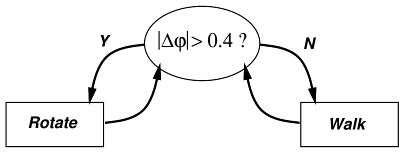

Figure 3.5. Gait Change Policy for Example Task

Using this control scheme, the robot was placed in an unknown flat terrain environment with the task of reaching a specified goal. Utilizing the IR sensors and odometry of the system (see chapter 2), the robot builds a map of the local environment as shown in the left column of Figure 3.6. This map, which models a $9m^2$ region is internally represented as a 32x32 cell grid and used by the path controller. When an obstacle is detected, the corresponding location is marked on the grid and dilated appropriately to reflect the geometry of the robot. All movements of the robot, including foot placements and global navigation, are generated reactively on-line by the activated controllers in the control gait without the need for off-line planning. This allows the system to react very quickly to the detection of new obstacles without the need for frequent gait transitions.

Figure 3.6 shows the robot's state at six points during execution. Panels in the left column show the robot's internal map, and panels in the right column show the corresponding actual state of the robot and environment. The markers in the top panel of the robot's map represent the goal (on the left) and the starting point (on the right) and correspond to the two crosses seen in the images in the right hand column. Since no obstacles other than the boundaries of the internal map have been mapped for the path controller, the robot initially heads directly from start to the goal. When it encounters an obstacle, it is included in the map, as shown in the second panel, and the grid is relaxed to compute a new harmonic surface, providing a new desired heading for the path controller. Grid relaxation is repeated each time a new obstacle is found.

The gradient of the harmonic function influences the path of the robot in two ways: indirectly through the gait switching policy of Figure 3.5 and directly by means of the activated instances of the path controller in the forward walking gait. This allows the robot not only to turn away from newly discovered obstacles but also to follow smoothly curved trajectories leading from the current location to the goal. For example, when most of the obstacle barrier has been detected, the best path to the goal becomes the long arc around the entire barrier, which is the path finally taken, as shown in the bottom panel.

Throughout the total walking time of 20 minutes, the robot activated the

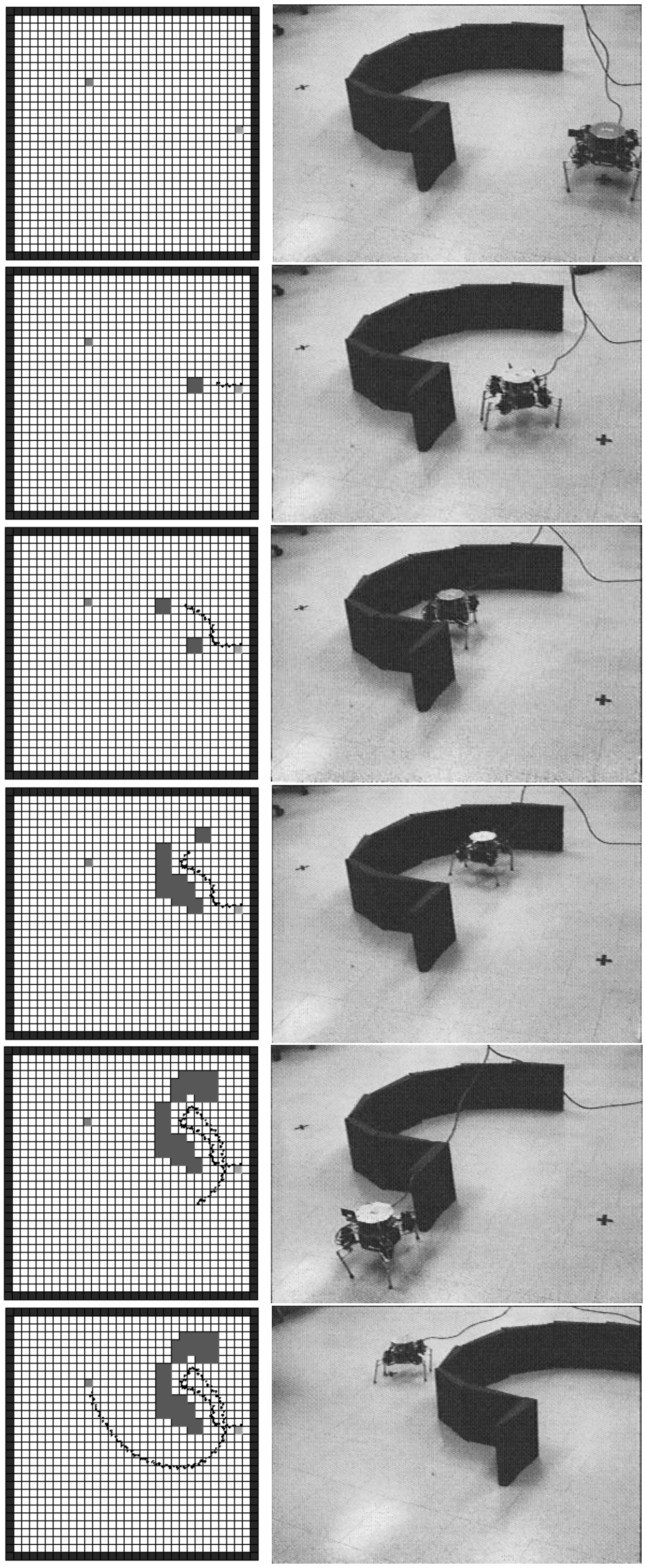

Figure 3.6. Walking Task in an Unknown Environment. Start and goal on the internal map (left) and actual robot (right) are indicated by initially shaded grid cells and crosses on the floor, respectively. The path of the robot is marked by black dots corresponding to the location of the center of mass.

rotation gait only 3 times, traveled approximately $5m$, and accrued a translational odometry error of approximately $0.2m$.

Chapter 4

Terrain Mapping

There are two natural ways for a walking robot to navigate [10]. The first is by means of sensors that directly provide an image of the terrain, such as that obtained by a vision system. This sensor modality is heavily relied on by humans, but tends to be very computationally expensive in machines. The second way is for the robot to blindly *feel* its way over and around obstacles by means of contact sensors on the legs and body of the robot. This is the approach discussed here.

The extent to which force information can be used varies from simply verifying foot-holds to actually creating internal maps of the terrain. For example, Nagy *et al* [27] compare actual foot forces (measured with six-axis force sensors on the feet) with expected foot forces (based on predictive models) to determine the quality of the foot-holds and learn the "walker/terrain interaction." A central region on the foot is designated. The resultant force vector on the foot due to the load of the robot must pass through this region for the foot-hold to be called safe. Otherwise it is called a "toe-hold". When a foot is in a toe-hold, the force applied to that leg is limited to a certain safe threshold in order to avoid slippage. Then the foot can be pre-loaded to further check the foot-hold. Once the foot-hold has been evaluated, the planned gait can proceed.

