A MOVEMENT PLANNING HEURISTIC PACKAGE FOR QUADRUPED AMBULATION OVER ROUGH TERRAIN

Kenneth M. Buckland
B. Sc. (Electrical Engineering) University of Calgary

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Master of Applied Science
in
THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
August 1989

© Kenneth M. Buckland, 1989
In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering
The University of British Columbia
1956 Main Mall
Vancouver, Canada

Date: Oct 11/89
Abstract

Legged walking machines offer a number of advantages over conventional wheeled or tracked vehicles. They can reach terrain that is inaccessible to conventional vehicles, and they leave fewer traces of their crossing behind on terrain that has not been prepared for transportation. This thesis details the development of a heuristic package that can be used to determine the movement plan of a longitudinally oriented quadruped walking vehicle ambulating over rough terrain. An investigation of walking machine research to date revealed that such work had not been done for quadrupeds that adjust their attitude to follow the terrain. This terrain following strategy was employed in the development of these heuristics because it provides superior movement capabilities on sloped terrain. The heuristics were tested extensively on a variety of simulated terrain conditions. These conditions consisted of different combinations of underlying terrain attitudes and terrain roughness levels. The heuristics performed well during the simulations, and the simulated machine responded as the theoretical analysis had indicated.

During the course of this work, a number of new ideas related to quadruped ambulation were developed. These included how to generate the overall movement plan of a quadruped walking machine, how foothold locations can be guided to best facilitate smooth machine turning, how the speed of a walking machine can be maintained at consistent levels, and how to prevent any possibility of leg collisions. In addition, in-depth analyses were made on how stride length affects quadruped machine speed, what conditions force such machines to halt their forward movement, and how a practical machine should be designed.
# Table of Contents

Abstract ii  
List of Figures viii  
List of Variables x  
Acknowledgements xii  

1 Introduction 1  
1.1 Statically and Dynamically Stable Walking 1  
1.2 Hexapods, Quadrupeds, and Quasi-Dynamic Walking 2  
1.3 The Purpose of This Study 3  
1.4 The Structure of This Paper 4  

2 Walking Machine Research Efforts 5  
2.1 Work at Ohio State University 5  
2.1.1 Foot placement strategies 6  
2.1.2 Free gaits and wave gaits 7  
2.1.3 Body movement planning 7  
2.2 Work at the Tokyo Institute of Technology 8  
2.2.1 Body movement planning 9  
2.3 “Approximate Straight Line Mechanism” Work 10  
2.4 Work at Odetics 11  
2.5 Other Work 12  
2.6 Summary 13  

3 General Characteristics of Quadruped Ambulation 14  
3.1 The Walking Machine Design Used in the Study 14  
3.2 The Quadruped Crawl 15
### 3.2.1 Description of the quadruped crawl

### 3.2.2 Difficulties with free gaits

### 3.3 “Follow The Leader” Gaits

### 3.4 Stability Triangles

### 3.5 Control of Movement

### 3.6 Speed Considerations

#### 3.6.1 The ideal gait

#### 3.6.2 The effect of stability margins

#### 3.6.3 The effect of swing overhead time

#### 3.6.4 The relationship between swing time and swing distance

#### 3.6.5 The effect of altering the machine stance width

### 4 Heuristic Specification

#### 4.1 The Overall Heuristic

#### 4.2 Coordinate System Specification

#### 4.3 The Orientation Vectors

#### 4.4 The Assess Movement Heuristic

#### 4.5 The Calculate Predicted Body Orientation Vectors Heuristic

#### 4.6 The Update Swing Leg Heuristic

#### 4.7 The Half-Cycle Movement Planning Heuristic

##### 4.7.1 Trajectory planning

##### 4.7.2 Attitude and orientation planning

#### 4.8 The Calculate Target Angles Heuristic

#### 4.9 The Foothold Determination Heuristic

##### 4.9.1 Rear foot placement

##### 4.9.2 Front foot placement general requirements

##### 4.9.3 Front foot placement strategies

##### 4.9.4 Determining the placement plane

##### 4.9.5 Determining the search starting point

##### 4.9.6 Determining the maximum forward placement distance

##### 4.9.7 Scanning system following of the placement rules
4.10 The Check Stability Heuristic ........................................... 53
4.11 The Speed Calculation Heuristic ........................................ 54
  4.11.1 Determining the termination point ............................... 55
  4.11.2 Speed calculation during the front leg swing phase ........... 56
  4.11.3 Speed calculation during the rear leg swing phase .......... 56
  4.11.4 Speed calculation during the creeping phase .................. 59
  4.11.5 Speed control strategies ........................................... 61
4.12 The Swing Leg Kinematics Heuristic ..................................... 62
  4.12.1 Determining the placement state .................................. 62
  4.12.2 Hovering ............................................................. 64
  4.12.3 Calculating the joint angles with a valid placement state .. 64
4.13 The Approximate Step Times Remaining Heuristic .................... 66
4.14 The Adjust Attitude Heuristic ........................................... 66
  4.14.1 Preventing failures by making footholds accessible .......... 67
  4.14.2 Adjusting the attitude to facilitate front foot placement ... 67
  4.14.3 Adjusting the rotational orientation to facilitate rear foot placement ... 69
  4.14.4 Halting conditions and the applicability of the Adjust Attitude heuristic .... 70
  4.14.5 Machine failure summary ............................................ 73
4.15 The Body Kinematics Heuristic ........................................... 74
4.16 The Prevent Leg Collision Heuristic .................................... 75
  4.16.1 Preventing contact when both feet were placed ............... 75
  4.16.2 Preventing contact during the rear leg swing .................. 77

5 Practicalities of Quadruped Design ......................................... 81
  5.1 Machine Dimensions .................................................... 81
    5.1.1 Leg dimensions ............................................... 81
    5.1.2 Platform dimensions ............................................ 82
    5.1.3 Model dimensions .............................................. 83
  5.2 Angular Range Limitations ............................................ 85
    5.2.1 Factors in the angular range limitation ....................... 85
    5.2.2 Angular range limitations of the model ....................... 85
List of Figures

3.1 The Structure of the Machine Used in the Study ............................................. 15
3.2 The Quadruped Crawl ..................................................................................... 17
3.3 Linked Stability Triangles ............................................................................... 20
3.4 The Creeping Distances That Result From Addition of a Stability Margin .......... 26
3.5 Plot of $v = \frac{1}{k \cdot \frac{1-2c/s}{3+2c/s}}$ ................................................................ 28
3.6 Swinging of Legs About Their Shoulder ....................................................... 29
3.7 Plot of $\frac{dx}{dt} = \frac{1}{k \cdot \frac{1}{1+(1/RE)}}^3$ ......................................................... 30
3.8 The Effects of Stance Width on the Creeping Distance ................................... 32

4.1 The Coordinate System Used .......................................................................... 37
4.2 Conversion of Joint Angles Into Shoulder to Foot Offset Distances ................. 40
4.3 Compensation of the Lateral Foothold Spacing Under Roll Conditions .......... 46
4.4 Minimum Stability Triangle Size ..................................................................... 49
4.5 Stability Triangle Sizing During Machine Turning ........................................... 52
4.6 Positioning Relationships When the Rear Foot is Behind the Front Foot .......... 59
4.7 Calculation of the Vertical Angle .................................................................... 75
4.8 Configuration for Leg Contact Prevention ..................................................... 76
4.9 Rear Leg Swing Contact Prevention ............................................................... 80

5.1 The Effects of Differing Leg Segment Lengths ................................................ 82
5.2 The Kaiser Spyder X5M ................................................................................ 84

6.1 Foothold Perturbation Determination ............................................................. 97
6.2 Performance During the No Roughness Benchmark Simulation ....................... 101
6.3 The Effects on Completion Rate of Varying the Terrain Roughness ............... 103
6.4 The Effects on Distance Travelled of Varying the Terrain Roughness ............ 104
6.5 The Effects on Speed of Varying the Terrain Roughness ................................ 105
6.6 The Effects on the Tilt and Roll Errors of Varying the Terrain Roughness ....... 106
6.7 The Effects on Completion Rate of Varying the Terrain Tilt ........................ 108
6.8 The Probability of Adjustments and Halts for Different Terrain Tilts ............. 109
6.9 The Effects on Completion Rate of Varying the Terrain Roll ........................ 110
6.10 The Probability of Adjustments and Halts for Different Terrain Rolls ............. 111
6.11 The Effects on Completion Rate of Varying the Terrain Tilt and Roll Together ... 112
6.12 The Probability of Adjustments and Halts for Different Terrain Tilts and Rolls ... 113
6.13 The Effects on Speed of Varying the Terrain Tilt and Roll Together ............... 114
6.14 The Response to Various Turning Rates Per Half-Cycle ............................ 116
6.15 The Effects on Speed of Turning ....................................................... 117
6.16 Response to Varying Slope Attitudes While Turning at 5° per Half-Cycle ........ 119
6.17 The Effects on Completion Rate of Various Terrain Tilt Changes .................. 121
6.18 The Average Number of Adjustments and Halts During Flat to Tilted Changes .... 122
6.19 The Average Number of Adjustments and Halts During Tilted to Flat Changes ... 123
6.20 The Effects on Completion Rate of Various Terrain Roll Changes .................. 124
6.21 The Average Number of Adjustments and Halts During Flat to Rolled Changes .... 125
6.22 The Average Number of Adjustments and Halts During Rolled to Flat Changes ... 126
6.23 The Relationship Between Front Foot Travel and Stride Length ........................ 129
6.24 The Effects on Completion Rate of Varying the Stride Length ....................... 130
6.25 The Effects on Speed of Varying the Stride Length .................................. 131
6.26 The Effects on Completion Rate of Varying the Stance Width ........................ 132
6.27 The Effects on Completion Rate of Varying the Stance Height ....................... 134
6.28 The Effects of Roughness on the Completion Rate of Different Terrain Conditions ... 135
List of Variables

\[ \theta \] the horizontal angle of the leg of a machine (see Figure 3.1).

\[ \phi \] the vertical angle of the leg of a machine (see Figure 3.1).

\[ \epsilon \] the extension angle of the leg of a machine (see Figure 3.1).

\[ \omega_{\text{max}} \] the maximum angular change that can occur in a joint per sample time.

\[ \psi \] the tilt angle of a machine platform (see Figure 4.1).

\[ \sigma \] the roll angle of a machine platform (see Figure 4.1).

\[ \rho \] the rotation of a machine platform.

\[ s \] the stride length.

\[ h \] the height of a machine stance.

\[ w \] the width of a machine stance.

\[ P_p \] the planned center of gravity of a machine at the end of a half-cycle (see Section 4.7).

\[ P_f \] the location of the next planned foothold.

\[ \vec{A} \] the forward orientation vector (see Section 4.3).

\[ \vec{S} \] the lateral orientation vector (see Section 4.3).

\[ \vec{N} \] the platform normal orientation vector (see Section 4.3).

\[ \vec{J} \] a vector from the center of a machine to one of its shoulders.
\( \overline{D} \) a vector from one of the shoulders of a machine to the corresponding foot.

\( d_s \) the distance between the shoulder and foot of a leg in the forward direction.

\( d_l \) the distance between the shoulder and foot of a leg in the lateral direction.

\( d_n \) the distance between the shoulder and foot of a leg in the vertical direction.

\( v \) the velocity of a machine.
Acknowledgements

I would like to thank the mountains north of Vancouver, and all those who challenged them with me. I would also like to thank the ocean. In addition, I would like to thank my incredible wife for not smashing that one last glass, for typing in some of the references, and for consistently providing a great time. I would like to thank Dr. Peter Lawrence for his guidance, and for allowing me a great deal of freedom (but not so much freedom that I never produced a thesis about which acknowledgements could be made). Finally, I would like to thank my Mom for the Mykit System 7 150-in-1 Electronic Project Kit that she gave me Christmases ago. This would be an entirely different thesis if she had given me a lump of coal like she threatened. I'd probably be a Geologist like my sister.
Over the past 25 years there has been considerable research interest in constructing artificial walking machines. Such machines would be of use in many military, industrial, and space applications, as they can cross terrain that is impassable to conventional wheeled and tracked vehicles. The range of mobility is increased with legged vehicles because they require only a limited number of footholds, not a smooth path, to facilitate their movement. One need only consider the terrain that can be reached by a mountain goat to realize the superior characteristics of legged vehicles. At present, no one in the field of walking machines would propose that these machines could access the difficult terrain of mountain goats, but there are still many areas they could reach that, without proper route preparation, conventional vehicles could not. In addition, the limited number of footholds required by walking machines results in less disturbance of the terrain that they walk over. So even though both legged and conventional vehicles may have the ability to cross a given field for example, the legged vehicle would leave far fewer traces of its crossing behind. Thus walking machines could be of value in applications where decreased environmental impact is desired.

1.1 Statically and Dynamically Stable Walking

Walking machine research has traditionally been divided into two basic fields, those that involve either statically or dynamically stable machines. Static stability refers to maintaining machine stability by continually keeping the center of mass of the machine inside a stability region defined by the convex hull of the horizontal projection of all its supporting feet (Song, 1987). If a statically stable machine stops its movement at any time, it remains standing.

Dynamic stability relies on the active modulation of leg forces to maintain balance. These machines are effectively always “falling”. Completion of their fall is prevented by the movement of their legs into new positions, and the careful application of leg forces. Dynamically stable walking machines have been designed and constructed (Raibert, 1983, Miura, 1984, Takanishi, 1984), but this type of stability was
not considered in this study. Dynamic stability, at least in the research to date, results in a very rough "ride", and is not very applicable to heavy machinery, the kind that would be of use in most practical applications.

1.2 Hexapods, Quadrupeds, and Quasi-Dynamic Walking

In the realm of statically stable machines, most of the early research was on quadruped, or four-legged, vehicles. This may have been because four-legged creatures are very familiar to us. The concern with quadrupeds continued until Bessonov and Umnov (1973) released a paper describing the superior stability characteristics of hexapod, or six-legged, vehicles. This superiority is due to the fact that the six legs can be lifted in a variety of combinations that maintain a very stable base, even when more than one leg is being lifted. Another advantage of hexapods is that they can walk with four or more legs on the ground, allowing for redundant support should one of the legs give way. It is interesting to note that insects, creatures that typically restrict their walking to statically stable gaits, always have six or more legs (if they have any at all). Conversely land mammals, animals that typically have four legs (this researcher not included), use dynamically stable gaits. Insects may make use of six-legged statically stable gaits because these gaits make more restricted requirements on their simple nervous systems. Therefore hexapod walking could be seen as a more primitive form of walking, and as such, should probably be investigated first. Possibly for this reason, and definitely for the reason of improving static stability, most of the significant statically stable walking machine research in North America and Europe concentrated on hexapods after 1973.

In the past decade, work on quadrupeds has reemerged in Japan (Hirose, 1984a and 1984b, Adachi, 1987) and Tiawan (Lee, 1986a and 1986b). The motivation for this work may be partially, as described in Hirose et. al. (1978), to decrease machine complexity. Making use of quadrupeds not only reduces the control requirements, it also decreases the number of legs which would have to be mechanically maintained in practical applications. A more significant justification for quadrupeds however, may be their greater suitability to dynamically stable operation. Work in this regard has been done by Adachi et. al. (1987). This work has not gone to the extent of replicating galloping animals in the manner of Raibert (1983), but it has extended the capabilities of statically stable machines. It has involved traditional statically stable machine designs that are in no way similar to those used by Raibert. Such designs, with the current understanding of machine walking, cannot incorporate a significant amount of
dynamically stable operation. They can however, use some dynamic balance at the “edges” of their static balance range. This could include going slightly beyond the statically stable region when a swinging foot that will restore stability is soon to be placed, or using dynamics to compensate for miscellaneous forces affecting the machine. In general, the inertia of the machine’s body is used to reduce the rigid stability constraints placed on the operation of the machine. It is in the potential for this “quasi-dynamic” walking that quadrupeds have the advantage. Four legs can more easily be controlled for this type of ambulation. Hexapods walking quasi-dynamically would require much more complex control systems to keep all of their legs functioning smoothly, and to prevent collision between the legs.

1.3 The Purpose of This Study

When quadrupeds were abandoned in 1973 as a viable walking machine solution, the work on four-legged gaits at that time had progressed only as far as to specify generally the patterns that the gaits should follow. No work at all had been done on the complexities of gait planning over rough terrain. Recent work in Japan and Taiwan has either not considered rough terrain or, in the case of Hirose et al. (1984a and 1984b), has done so in a manner that is rather restricted. The machines that were designed by Hirose et al. handled rough terrain walking in the manner of expert systems, not “terrain followers”. That is, when these machines were presented with terrain that had many inaccessible areas and varying foothold heights, they carefully determined where the best foot placement locations were. They moved slowly, chose their footholds precisely, and maintained a perfectly level attitude, regardless of terrain. This method of walking was felt by this researcher to be too restrictive in terms of adapting to the terrain. To best cross uneven terrain, a vehicle should adjust its attitude to match that of the terrain, and move on a more or less constant trajectory defined by the longitudinal axis of its body. Without such adjustments, extremes of terrain slope become unnegotiable due to the simple fact that the down hill legs get extended drastically, and the up hill ones get pulled in too close to the machine. If walking continues in this manner, leg range limits are eventually reached, and the machine must halt. The terrain following strategy was used in most of the work done on hexapods, as will be detailed in the next chapter.