4.1 Disturbance Suppression

Klein *et al* [23] use force information to realize active-compliance in the feet. The method uses a (virtual) spring-damper suspension configuration (Figure 4.1) augmented to a fixed smooth terrain gait. Disturbances due to terrain irregularities are damped in a similar way to the function of an automobile's suspension system.

This is a passive approach to dealing with irregularities in the terrain. The bumps and contours are not explicitly mapped; instead, their influence on the robot's gait is suppressed.

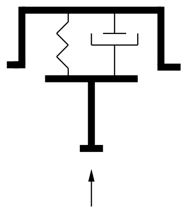

Figure 4.1. Active Compliance at the Feet.

Yoneda *et al* [35] use a similar approach that breaks walking control into two parts which they call "feed-forward" and "feedback". The feed-forward controller drives the legs to achieve a suitable force distribution, assuming flat terrain. These force commands are then modified based on sensor data in the feedback controller, which tries to maintain a level body posture in the presence of disturbances by setting a "virtual impedance" in absolute coordinates. The robot appears as though it is suspended from an absolute frame by springs and dampers (the approach was coined "sky-hook suspension"). Gaits used by this group included dynamic walking (*i.e.* in some cases the gaits included periods of only two-legged support).

In sky-hook suspension control, each leg's force is controlled only during its supporting phase. The legs are position-controlled during the lift and swing phases, and switched to force control at the instant of touch-down determined by the presence of a small normal force on the foot. Force control is then switched off when the foot is lifted off the ground. Using this method, terrain height does not need to be known *a priori*, however the foot must always contact the terrain from above (contact in the swing or lifting phases will not be sensed).

Approaches that suppress disturbances in the terrain have been successfully demonstrated on simple irregular terrain [10, 35]. They work as long as the corrective measures do not threaten the stability or kinematic conditioning of the robot.

In irregular terrain, it may be necessary to modify the original gait plan in order to find a solution. For example, a 1324 periodic gait would run into trouble when, say, leg 2 could not be placed to set up the next tripod on legs $1, 3$, and 2 This could be due to a large obstacle or kinematic constraints. Schemes that rely on suppressing disturbances so that a fixed supervising gait can be used would fail in this situation. The robot could loose stability due to a foot placed where it happened to end up while complying to the terrain. Another scenario is deadlock later in the gait cycle due to the kinematic space that was used up in the placement of the trouble leg. For example, a leg that was left at the rear of its joint limits due to compliance with an obstacle that was in the way will restrict the forward motion of the body later in the gait cycle.

Reactive and behavior-based methods often use force information to initiate fixed reactive behaviors [11, 15, 18]. The forces are checked against a fixed threshold. When forces above this threshold are sensed, a behavior is initiated. This can lead to modified gaits, as opposed to just modified foot placement, however, behavior-based responses to irregular terrain cannot be guaranteed to find a stable gait.

4.2 Mapping Facets with Contact Normals

An alternative approach is to *map* the local terrain and use this information to modify or generate a gait. For example, Adachi *et al* [1] use force sensors to measure a global characteristic of the terrain—its slope. This information is then used to modify the planned geometric gait.

More detailed mapping of the terrain local to each foot can provide a means of planning gaits on-line. This local mapping can be achieved through interaction with the terrain. One type of interaction is to obtain surface normals through multiple contacts with the terrain. Given the contact normals, local planar facets can be postulated. These facets can then provide useful information to a gait planner.

Contact normals can be approximated as the reactive force measured when the foot contacts the ground, as shown in Figure 4.2. When friction at the contact point is non-zero, the normal approximation can only be estimated to within a conical region around the point of contact (the *friction cone*), as shown by the dashed lines.

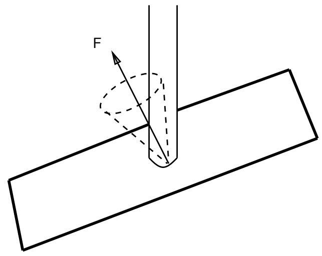

Figure 4.2. Sensing Surface Normals at the Foot.

In real terrain, friction between the foot and ground will, in general, have large friction coefficients, causing large friction cones and poor normal data. In this case, techniques have been developed—mostly in the domain of object recognition with robotic hands—that can extract the surface normals [5], however enhanced sensors are required at the fingertip (or foot).

4.3 Mapping Facets with Linear Regression

If accurate contact *position* data is available (*e.g.* through joint encoders), then facets can be estimated with multiple contact points using linear regression. This is the technique used with Thing.

Contact is determined based on sensing a force above a certain threshold at the foot. Given a set of contact points, a weighted least-square approximation to the facet is made as shown in Figure 4.3. Least-square linear regression will minimize the sum of the squared error between the points and the postulated facet.

By weighting each point's error by its variance—predicted based on joint errors and leg configuration—outliers will have reduced influence.

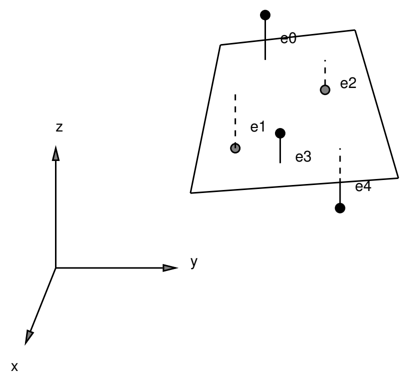

Figure 4.3. Least-Square Approximation of Plane Minimizes $\sum_{i=0}^{n} e_i^2$.

Since most of the surfaces being approximated will be near-horizontal, just error in the z direction will be minimized. Given the equation of a plane:

$$z = ax + by + c,$$

the error for point i in the z direction will be:

$$e_i = Z_i - z_i,$$

where Z is the measured height, and z is the height predicted by the estimated plane. The weighted sum of the square errors is:

$$S = \sum_{i=0}^{n} e_i^2 w_i = \sum_{i=0}^{n} (Z_i - ax_i - by_i - c)^2 w_i \tag{4.1}$$

where n is the number of points, and w_i is the weight assigned to point i. To find the minimum, we differentiate with respect to each of the planar parameters and set them to zero:

$$\begin{aligned} \frac{dS}{da} &= \sum_{i=0}^{n} 2(Z_i - ax_i - by_i - c)(-x_i)w_i = 0 \\ \frac{dS}{db} &= \sum_{i=0}^{n} 2(Z_i - ax_i - by_i - c)(-y_i)w_i = 0 \end{aligned}$$

$$\frac{dS}{dc} = \sum_{i=0}^{n} 2(Z_i - ax_i - by_i - c)(-1)w_i = 0$$

The *normal equations* are then:

$$\begin{aligned}
a\sum_{i=0}^{n} x_i^2 w_i + b\sum_{i=0}^{n} x_i y_i w_i + c\sum_{i=0}^{n} x_i w_i &= \sum_{i=0}^{n} x_i Z_i w_i \\
a\sum_{i=0}^{n} x_i y_i^2 w_i + b\sum_{i=0}^{n} y_i^2 w_i + c\sum_{i=0}^{n} y_i w_i &= \sum_{i=0}^{n} y_i Z_i w_i \\
a\sum_{i=0}^{n} x_i^2 w_i + b\sum_{i=0}^{n} y_i w_i + c\sum_{i=0}^{n} w_i &= \sum_{i=0}^{n} Z_i w_i.
\end{aligned}$$