The purpose of this study was to develop a collection of walking machine control heuristics that applied general terrain following hexapod walking strategies to quadrupeds. These heuristics were designed to generate an attitude adapting, longitudinally oriented gait plan that provides a smooth, consistent
Chapter 1. Introduction

"ride" over rough terrain. Such work has never been performed on quadrupeds, as the groups that have developed longitudinally oriented, terrain following gait planners have concentrated on six-legged machines. As stated previously, the advantage of having four instead of six legs, is the ease with which the former could be converted to quasi-dynamic operation when the requirements of such become better known. This study did not actually include quasi-dynamic walking strategies, its purpose was instead to do a thorough investigation of statically stable quadruped walking that would further the progress towards towards quasi-dynamic ambulation.

More specifically, this study resulted in the development of a complete quadruped movement planner consisting of a number of heuristics. The collection of heuristics was entitled, "Longitudinally Oriented Gait Heuristics for Ambulating Quadrupeds" (LOGHAQ). These heuristics did not provide dynamic control. They simply generated a gait plan that a dynamic controller could follow. It was not assumed that this plan would be followed precisely however. The heuristics completely reconstructed their plan from the current machine state every sample time to facilitate a realistic response to machine performance.

1.4 The Structure of This Paper

This paper will first discuss historical walking machine work and its relationship to the heuristics. Next the general operation of the heuristics will be described, including how they were developed from consideration of the characteristics of quadruped walking. This will be followed by more specific details of the heuristics' operation. Then general guidelines for using the heuristics on an actual machine will be discussed. In parallel with this, the development of a simulation model will be detailed. Finally, simulations performed to evaluate the heuristics will be described, and conclusions drawn.
Chapter 2

Walking Machine Research Efforts

This chapter will discuss the walking machine research that has occurred over the past 15 years. The purpose of this discussion is to provide a background for the work done in this study, and to explain how it helps further the understanding of the field. The time period of 15 years was selected as it was in this period that walking machines evolved to the point where they could handle more than smooth, flat terrains that could be easily traversed with inflexible gaits. The reason for this evolution was a combination of simple advancement based on previous work, and the availability of portable, powerful microprocessors.

2.1 Work at Ohio State University

Extensive walking machine research has been carried out at Ohio State University. O.S.U. has had the most comprehensive agenda in this field of any institution. The work at O.S.U. has centered around two machines, the O.S.U. Hexapod and its successor, the Adaptive Suspension Vehicle (ASV). The six legs of the Hexapod were powered by electric motors, and were “reptilian” in design (Klein, 1983, Messuri, 1985, McGhee, 1984). By “reptilian” it is meant that the legs had two degrees of freedom at the shoulder, and thus the upper leg segment could move freely about this joint. In addition it had a knee joint, providing another degree of freedom, that allowed the lower leg segment to swing in the vertical plane defined by the upper leg segment. In general then, the upper leg segment remained mainly horizontal, and the lower remained vertical. (Figure 3.1 illustrates four reptilian legs.) The two segments rotated in a vertical plane around the shoulder to facilitate movement. This leg construction, and in fact the machine itself, is a common design for hexapods and will be referred to henceforth as that of a “standard hexapod”. Another characteristic of both the Hexapod and standard hexapods is that they have a longitudinal leg configuration. That is, the legs are attached in two groups of three down each side of a long body. This design differs from one where the legs are spaced equally and symmetrically around a roughly circular body. The comparison is similar to that of an ant and an idealized spider.
Chapter 2. Walking Machine Research Efforts

The ASV, as stated above, was the successor to the Hexapod, and while still six-legged and longitudinally configured, it incorporated a number of advances in leg design (Waldron, 1984 and 1986). In order to increase energy efficiency, the legs used in the ASV were designed to isolate the horizontal and vertical leg forces. This allowed for the forward driving forces to be kept separate from those used to simply keep the machine standing, and decreased the overall power consumption (Waldron, 1984). The legs looked like those of a large mammal. The leg segments were both vertically oriented in a plane that ran parallel to the length of the machine’s body. There was a knee joint with one degree of freedom, a shoulder joint that allowed rotation of the leg in the plane parallel to the body, and a third joint that allowed the leg plane to tilt a limited amount away from vertical. Not only did this design result in an energy efficient stance similar to that of a large mammal, it had a narrower profile, allowing for the easier negotiation of obstacle filled terrain. Of course this narrower profile also decreased the lateral stability of the machine.

2.1.1 Foot placement strategies

While the two O.S.U. machines were somewhat different in design, they were similar in how their feet could be placed and moved relative to their bodies. The legs in both machines had three degrees of freedom. So the gaits that were developed for use on the Hexapod were easily extended and enhanced for use on the ASV. A few of the many gait strategies that were used on the two machines will be described below.

One of the first considerations in rough terrain negotiation is that of foot placement. This problem was solved, in one strategy, by letting the legs move to wherever the ideal foothold location was, and using proximity sensing to control the downward placement of the feet (Tsai, 1986). This method of placement assumes that the terrain, regardless of its roughness, has no regions were feet cannot be placed. The foot placement problem was also solved by allowing the machine operator to take over the exact placement of each foot (Messuri, 1985). Another solution involved using a vision system to locate potential footholds indicated by the operator with a laser beam (Ozguner, 1984, McGhee, 1984). The concept of “constrained working volumes” was employed to limit the area of foot placement to one that ensured machine stability (Lee, 1986c). The ideal foot placement was then defined by the point on the front edge of the working volume that the planned leg trajectory passed through. Placement locations of feet other than the front two were often determined using a “follow the leader” strategy that placed these feet in or near the footholds used by the front feet (Ozguner, 1984, McGhee, 1984). This strategy
eliminates the need to identify footholds for each foot, and allows knowledge obtained by front foot placements, such as ground firmness, to be made use of.

2.1.2 Free gaits and wave gaits

The actual order of foot placement is a major issue, and was determined in the O.S.U. machines using both the “wave gait” patterns studied extensively by McGhee (1984), and “free gait” heuristics (McGhee, 1979). The former of these involves fixed foot placement orderings that result in a high degree of vehicle stability, and often reflect animal gaits. The latter involves the continuous determination of the optimal foot to move at any time to ensure maximum stability and smooth body movement. One goal of free gaits is to keep as many feet off the ground as possible. The more feet that are off the ground, the more rapidly new support areas can be established, and the faster the machine can move. The fastest wave gait is the “alternating tripod” gait, one that involves having three legs, the maximum number, of a hexapod off the ground simultaneously.

2.1.3 Body movement planning

Numerous strategies were developed at O.S.U. to determine body movement once the foothold selections had been made. One was to define a new body centroid based on where the feet would be when those off the ground were placed (McGhee, 1984). A movement plan was then developed that would take the machine’s center of gravity to that predicted centroid. This plan, like all of those used, also reoriented the machine heading to a more appropriate direction. Another strategy was developed that determined the radius of curvature for a desired center of gravity path (Orin, 1982). The radius of curvature calculation was made based on direction specifications made by the machine operator and could, of course, be infinite. The algorithm used then supervised movement along this path. A third strategy was to determine, using relaxation techniques, an ideal body position that maximized the “energy stability margin” (Messuri, 1985). The energy stability margin is determined by the minimum energy required to tip a machine over when it is in a given state, and is possibly a better measure of stability than that normally used, the distance from the center of gravity to the nearest edge of the stability region (the horizontally projected region whose corners are defined by the placed feet). Movement of the machine into the most energy stable body position was accomplished by following a calculated maximal stability path. The energy stability technique is calculation intensive, and the high degree of stability it produces may not be necessary under the normal operating conditions of a walking machine. It would be
most useful in extreme terrain slope situations or when solid positioning is needed to do work that will subject a machine to various disturbing forces. Many other movement plans were developed at O.S.U., but those presented previously seem to be the ones that have the most general application and are the most representative of particular methods of approach.

Another aspect of motion planning that should be considered is body attitude control. Body attitude is adjusted to provide the largest possible stability margins for varying terrain attitudes. Terrain attitude was typically determined at O.S.U. by somehow averaging the elevations of the machine's feet. The averaging sometimes considered intended footholds that had been chosen, but not yet reached (Lee, 1986c). A number of strategies were adopted to match the body attitude to the perceived terrain one (Lee, 1986c, Waldron, 1984, Orin, 1976). All of these strategies included restrictions that kept the tilting of the body within reasonable ranges. Strategies that maintained a horizontal attitude were used as well. Many of the attitude controlling strategies also incorporated height control. Lessening the machine height decreases ground clearance, but increases stability when the machine body is tilted (Waldron, 1984).

In addition to the work described above, additional walking machine capabilities were developed at O.S.U. These included strategies for obstacle crossing (Waldron, 1984, Choi, 1988), turning in place (Orin, 1982), side-stepping (Orin, 1982), and foot slippage control (Wong, 1988). The work done on the Hexapod and the ASV thoroughly considers the problems of ambulation confronted by hexapod machines.

While both of the machines developed at O.S.U. were hexapods, much of the work done there was considered while developing the heuristics described in this paper. These machines were similar in two fundamental ways that allowed for translation of ideas. The machines had three degree of freedom legs, and they were longitudinally oriented. The existence of two extra legs made some of the schemes developed for these machines inappropriate, but many O.S.U. strategies were developed for use on the LOGHAQ heuristics. The references to such will be apparent through the paper. The fact of the matter is that no greater body of research exists for longitudinally oriented machine walking.

### 2.2 Work at the Tokyo Institute of Technology

Three walking machines have been developed at the Tokyo Institute of Technology. These are the KUMO, the PVII (Perambulating Vehicle II), and the TITAN-III (Hirose, 1984a and 1984b). The former of these
was an early, simple design and will not be considered. The TITAN-III was an improved version of the PVII that was capable of supporting its own weight (the PVII was supported by a cable while walking). Both machines were quadrupeds with legs that were very long relative to their body size. The legs were located at the corners of basically square bodies and could swing more or less equally in either direction. The design was therefore not a longitudinally oriented one, but of an idealized spider nature. The legs themselves were reptilian, and had a knee joint with one degree of freedom and a shoulder joint with two degrees of freedom. Like the O.S.U. Hexapod, the two segments of the legs defined a vertical plane, and that plane could swing about the shoulder joint. The upper segment was kept mostly horizontal, the lower vertical. Much of the work done on these machines was concerned with energy efficiency. The legs made use of a pantograph design that isolated the vertical and horizontal force components. This allowed the horizontal driving forces required to move the machines to be kept separate from the vertical support forces. Thus the vertical support forces could be made mechanically passive. That is, the vertical support mechanisms could be locked, and energy expenditures to support the machine thereby eliminated.

2.2.1 Body movement planning

The movement control of these two machines involved a free gait heuristic. This heuristic operated by making a movement plan based on the required general direction of movement. It then went through a thorough evaluation process to determine how this movement should be made. It carefully selected the best leg to swing at any given time to best facilitate continued smooth motion, it chose the best new foothold for that leg based on those that were available in the environment, and it calculated the best body movement path. The gait, given terrain where it could place its feet almost anywhere, eventually took on the pattern of the quadruped crawl, but it was not forced to do so. The quadruped crawl is the most practical wave gait available to four-legged walking machines. Overall, the gait determination system had more of an expert system nature than the more structured gait systems used on the O.S.U. machines. The PVII and TITAN-III, possibly as a result of having two fewer legs, walked in a manner that emphasized careful foothold selection, as opposed to continued longitudinal movement. In fact, the orientation of the machines' bodies was rarely associated with the direction of movement. Also, the machines always assumed a perfectly horizontal attitude from which deviations were prevented. Hirose (1984a and 1984b) argues that this is desired for transporting people and goods in a smooth fashion, but it does limit access to more highly sloped terrain. It seems possible that either machine could have
adopted nonhorizontal attitudes mechanically, but the control strategy would in no way have been able to handle this. This characteristic of being limited to horizontal attitudes only, highlights again that these machines were more oriented to choosing footholds carefully in terrain that did not have a lot available, than they were to terrain following in the manner of the O.S.U. machines.

Another disadvantage of these machines was that, to function well, they required legs that could swing widely in either direction about the corners on which they were mounted. This was necessary so that they were not restricted to longitudinally oriented walking, and could move in any direction. While this type of operation does have its advantages, the complexity of the mechanisms required, and the load bearing problems associated with them, can be eliminated by simply using a longitudinally oriented design. Such designs require only a restricted range of leg motion. So while these machines were quadrupeds, not much of the gait strategy developed for them was of use in the heuristic described in this paper. Their existence does prove however, the feasibility of four-legged walking.

2.3 "Approximate Straight Line Mechanism" Work

A series of six-legged walking machines, MELMARK Marks I, II and III, were built at the Mechanical Engineering Laboratory in the Tsukuba Science City in Japan. These machines made use of "approximate straight line mechanisms" that mechanically ensured that their foot movements occurred along straight lines parallel to their longitudinal axes. These mechanisms were used on each of the machines’ six legs. The legs were configured three per side on a longitudinally oriented body, and were very close to the body. The effect of the approximate straight line mechanisms, that of isolating the horizontal driving forces from the vertical support ones, was similar to that of the pantographs used by the ASV, PVII and TITAN-III, but the mechanisms involved were quite different. The reason for the design was different too. While this arrangement may have increased energy efficiency, the main purpose of it was to provide decoupled degrees of freedom in the machines. That is, by isolating the horizontal movement component in the legs, and mechanically connecting it in all of them, there was effectively a single degree of freedom in the direction parallel to the longitudinal axes of the machines. Thus the mechanism eliminated the complexity of dealing with the extensive number of degrees of freedom found in walking machines that have three degrees per leg.

It is interesting to consider the evolution of the MELMARK series. Mark I had only one degree of freedom, its straight line mechanisms moved all six legs simultaneously in the same direction. Mark II
was designed so that two groups of three legs, each group consisting of one middle leg and the two end legs on the opposite side of the body, could rotate relative to each other about a vertical axis in the center of the machine. It could thus turn as well as move forward and backward, and had two degrees of freedom. Mark III, designed for rough terrain, had eight degrees of freedom. It incorporated leg length variation in each of the six legs. So its body could turn, move in a given direction, and maintain its stability in that direction by adjusting the leg lengths to match the ground height variations. The design of Mark III was extremely simple, and it could handle rough terrain, but it has to be questioned whether all this simplification was necessary. The designers’ justification was that it decreased computing complexity, but that is not really a great concern with modern computers. The price paid for the control simplicity was a significant decrease in adaptability. The Mark III could have only one gait, the possible placement locations of its feet were highly restricted, and it could not readily change its attitude. Unlike the PVII and TITAN-III, this last problem was not due to limitations of the movement planner. The mechanism itself could not change attitude unless foot slippage occurred. So the limitations of the Mark III in rough terrain conditions were quite extensive, and none of this work was really considered in the heuristic design.

Adachi, Nakano and Koyachi (1987) developed a quadruped machine that also made use of approximate straight line mechanisms. This machine, the TURTLE-1, was similar to the Mark I in that it could only move forward and backward, but as the legs were not mechanically connected, it was not restricted to one overall degree of freedom. The TURTLE-1 had the ability to move in a quasi-dynamic fashion. It used a trotting gait to do so. It also had the ability to follow a fairly structured quadruped crawl gait. The TURTLE-1 has not yet been developed to handle uneven terrain. It does however, provide an example of what may be possible in terms of quasi-dynamic walking once an adaptive rough terrain gait for quadrupeds has been developed.

2.4 Work at Odetics

The Advanced Intelligent Machines Division of Odetics Inc. has developed a series of similar walking machines for use in nuclear installations (Bartholet, 1986). These ODEX machines take full advantage of the extent of present technology in order to have high power efficiency and high strength to weight ratios. They are the only true walking machines being used in practical applications. These machines all have a very tall and narrow aspect ratio that allows them to go where humans on foot can. They have six
Chapter 2. Walking Machine Research Efforts

legs distributed uniformly around a vertical tubular body, and are thus configured like idealized spiders. Each leg has three degrees of freedom, and operates in a reptilian fashion where the two segments rotate together in a vertical plane. The legs make use of a mechanical linkage design that isolates the vertical support and horizontal driving forces required to operate them. Different capability motor and gearing systems are then applied in these two directions to provide very strong lifting capability and very energy efficient forward movement. Under normal operating conditions these machines can run for one hour using their on-board battery system. Separating the two force directions in this fashion requires that the machine bodies not be oriented in any but a straight vertical posture, but this is not such a shortcoming with these machines. Their tall, vertical nature makes it undesirable to tilt them anyway, and their narrow horizontal dimensions greatly decrease the negative affects of not having the machine body attitude match that of the terrain upon which it is walking. All of the ODEX machines, excluding the first version of the first model, have had vision systems and arm/grippers attached. The vision systems provide for teleoperation of these machines, a feature first implemented over a radio link, and later over an automatically played out and retrieved umbilical cord. This link did not connect the machines to an external processor, all of the processing necessary for walking was done on-board, but it did allow remote specification of machine velocity, heading and rotation rate. All other aspects of walking are automatic, although factors such as height, and width in the direction of travel, can be specified. These machines make use of foot and shin sensors to feel their way through uneven terrain. They can negotiate stairs automatically. The gait that they make use of is the alternating tripod gait, the fastest member of the hexapod wave gait class. Bartholet (1986) does not discuss the gait determination scheme used in great detail. It is clear however, that it was one that followed closely the commands given by the operator, and that it assumed that the footholds it sought, while not necessarily at a consistent elevation, would always be available.