Rewriting in matrix form,

$$a\begin{bmatrix} \sum x_i^2 w_i \\ \sum x_i y_i w_i \\ \sum x_i^2 w_i \end{bmatrix} + b\begin{bmatrix} \sum x_i^2 y_i w_i \\ \sum y_i^2 w_i \\ \sum y_i w_i \end{bmatrix} + c\begin{bmatrix} \sum x_i w_i \\ \sum y_i w_i \\ \sum w_i \end{bmatrix} = \begin{bmatrix} \sum x_i Z_i w_i \\ \sum x_i Z_i w_i \\ \sum Z_i w_i \end{bmatrix},$$

which can be written:

$$a\vec{a} + b\vec{b} + c\vec{c} = \vec{d},$$

or

$$\left[\, \vec{a} \mid \vec{b} \mid \vec{c} \,\right] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \vec{d}$$

$$L\vec{p} = \vec{d}.$$

Finally, solving for the parameters of the plane,

$$\vec{p} = L^{-1}\vec{d}.$$

The weights used in equation 4.1 are derived from the magnitude of the variance vector:

$$v = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

as follows:

$$w_i = e^{-\gamma v} \tag{4.2}$$

where γ is is a scaling factor that determines how much influence the weights will have. The variance in the position of the foot is approximated as the joint error (due to gear backlash, discretization in the low-level controllers, and accuracy of the position sensors) multiplied by the Jacobian of the leg:

$$\vec{v} = \vec{\Delta x} = J\vec{\Delta\theta}$$

4.3.1 Results

The facet mapping scheme outlined above, was tested on a number of canonical terrain types. The probing pattern used in evaluating the system is shown in Figure 4.4. The perspective of Figure 4.4 is looking down $\hat{z}$ at the top of the leg and the terrain probe points. A single probe point consists of lowering the foot along $\hat{z}$ until a threshold force is reached and then recording the position of the contact point.

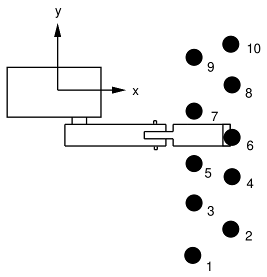

Figure 4.4. Probing Pattern Looking Down $\hat{z}$ (top of leg shown).

Figures 4.5 through 4.7 show surfaces incrementally fit to multiple probe points. The box around each point shows the approximate configuration-dependent variance of the point's position. The vertical projecting lines represent the normal measured at each point. Planes are mapped in 3 dimensions and shown from two different perspectives. Notice that position data is fairly accurate, while measured normal data is not. This is due to the inherently high friction coefficient of the terrain which prevents accurate resolution of surface normals.

In the results of Figures 4.5 through 4.7, new points are checked to see if the point statistically predicts the facet being formed. If the variance of the point does

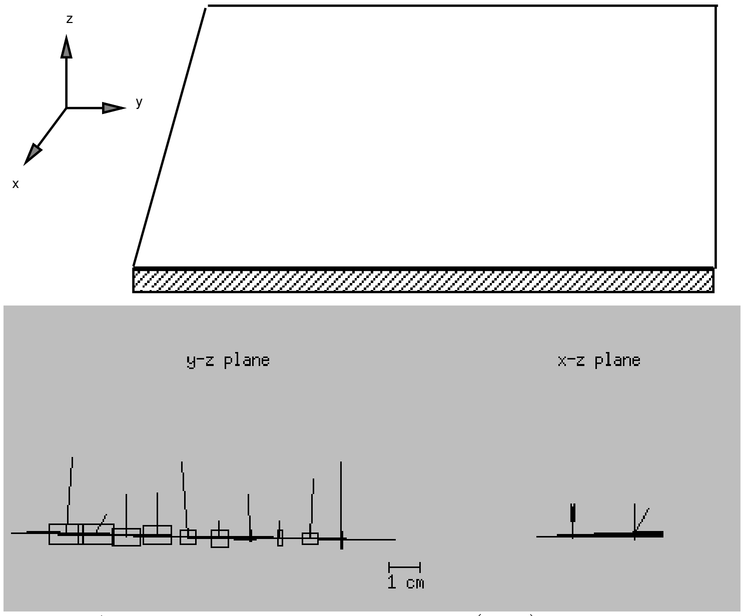

Figure 4.5. Actual Flat Level Terrain (top) and the Resulting Mapped Terrain (bottom).

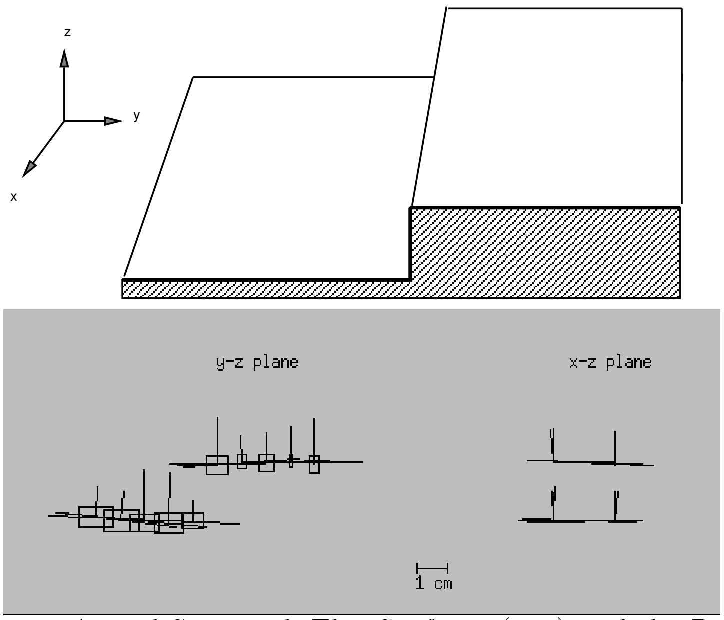

Figure 4.6. Actual Step with Flat Surfaces (top) and the Resulting Mapped Terrain (bottom).

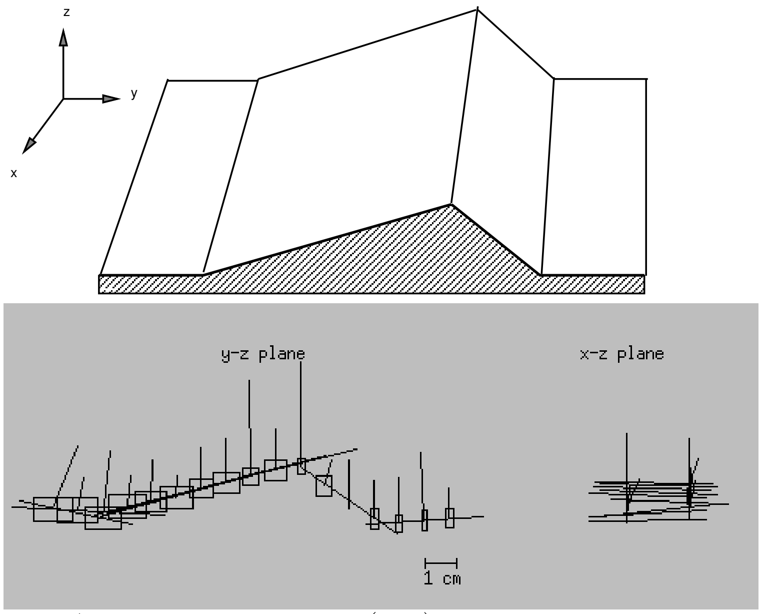

Figure 4.7. Actual Ramp Terrain (top) and the Resulting Mapped Terrain (bottom).

not contain the facet, then at least two additional points are taken, and a new facet is mapped.

An alternative method is to either include all past points when fitting a new facet, or to maintain only a finite history of points (greater than 3), and use them in the regression. This results in essentially running a low-pass spatial filter on the terrain data, as shown in Figure 4.8 for the step case. Here, a finite history of four points was used. Notice how the sharp edge is smoothed by the averaging nature of the least-square regression.

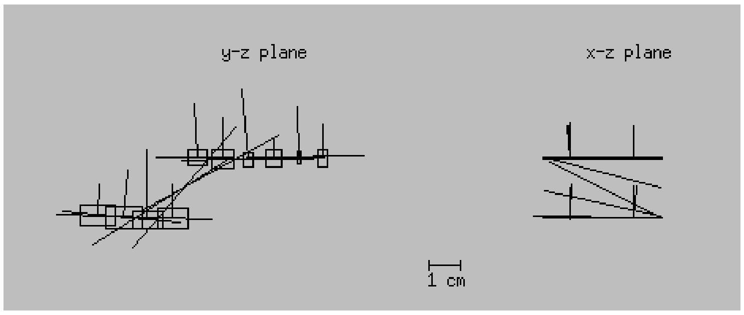

Figure 4.8. Mapping Terrain Using a Finite History Least-Square Linear Regression.

Since the configuration-dependent variances vary smoothly along $\hat{x}$, as can be seen in Figure 4.5, data points in proximity to one another do not get distinguished significantly in terms of their influence on the least-square regression (determined by equation 4.2). However, when large regions of many points are used to fit a single facet, as with areas of strictly flat terrain, then the weighting scheme has more influence; the facet will be less distorted. In addition, if the configuration of the robot were such that variances of points in a local region were significantly different, then even the smaller facets would benefit from less distortion caused by outliers.

When probing a region of terrain, it will not always be the case that the progression of points will correspond to the gradient of the terrain, as in Figures 4.6 and 4.7. In addition, it may not be convenient to take successive probe points along a line like in Figure 4.4. In these cases, the data can be represented in parameter space instead of Cartesian space, where peaks in a Hough transform signify different facets contained in the data. For example, Figure 4.9 shows a hough transform for the step geometry of Figure 4.6. To generate this density plot, 17 probe points were taken. The facet parameters used were the spherical coordinates, ρ, θ, ϕ, of the normal to the facet. These parameters uniquely define infinite planes in 3D. They are determined as shown in Figure 4.10. In Figure 4.9, ρ is plotted on the horizontal axis and ϕ on the vertical axis. Notice the bimodal distribution indicating the existence of the two different facets.

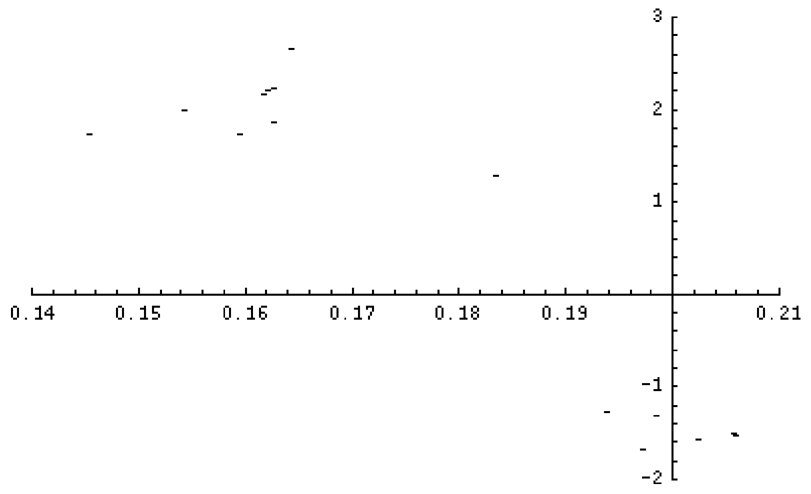

Figure 4.9. Hough Transform on Sample Probe Data (ρ is plotted horizontally and ϕ is plotted vertically).

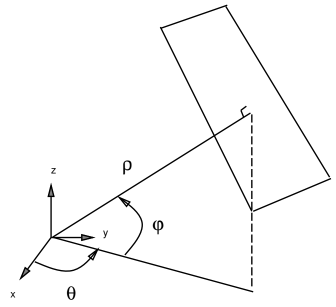

Figure 4.10. Determining the Spherical Parameters of a Facet's Normal.

Chapter 5

Walking on Irregular Terrain

All previous work with Thing has concentrated on locomotion over smooth terrain, as described in chapter 3. However, the control basis approach, was designed to be flexible across task domains. Solutions to tasks are not defined in terms of geometric foot placements or low-level behaviors but rather as stability, kinematic conditioning, and navigational objectives. These objectives do not change when walking on irregular, instead of flat, terrain.

5.1 Control Composition for Irregular Terrain

The control composition shown in Figure 3.3 was built by identifying convergence events that would result in stable locomotion on flat terrain. The state transitions (caused by controller convergence) are not *guaranteed* to occur, but *do* in the case of flat terrain. When the robot was placed on irregular terrain, it failed to assert all of the predicates necessary for state transition (due to the constituent controllers not converging), causing the robot to halt or—if the transition was allowed to occur anyway—become unstable. New performance criteria needed to be accounted for in the composition.

By identifying behaviors of the machine that fail and global characteristics of the performance that need improvement, changes to the composition can be postulated. If modifications are made within the frame-work of legal composition combinations, given the system and task constraints, then the controller can be guaranteed to run correctly; although, it will not necessarily achieve the desired task—that is up to the composition engineer. Walking over irregular terrain can be accomplished using the *same* control basis as for flat terrain with some slight modifications to the control composition.