2.5 Other Work

The National Chiao Tung University in Taiwan has constructed a walking machine called the N.C.T.U. Quadruped-1 (Lee, 1986a and 1986b). This four-legged machine has the popular reptilian style legs with three degrees of freedom. These legs are distributed uniformly around a circular body in an idealized spider fashion. While this machine has the capability to ambulate over rough terrain, and while nothing about its design prevents it from adopting varying attitudes, this aspect of performance has not yet been
investigated. Instead the work by Lee and Shih (1986a and 1986b) has focused on the mathematical determination of ideal walking performance over smooth terrain. This work was of use in characterizing the fundamental gait of the heuristics described in this paper. Lee has also worked with others (1985 and 1988) analyzing, in a similar manner, the alternating tripod gait in simulated hexapods.

Joint work done at the U.S.S.R. Academy of Sciences and Moscow State University (Gurfinkel, 1981), and work at the University of Paris (Kessis, 1981), has involved hexapod walking machines. The former made use of a standard hexapod design. The latter a longitudinally oriented machine with two degree of freedom pantograph legs whose plane of operation was parallel to the longitudinal axis of the machine. Both machines made use of rather structured gaits, and neither has been developed to operate on uneven terrain.

Sutherland developed a standard hexapod that was large enough to transport a man (Raibert, 1983). This machine made use of an alternating tripod gait, and maintained its stability by tightly controlling the freedom of movement of the hardware. Each leg had three degrees of freedom, but walking was limited to small steps. The machine was powered by a hydraulic system that effectively linked the legs together. Further, the feet of this machine were very wide, adding to its inherent stability. This machine did walk, but in a very structured manner that did not respond to the terrain conditions. It was not a design that could be easily extended to rough terrain applications.

2.6 Summary

In summary, while the quadruped work of Hirose et. al. (1984a and 1984b), Lee and Shih (1986a and 1986b), and Adachi et. al. (1987) provided inspiration for this study, it was the hexapod work at O.S.U. that was found to be the most valuable during the development of the LOGHAQ heuristics.
General Characteristics of Quadruped Ambulation

This chapter will consider the general characteristics of statically stable quadruped ambulation. To remain statically stable, a four-legged machine must operate under a large number of movement restrictions. These restrictions result in a well defined general movement pattern. The factors that determine this pattern, and how the design of the LOGHAQ heuristics was affected by them, will be described in the following sections.

3.1 The Walking Machine Design Used in the Study

The LOGHAQ heuristics were designed to be of use on a machine similar to that shown in Figure 3.1. The actual dimensions of the machine and its legs are not of great importance as long as the feet on either side have the “access range” to be placed near each other throughout a wide volume of space on their side. The center of gravity was placed, throughout the study, in the center of the plane defined by the shoulders of the machine. This was not a requirement however, it could have been placed anywhere near the center of the machine. Each of the legs on the machine must have three degrees of freedom, as shown in Figure 3.1. The leg design used for the study was the popular reptilian one (see Section 2.1). It had a shoulder that could swing in two directions. These were horizontally (relative to the machine) about an axis perpendicular to the machine platform (specified by $\theta$), and vertically (relative to the machine) about an axis parallel to the side of the machine (specified by $\phi$). The third degree of freedom was in the knee joint extension angle (specified by $\epsilon$), which allowed rotation of the lower leg segment in the vertical plane (relative to the machine) defined by the upper segment. The two leg segments were, for this study, made the same length. The overall leg design was necessary for the correct operation of the heuristics. With some modification however, any design with three degrees of freedom could be used.

\footnote{The horizontal and vertical lines of the $\theta$ and $\phi$ symbols can be used as mnemonics.}
3.2 The Quadruped Crawl

The first consideration in developing the LOGHAQ heuristics was determining what type of gait should be used. The possibilities were to have either free or structured gaits, and it was decided that a structured gait was more appropriate. Structured gaits have a fixed and recurring leg lift pattern that is followed by the machine as it moves forward. There are only six feasible recurring leg lift patterns for quadrupeds (McGhee, 1968). Of these, the “quadruped crawl” is the one that affords the largest stability margins over the complete walking cycle (McGhee, 1968). It is also the gait used by mammals walking at slow speeds. This gait is uniquely suited to statically stable, longitudinally oriented, quadruped walking. Thus the quadruped crawl was chosen as the overall gait pattern for the LOGHAQ heuristics.

3.2.1 Description of the quadruped crawl

The quadruped crawl is illustrated in Figure 3.2. It operates as follows. In position ‘A’ leg 4 is the next to be swung forward. This cannot occur though, until the center of mass of the machine, and its surrounding “stability margin”, move into the “stability region” defined by the three front feet. So the
machine must "creep" forward until this requirement is met.\(^2\) Once this has occurred, at position 'B', leg 4 can safely be lifted and swung forward. As this takes place, the machine can continue moving forward, possibly to a location around 'C'. At this point foot 4 can be placed behind foot 2. When this occurs, leg 2 can immediately be lifted and swung forward. The stability region, due to this exchange of supporting feet on the right side, does not change much, so stability is maintained. As leg 2 is swung forward, the machine can continue its forward movement, as long as its center of gravity does not go past the front edge of the stability region. When foot 2 is placed, the machine should ideally have moved forward to position 'D', a location just within the stability region. At this point the stability region has been extended, and the machine is in a configuration that is a mirror image of its starting one. Legs 3 and 1 can then be moved in a similar manner, but not until the machine has again moved forward into the stability region defined by the three front legs. The movement of the machine that occurs when the two legs on either side are swung forward in turn, will be referred to throughout this paper as a "half-cycle".

### 3.2.2 Difficulties with free gaits

Free gaits, as described in Section 2.1, operate in the manner of an expert system, where the leg to be moved at any one time is chosen carefully based on a number of criteria. Hirose et. al. (1984a and 1984b), as stated Section 2.2, used such a gait on their quadruped, but it did not move in the longitudinally oriented fashion desired by this researcher. Other work on free gaits has involved hexapods, where longitudinally oriented walking is made use of, but where there is a more significant question of which leg should be lifted at any given time. To maintain static stability, any given walking machine must have at least three legs on the ground all of the time. So unlike quadrupeds, which can only lift one leg at a time, hexapods can lift as many as three. The advantage of using a free gait strategy on such a machine is that the free gait controller can determine which leg should be lifted among a number of possibilities. This is not necessary with quadrupeds. The next leg that a quadruped should lift is determined most often by which leg has moved farthest backward, and thus follows a natural pattern. At certain points during quadruped ambulation it may be advantageous to break the pattern to increase stability in the short term, but the resulting interruption in the smooth flow of the gait could eventually lead to gait failure. It would be extremely difficult to determine beforehand what the long term effects of breaking the pattern will be, so doing so is inadvisable. Thus the decisions made by free gait controllers about

\(^2\) "Creeping", throughout this study, refers to movement with all four legs supporting and none in the swing phase.
Chapter 3. General Characteristics of Quadruped Ambulation

Figure 3.2: The Quadruped Crawl
Chapter 3. General Characteristics of Quadruped Ambulation

which leg should be lifted are clearly useful for hexapods, but they are of little use, and could potentially be disastrous if employed on quadrupeds.

Less structured gaits are of use on quadrupeds when turning in place, or large obstacle avoidance are required. They are also of use when movement is desired in directions other than that defined by the longitudinal axis of the machine (Hirose, 1984a and 1984b). This study concerned heuristics that produced continuous, longitudinally oriented, forward motion however, and thus did not consider these capabilities. The main goal of the study was to develop techniques that maintained forward motion over rough terrain. It was envisioned that, once a complete forward motion heuristic package was developed, these extended capabilities could be added to produce a final system.

3.3 “Follow The Leader” Gaits

One interesting walking technique that many animals use, especially insects, is the “follow the leader” gait (Pearson, 1984, McGhee, 1984). Follow the leader gaits were also used on the two Ohio State University machines, as described in Section 2.1 (McGhee, 1984). This technique simply involves placing the back feet on both sides of the vehicle in, or near, the footholds used by the front feet on that side. Restricting them to nearby placements is necessary when these footholds are already occupied. The advantage of this technique is that footholds used for the back feet in this manner, have already been investigated, both visually and through actual placement of the front feet, and are known to be good locations. This removes the burden of finding new footholds for each foot, and requires only a system to locate footholds for the two front feet. (It is interesting to note that large cats use follow the leader gaits while stalking prey so that they can be assured of silent foot placement.) This technique, for the reasons given, was adopted in the LOGHAQ heuristics.

There are in fact, disadvantages to not using a follow the leader strategy on a quadruped machine. If the rear footholds are chosen at points other than near the front footholds, they should be located either ahead of, or behind, them. Placing them in positions laterally offset from the front footholds is also possible, but serves no purpose and just adds the foothold locating burden with no compensatory benefits. Placing the rear feet ahead of the front ones pushes forward the front edge of the stability region and allows the machine to travel further before the front foot is placed, thus hastening the entire movement. This is not possible however, unless a considerable sacrifice is made. This is due to the fact that the rear feet of a quadruped are placed before the front ones are lifted from the same area,
so extensive computations would have to be made to find footholds that will not result in the legs colliding. Collisions would have to be prevented while the rear foot is being placed, when both feet are on the ground, and when the front foot starts its swing forward. It will be shown in Section 4.16 that collisions can be easily prevented as long as the rear foot never goes further forward than the front one. Thus placing the rear feet ahead of the front ones is inadvisable. For reasons identical to those above, placement of the rear feet behind the front ones results in a decrease in speed. Such placements move the front edge of the stability region back, and thus decrease the travelling range. So this results in no benefits, and again requires extra foothold locating effort.

3.4 Stability Triangles

If the stability regions of a quadruped crawl that uses a follow the leader gait are mapped out over a number of steps, an interesting pattern develops, as shown in Figure 3.3. Each movement half-cycle takes place within a triangular stability region that borders those in front of, and behind it. This creates a continuous track of "stability triangles". The LOGHAQ heuristics made extensive use of this linking triangle pattern. Machine movement was considered in each of these triangles separately. That is, complete movement plans were developed to determine body motion through each triangle, and to control the swinging of the two legs that had to be relocated during the crossing of the triangles. Further, the direction of movement, and the machine attitude and orientation were determined by the stability triangles. The vector between the two unmoving legs on the external side of the triangles was used to define the trajectory that the machine was to adopt during each half-cycle. The machine adjusted its tilt and rotational orientation to match that of the vector. That is, the machine oriented itself so that its longitudinal axis would be parallel to this vector. Its platform attitude was set to match that of the apparent ground planes defined by the three corners of the triangles. McGhee et. al. (1984) averaged all of the foothold elevations of the O.S.U. Hexapod to determine its ground plane in a similar manner.

It was considered whether or not using just three footholds to determine the apparent ground plane, and thereby the machine attitude and orientation, was wise. Doing so could result in exaggerated reaction of the machine to new footholds. Past footholds, or even the next forward foothold, could have been averaged in to the ground plane determination to decrease this effect. It was decided however, that past footholds were past, they reflected conditions which were no longer prevailing. It was the reality of the
Figure 3.3: Linked Stability Triangles
current machine situation that was of the most importance. The next forward foothold location, while possibly of use, was not known when the movement plans for the half-cycle were made. Further, it was felt that in general the three footholds best reflected what was actually happening to the machine. While these footholds might not have accurately indicated the attitude of the ground underneath the machine, and while they did not indicate upcoming radical terrain changes that should have been prepared for, they did specify the conditions that the machine currently had to work with. If they were not used as a guide, the machine might have adopted orientations that were awkward relative to its foot positioning. Thus it was decided to make use of only these footholds. Possible exaggerated reactions to them were dampened by using "settling factors" to promote smooth transitions from one prescribed trajectory and attitude specification to the next. Doing this did incorporate information about old footholds into the motion plan, but in an indirect manner. The exact operation of the settling factors is described in Section 4.7.

The stability triangles were defined by the foothold locations, and these triangles in turn determined the movement of the machine. So the operation of the heuristics depended entirely on the series of footholds presented to them. No other information was available. This was the general philosophy of the heuristic design. The movement planning problem centered on how to best move along a path in space determined by the isolated footholds. Other aspects of the terrain, such as the location of large obstacles, were not considered. It was assumed that the operator of the machine would prevent them from becoming problems. This does not mean that the LOGHAQ heuristics unconditionally accepted any footholds presented to them however. Section 4.9 details how the Foothold Determination heuristic guided the selection of the next foothold so that it would best facilitate the movement desired by the operator. Thus some control was exerted over the actual positioning of the foothold series, and thereby the stability triangle pattern.

3.5 Control of Movement

As stated in the Introduction, the purpose of the LOGHAQ heuristics was to generate a longitudinally oriented movement plan for a quadruped crossing rough terrain. The heuristics were designed to be used on a machine that had an operator (or teleoperator) who specified the general direction of movement by setting a desired turning rate, \( \rho_c \). This turning rate indicated the amount that the machine should rotate about the platform normal every half-cycle. Given the footholds available in the direction specified, the
heuristics then calculated the desired joint angles to facilitate smooth walking. The heuristics did not choose a course around major obstacles, this was perceived as the task of the operator. The heuristics were designed to work out the details of the walking motion under general direction. Further, they were not designed to be of use in extremely difficult terrain conditions where each leg must be placed carefully in awkward foothold locations. The main application of the heuristics was over rough, but not impassable terrain, where, under general guidance, they provided the plan necessary for a smooth, steady gait.

Aside from specifying the turning rate, the operator could also set other heuristic parameters. These included control values that could be altered occasionally, but did not require constant adjustment. They were the height of the machine above the ground, $h$, the width of the stance, $w$, the "stride length", $s$, and a number of settling factors (described in Section 4.7) that determined how quickly the machine adjusted to new conditions.

The LOGHAQ heuristics, for the purposes of this study, did not include speed control. One of goals of the investigation was to observe, given a set of maximum joint angle velocities, what speeds a simulated machine could maintain. Slower speeds were not of interest, as operating at reduced speeds did not affect machine stability or any other aspect of performance. In a practical application, speed control could easily be added to the heuristics. In addition, the heuristics did not require, or even allow, operator control of the machine attitude in the manner of Waldron et al. (1986). Another goal of the study was to observe how well the heuristics performed this function themselves. Attitude control could also be easily added in a practical application.

3.6 Speed Considerations

The main objective of the LOGHAQ heuristics was to maintain a smooth ride over rough terrain. This goal could have been accomplished by making extremely slow movements and allowing plenty of time for leg repositioning, but this would hardly have been a significant accomplishment. So a further goal of the heuristics, as mentioned in the previous section, was to maintain the highest speed possible. This section will consider some of the factors that influence the operational speed of quadruped machines, and examine how these affected the heuristic design.

It should be noted that this section departs from conventional walking machine gait analysis. The reason for this is that most gait analysis work (ie.: Song, 1987, Lee, 1986a) has concerned how the
stability margins are affected by different aspects of the machine design and performance. This was not a concern with the LOGHAQ heuristics. The minimum stability margin was a defined constant in these heuristics, and was always maintained. This was considered a safe and practical way of implementing a gait planner. As a result, while traditional analyses have considered how to improve the stability margin for a machine walking with a given leg duty cycle (effectively a defined speed), this analysis will consider how the top machine speed was improved with a given stability margin.

Like most gait analyses, this one considers only the locations of the feet and the center of gravity of a machine. It is never concerned with the locations of the shoulders of a machine. This may seem unusual, as the shoulders would seem to be intimately involved in the walking process, but they are not really of concern from a speed analysis point of view. It is only the stability triangles, as defined by the footholds, that need to be considered. The practical aspects of a machine design, including the relationship between the shoulders and feet, will be discussed in Chapter 5.