5.1.1 Terrain Probing

When tested on irregular terrain, the robot exhibited a number of deficiencies. Perhaps the most salient was the need to sense the terrain. As discussed in chapter 4, local planar facets can be mapped by probing the terrain. This facet probing was added to the contact controller, which chooses footholds in order to achieve stable stances. The controller positions the feet based on the locally mapped facet and therefore depends on the probing for accurate facet parameters.

Most of the testing of Thing was done on a piece-wise planar terrain that consisted of discrete vertical steps. The step edges posed problems in that often (more often than you might think) a foot-hold would end up right on an edge. This type of "toe-hold" would inevitable slip off later in the gait, as the weight of the robot was shifted around. To overcome this problem, edges were treated as obstacles and path planar was employed to guide the foot away from the edge based on a harmonic gradient. This correction in foot placement was "subject to" the desired stable stance. For example, in state (1) of Figure 3.3, the controller hierarchy was changed to:

$$\left(\Phi_2 \frac{3}{\underline{3}} \triangleleft \Phi_0 \frac{3}{\underline{3}} \triangleleft \Phi_1 \frac{0,1,3}{\underline{3}} \right)$$

Notice that the posture controller is still included, but now it runs "subject to" the contact controller *and* the path controller. This is a different use of the path controller than in states (3) and (7), where it is directing the body of the robot along a harmonic gradient that avoids large obstacles. In this case, it is being used to direct only the placement of a single leg.

5.1.2 Z-Posture Control

The next most significant addition to the system was vertical body posture control. As the feet are placed higher or lower with respect to the body due to terrain inclination or declination, workspace is reduced and eventually the legs are at there joint limits, as shown in Figure 5.1.

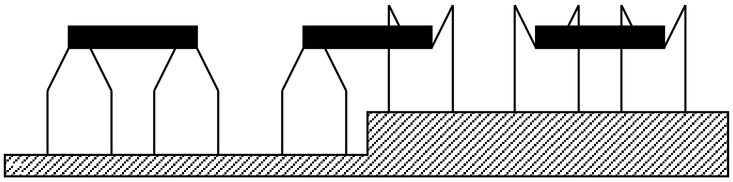

Figure 5.1. Effect of Neglecting Body Posture in the Z-Direction .

In this case, activating the posture controller with the z position of the center-of-mass of the body as the output resource solves the problem: $\Phi_2{}_{\underline{z}}^{\underline{0},1,2,3}$. Now, body posture is adjusted vertically in order to maximize workspace in the legs. Notice that, like the activation of the posture controller with center-of-mass *rotation* as the output, $\Phi_2{}_{\underline{\varphi}}^{\underline{0},1,2,3}$, the output of this new activation is orthogonal to that of the other controller activations and, therefore, can operate without the “subject to” constraint. This allows the robot to adjust its height at the same time as all the other controllers are running. So, for example, as a leg is lowered below the currently mapped facet, the body will be lowered, effectively giving the leg a longer reach.

To solve this same problem Celaya and Porta [11] designed a new behavior, $\beta - balance$, for their behavior-based walker. This new behavior had to be carefully designed not to interfere adversely with the other behaviors controlling the robot. When using a control basis, the capability of the system is already built into the basis controllers. All that need be done is to discover and activate the particular instance of the controller and necessary input/output resources that solves the problem.

5.1.3 Foot Drag

Failure of controllers to converge normally occurs when legs run into kinematic constraints (*i.e.* joint limits). The most common points of failure were in states (0) and (4) of Figure 3.3. In these states, the body is propelled forward, and stability must be achieved by servoing the center-of-mass of the robot into the next stable tripod, requiring significant motion in all legs; whereas in the other states (particularly (1) and (5)), at most a single leg must be moved to achieve stability. For this reason, most of the discussion in this section and section 5.1.4 will concentrate on states (0) and (4).

The posture controller in every state attempts to maintain the robot's joints away from their joint limits; however, this is not always possible on unknown irregular terrain. A scalar posture metric was used to evaluate the kinematic conditioning of the robot. The metric used was:

$$m = \prod_{i=0}^{i<4} \left[p_i \prod_{j=0}^{j<3} \cos(\theta_{\epsilon_{ij}}) \right] \tag{5.1}$$

where $\theta_{\epsilon_{ij}}$ is the position of the j^{th} joint of leg i between its joint limits scaled to be $-\pi \leq \theta_\epsilon \leq \pi$, and p_i is the *manipulability* of leg i, a common posture metric derived from the Jacobian, J, as:

$$p_i = \sqrt{det J_i J_i^T}$$

The posture metric m is large when all of the robot's joints are far from their joint limits (measured by the cosine term in equation 5.1, and the configuration of the leg is such that it can execute large velocities in all directions (measured by the manipulability term, p). Figure 5.2 shows a running evaluation of this metric while walking on flat and irregular terrain. The data was collected over two gait cycles in the middle of a larger walking segment. The irregular terrain was as shown in Figure 5.5.

Clearly kinematic conditioning is better on flat terrain than on irregular terrain, as expected. Since posture is already being explicitly optimized within the constraints of the other controllers, it is not clear how additional instantiations of the posture controller (with different resource combinations) within the existing states would improve performance.

One solution is to allow joints to move as close to the desired location as possible and then stop. This, however, results in internal force build up in the links and foot slippage. The foot slippage can be tolerated if it is explicitly incorporated

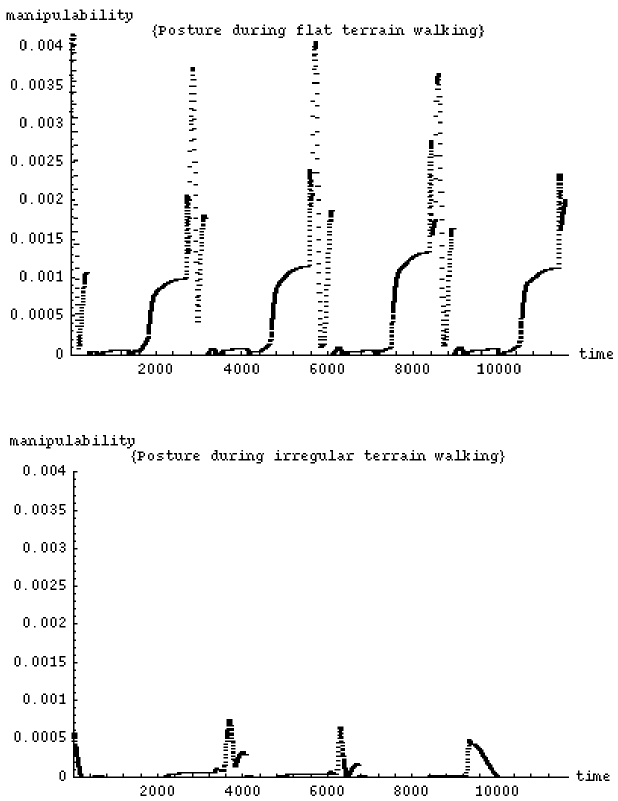

Figure 5.2. Body Posture During Flat and Irregular Terrain Walking.

into the controller. One way to do this is through the "subject to" constraint. For example, in state (4) of Figure 3.3, the contact controller, Φ_1, is moving the center-of-mass into a stable three-legged stance on legs $1, 2$ and 3—moving away from leg 0. Often leg 0 runs into its joint limits since it was placed furthest in the past (leg 0 is in the air, being posture-controlled, in state (5)), preventing the center-of-mass from reaching the supporting triangle, as shown in Figure 5.3a.

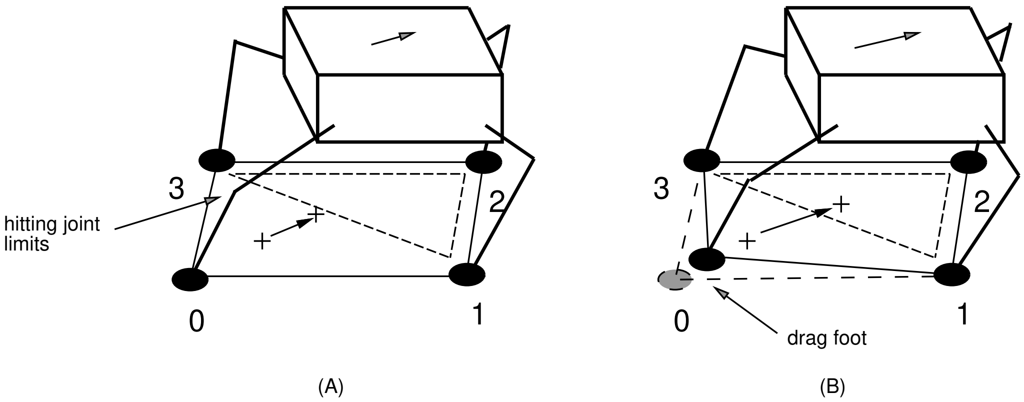

Figure 5.3. (A) Joint Limits Preventing Stable Stance. (B) Dragging Leg 0 Allows Stable Stance.

If leg 0 is allowed to be *dragged*, then the stable tripod can be reached. This can be governed by the composition hierarchy:

$$\left(\Phi_1 \frac{1,2,3}{\underline{x,y}} \triangleleft \Phi_1 \frac{0,1,2,3}{\underline{x,y}} \right)$$

The first instance of Φ_1 drives the robot into the three legged stance, possibly dragging leg 0, while the second instance makes sure the robot is stable during the move and slippage of leg 0. Note, state (0) can benefit in the same way, this time with leg 3 being dragged. Adding the above changes dramatically improved performance on irregular terrain.

5.1.4 Alternative State Transitions

In some cases, allowing the trailing foot to slip still does not allow stability to be achieved. In these cases, new state transitions can solve the problem. For example, $\neg p_1$ (failed convergence of controller Φ_1) in state (4) can cause a transition to an alternative state (4.1), as shown in Figure 5.4. State (4.1) establishes a stable stance on legs $0, 2, 3$ by moving the center of mass *backwards*, and then state (4.2) places leg 1 in order to achieve the stable $1, 2, 3$ stance that state (4) was not able to achieve. Then the gait proceeds as before by transitioning to state (5). The same type of alternative state transitions can be added coming from state (0).

The final gait diagram, including terrain probing, z-posture control, foot dragging, and alternative state transitions, for walking on irregular terrain is shown in Figure 5.4.

5.2 Walking Experiment on Irregular Terrain

Figure 5.5 shows a demonstration of the gait in Figure 5.4. Steps (verticals painted white) are $1cm$ high, and there are a total of five of them. The tops of the steps are flat horizontal planes, however, the step edges are at random angles.

At different times during this walking segment, the legs were spanning three steps—more than 50% of the robot's maximum step height. The entire experiment took approximately 13 minutes with the walking gait generated on-line by a Sun SPARC 10 workstation. In multiple runs, when system failure occurred, it was most often due to feet slipping off of edges. This was caused in part by the $1.3cm$ diameter flat feet being modeled as point contacts by the controllers.