3.6.1 The ideal gait

Following Song and Waldron (1987), an ideal quadruped crawl gait will first be considered. This gait has the connected stability triangle pattern shown in Figure 3.3, but it does not have any creeping period. It is so well timed that the crossing of each of its stability triangles coincides exactly with the placement of a forward foot that defines a new one. No stability margin is used. The machine moves right to the edge of the stability region and makes use of precise timing to constantly remain (theoretically) statically stable. With an ideal gait the machine body maintains a constant speed. One leg is always swinging, and all of the legs swing forward the same amount. The uniform swing distance is due to the fact that each foothold is spaced the same distance from the one behind it, and the fact that the body moves forward exactly the same amount for each swing period. The latter of these two conditions is true because each leg does swing forward the same amount, thus taking the same amount of time, and resulting in the same machine movement distance at the constant speed. So a complete cycle of all four legs consists of four equal duration periods in which each of the legs swings forward the same amount.

The time required for each of them to swing forward is:

\[ t_s = \frac{\theta_f - \theta_i}{\omega_{\text{max}}} \]  

(3.1)

Where \( \omega_{\text{max}} \) is the maximum angular velocity of the forward swing, and \( \theta_f \) and \( \theta_i \) represent the final and initial leg horizontal angles respectively.
Chapter 3. General Characteristics of Quadruped Ambulation

The length of time that each leg supports the machine while rotating backward, \( t_i \), is 3\( t_s \). This is because during the support period the other three legs, one at a time, all swing forward. Now it is apparent that the amount that the legs swing forward is exactly equal to the amount that they rotate backward as the machine moves ahead. So:

\[
\omega_{\text{backward}} = \frac{\theta_f - \theta_i}{t_i} = \frac{\theta_f - \theta_i}{3t_s} = \frac{\theta_f - \theta_i}{3(\theta_f - \theta_i)/\omega_{\text{max}}} = \frac{\omega_{\text{max}}}{3}
\]

Therefore, unless there is some mechanical limitation that forces the backward leg rotation to be much slower than the forward swing, it will always be the maximum forward angular velocity that limits the speed of the machine. That is, there is always plenty of time for the machine to move forward as a leg swings, so it is the swing time itself that is the limiting factor. It is this time that determines, as one leg after another swings, how long it will take the machine to cross each stability triangle.

Considering the total time required for one leg to complete its entire cycle, including its backward drive and forward swing, the following is apparent:

\[
\text{duty cycle} = \frac{\text{support time of leg}}{\text{total cycle time}} = \frac{t_i}{t_i + t_s} = \frac{3t_s}{4t_s} = 0.75
\]

This result agrees with McGhee's (1968) conclusions, from similar ideal gait consideration, that the lowest possible duty cycle for a quadruped is 0.75. As the gait moves farther away from an ideal one, \( t_i \) grows relative to \( t_s \), driving the duty cycle up.

Now the distance travelled during the total cycle must be such that it returns the machine to the same position relative to its feet. The front feet, when they are placed, are spaced apart in the forward direction of the machine by a distance \( s \) (the stride length). As two front foot movements take place in one complete cycle, the total distance travelled is 2\( s \). So the machine velocity is:

\[
v = \frac{2s}{t_i + t_s} = \frac{2s}{4t_s} = \frac{s}{2t_s}
\]

Now the time required to swing a leg forward is related to the distance that it has to swing. The exact relationship is difficult to determine however, as it depends on details of the machine design. For the sake of clearer analysis and, as will be shown later, because doing so has no significant detrimental effect on these calculations, a linear relationship will be assumed between the swing time and the swing distance of a leg. With this assumption:
Chapter 3. General Characteristics of Quadruped Ambulation

\[ t_s = k\left(2s - \frac{1}{4}2s\right) = \frac{3ks}{2} \]  

(3.5)

Where \( k \) is the constant of the linear relationship, and the swing distance is calculated by subtracting the distance that is travelled by the machine during the swing from the distance between the footholds. Combining this with Equation 3.4:

\[ v = \frac{s}{2 \frac{3ks}{2}} = \frac{1}{3k} \]  

(3.6)

So the velocity of the machine's movement, given the assumption of a linear relationship between the leg swing distance and the swing time, is entirely determined by the constant of proportionality between the two. The faster the swing leg can cover a given relocation distance, the faster the machine can move. This emphasizes the fact that it is the leg swing time that determines the speed of the machine, and reveals the rather interesting fact that theoretically, the stride distance does not affect it at all. This finding agrees with the ideal gait analysis done by Song and Waldron (1987). They found that, "for any [stride], no matter how small, the vehicle can move at a speed determined by the stepping frequency" (p. 68). In practice this is not the case however, and the next subsection will detail further investigations into the realities of the effects of stride length on walking speed.

### 3.6.2 The effect of stability margins

Now that it has been used to establish the basic relationships, the ideal gait must be discarded to permit a more practical analysis. To prevent machine collapse, stability margins must be added. Doing so requires that creeping movements occur in the regions near the edges of the stability triangles. This is shown in Figure 3.4. In this figure, \( c \) is the distance that the machine must creep before and after its stability margin circle crosses into a new stability triangle. A result of adding the creeping distance is that the distance the machine travels during its swing periods is reduced. Equation 3.5 must be altered accordingly:

\[ t_s = k\left(2s - \frac{1}{4}(2s - 4c)\right) = k\left(\frac{3s}{2} + c\right) \]  

(3.7)

Now it is assumed that the velocity of the machine as it passes through the creeping regions is the same as that during the leg swing periods, so the time required to cross these regions is:
Chapter 3. General Characteristics of Quadruped Ambulation

Figure 3.4: The Creeping Distances That Result From Addition of a Stability Margin

\[ t_c = \frac{c}{v} \]  

(3.8)

Using these time requirements, a new velocity equation can be calculated:

\[ v = \frac{2s}{4t_s + 4t_c} \]  

(3.9)

\[ v = \frac{s}{k(3s + 2c) + 2v} \]  

(3.10)

\[ v = \frac{s - 2c}{k(3s + 2c)} \]  

(3.11)

\[ v = \frac{1 - 2\frac{c}{s}}{3k + 2k\frac{c}{v}} \]  

(3.12)

Where \( c \) is restricted to a range of:
So it can be seen that the velocity of a machine is affected by the ratio of the creeping distance to the stride length. As shown in Figure 3.5, as that ratio gets larger, the velocity decreases. Thus it would be best, given a fixed creeping distance, to make the stride length as large as possible. If the stride length is made so much larger than the creeping distance that \( \frac{c}{s} \) approaches zero, the velocity becomes the familiar:

\[
v = \frac{1}{3k}
\]  

(3.14)

If the stride length is reduced to its minimum value of \( \frac{4c}{2}, \) or \( 2c, \) which would correspond to the elimination of any area over which the machine could travel while swinging its legs, the resulting velocity is zero. This is because the velocity during the creeping periods is restricted to that of the swinging periods. If the velocity of the creeping periods were allowed to be infinite (accomplished by setting \( t_c \) to zero), the velocity equation would become:

\[
v = \frac{1}{3k + 2k \frac{c}{s}}
\]  

(3.15)

And it would still be best to make the creeping distance to stride length ratio as small as possible.

None of the results of adding the stability margin to the ideal gait are particularly surprising. It follows intuitively that, since it is the leg swing time that fundamentally determines machine velocity, the creeping time during which swinging does not occur should be minimized. As two fixed sized creeping distances must be crossed in every stability triangle, this is accomplished by increasing the size of the stability triangles (a direct result of the stride distance) and thus reducing the number of creeping distances over a given distance of travel.

3.6.3 The effect of swing overhead time

Another factor that determines how the stride length affects the velocity is the leg swing overhead time. A real walking machine will have extra time delays at the start and finish of its leg swings to provide for initial ground clearance and accurate foot placement. These time delays will add directly to the total swing time required, and will not follow the linear relationship that was assumed between the swing time and the swing distance. To minimize the effect of this overhead, the stride length should be made
Figure 3.5: Plot of $v = \frac{1}{k} \cdot \frac{1 - \frac{2s}{k}}{3 + \frac{2s}{k}}$
3.6.4 The relationship between swing time and swing distance

As stated previously, the exact relationship between the time and distance of a leg swing is difficult to determine. It is affected by specifics of the walking machine design and the machine state at the beginning and end of the swing. The swing will generally be circular in nature however, so the following analysis can be made. Although it is not required, to most easily access the greatest range, each leg of a walking machine should swing more or less symmetrically about its shoulder under normal operation. Thus, as shown in Figure 3.6, a typical swing would start at a point behind the shoulder and end at a point roughly an equal distance in front of it. Now from Figure 3.6:

\[ \alpha = \tan^{-1} \left( \frac{s}{R} \right) \]  
(3.16)

\[ \frac{d\alpha}{ds} = \frac{R}{R^2 + s^2} \]  
(3.17)

So, as shown in Figure 3.7, as \( s \) increases, \( \frac{d\alpha}{ds} \) decreases, indicating that as the stride is made larger, the resulting increase in the angular movement of the swing does not grow as quickly. This means that the swing time grows more slowly as well, and again points to making the stride length as long as possible. With fewer steps over a given distance of travel, the resulting total overhead will be less.
Chapter 3. General Characteristics of Quadruped Ambulation

Figure 3.7: Plot of $\frac{d\theta}{ds} = \frac{1}{R} \cdot \frac{1}{1+\left(s/R\right)^2}$

possible. Further, it shows why consideration of the actual swing time to swing distance relationship was not necessary throughout this analysis. Such consideration would have made the analysis unnecessarily complex, and when it was made, merely added to the general trend of the findings.

It is interesting to note that $\frac{d\theta}{ds}$ also decreases if $R$ is increased. This is because, with a longer radius of swing, the same angular change results in a larger forward foot movement. Such a change in the design of a machine involves other considerations however, such as the additional force required to support the longer extension. These issues are discussed in Chapter 5.

Based on all of the results considered above, it was felt that the best strategy for determining the stride length was to make it as long as possible within the physical constraints of the walking machine concerned.
3.6.5 The effect of altering the machine stance width

Increasing the stride length resulted in the longest possible external edges on the connected stability triangles. Now consideration will be made of how the maximum speed of the machine is affected by altering the lengths of the other two edges, the front and back ones. These edges are lengthened simultaneously by increasing the width of the machine stance, $w$. Changing the width of the stability triangles does not have a significant effect on the stability triangle pattern, and has no effect at all on the distance travelled across each triangle. The pattern becomes wider as the stance does, but the number of triangles necessary for a given distance of forward movement is not altered. Thus the number of creeping distances that must be crossed is not altered either. Widening the stance does affect the creeping distance itself however, as seen in Figure 3.8. As the stability triangle becomes wider, its front and back edges tend toward being perpendicular to the direction of travel. As this happens, the creeping distances, which are a result of the contact between the circular stability margins and the edges, decrease in size. The extreme result of this process is that the creeping distances become simply the radial line from the center of the stability margins to their contact point on the stability triangle edges. In this case the creeping distances become equal to the stability margin. Using similar triangles, the relationship between the various lengths is:

$$\frac{c}{\text{stability margin}} = \frac{\sqrt{s^2 + w^2}}{w}$$

(3.18)

Which can be altered to:

$$c = \text{stability margin} \cdot \sqrt{\left(\frac{s}{w}\right)^2 + 1}$$

(3.19)

Thus the larger $w$ is made in relation to $s$, the smaller $c$ becomes. The benefits of decreasing $c$ in relation to $s$, in terms of increasing maximum velocity, were described earlier in this section. The changes that are possible in the size of $c$ are somewhat restricted however, and making them by increasing $w$ may not be as effective in increasing velocity as simply extending the stride length. So while having a wide stance is preferable, it is not a high priority. Other reasons for altering the stance, such as improving lateral stability, would be of more importance in a practical machine design. These issues will be discussed further in Chapter 5.
Figure 3.8: The Effects of Stance Width on the Creeping Distance
Chapter 4

Heuristic Specification

This chapter will explain the details of the LOGHAQ heuristic package, beginning with the overall heuristic structure. The heuristics operated in a cyclic fashion performing the following basic functions:

1. Determine the joint angle changes made by the machine in the last sample time, and assess the effects of these changes on the position of the machine in space.

2. Check to see if the swinging leg has touched down. If so, plan the overall movement of the machine accordingly and determine the foothold for the next swing leg.

3. Check the stability status of the machine. If it is about to become unstable, halt it. If it has entered a new stability triangle, lift its swing leg.

4. Use the overall movement plan to calculate the machine speed over the next sample time.

5. Use the planned speed to determine the swing leg joint angles that will move the leg smoothly towards the foothold. Calculate the number of sample times required for this to take place.

6. Use the planned speed to determine the body translation. Use the number of remaining sample times to determine the body attitude and orientation changes.

7. Determine the joint angles necessary in the support legs to produce the desired translation, and the desired attitude and orientation changes.

8. Use the desired joint angles calculated to drive a machine dynamic controller.

The second function performed, that of planning the overall movement and determining the new foothold, was the only one that involved any long term planning. This function was used only when a foot had just been placed, an event that resulted in an entirely new machine state. The rest of the functions were performed continuously, every sample time, and used the overall movement plan as a guide. These functions used the current machine state to calculate a movement plan for just the next
sample time. It was not assumed that the short term plans of these functions would be followed exactly by the machine dynamic controller. These plans were completely recalculated every sample time.

4.1 The Overall Heuristic

More specifically, the overall heuristic operated as follows:

repeat:

Assess Movement, both translational and rotational, from last sample,
Calculate Predicted Body Orientation Vectors from target angles
if Swing Leg Kinematic section previously indicated possible touchdown:
    check foot sensors to see if this has occurred
if touchdown has occurred:
    set LIFTED to false
    set HALTED to false
Update Swing Leg
if either front leg was swinging:
    do Half-Cycle Movement Planning
Calculate Target Angles for body when new foot is placed
re-Calculate Predicted Body Orientation Vectors from target angles
make new Foothold Determination
if machine is not HALTED:
    Check Stability
    if machine has reached edge of stability region:
        set HALTED to true
        if machine was creeping and reached new stability triangle:
            set LIFTED to true
    if machine is still not HALTED:
        make Speed Calculation
otherwise:
    set speed to 0
if a leg is LIFTED:
    calculate Swing Leg Kinematics at specified speed
    (while doing so return number of sample times remaining for swing)
otherwise:
    Approximate Step Times Remaining in half-cycle movement
if swinging leg cannot reach foothold and machine is HALTED:
    Adjust Attitude to reach foothold
    calculate Body Kinematics using speed and remaining sample time values
if the machine cannot move because a leg is at the end of its movement range:
    set HALTED to true
if either rear leg is swinging:
    Prevent Leg Collision from resulting from planned changes

Each of the tasks indicated with capital letters in the above pseudo-code constituted a component heuristic of the LOGHAQ package and will be elaborated on in the following sections. To begin with, the coordinate system used will be described, as well as the extensively used “orientation vectors”. It should be noted that, for the purpose of clarity, many of the small details of the heuristics have been omitted in the following sections. The complete heuristics, in the form of “C” code, can be found in Appendix C.

4.2 Coordinate System Specification

All points of concern to the LOGHAQ heuristics were specified in a Cartesian coordinate system with its origin maintained at the center of the machine, as shown in Figure 4.1. The z and y axes of this coordinate system remained horizontal at all times, regardless of the attitude of the machine, and the y axis always pointed in the direction of the longitudinal axis of the machine. The attitude of the machine was described with two angles, \( \psi \) and \( \sigma \). \( \psi \) was the angle between the longitudinal axis of the machine and the y axis, and was positive when the front of the machine was tilted upward. \( \sigma \) was the angle of rotation of the machine body about its longitudinal axis. It was zero when a line running across the breadth of the machine was horizontal, and became positive as the right side of the machine rotated higher than the left side. These two angles were specified to match what the output of a gyroscopic leveling device would be if its outer axis were made parallel to the longitudinal axis of the machine.
Its outer axis rotation would then be $\sigma$, and its middle axis rotation $\psi$. Its inner, spin axis would be vertical. The heuristics made use of no other coordinate systems.

To maintain the validity of this coordinate system as the machine moved, all points of concern to the heuristics were recalculated each sample time to compensate for any translational or rotational movement. This was not an excessively time consuming task, as there were only two points of significance. These were the new front foothold location, $P_f$, and the planned final location of the machine at the end of the half-cycle, $P_p$ (to be described in Section 4.7). There were also two orientation specifying angles that required only rotational compensation.

### 4.3 The Orientation Vectors

The orientation vectors were used extensively throughout the LOGHAQ heuristics. They consisted of three orthogonal unit vectors that specified the attitude of the machine. One ran forward along the longitudinal axis of the machine, $\hat{A}$, another ran laterally across the machine from the left to the right side, $\vec{S}$, and the third was normal to the machine platform, $\vec{N}$ (see Figure 4.1). These vectors were calculated from the attitude specifying angles as follows:

\begin{align}
\vec{t}_a &= 0 \\
\vec{j}_a &= \cos \psi \\
\vec{k}_a &= \sin \psi \\
\vec{i}_s &= \cos \sigma \\
\vec{j}_s &= \sin \sigma \sin \psi \\
\vec{k}_s &= -\sin \sigma \cos \psi \\
\vec{N} &= \vec{S} \times \hat{A}
\end{align}

### 4.4 The Assess Movement Heuristic

After machine movement occurred during a sample time, the Assess Movement heuristic determined the changes to the position of the machine in space. It recalculated the positions of the feet of the machine from the leg joint angles, used the new positions to determine how the machine had moved, and performed compensatory translation and rotation of various points.
Figure 4.1: The Coordinate System Used
The only information the LOGHAQ heuristics had about the position of the machine in space was the gyroscopic attitude indication, and the location of the supporting feet. It was assumed that the feet were firmly planted and could only move insignificant distances. Using this assumption, the Assess Movement heuristic determined the machine movement by comparing foot locations before and after the sample time. The resulting movement estimate was used to do the compensatory translation and rotation of the relevant values. The heuristic only considered the three firmly planted supporting feet, not the foot of a swinging leg or a leg that had just finished swinging. This latter specification ensured that the feet it used for its calculations had as much time as possible to settle firmly into new footholds. The heuristic operated as follows:

The Assess Movement Heuristic

calculate the orientation vectors with the current $\psi$ and $\sigma$
calculate the new location of each foot:

\[ \alpha = \pi - \epsilon - \phi \quad \text{(see Figure 4.2 for clarification)} \]

\[ d_a = (L_U \cos \phi + L_L \cos \alpha) \sin \theta \quad \text{($L_U$ and $L_L$ are the upper and lower leg segment lengths)} \]

\[ d_z = (L_U \cos \phi + L_L \cos \alpha) \cos \theta \]

\[ d_n = L_U \sin \phi - L_L \sin \alpha \]

\[ J = \bar{\bar{A}} \cdot \text{forward distance to shoulder} + \bar{S} \cdot \text{lateral distance to shoulder} \]

\[ D = \bar{\bar{A}} d_a + \bar{S} d_z + \bar{N} d_n \]

\[ P_{\text{foot}}' = J + D \]
calculate the horizontal angular difference between vectors running between pairs of new support foot locations and old support foot locations, where $z_n$ and $y_n$ are foot coordinates:

\[ \Delta \rho = \tan^{-1}[\frac{(y_n^2 - y_n^1)}{(x_n^2 - x_n^1)}] - \tan^{-1}[\frac{(y_1^2 - y_1^1)}{(x_1^2 - x_1^1)}] \]

average the three differences calculated for an overall rotational estimate
deduct this amount from the planned and target rotational angles required in the remainder of the half-cycle (to be described in Sections 4.7 and 4.8 respectively)
do a compensational rotation of $P_f$, $P_p$, and the old support foot locations by this amount
calculate three dimensional translation between old (compensated) and new support foot locations
average these three translation values for an overall translation estimate
do a compensational translation of $P_f$ and $P_p$ by this amount
obtain speed measurement by determining magnitude of translation
The rotations were accomplished by setting:

\[ x' = \sqrt{x^2 + y^2} \cos \left( \tan^{-1} \left( \frac{y}{x} \right) + \text{desired rotation} \right) \]  
(4.8)

\[ y' = \sqrt{x^2 + y^2} \sin \left( \tan^{-1} \left( \frac{y}{x} \right) + \text{desired rotation} \right) \]  
(4.9)

### 4.5 The Calculate Predicted Body Orientation Vectors Heuristic

This heuristic was responsible for calculating the predicted orientation vectors of the machine. That is, it determined what these vectors would be if the machine achieved, when its next footfall occurred, the target attitude and orientation angles, \( \psi_t, \sigma_t \) and \( \rho_t \) (see Section 4.8). These vectors were used in many of the other heuristics and had to be kept up to date. Machine rotations caused them to become invalid after every sample time. They could have been rotationally compensated in the Assess Movement heuristic, but they were used so widely that it was felt best to recalculate them. This also prevented cumulative errors. In addition, limitations were placed on the angular velocities of the machine movement during each sample time. If, with these limitations, there were not enough sample times left in any phase of machine motion to reach the target angles, this heuristic compensated accordingly. It determined the angles that the machine could reach in the remaining sample times, and it used these to calculate the predicted orientation vectors. The remaining sample times in the movement phase were calculated in the Swing Leg Kinematics heuristic, and passed to this heuristic as the variable \( \text{"REMAINING"} \). The calculation of the predicted orientation vectors was made using Equations 4.1 to 4.7, and the achievable target attitude angles. After this was done, each of the three vectors was rotated about the \( z \) axis by the target orientation angle, \( \rho_t \) (using Equations 4.8 and 4.9).

This heuristic also determined the predicted shoulder location of the swing leg when the next footfall occurred. This was calculated by simply multiplying the predicted body orientation vectors by the appropriate machine dimensions and adding them together for each leg (see Figure 3.1):

\[ \bar{J}_p = \bar{A}_p \cdot \text{forward distance to shoulder} + \bar{S}_p \cdot \text{lateral distance to shoulder} \]  
(4.10)

The predicted swing leg shoulder location was used by both the Speed Calculation and Swing Leg Kinematics heuristics.
Figure 4.2: Conversion of Joint Angles Into Shoulder to Foot Offset Distances
4.6 The Update Swing Leg Heuristic

When a foot touchdown occurred this heuristic determined, using the quadruped crawl sequence, which leg should next be lifted. In addition, it maintained a running estimate of the expected final horizontal angles when the front feet touched down. This estimate was used in the Speed Calculation heuristic. When a front foot was placed, the horizontal angle at which the placement occurred was merged into the running estimate as follows (where the prime indicates the new value of the variable):

\[ \hat{\theta}_j' = \hat{\theta}_j + \left( \hat{\theta}_j - \hat{\theta}_j \right) \cdot \text{learning rate} \]  

(4.11)

This was done independently for each of the two front legs.

4.7 The Half-Cycle Movement Planning Heuristic

4.7.1 Trajectory planning

As described in Section 3.4, the machine movement plan over a half-cycle was based on the triangle defined by the three footholds of the stability triangle. There were two aspects of movement planning that had to be considered, the trajectory of the movement, and the final attitude of the machine body. The trajectory was calculated by first determining the ideal final location, \( P_i \), of the machine at the end of the half-cycle. The ideal location was to have the machine center of gravity right at the front edge of the stability triangle. To keep its lateral movements reasonable, the point halfway along the front edge of the stability triangle was chosen as \( P_i \). So \( P_i \) was calculated by simply averaging the coordinate values of the two footholds that defined the front edge of the stability triangle. The desired machine height, \( h \), was added to the \( z \) value thus obtained to keep the machine that distance above the ground. It should be noted that adding to the \( z \) value in this manner foreshortened the actual ground clearance distance when the machine was on anything but level ground, but, as described by Lee and Orin (1986c), this increased the stability of the machine when crossing sloped terrain. The following were the calculations required:

\[ x_i = \frac{x_2 + x_1}{2} \]  

(4.12)

\[ y_i = \frac{y_2 + y_1}{2} \]  

(4.13)

\[ z_i = \frac{z_2 + z_1}{2} + h \]  

(4.14)
Once $P_i$ had been determined, it was merged with a point reflecting the old trajectory of the machine, $P_0$. Doing so prevented the machine from making drastic alterations to its trajectory as a result of one inconsistent stability triangle. $P_0$ was determined by first rotating the current forward orientation vector, $\vec{A}$, about the normal orientation vector, $\vec{N}$ (these were calculated using Equations 4.1 to 4.7). The amount of rotation was the turning rate specified by the operator, $\rho_c$. The three dimensional rotation of the orientation vectors was accomplished with the following:

$$\vec{A}' = \frac{\vec{A} - \vec{S}\tan \rho_c}{|\vec{A} - \vec{S}\tan \rho_c|} \quad (4.15)$$

$$\vec{S}' = \frac{\vec{S} + \vec{A}\tan \rho_c}{|\vec{S} + \vec{A}\tan \rho_c|} \quad (4.16)$$

These equations were made simpler by the fact that the orientation vectors involved were unit vectors. The results of the calculations were also unit vectors.

The vector generated by the rotation was then multiplied by $|P_i|$ to produce $P_0$. The coordinate values of $P_0$, thus determined, were merged with those of $P_i$ using the following settling formulas:

$$x_p = x_o + (x_i - x_o) \cdot \text{movement settling factor} \quad (4.17)$$

$$y_p = y_o + (y_i - y_o) \cdot \text{movement settling factor} \quad (4.18)$$

$$z_p = z_o + (z_i - z_o) \cdot \text{movement settling factor} \quad (4.19)$$

With the result being, $P_p$, a point that indicated the planned center of gravity location at the end of the half-cycle, and thus the planned trajectory.

### 4.7.2 Attitude and orientation planning

A planned machine body attitude for the end of the half-cycle was also determined by this heuristic. As stated in Section 3.4, the desired attitude and rotational orientation of the machine were determined by the stability triangle. The ideal rotational change angle, $\rho_i$, and tilt angle, $\psi_i$, were calculated by considering the vector that ran between the two feet on the external (or unmoving) side of the stability triangle, $\vec{L}$. This vector specified the orientation that machine body should have at the end of the half-cycle. From it, $\rho_i$ and $\psi_i$ were calculated as follows:

$$\rho_i = \tan^{-1} \left( \frac{-\vec{L}_i}{\vec{L}_i} \right) \quad (4.20)$$
Chapter 4. Heuristic Specification

\[ \psi_i = \tan^{-1} \left( \frac{k_i}{\sqrt{\ell_i^2 + J_i^2}} \right) \]  

(4.21)

To calculate the ideal roll angle, \( \sigma_i \), another vector was generated, \( \vec{C} \). This vector crossed from the rear foot on the unmoving side of the machine to the front foot on the other side. From it, \( \sigma_i \) was determined as follows:

\[ \vec{U} = \vec{C} \times \vec{L} \]  

(4.22)

\[ \vec{H} = \vec{L} \times |0, 0, 1| \]  

(4.23)

\[ \vec{V} = \vec{H} \times \vec{L} \]  

(4.24)

\[ \sigma_i = \tan^{-1} \left( \frac{(\vec{U} \cdot \vec{H})/|\vec{H}|}{(\vec{U} \cdot \vec{V})/|\vec{V}|} \right) \]  

(4.25)

The three ideal angular values were then merged with angles representing the old machine attitude and orientation. These latter angles were also calculated using Equations 4.20 to 4.25, but \( P_o \) was used for the orientation vector, \( \vec{L} \), and the normal orientation vector, \( \vec{N} \), was used for \( \vec{U} \). The ideal and old angles were merged as follows:

\[ \rho_p = \rho_o + (\rho_i - \rho_o) \cdot \text{rotational settling factor} \]  

(4.26)

\[ \psi_p = \psi_o + (\psi_i - \psi_o) \cdot \text{tilt settling factor} \]  

(4.27)

\[ \sigma_p = \sigma_o + (\sigma_i - \sigma_o) \cdot \text{roll settling factor} \]  

(4.28)

4.8 The Calculate Target Angles Heuristic

The target angles, \( \psi_t \), \( \sigma_t \) and \( \rho_t \), were the attitude and rotational orientation specifying angles that the machine attempted to adopt before each new leg placement was made. They were derived simply from the planned angles, \( \psi_p \), \( \sigma_p \) and \( \rho_p \), that were calculated in the Half-Cycle Movement Planning heuristic. The target angles for when the rear foot was placed were defined as angles half way between the current ones and the planned ones. The target angles for when the front foot was placed, and the half-cycle ended, were defined as the planned angles.
Chapter 4. Heuristic Specification

4.9 The Foothold Determination Heuristic

As described in Section 3.4, the movement plan that the LOGHAQ heuristics developed was based entirely on the footholds presented to them. The location of these footholds was thus of critical importance. This section will consider how the Foothold Determination heuristic guided a foothold scanning system so that reasonable foothold choices were made.

4.9.1 Rear foot placement

The follow the leader gait eliminated the need to consider the footholds required for the rear feet, they simply fell in behind the front feet. Their location was determined by moving back from the front feet a short distance in the reverse direction of the predicted forward orientation vector. This distance, the "following distance", was empirically chosen so that the feet were as close together as possible, with a large enough margin between them to prevent the two legs from striking each other. If leg 3 was being lifted, the foothold for it was calculated as follows:

\[ P_f = P_1 - \bar{A}_p \cdot \text{following distance} \quad (4.29) \]

4.9.2 Front foot placement general requirements

The foothold locations for the front feet were a much larger concern. They had to be chosen to facilitate continuous smooth, stable movement of the machine. The footholds could not be chosen too far forward, or too close, and they had to produce a series of stability triangles that at least generally lead in the direction chosen by the operator. The footholds also had to be placed within a reasonable lateral proximity of the machine. Meeting all of these criteria, especially when the machine was being instructed to turn, was the most difficult part of the work done in this study. The movement of the machine body followed from the footholds, and if they were not placed so that a smooth flow of motion could be maintained, the machine inevitably reached an unworkable situation.

The main difficulty in determining the footholds was that they could not just be chosen as desired. The assumption was made, throughout the heuristic development, that some sort of scanning system would actually make the choice. This system would operate in parallel with the LOGHAQ heuristics, searching for reasonable footholds in the general area desired. The task of the Foothold Determination heuristic was to guide the scanning system's search. To do this it provided three pieces of information to
the scanning system. One was the normal vector to a plane, $\vec{K}$. This plane was more or less vertically oriented, and the foothold search was to take place in it. The second piece of information was an actual point in this plane, $P_z$, that indicated the lateral displacement of the plane from the machine, and the point from which the search should proceed forward. That is, the scanning system had to return a foothold that was ahead $P_z$. The last piece of information was a maximum distance, $d_m$, that the foothold could be away from $P_z$. After the search was completed using this information, the scanning system simply returned the new foothold location, $P_f$.

### 4.9.3 Front foot placement strategies

Two main strategies were used in guiding the front foothold selection. One was that the foothold should be located ahead of the other front foot by roughly the stride length, $s$. Not having such relative positioning between the two front feet resulted (in earlier versions of this heuristic) in the separation between them decaying as placement errors accumulated. This eventually lead to distortion of the stability triangle pattern. Such a condition occurred for example, if the forward placement distance was made relative to the old foothold on the same side of the machine. The footholds on the two sides of the machine eventually got out of alignment. So to maintain a consistent stability triangle pattern, it was critical that new footholds be located relative to the front foot on the other side of the machine.

The other strategy that was used was to locate the footholds at a fixed "lateral foot placement distance" from the machine, $d_l$. This lateral spacing was not made from the center of the machine, but from a point under the center by the distance $h$. As shown in Figure 4.3, if the machine had any roll to it, to keep the horizontal projection of the center of gravity half way between the horizontal projections of the feet on both sides, the footholds on the machine’s lower side had to be further away from it than those on its upper side. Placing the footholds at lateral distances relative to the point under the machine ensured this. The equal spacing of the horizontal projections was necessary to maintain a consistent stability triangle pattern.

### 4.9.4 Determining the placement plane

Immediately after each front foot placement was made, a new stability triangle was formed, and the machine started a new half-cycle in which it crossed this triangle. To guide the choice of this foothold location, the Foothold Determination heuristic had to determine ahead of time what the trajectory of the machine would be in the new half-cycle. The planned trajectory for the new half-cycle was determined
Figure 4.3: Compensation of the Lateral Foothold Spacing Under Roll Conditions
by the stability triangle corner locations, as explained in Section 4.7, but it was determining one of these corner locations that this heuristic was concerned with. So the planned trajectory could not be known. It could be estimated however. When the transition was made from the old half-cycle to the new one, the trajectory of the machine was defined by the predicted forward orientation vector, $\hat{A}_p$. So to estimate the trajectory in the new half-cycle, the predicted forward orientation vector was rotated about the predicted normal orientation vector by the desired turning rate specified by the operator, $\rho_C$ (Equations 4.15 and 4.16 were used to do this). The result was $\hat{A}$.

Estimating the trajectory in the new half-cycle in this fashion was reasonable. This is because the Foothold Determination heuristic was mainly concerned with the rotational orientation of this trajectory, and the rotational orientation was not greatly affected by the actual foothold location chosen. There were two reasons for this. The first was that, as will be explained in Section 5.6, the rotational settling factor used in the Half-Cycle Movement Planning heuristic was kept very small. Thus the rotational orientation of the trajectory was determined almost entirely by the desired turning rate specified by the operator. The location of the new front foothold affected it little. The second reason was that this heuristic attempted to place the foothold so that it would be in line with the estimated future trajectory. Thus the foothold was even less likely to perturb the actual trajectory away from the estimated one.