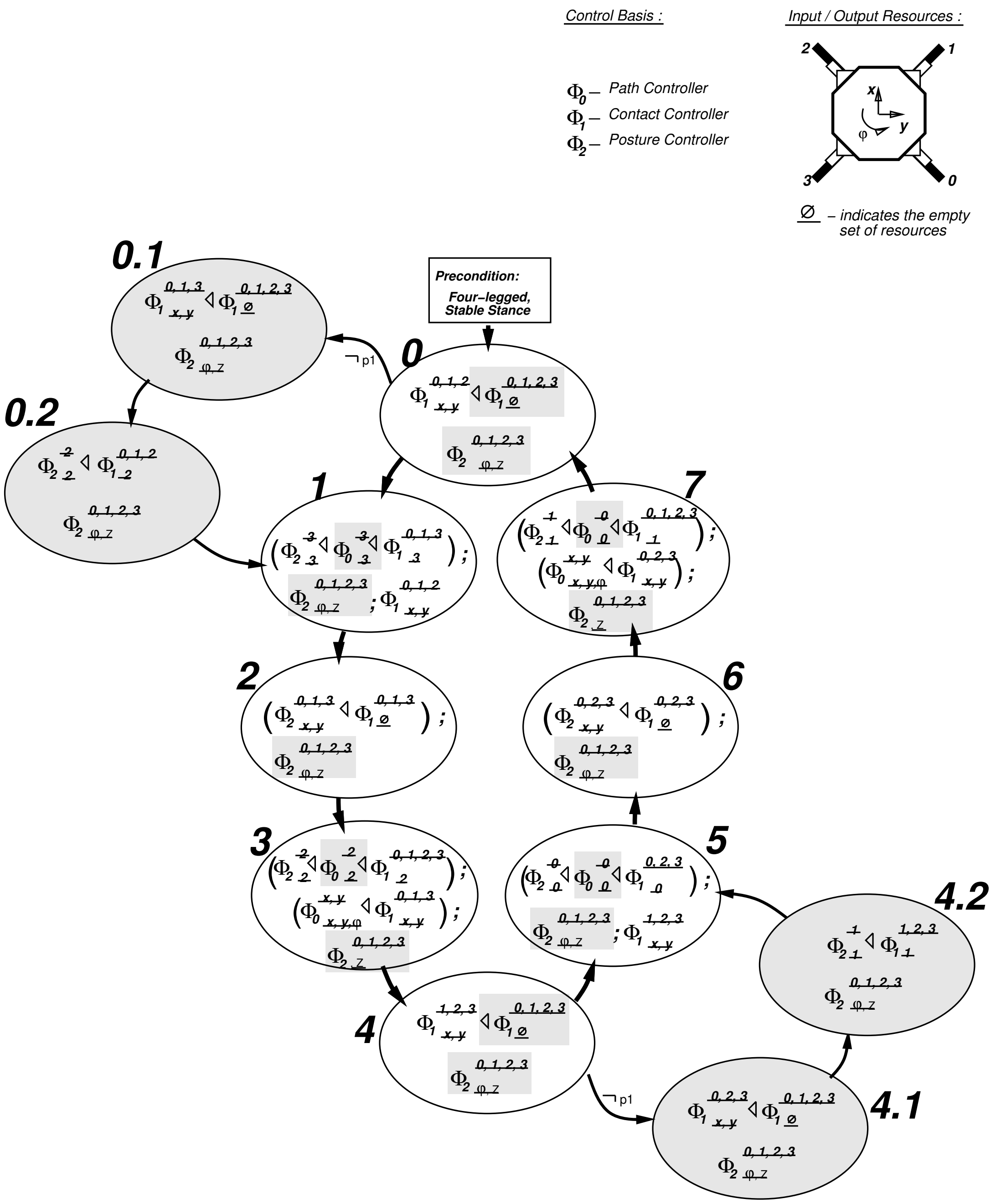

Figure 5.4. Control Gait for Walking on Irregular Terrain. Controller instantiations that are new or modified from the flat terrain gait are highlighted.

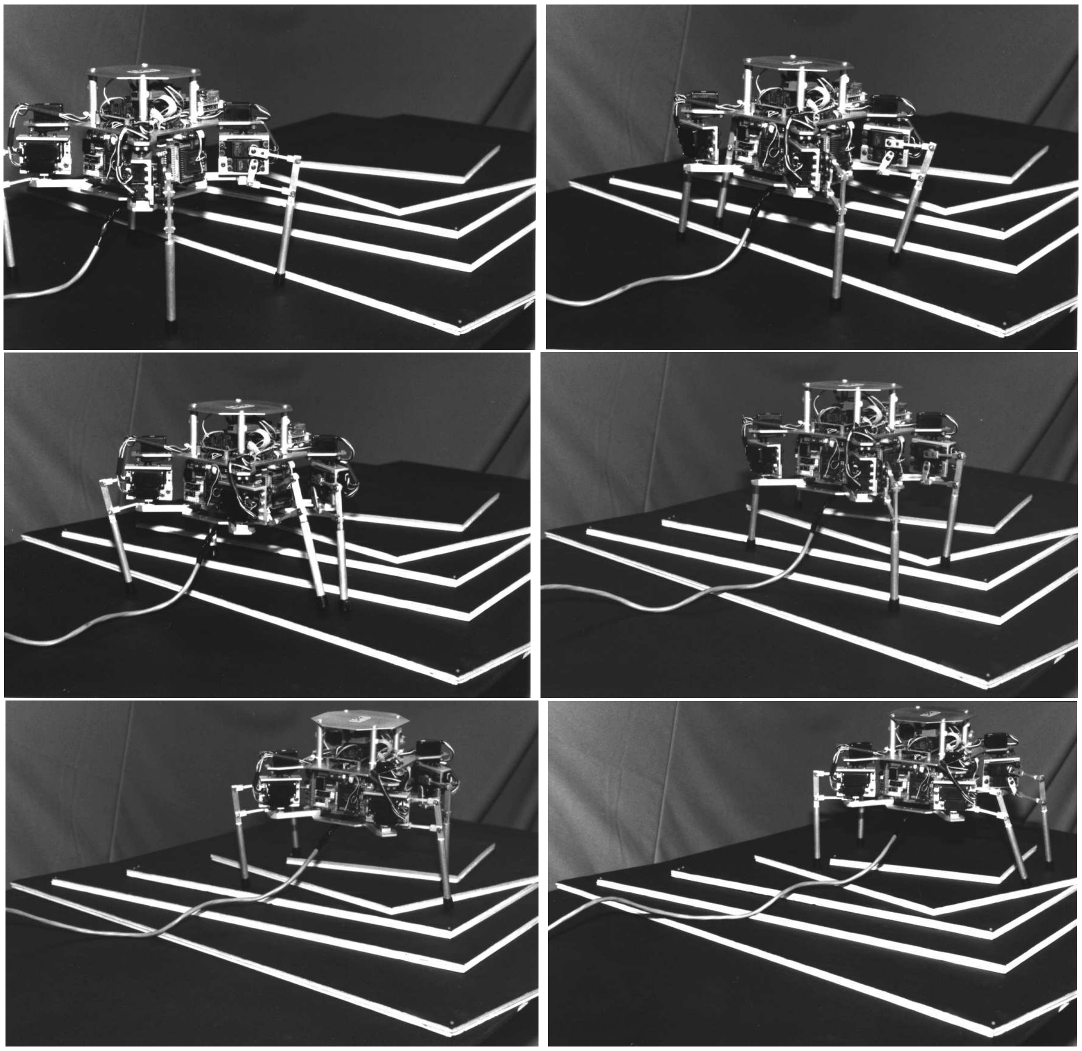

Figure 5.5. A Demonstration of Walking over Irregular Terrain. Steps (white) are $1cm$ high.

Chapter 6

Conclusion

Despite a continuous stream of research in walking robots, the state of the art leaves much to be desired. A large portion of research has concentrated on geometric aspects of walking that in order to be of practical use would require more accurate sensing and more powerful computers (which even when they arrive may still be insufficient). Rather than trying to squeeze the natural terrain that our robot encounters into a pre-determined form that we can develop gaits for, controllers should be developed that can interact with and adapt to real terrain in real-time. Uncertainties should not cause a gait to become incorrect and useless, but should result in an incremental modification to the gait.

Behavior-based methods attempt to piece together gaits in real-time, as suggested above. Unfortunately, such controllers can not be guaranteed to find a solution. In addition, the underlying interaction of behaviors is difficult to understand and predict due to the *ad hoc* nature of the design of such systems.

The control-basis approach attempts to take the best of both worlds by generating gaits on-line that can be guaranteed to converge within the capabilities of the underlying controllers and robot hardware.

In the work presented in this thesis, a control basis previously used for legged traversal of flat terrain was used for traversing irregular terrain. To achieve this, only minor changes in the control composition were made—the control *basis* was not changed at all.

Force sensors were added to Thing to allow it to sense the terrain and build local maps. Posture control of the body in the z-direction, allowing trailing feet to be dragged, and providing alternative state transitions provided Thing with the ability to climb over a piece-wise planar terrain.

To the author's knowledge *nobody* has been successful in building a system capable of what animals do so easily—autonomously traversing natural irregular terrain in a robust and flexible manner. The solution must lie in a combination of advances in hardware, reflexive behaviors, sensor interpretation, planning, and the composition of all of these pieces into a single system.