Determined in a manner identical to the estimated trajectory, the estimated lateral orientation vector, $\hat{S}$, was normal to the plane in which the machine would move in the new half-cycle. If it followed the estimated trajectory, the center of the machine would move forward in this plane from around the $P_p$ of its current half-cycle towards the $P_p$ of its new half-cycle. For the new foothold to be a fixed lateral distance from the center of the machine, it had to fall in a plane parallel to the one that the center of the machine was going to move through. Thus:

$\hat{K} = \hat{S}$

(4.30)

4.9.5 Determining the search starting point

To ensure that the new foothold was the stride distance ahead of the front foot on the other side of the machine, the point in the placement plane defined by $\hat{K}$ had to be determined that was laterally opposite the other front foot. This point, $P_s$, was along a line in the direction of $\hat{K}$ that started at the other front foot. Now, as stated above, the footholds had to be chosen at distances laterally offset from a point under the center of the machine by a distance of $h$. This point effectively defined a center
plane, whose normal was \( \vec{K} \), that the footholds had to be spaced laterally from. So this center plane was parallel to the placement plane, and the two were separated by the lateral foot placement distance, \( d_l \). Figure 4.3 clarifies this. The line from the other front foot intersected the center plane perpendicularly and at point \( P_c \). If foot 2 was to be placed, \( P_e \) was calculated as:

\[
P_e = P_l + \vec{K}((P_v - h\vec{k}) - P_l) \cdot \vec{K}
\]  

(4.31)

Where \( \vec{k} \) was a unit vector in the vertical direction. From \( P_e \), \( P_o \) was calculated:

\[
P_o = P_e + \vec{K}d_l
\]  

(4.32)

Equations 4.31 and 4.32 were combined to produce:

\[
P_o = P_l + \vec{K}((P_v - h\vec{k}) - P_l) \cdot \vec{K} + d_l
\]  

(4.33)

Once \( P_o \) had been determined, \( P_s \) had to be calculated from it. The initial strategy used was to simply set \( P_s \) equal to \( P_o \) and set the maximum separation distance, \( d_m \), to \( s \). The foothold search then began from \( P_o \) and was constrained to be within the stride length. This did not work however, because there was a minimum distance that \( P_s \) had to be in front of the last foothold on its side of the machine. If the footholds were too close, as shown in Figure 4.4, the stability triangle could not support the machine. There had to be enough space in the center of the triangle for the complete circular stability margin and, as will be explained in the next section, the following distance. If there was not, the machine would stop at the front edge of the new stability triangle, and would still require the old triangle for stability. Thus the rear leg that defined that triangle could not be lifted.

To ensure that the minimum distance requirements were met, the horizontal projection of the line between the two feet that defined the back end of the stability triangle was determined. If foot 2 was being placed:

\[
y = \frac{y_4 - y_1}{x_4 - x_1} (x - x_1) + y_1
\]  

(4.34)

\[
y = m(x - x_1) + y_1 \quad \text{if} \quad m = \frac{y_4 - y_1}{x_4 - x_1}
\]  

(4.35)

The line running from \( P_o \) in the direction of the horizontal projection of \( \vec{A} \) was specified parametrically as:
Figure 4.4: Minimum Stability Triangle Size
\[ z = i_at + x_0 \quad \quad y = j_at + y_0 \]  

(4.36)

The intersection of the two lines was then determined by combining the two equations:

\[ t = \frac{mz_0 - mz_1 - y_0 + y_1}{j_at - m_i_t} \]  

(4.37)

Referring to Equation 4.36, it is apparent that \(-t\sqrt{t_a^2 + j_a^2}\) was the distance from \(P_t\) to the back side of the stability triangle along the horizontal projection of \(\overline{A}\). Now, by considering similar triangles, the total distance from the back side of the stability triangle to \(P_s\) along the horizontal projection of \(\overline{A}\) had to be roughly:

\[ 2(2 \cdot \text{stability margin} + \text{following distance}) \]  

(4.38)

From this, the distance that the two footholds were already separated could be subtracted. This produced a required minimum distance along \(\overline{A}\) of (with the following distance added in twice more for an extra margin):

\[ d_{co} = \frac{4(\text{stability margin} + \text{following distance}) - t\sqrt{t_a^2 + j_a^2}}{\sqrt{t_a^2 + j_a^2}} \]  

(4.39)

\[ d_{co} = \frac{4(\text{stability margin} + \text{following distance})}{\sqrt{t_a^2 + j_a^2}} + t \]  

(4.40)

Where the denominator in Equation 4.39 converted the horizontally projected separation distance into a distance actually along \(\overline{A}\). Finally:

\[ P_s = P_o + \overline{A}d_{co} \]  

(4.41)

4.9.6 Determining the maximum forward placement distance

With \(P_s\) and \(\overline{K}\) determined, it remained only to calculate \(d_m\), the maximum distance that \(P_f\) could be away from \(P_s\). The determination of \(d_m\) depended on whether the foot that was being placed was on the inside of a machine turn or not. As shown in Figure 4.5, during machine turning, the inside footholds had to be closer together. This was because the two sides of the machine were following circles of different radius. The strategy adopted during turning was to space the outside footholds apart by twice the stride.
length, and to space the inside footholds by less than 2s. Now assuming that the stability triangles on
the outside of a turn were perfectly shaped, from Figure 4.5:

\[ \alpha = \tan^{-1}\left(\frac{s}{2d_i}\right) \]  
(4.42)

So, for an inside triangle:

\[ \beta = \frac{\pi}{2} - \left(\frac{\pi}{2} - \alpha\right) - \rho_c \]  
(4.43)

\[ \beta = \tan^{-1}\left(\frac{s}{2d_i}\right) - \rho_c \]  
(4.44)

\[ d_m = b \tan(\beta) - d_{co} \]  
(4.45)

\[ d_m = b \tan(\tan^{-1}\left(\frac{s}{2d_i}\right) - \rho_c) - d_{co} \]  
(4.46)

Where \( b \) was simply derived from Equation 4.33 as:

\[ b = \tilde{R}\left(((P_p - h\tilde{k}) - P_{\text{front foot}}) \cdot \tilde{R} + d_i\right) \]  
(4.47)

To compensate for any movement of \( P_s \) away from \( P_o \), \( d_{co} \) was subtracted in Equations 4.45 and 4.46.

When \( d_m \) had been calculated, it was checked to see if it was less than zero. If so, the operator
specified turning rate was requiring that the foothold be placed behind \( P_s \). This was not allowable, so
\( d_m \) was set to zero and a new turning rate was calculated by reversing the process above:

\[ \rho_a = \tan^{-1}\left(\frac{s}{2d_i}\right) - \tan^{-1}\left(\frac{d_{co}}{b}\right) \]  
(4.48)

This revised turning rate, \( \rho_a \), was temporarily used by the Half-Cycle Movement Planning heuristic
in place of \( \rho_c \), when the next half-cycle began. This allowed this heuristic to make a movement plan
that better suited the foot placement that had been made. When the turning rate was modified in this
fashion, a message was sent to the operator advising him that he was attempting too difficult of a turn
on the current terrain.

If \( 2d_i \) and \( b \) were set to \( w \), Equation 4.48 could also used to determine the maximum rate of turning.
On flat terrain, with perfectly shaped stability triangles, this was defined when:

\[ d_{co} = 2(\text{stability margin} + \text{following distance}) \]  
(4.49)
Figure 4.5: Stability Triangle Sizing During Machine Turning

\[ \theta = \beta + \left( \frac{\pi}{2} - \alpha \right) + \frac{\pi}{2} + \rho_c \]

\[ \beta = \alpha - \rho_c \]
If the foot placement was occurring on the outside of a machine turn, or if no machine turning was taking place, \( d_m \) was set to:

\[
d_m = s - d_{eo}
\]  

(4.50)

Where \( d_{eo} \) was again subtracted to compensate for any movement of \( P_s \) away from \( P_o \). A check was also made to ensure that the new foothold would not be too far away from the last one on that side:

\[
d_h = (P_s - P_{last foothold}) \cdot \hat{A}
\]

(4.51)

if: \( d_h + d_m > 2s \) then \( d_m = 2s - d_h \)  

(4.52)

4.9.7 Scanning system following of the placement rules

In the manner described above, \( \vec{K}, P_s \) and \( d_m \) were calculated. These values were passed on to a scanning system that located the actual foothold, \( P_f \), and returned its position. The scanning system sought the foothold in the vertical plane defined by \( \vec{K} \) and \( P_s \), and it restricted its search to the area in front of \( P_s \). It attempted to locate the foothold at a distance \( d_m \) away from \( P_s \). These rules did not have to be strictly followed however. Realities of the terrain, such as roughness, small obstacles, and unfirm areas, could result in footholds being chosen that were not in the specified plane. The distance to the foothold could also be highly variable. Although it was preferred if \( P_f \) was exactly \( d_m \) in front of \( P_s \), it could be closer. This was expected to occur. \( P_f \) could also be further away than \( d_m \), although this was less desirable, as the risk of a leg over-extension increased. One rule that could not be broken was that \( P_f \) always had to be in front of \( P_s \). If it was not, a machine failure would inevitably result.

4.10 The Check Stability Heuristic

This heuristic simply determined whether the machine was within the stability region defined by its supporting feet. To be stable, the horizontal projection of the center of gravity of the machine had to be within the horizontal projection of the region defined by the supporting feet. Given an indication of which legs were actually supporting the machine, this heuristic returned a binary value indicating whether the machine was stable or not. To determine this, it performed simple vector calculations to find the distance from the machine's center of gravity to the closest stability region edge. For example,
consider how the calculation was made for the edge running between feet 1 and 3. The horizontal projection of the vector defining this edge was:

\[ \vec{t}_e = x_3 - x_1 \quad \vec{f}_e = y_3 - y_1 \]  

(4.53)

Now the perpendicular to this edge was:

\[ \vec{t}_p = \vec{f}_e \quad \vec{f}_p = -\vec{t}_e \]  

(4.54)

So the distance to this edge along the perpendicular to it, was determined by projecting a vector to either of the two feet defining the edge unto \( \vec{P} \):

\[ m = \frac{\vec{V}_1 \cdot \vec{P}}{|\vec{P}|} = \frac{x_1 \vec{t}_p + y_1 \vec{f}_p}{\sqrt{\vec{t}_p^2 + \vec{f}_p^2}} = \frac{x_1 \vec{f}_e - y_1 \vec{t}_e}{\sqrt{\vec{t}_e^2 + \vec{f}_e^2}} \]  

(4.55)

These distances were calculated for each supporting edge, and the smallest distance then had to be greater than the prescribed stability margin for the machine to be verified as stable.

There was one special case dealt with by this heuristic. When the machine was approaching the edge defined by the two front feet while one of its rear feet was being swung, the required distance to that edge was increased. That was because when the rear foot placement was made, the foot was placed behind the corresponding front foot by the following distance. When the front foot was lifted, this rear foot location defined the new front edge of the stability region. This had the effect of shifting the front edge back slightly (Figure 3.2 clarifies this). So if the machine had moved right up to the front edge of the stability region before the exchange of the rear and front feet, it would have been immediately defined as unstable. To prevent this, the required stability margin for this edge was increased by the following distance. This extra margin was more than enough because the center of the stability region edge, the place where the machine would likely contact it, only shifted back by roughly half the following distance.

### 4.11 The Speed Calculation Heuristic

The three heuristics that performed the bulk of the calculations necessary to determine the desired joint angles for each sample time were the Speed Calculation, Swing Leg Kinematics, and Body Kinematics heuristics. The rest of the heuristics were concerned with longer term planning, determination of machine
status, or simply aided these three heuristics. The Speed Calculation heuristic only determined one value, the desired speed of the machine, \(v_i\), but all of the other sample time motion planning was based upon this. To determine the desired joint angles of both the swinging and supporting legs, the speed, and thus the time required before the machine reached various target points, had to be known. The speed also had to be carefully determined to ensure smooth body motion.

For a given half-cycle of movement, the speed calculation was largely based on the time required to swing the two legs into place. As the distance to \(P_p\), the planned center of gravity location at the end of the half-cycle, was known, the time required to cover this distance had to be determined. This time, as discussed in Section 3.6, depended entirely on the time required to swing the two legs. So accurate estimates of the swing times were essential.

### 4.11.1 Determining the termination point

The first point considered in the speed calculation was that the defined final center of gravity position, \(P_p\), was not a realistic one. It was a location on, or near, the edge of the stability triangle and could not be approached at the end of the half-cycle. If the machine had done so, it would simply have had to halt some distance away from it to avoid becoming unstable. Halting in this manner every half-cycle would have resulted in a very inconsistent "ride". So a new termination point, \(P_t\), was calculated that was a "buffer distance" back from the stability region edge, but in the same direction as \(P_p\). The buffer distance was empirically chosen so that it was large enough to prevent the machine from having to halt, but not so large that the machine ended the half-cycle at a point far behind where it could have safely moved:

\[
P_t = \left( |P_p| - \frac{\text{buffer distance}}{\cos \psi} \right) \frac{\vec{P}_p}{|P_p|} \tag{4.56}
\]

The termination point could have been calculated in a different manner. It could have been determined once when \(P_p\) was, and then moved relative to the machine by the Assess Movement heuristic in the same manner as \(P_p\). It was felt however, that even if the machine moved laterally and vertically off the original trajectory defined by \(P_p\), it should always keep heading towards this point. Thus \(P_t\) constantly had to be recalculated to be in the direction of \(P_p\).
4.11.2 Speed calculation during the front leg swing phase

Once $P_t$ was determined, one of three strategies was followed to calculate the leg swinging time. This depended on what phase of the half-cycle movement the machine was in. The first phase was when the machine was creeping forward so that it could swing its rear leg. The second was when this leg was being swung. The third was when the front leg was being swung. If the front leg was being swung, the ideal speed of the machine was simply that which would bring it to $P_t$ in exactly the time required to complete the leg's forward swing. The forward swing time of the leg depended on its horizontal angular change. As the current position of the leg was known, only the final angle of the leg had to be determined. This was calculated by first constructing a vector from the predicted shoulder location, when the foot touched down, to the foothold (see Figure 3.1):

$$\vec{D}_p = \vec{P}_t - (\vec{P}_f + \vec{j}_p)$$

(4.57)

Then $\theta_f^i$ was determined as follows:

$$\theta_f^i = \tan^{-1} \left( \frac{\vec{D}_p \cdot \hat{A}_p}{\vec{D}_p \cdot \hat{S}_p} \right)$$

(4.58)

Including the calculation of the time required for the leg to swing forward, the ideal speed was calculated as:

$$t_f^i = \frac{\theta_f^i - \theta_f^l}{\omega_{\text{max}}}$$

(4.59)

$$v_i = \frac{|P_t|}{t_f^i}$$

(4.60)

4.11.3 Speed calculation during the rear leg swing phase

If the rear leg was being swung forward, a more complex calculation had to be made. This complexity resulted from the fact that there was no specified position that the center of gravity had to be in when that leg touched down. It was possible to determine this position however, so that a constant speed would be maintained throughout both leg swinging phases. During the rear leg swing phase, the center of the machine moved an arbitrary distance, $d_r$, towards the termination point, $P_t$. During the front leg swing phase, its center moved the remaining distance, $d_f$, to the termination point:

$$|P_t| = d_r + d_f$$

(4.61)
So to maintain constant speed:

\[
\frac{d_f}{t'_f} = \frac{d_r}{t'_r}
\]  

(4.62)

Which was manipulated into a proportionality equation:

\[
p = \frac{d_r}{|P_1|} = \frac{t'_r}{t'_f + t'_r}
\]  

(4.63)

Now if a future body position was specified for when the rear leg touched down, it was trivial to calculate the distance that had to be travelled to that position, \(d_r\). The swing time of the rear leg, \(t'_r\), could also be calculated, in a manner identical to that of the front leg swing time calculation (using Equations 4.57 to 4.59). The difficulty lay in determining the total swing time, as the front leg swing time, \(t'_f\), was not known and could not be known. This was because the final angle of that swing depended on where the new front foothold was to be, a determination not made at this point. It was possible to calculate the starting angle of the front leg swing from the rear leg placement angle, but the final angle had to be estimated. This was done by considering previous front leg placement angles. Over terrain of consistent attitude, it was expected that there be some consistency in the final angles. This was because the machine, with its relatively fixed stride length, should pass through roughly similar states as it completes each walking cycle. So an estimate of the typical final angles was maintained, in the Update Swing Leg heuristic, for each of the two front legs.

There was some question as to whether using an estimated value for \(\theta'_f\) was reasonable. There was little choice though, as it was not possible to get an actual value. Further, the speed calculation that used this value did not have to be highly accurate during the rear leg swing phase, so errors in the estimate were not an enormous concern. This was because, as stated, the machine position at the end of this phase was arbitrary. It was only the speed consistency between the rear and front foot swing phases that was affected by speed miscalculations.

The starting angle of the front leg swing was calculated from the predicted final angle of the rear leg swing, as shown in Figure 4.6. Ignoring the slight difference in the position of the two feet, the follow the leader gait ensured that:

\[
d'_f = d'_r = \bar{D}_p \cdot \bar{S}_p
\]  

(4.64)
Chapter 4. Heuristic Specification

\[ d'_d = \text{distance between shoulders of two legs} - d'_a \]  
\[ d'_a = \text{distance between shoulders of two legs} - \tilde{D}_p \cdot \tilde{A}_p \]

From these values the following was calculated:

\[ \theta'_i = \tan^{-1} \left( \frac{d'_i}{d'_a} \right) \]  
\[ t'_i = \frac{\theta'_i - \theta'_0}{\omega_{\text{max}}} \]

So given a machine body position where the rear leg was to touch down, and the equations derived above, the swing time values that would result from this position could be determined. The inverse was not true however, as there was a nonlinear relationship, due to the inverse tangent calculation, between the distance and time equations. That is why Equation 4.63 was put in a proportional format. The proportional value, \( p \), had to be solved iteratively. This was accomplished with a binary search algorithm that went through 20 cycles. The algorithm attempted to match \( p_d \), the distance proportionality factor, and \( p_t \), the time proportionality factor. It operated as follows:

**The Algorithm to Determine Speed When a Rear Leg is Swinging**

set UPPER to 1, LOWER to 0

repeat 20 times:

\[ p_d = \frac{1}{2}(\text{UPPER} + \text{LOWER}) \]

set \( \tilde{P}_i \) to \( p_d \tilde{P}_i \) (effectively \( d_r \)) in Equations 4.57 to 4.59 to determine \( t'_* \)

use Equations 4.64 to 4.68 to determine \( t'_i \)

\[ p_t = t'_* / (t'_* + t'_i) \]

if \( p_t < p_p \):

the machine travelled too far resulting in swing being too small,

so set UPPER to \( p_p \) to keep search below this value

otherwise set LOWER to \( p_p \) for converse reason

After this search ended, the final values of \( t'_* \) and \( p_d \tilde{P}_i \) (effectively \( d_r \)) were used to determine the ideal speed in Equation 4.60. The speed calculated in this manner should remain consistent throughout the remainder of the half-cycle.
It should be noted that the iterative search covered a complete range. It sought a proportionality value between 0 and 1, covering all of the possible machine positions along the line to \( P_t \). A valid position was sure to exist along this line as long as the initial location of the rear foot was behind its new foothold. In fact, there were an infinite number of positions available, each of which represented various combinations of machine and swing leg motion. The purpose of the iterative search was to find the one combination that met the requirement of producing consistent speed throughout the remaining two phases of the half-cycle.

4.11.4 Speed calculation during the creeping phase

The final phase of machine motion that had to be considered during the half-cycle was when the machine was creeping. During this period, the machine was moving forward so that the rear leg could begin its swing. The creeping phase was one in which very high forward speed was possible. As all four legs were placed, the forward swinging time of a leg that was repositioning was not the limiting factor. The four legs could have driven the machine forward at near their maximum angular speeds. As discussed in Section 3.6, this would have resulted in a machine speed roughly three times that of when a leg was being swung. It was decided that operating the machine at such speeds was not advisable however. Doing so
would have resulted in large accelerations at the beginning and end of the creeping phase, disrupting any speed consistency. The strategy adopted was to set the speed during this phase to the higher of the two speeds before and after the phase. As the speed before the creeping phase was known, it remained to estimate the speed during the upcoming rear leg swing phase.

The speed during the rear leg swing swing phase, \( v_r \), was estimated by first determining where the machine would be when that phase began. This point, \( P_r \), was determined by doing a simple vector calculation to find where the horizontal projection of the line to \( P_t \) intersected the horizontal projection of the rear edge of the next stability triangle. If leg 3 was to be lifted when the machine entered the next stability triangle, the horizontal projection of the rear edge of that stability triangle was defined by the line:

\[
y = \frac{y_4 - y_1}{x_4 - x_1} (x - x_1) + y_1 \quad (4.69)
\]

\[
y = m(x - x_1) + y_1 \quad \text{if:} \quad m = \frac{y_4 - y_1}{x_4 - x_1} \quad (4.70)
\]

The horizontal projection of the line running to \( P_t \) was specified parametrically as:

\[
x = x_t t \\
y = y_t t \quad (4.71)
\]

The two equations were combined to produce:

\[
t = \frac{y_1 - m x_1}{y_t - m x_t} \quad (4.72)
\]

Referring to Equation 4.71, it is apparent that \( t \sqrt{x_t^2 + y_t^2} \) was the distance to \( P_t \). Now the machine would not start its rear leg swing until it had passed the rear edge of the next stability triangle by the stability margin distance. This distance was also a horizontally projected one, so it was added in as follows:

\[
P_r = \frac{\vec{P}_t}{|P_t|} \left( t \sqrt{x_t^2 + y_t^2} + \text{stability margin} \right) \frac{|P_t|}{\sqrt{x_t^2 + y_t^2}} \quad (4.73)
\]

\[
P_r = \vec{P}_t \left( t + \frac{\text{stability margin}}{\sqrt{x_t^2 + y_t^2}} \right) \quad (4.74)
\]

Where the last multiplicand in Equation 4.73 converted the horizontal projection total into a distance along the line to \( P_t \).
Once \( P_r \) had been found, it was used as the current position in the rear leg lift phase speed calculations presented in the previous subsection. These calculations also required a value for the initial horizontal angle of the rear leg, \( \theta_i^r \). During the actual rear leg swing phase, the current value of that angle was used, but in this case it had to be estimated. This was done as follows:

\[
\begin{align*}
\vec{D} & = \vec{P}_{\text{swing foot}} - (\vec{P}_r + \vec{J}_p) \\
\theta_i^r & = \tan^{-1} \left( \frac{\vec{D} \cdot \vec{A}_p}{\vec{D} \cdot \vec{S}_p} \right)
\end{align*}
\] (4.75) \hspace{1cm} (4.76)

Using \( P_r \) and \( \theta_i^r \) in the search algorithm above provided an estimate of the speed that the machine would adopt when it began the rear leg swing.

When the speed at the end of the creeping phase had been estimated, \( v_r \), the higher of it and the current speed was made the ideal speed, \( v_i \). If \( v_r \) was chosen, the machine accelerated towards it. If not, the machine maintained its pace. In the latter of these two cases, extra calculations were made to prepare the machine for the end of the creeping phase. As will be described in the next subsection, the LOGHIAQ heuristics were subject to maximum acceleration limits. So to ensure that the machine speed was near \( v_r \) when the rear leg swing phase began, the machine was made to accelerate negatively during the end of the creeping phase. The acceleration required to meet the new speed specification was calculated as follows:

\[
\frac{dv}{dt} = \frac{v_r^2 - v^2}{2|P_r|}
\] (4.77)

When the magnitude of this acceleration was greater than 0.9 times the maximum acceleration per sample time, \( v_i \) was set to \( v \) less the maximum acceleration per sample time.

A number of other strategies were explored to determine speed during the creeping phase. Most allowed greater speeds than the finally selected strategy. This was done to take advantage of the high speeds possible during this phase. All of the strategies were found to result in highly inconsistent speeds however, and were thus discarded as viable alternatives. The final strategy selected performed extremely well, except, as will be noted in Chapter 6, when the machine was turning.

### 4.11.5 Speed control strategies

The speed calculations involved one final stage that ensured reasonable machine performance. This stage limited the overall speed changes that could occur during any sample time to defined maximum
accelerations. If, as a result of the calculations described in this section, \( v_i \) differed from the current speed, \( v \), more than was permitted, it was set to \( v \) plus the appropriate maximum acceleration value. In addition, this stage limited the minimum speed to values based on the maximum acceleration. The creeping speed was maintained at three times the maximum acceleration, and the speed during leg swings was maintained at the maximum acceleration. These lower limits were chosen empirically.

4.12 The Swing Leg Kinematics Heuristic

Once the ideal speed of the machine had been determined by the Speed Calculation heuristic the position of the machine in the future became known. The machine would simply be a distance along the line to \( P_x \) determined by the ideal speed and the time it had to move in. Thus the desired joint angles of the swinging and support legs could be calculated to match this motion. This heuristic determined the desired joint angles of the swinging leg. If the machine was in a creeping phase, this heuristic was not used.

4.12.1 Determining the placement state

The swing leg calculations first involved estimating how many more sample times were needed to complete the current leg swing. The basic strategy behind the swing leg control was to swing the leg in the horizontal direction at the maximum angular velocity, \( \omega_{\text{max}} \). The extension and vertical angle movements were then calculated to match the horizontal swing. At the start of the swing, the leg was lifted rapidly to a clearance height. It was kept there until the horizontal swing neared its end, and then dropped down so that the foot touched the ground just as the horizontal rotation finished. Further details on how the three angular changes were made relative to each other will be provided later in this section. The horizontal angle typically underwent the largest change though, so the time required to complete this change had to be determined first. The heuristic did this by assuming that the idea speed determined in the Speed Calculation heuristic would be constantly maintained. Then an iterative approach was taken to determine when the foot would touch down with the machine travelling at that speed:
The Algorithm to Determine When the Swing Leg Will Touch Down

set VALID to false

set REMAINING to 1

set $\theta_{\text{trial}}$ to $\theta$ of swing leg

$P_{\text{trial}} = [0, 0, 0]$ (the present location)

repeat until a position is VALID or $\theta_{\text{trial}}$ goes well beyond the horizontal angle limit:

$P'_{\text{trial}} = P_{\text{trial}} + v_t \cdot \text{duration of each sample time}(P/P_p)$

$\vec{D} = P'_{\text{trial}} - (\vec{P}_t + \vec{J}_p)$

$d_a = \vec{D} \cdot \vec{A}_p$ (see Figure 4.2 for clarification)

$d_s = \vec{D} \cdot \vec{S}_p$

$d_n = \vec{D} \cdot \vec{N}_p$

$\theta_p = \tan^{-1}(d_a/d_s)$

$\epsilon_p = \cos^{-1}((L_U^2 + L_L^2 - |D|^2)/(2L_UL_L))$ (cosine law)

$\phi_p = \tan^{-1}(d_n/\sqrt{d_a^2 + d_s^2}) + \sin^{-1}(L_L \sin \epsilon_p/|D|)$ (see Figure 4.7 for clarification)

if all of these angles are in the leg movement range set VALID to true

$\theta'_{\text{trial}} = \theta_{\text{trial}} + \omega_{\text{max}}$

if $\theta'_{\text{trial}} < \theta_p$ set VALID to false

(this is the case when there has not been enough time for the leg to swing into place)

if VALID is false increment REMAINING

To summarize the above, the iterative scheme operated by starting at the current center of gravity location and successively adding, in the direction of travel towards $P_p$, units of distance equal to those which could be traversed in one sample time at the ideal speed. Simultaneously, the maximum angular change that could occur in one sample time was repetitively added to the initial horizontal angle. Thus the state of the machine at points in the future was estimated. When a state was reached where all the joint angles were within the range of motion of the swing leg, and $\theta_p$ was less than that of the accumulated horizontal swing, the search was terminated. The first state had been found in which the foot placement could occur.

The heuristic also handled cases in which backwards swinging was necessary. It did this by first checking to see if the swing leg, with the machine in its current position, should be swung backwards to reach the foothold. If so, it used the same iterative search algorithm, but subtracted $\omega_{\text{max}}$ from $\theta_{\text{trial}}$. 

during each cycle instead of adding it. The verification of the size of \( \theta_{\text{trial}} \) was also reversed.

If the swing leg was being swung forward, and the foothold was within reach of the machine, a valid foot placement state would almost always be found that entailed continuous forward swinging. This was because the foot movement can be seen as being composed of two different actions. One was movement in a lateral plane so that the foot was positioned directly behind the foothold, and the other was forward movement to meet the foothold. The latter movement consisted of both the forward leg swinging and the actual machine motion. Now as long as the swing time was large enough for the former movement to occur, there was no way for the foothold to be missed. With the lateral preparations made, the foot moved forward towards the foothold with a combination of machine movement and leg swinging, and when it got to it, simply dropped down into place. Unless the foothold was displaced laterally a large distance, it could not miss the foothold.

4.12.2 Hovering

If no valid future state was found in which the foot placement could occur, this heuristic operated under the assumption that as the machine moved forward, the foothold may become accessible. So as the machine continued on at the ideal speed, this heuristic swung the leg into a forward position that would best prepare it for when the foothold was accessible. The extension angle, \( \epsilon \), was left as it was when the foot started its swing. The vertical angle, \( \phi \), was set to an empirically defined "ground clearance angle". If either front leg was being swung, \( \theta \) was set as close to \( \theta_p \) as possible within the movement limits of the leg. These settings made the leg "hover" over the foothold preparing to reach for it when it became accessible.

4.12.3 Calculating the joint angles with a valid placement state

If a valid foot placement state was found, the three swing leg angles were controlled carefully to ensure a smooth swing with a well timed placement. One objective of this control was to keep the leg as high above the ground as possible while it swung, without allowing the vertical movement times to affect the overall swing time. This portion of the heuristic operated as follows, where \( \text{REMAINING} \) was the number of sample times that were required to complete the horizontal swing (calculated as described previously):
The Algorithm to Determine the Swing Leg Joint Angles

if the leg has just lifted, increase $\phi$ maximally
(the other angles were kept as they were to insure ground clearance)
otherwise:
  calculate sample times needed to make $\epsilon$ change to $\epsilon_p$
  if this is $> REMAINING$, reset $REMAINING$ to this value
  calculate sample times needed to make $\phi$ change to $\phi_p$
  if this is $> REMAINING$, reset $REMAINING$ to this value
  (these would be unusual, but not impossible events)
  increase or decrease $\epsilon$ maximally until it reaches $\epsilon_p$
if $REMAINING$ is 1 (foot is ready for placement) or if touchdown already initiated:
  (initiate or continue touchdown sequence)
  set $\theta$ to $\theta_p$
  decrease $\phi$ maximally to drop leg
  activate touchdown sensing
otherwise:
  increase or decrease $\theta$ maximally until it reaches $\theta_p$
  if the sample times required to decrease $\phi$ to $\phi_p$ are $> REMAINING$ — margin:
    decrease $\phi$ maximally
  otherwise if $\phi <$ empirically determined ground clearance angle:
    increase $\phi$ maximally
  correct $\phi$ if it has exceeded either of its movement limits

The "margin" included above ensured that the leg began dropping into place a short time before it was absolutely necessary for it to do so. This prevented any movement errors from resulting in a situation where changes to $\phi$, and not $\theta$, determined the time required for the swing to complete. The margin was set to 2.5 after empirical observation of performance.

To prevent jerky motion of the leg in the vertical plane, changes to $\phi$ were passed through an equation that effectively lowpass filtered them:

$$\theta' = \theta + \frac{(\theta_d - \theta) + \Delta \theta}{2}$$  (4.78)
Where $\theta_d$ was the desired joint angle, and $\Delta \theta$ was the difference between $\theta'$ and $\theta$ during the last sample time.

In this manner, given only a speed specification, the Swing Leg Kinematics heuristic determined how the joint angles of the swinging leg should be altered to provide a fast, smooth, ground clearing transition. The heuristic also calculated the estimated number of sample times remaining before foot placement, $REMAINING$. This value was used in the Body Kinematics heuristic, as will be described Section 4.15.

### 4.13 The Approximate Step Times Remaining Heuristic

During the creeping phase of a half-cycle, no leg was being swung, so the Swing Leg Kinematics heuristic was not used. Thus no value was calculated for the number of sample times remaining before the next foot placement. This value was needed by the Body Kinematics heuristic, and so had to be determined somehow. The Approximate Step Times Remaining heuristic approximated it as follows:

$$REMAINING \approx \frac{|P_t| - \text{buffer} \cdot \cos \psi}{2v_i}$$

(4.79)

### 4.14 The Adjust Attitude Heuristic

The goal of the heuristics described in this paper was to facilitate continuous smooth walking over rough terrain with the highest possible speed. It was possible for the heuristics to fail however. They could place the machine in a state where it had to halt all further forward progress, and where there were no options available to restart movement. With the many degrees of freedom of a quadruped, it would be an enormous computational task to prevent all possible undesirable states. Doing so was not the goal of the LOGHAQ heuristics, instead it was to guide the machine in a manner that kept it away from such states. This section will consider in detail the conditions that brought about failure and how these were avoided. It will also describe how some failure conditions were solved using the Adjust Attitude heuristic.
4.14.1 Preventing failures by making footholds accessible

Failures were defined as occurring when a complete halting of the machine took place. That is, when all leg movements had stopped. As the goal of the LOGHAQ heuristics was to maintain smooth walking, this was really the only absolute form of failure that could occur. Inconsistent or excessively slow movement could have been produced by the heuristics, and such was not desired, but as long as movement continued, the heuristics had not completely failed. To be fully halted, the machine must have stopped all movement. In addition, if any leg was swinging, that leg must have halted its swing. This could only occur when the leg could not reach the foothold chosen for it. That is, when the Swing Leg Kinematics heuristic was not able to access the foothold, and had moved the leg to a hovering position above it. If the swing leg could reach its foothold, it moved continuously until it did so. Thus, if a leg was swinging, failure could only occur if that leg was unable to reach its foothold. The purpose of this heuristic was to adjust the attitude of the machine in this case, in an attempt to make the foothold accessible. If this was accomplished, the foot could be placed, and movement could continue. This would prevent a failure.