6.1 Future Work

In the work presented here, walking behavior was observed, and when deficiencies in the control composition were identified, it was modified. This type of "manual" composition design is not really necessary since it can be formulated as a machine learning problem. The control basis contains all of the capability of the system and combinations of controllers must only observe fundamental constraints of the system. The learning would, then, effectively conduct a search over all possible combinations, eventually finding a set of controller instances, resources, and states that achieve the task in the most successful manner. This may allow the robot to traverse even more severe terrain.

Bibliography

[1]H. Adachi, N. Koyachi, T. Nakamura, and E. Nakano. Adaptive gait for quadruped walking robot using force sensor. In *Proc. Int. Aut. Sys.*, pages 54–63, Pittsburg, PA, Feb 1993.

[2]C. Angle. Ghengis, a six legged autonomous walking robot. Undergraduate thesis, dept of EE and CS, MIT, March 1989.

[3]C. Angle. Design of an artificial creature. Masters thesis, dept of EE and CS, MIT, June 1991.

[4]K. Arikawa and S. Hirose. Study of walking robot for 3 dimensional terrain. In *ICRA*. IEEE, June 1995.

[5]A. Bicchi, J. Salisbury, and D. Brock. Contact sensing from force measurements. *Int. J. Robotics Res.*, 12(3):249–262, 1993.

[6]E. Bizzi. *"Central and Peripheral Mechanisms in Motor Control" in Stelmach and Requin (eds) Tutorials in Motor Behavior.* North-Holland, Amsterdam, 1980.

[7]V. Braitenberg. *Vehicles–Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA, 1984.

[8]R. A. Brooks. A robust layered control system for a mobile robot. *ICRA*, 2(1):14–23, March 1986.

[9]R. A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. *Neural Computation*, 1(2):355–363, 1989.

[10]D. P. C. Klein, K. Olson. Use of force and attitude sensors for locomotion of a legged vehicle over irregular terrain. *Int. J. Robotics Res.*, 2(2), 1983.

[11]E. Celaya and J. M. Porta. Force-based control of a six-legged robot on abrupt terrain using the subsumption architecture. In *Int. Conf. on Adv. Rob.*, Sant Feliu de Guixols, Spain, 1995.

[12]J. A. Coelho Jr. and R. A. Grupen. Effective multifingered grasp synthesis. Technical Report 94-12, CMPSCI Dept., Univ. of Mass., Amherst, February 1994.

[13]C. I. Connolly and R. A. Grupen. The applications of harmonic functions to robotics. *J. Robotic Sys.*, 10(7):931–946, October 1993.

[14]P. de Santos and M. Jimenez. Generation of discontinuous gaits for quadruped walking vehicles. *J. Robotic Sys.*, 12(9):599–611, 1995.

[15]C. Ferrell. Robust agent control fo an autonomous robot with many sensors and actuators. Masters thesis, dept of EE and CS, MIT, May 1993.

[16]R. A. Grupen, M. Huber, J. A. Coelho Jr., and K. Souccar. A basis for distributed control representation for manipulation tasks. *IEEE Expert*, 10(2):9–14, April 1995.

[17]M. Guihard, P. Gorce, J. G. Fontaine, and N. M'Sirdi. A solution to control the dynamic behavior of a pneumatic quadruped robot. In *ICRA*, pages 1000–1005, Nagoya, Japan, May 1995. IEEE.

[18]S. Hirose. A study of design and control of a quadruped walking vehicle. *Int. J. Robotics Res.*, 3(2):113–133, 1984.

[19]S. Hirose, K. Yoneda, R. Furuya, and T. Takagi. Dynamic and static fusion control of quadruped walking vehicle. In *International Conference on Intelligent Robots and Systems*, pages 199–204, 1989.

[20]M. Huber and R. A. Grupen. 2d contact detection and localization using proprioceptive information. *IEEE Trans. Robotics Automat.*, 10(1):23–33, 1994.

[21]M. Huber, W. S. MacDonald, and R. A. Grupen. A control basis for multilegged walking. In *ICRA*, volume 4, pages 2988–2993, Minneapolis, MN, April 1996. IEEE.

[22]K. Jeong, T. Yang, and J. Oh. A study on the support pattern of a quadruped walking robot for aperiodic motion. In *Proc. IROS*, pages 308–313, Pittsburgh, PA, August 1995.

[23]C. Klein and R. Briggs. Use of compliance in the control of legged vehicles. *IEEE Trans. Syst. Man Cyber.*, 10(7):393–400, 1980.

[24]W. MacDonald. Design and implementation of a four-legged walking robot. BS thesis, ECE dept., Univ. of Mass., Amherst, May 1994.

[25]R. McGhee and G. Iswandhi. Adaptive locomotion of a multilegged robot over rough terrain. *IEEE Trans. Syst. Man Cyber.*, 9(4):176, 1979.

[26]D. Messuri and C. Klein. Automatic body regulation for maintaining stability of a legged vehicle during rough terrain locomotion. *IEEE Trans. Robotics Automat.*, RA-1(3):132–141, 1985.

[27]P. Nagy, W. Whittaker, and S. Desa. A walking prescription for statically-stable walkers based on walker/terrain interaction. In *ICRA*, Nice, France, 1992. IEEE.

[28]D. Pack and H. Kang. An omnidirectional gait control using a graph search method for a quadruped walking robot. In *ICRA*, volume 1, pages 988–993, Nagoya, Japan, May 1995. IEEE.

[29]M. Raibert. *Legged Robots that Balance.* MIT Press, Cambridge, MA, 1986.

[30]S.-M. Song and K. J. Waldron. *Machines That Walk: The Adaptive Suspension Vehicle*. MIT Press, Cambridge, MA, 1989.

[31]S. Sun. A theoretical study of gaits for legged locomotion system. Ph.d. dis., Ohio State Univ., Columbus, Ohio, 1974.

[32]A. Takanishi, J. Yamaguchi, and M. Iwata. Dynamic quadruped walking stabilized with trunk motion. In *International Conference on Intelligent Robots and Systems*, pages 165–172, Pittsburgh, PA, Aug 1995. IEEE.

[33]D. Wettergreen, H. Pangels, and J. Bares. Behavior-based gait execution for the dante ii walking robot. In *International Conference on Intelligent Robots and Systems*, pages 274–279, Pittsburgh, PA, August 1995. IEEE.

[34]K. Yoneda and S. Hirose. Dynamic and static fusion of a quadruped walking vehicle on a winding path. In *ICRA*, pages 143–148, Nice, France, May 1992. IEEE.

[35]K. Yoneda, H. Iiayama, and S. Hirose. Sky-hook suspension control of a quadruped walking vehicle. In *ICRA*, pages 999–1004, San Diego, CA, May 1994. IEEE.