When the machine halted with a swing leg in the air that could not access its foothold, one of two strategies was used to make the foothold more accessible. One strategy determined the attitude adjustments necessary when a front foot could not reach its foothold. The other determined the adjustments when a rear foot was unable to touch down. In either case, adjustments were made by replacing the target body attitude and rotational orientation angles with new ones that oriented the machine in a more advantageous direction. Neither strategy included translational movement. Possible adjustments could have included vertical movement, or "backing up", but the former was felt to be unwise without operator control, as the ground clearance was affected, and the latter would not aid in reaching footholds that were already further ahead than the machine could reach. So both strategies were restricted to rotation of the machine about its current center of gravity.

4.14.2 Adjusting the attitude to facilitate front foot placement

To make the front footholds more accessible, the following operations were performed:
Chapter 4. Heuristic Specification

The Algorithm to Determine Attitude Adjustments

set REMAINING to 1
(this is done because the attitude adjustment is calculated to happen one sample time at a time)
calculate the normal orientation vector (use Equations 4.1 to 4.7)
\[ P_1 = P_f + \hat{N}h \]
if leg 1 is being swung set \[ \sigma_t = \tan^{-1}\left(\frac{z_1}{|z_1|}\right) \]
otherwise set \[ \sigma_t = -\tan^{-1}\left(\frac{z_1}{|z_1|}\right) \]
ensure that \( \sigma_t \) does not differ from \( \sigma \) by more than one third of the maximum body roll rate
recalculate the complete set of orientation vectors using \( \sigma_t \) and the current tilt
\[ P_f = P_f + \hat{N'}h \] (using the new normal orientation vector)
\[ \bar{U} = \hat{S}' \times \bar{P}_1 \]
\[ \bar{L} = \bar{U} \times \hat{S}' \]
\[ \psi_t = \tan^{-1}\left(\frac{\hat{k}_t}{\sqrt{\hat{r}_t^2 + \hat{s}_t^2}}\right) \]
ensure that \( \psi_t \) does not differ from \( \psi \) by more than the maximum body tilt rate

This attitude changing strategy basically rotated the machine so that a vector from the center of the machine to the point \( P_1 \), at height \( h \) above the foothold would be in the plane defined by the platform of the machine. This positioning ensured that the shoulder was well situated for the leg to reach the foothold if such was possible at all. Under this one constraint however, the platform could adopt an infinite number of attitudes. This was because the plane could effectively rotate about the described vector. The heuristic operated by choosing as its final plane, the one that consisted only of machine rolling. The rate of rolling towards this plane was limited so that it occurred at one third of the maximum allowable roll rate. While the gradual rolling took place then, tilt angles were calculated that would adjust the platform attitude so that the vector to \( P_f \) was contained in it. The tilting required to do this was allowed to take place at the full allowable tilt rate. Thus the attitude adjustment initially consisted mainly of tilting towards \( P_f \). Later, as the roll increased, the tilting abated and the attitude adjustment consisted mainly of rolling towards \( P_f \). If the foothold become accessible at any point in this process, the attitude adjustment stopped. If the foothold never became accessible, the process completed and a failure was declared. If, in the course of adjusting the attitude, a leg was brought to the limits of its range, the process also stopped and a failure was declared.
It was preferred that the machine tilt towards the foothold more than it rolled towards it. Excessive
tilting could be recovered from more easily and did not have as drastic effects on the range of operation of
the supporting legs. As well, it was felt that should the machine platform contain a seated operator, this
operator would be able to handle tilting more easily than rolling. This heuristic adjusted the attitude
in a manner that made machine tilting more likely than rolling. It started off by mainly tilting, and
only if this did not make did the foothold accessible did it continue on and began rolling the machine
excessively.

The machine height, \( h \), was added to \( P_j \) in the direction of the normal orientation vector, not, as
would be expected, in the vertical direction. This was because the goal of this heuristic was to make the
foothold accessible. This was best accomplished by having the shoulder offset from the foothold by \( h \) in
the platform normal direction. If \( h \) had been added to \( P_j \) in the vertical direction, the offset would have
been foreshortened for any nonhorizontal machine attitudes, and the leg access range then have been
reduced.

4.14.3 Adjusting the rotational orientation to facilitate rear foot placement

To make the rear footholds more accessible, the machine could only be rotated. Adjusting the attitude
would have had little effect as the rear footholds were not out in front of the machine, they were along
side it. The front footholds were constrained by the Foothold Determination heuristic to be in the
forward range of the machine. When they were inaccessible, it was typically because they were out of
range laterally or vertically (relative to the machine). Thus adjusting the attitude in their direction was
beneficial. When rear footholds were inaccessible, it was typically because they were too far forward.
That is, the machine had not advanced enough to bring them in range. They were not likely to be out
of range in other directions, as they were located immediately behind the front feet, and the LOGHAQ
heuristic package was designed to ensure that once the feet were placed, they moved down the length of
the machine at a fairly close distance. So, when a rear foothold was inaccessible, the strategy adopted
was to move it further down the length of the machine. This was accomplished by rotating the back end
of the machine, on the side of the swinging foot, towards the foothold. The heuristic simply set the target
rotational angle each sample time to half of the maximum body rotation rate in the direction required.
The heuristic did not move the machine towards a final orientation goal as it did when adjusting the
attitude to reach front footholds. It just kept rotating the machine until either a leg reached its range
limit, in which case a failure was declared, or the foothold became accessible. This part of the heuristic,
like the part that facilitated front foot placements, made the orientation changes one sample time at a
time. Thus it also set \textit{REMAINING} to 1, indicating that the changes specified would take place in one
sample time.

4.14.4 Halting conditions and the applicability of the Adjust Attitude heuristic

As detailed above, if the machine halted and it had a swinging leg in the air that could not reach its
foothold, the Adjust Attitude heuristic attempted to alter the machine orientation to bring the foothold
in range. It will now be examined under what conditions the machine had to halt, and in which cases
the heuristic was useful.

There were two possible causes of forward body movement cessation:

1. The center of gravity (buffered by the stability margin) had reached the front edge of the stability
region.

2. A support leg had reached the limit of its range.

The former of these was most likely occur during the front leg swing phase of a half-cycle. The other
two phases of the half-cycle, the creep into the new stability triangle and the rear leg swing phase,
both took place earlier in the movement across the stability triangle and normally ended long before
the machine approached the front edge of the stability triangle. If the machine reached the end of the
stability triangle during the front leg swing phase, it stopped and awaited placement of the front foot.
Then, if the foot could reach its foothold, the next stability triangle became available, and the machine
continued on. If the foothold was not accessible, the Adjust Attitude heuristic attempted to bring it into
range. Failure of the heuristic to do so resulted in a general failure declaration. It was this situation,
that of the machine reaching the end of the stability triangle with the front foot unable to reach its
foothold, that most often resulted in the Adjust Attitude heuristic being used.

As stated previously, it was unlikely that the machine would reach the front edge of the stability
triangle during the rear leg swing phase. Normally this phase ended with the machine close to the center
of the triangle. The machine was most likely to completely cross the triangle only when it could not
make the rear foot placement. In that case, it continued moving, with the swing leg in the hovering
position, until the triangle had been crossed. At this point the Adjust Attitude heuristic was used to
facilitate placement. If it failed to do so, a general failure was declared. Although rare, it was possible
for the machine to cross the stability triangle before the rear foot had had time to reach its foothold.
This was most likely to occur when the stability triangle was small. In that case, the machine simply stopped and awaited placement.

The end of the stability triangle could not be reached during the creeping phase. The Foothold Determination heuristic ensured that the stability triangles were always large enough to prevent this from happening.

The second cause of machine body halting was when one of the machine’s supporting legs reached the limit of its range. Normally this meant that the leg was over-extended. That is, the leg had been forced to reach so far outward that one of its joints reached a limit. Typically over-extensions occurred in the backward direction. This was because the legs were extended forward the farthest they would be when their feet were placed, and this placement was not permitted to be made out of the leg range. After the placement was made, the machine moved forward, and the forward extension was continually lessened. It was the rear extension that grew as the machine moved forward, and it is this problem that will first be considered.

The leg that had the greatest backward extension at any time was most likely the one that was next to be swung forward. It was this leg that was farthest through its backward stroke. So in the first phase of a half-cycle, the creeping phase, it was the rear leg that was about to swing that was the most likely to over-extend. If this happened, it indicated that the rear boundary of the next stability triangle was too far forward, and that the feet had been too aggressively placed. That is, the stride length had been made too long relative to the range of the legs. If halting occurred for this reason, there was nothing that could be done, and failure was declared immediately.

Although such an occurrence was not likely, if any leg other than the next one to be swung reached the end of its range during the creeping phase, failure was again declared immediately.

After the creeping phase ended, the rear leg began its forward swing, and it was the leg in front of it that was then most likely to over-extend. If this happened, the machine halted, but the swing leg was able to continue forward. In fact, the more that the front leg was over-extended, the further its foothold, and the new rear foothold immediately behind it, had moved into the range of the rear leg. Thus the more likely it was that the rear foothold was accessible. If the foothold was inaccessible, the Adjust Attitude heuristic attempted to bring it into range, and failing that, a general failure was declared. If the rear foot could be placed, the over-extension of the front leg was no longer a concern, because that leg immediately began a forward swing. So a front leg over-extension in this phase of the half-cycle was not a problem unless, as a result of an unusual machine state, the rear leg could not reach its foothold.
The sudden halting due to such over-extensions resulted in inconsistent machine movement however, and was best avoided by adjusting the stride length to better suit the range of the legs.

If either leg on the other side of the machine over-extended during the rear leg swing phase, the machine again halted and awaited the foot placement. After this had occurred, the front leg began its swing, but the machine had to remain halted until that foot too was placed. At this point failure was declared. The reason for this was that the machine then had to creep into the next stability triangle to facilitate swinging of the other rear leg. It could not move however, because the over-extension problem still existed in one of the two legs on that side. This problem was not eliminated in the course of the two leg swings. In effect, this over-extension failure condition was an early occurrence of the creeping over-extension problem described earlier. Occurrence of this problem indicated that the stride length was far too aggressive.

Over-extensions could also occur when the half-cycle was in its last phase, the front leg swing phase. In this case, it was most likely the rear leg on the opposite side that over-extended. If this happened, it indicated that the stride length was far too aggressive. This over-extension problem was a milder form of that described above, and was again effectively the same as the creeping over-extension problem. When the swinging front leg was placed, the machine entered the creeping phase of the next half-cycle, but the over-extension condition still existed, so the machine could not move forward. Failure was then declared.

Over-extensions were not necessarily the cause of all the machine haltings due to legs reaching the end of their range. Under-extensions, which occurred when the legs were compressed inward too much, also resulted in halting. Under-extensions could occur at any time during a half-cycle, although they were most likely when the feet were laterally opposite their shoulders, as this was when the legs were their shortest. In a given set of conditions, the effects of under-extensions were exactly the same as over-extensions in terms of forcing halts and resulting in failures. They were more easily avoided though, as they were not as directly affected by the series of footholds presented to the machine. Over-extensions typically resulted from poorly positioned footholds that spread the feet out over too large of an area. Having the footholds laterally too close to the machine caused under-extensions, excessive foothold spread did not. The lateral proximity of the footholds was increased by widening the stance. Widening the stance also increased the extension of the legs when they were farthest back in their stride, and thus had to be accompanied by a decrease in the stride length. So trade-offs had to be made between the stride length and the stance width based on empirical analysis of machine performance.
Chapter 4. Heuristic Specification

In summary, if a leg reached the end of its range, a failure resulted in most cases. This was true even when a leg was being swung when the over-extension or under-extension occurred. If a leg was being swung when the machine reached the end of a stability region and halted, the placement of that foot could form a new stability region into which the machine could move. Having a new stability region did not solve leg extension problems. The only time a leg extension problem could be solved was when the leg that had reached the end of its range became the swing leg. This only occurred when a front leg had reached the end of its range, and it was waiting for the rear leg behind it to touch down before it began its swing. All other conditions required additional forward motion of the machine, which was impossible with a leg extension problem. So leg extension problems were serious in nature, and were best avoided by limiting the stride and keeping the stance wide.

In a more complete walking package than the LOGHAQ heuristics, leg over-extension problems could be corrected. A strategy that could be used would be to move the machine backwards somewhat, and then shuffle the feet to reduce their placement spread and make the next stability triangle more accessible. Poor foothold locations could thus be corrected. Doing this was not in the realm of this study however. The goal of this study was to develop a collection of heuristics that provided smooth forward movement. Shuffling the feet of a machine would result in a significant break in its forward flow. This is not an acceptable option if continuous forward motion is desired. The approach that was taken in this study was to avoid the conditions that required this response.

4.14.5 Machine failure summary

In general, failure of the LOGHAQ heuristics to provide continuous movement occurred mainly when an overly aggressive stride length had been used. Footholds chosen too far forward were out of range of the front feet. It was possible to correct this using the Adjust Attitude heuristic, but such action was not desirable. Additionally, a series of aggressive footholds often forced a support leg to over-extend as the machine moved forward, resulting in a more serious problem that could rarely be corrected. Too narrow of a stance resulted in under-extension failures. Widening the stance to solve this required a decrease in stride length. As explained in Section 3.6, it is best to make the stride as long as possible. But long stride lengths resulted in many problems that disrupted continuous machine motion. To prevent failures and to maximize speed, the stride length has to be chosen carefully.
4.15 The Body Kinematics Heuristic

The Body Kinematics heuristic calculated the desired angles of the supporting legs for the next sample time. It did so by determining where the shoulders of the machine would be if the desired movements were made, and then using vectors from the shoulders to the current foot locations to determine what combination of angles the legs should adopt to facilitate this change. The desired body movements consisted of translation, and attitude and orientation changes. The extent of the attitude and orientation changes was determined by the number of sample times remaining before the next footfall occurred. This value was passed to the Body Kinematics heuristic from the Leg Kinematics heuristic as REMAINING.

The heuristic operated as follows:

\[ P_d = v_i \cdot \text{duration of one sample time} \cdot (\vec{P}_f/|P_f|) \]  
\[ \rho_d = \rho_i/\text{REMAINING} \]  
\[ \psi_d = \psi + (\psi_i - \psi)/\text{REMAINING} \]  
\[ \sigma_d = \sigma + (\sigma_i - \sigma)/\text{REMAINING} \]  

Ensure that none of the attitude and orientation changes exceed their maximum amounts.

Calculate the orientation vectors with \(\psi_d\) and \(\sigma_d\) (use Equations 4.1 to 4.7).

Rotate orientation vectors around the vertical axis by amount \(\rho_d\) (use Equations 4.8 and 4.9).

Set \text{RESET} to false.

Calculate the joint angles for each supporting leg:

\[ \theta_d = \tan^{-1}(d_a/d_s) \]  
\[ \epsilon_d = \cos^{-1}((L_U^2 + L_L^2 - |D|^2)/(2L_U L_L)) \]  
\[ \phi_d = \tan^{-1}(d_n/\sqrt{d_a^2 + d_s^2}) + \sin^{-1}(L_L \sin \epsilon_d/|D|) \]  

(see Figure 4.2 for clarification)

If any of these angles are out of the leg movement range set \text{RESET} to true.

If \text{RESET} is true, cannot move, so:

Set all of the desired angles to their current values (for supporting legs only, not the swing leg)
Chapter 4. Heuristic Specification

Figure 4.7: Calculation of the Vertical Angle

\[ \gamma = \tan^{-1} \frac{d_n}{\sqrt{d_o^2 + d_s^2}} \]

\[ \frac{\sin \beta_o}{101} = \frac{\sin (\Phi_o + \theta)}{L_o} \]

The Body Kinematics heuristic was used every sample time, even if the machine was halted. That was because the machine could still make attitude and orientation changes in place.

4.16 The Prevent Leg Collision Heuristic

This heuristic ensured that the rear leg of a machine, as it swung forward, did not strike the front leg. This was the only time that the legs of a machine operating under the LOGHAQ heuristics could strike. The nature of the heuristics, combined with some machine design limitations (to be immediately described), prevented all other possibilities of leg contact.

4.16.1 Preventing contact when both feet were placed

If the two feet on one side of a machine are placed near each other, in the manner of a follow the leader gait, with the rear one slightly behind, the legs can contact each other. It does not seem at first that this could be the case. When seen from above, as shown in Figure 4.6, the two legs, and the machine side between them, simply form a triangle with one vertex at the feet. Should the knees of both legs extend outward beyond their feet though, they can strike. If only one knee is ever allowed to extend past its foot, the legs cannot contact. This is because, while one knee may extend past the vertex of the
triangle shown in Figure 4.6, the other leg will only butt up against this extra extension.

So when the two feet on a side are placed together, to prevent the two legs from striking as the machine moves, it only has to be ensured that at most one knee at a time can extend past its foot. This can be guaranteed if the walking machine is configured as shown in Figure 4.8. In this figure, both legs are rotated towards each other so that their knees are as close as they can be, and they are not touching. The front leg is at its maximum backward horizontal angle, the rear leg is at its maximum forward horizontal angle, and both have a vertical angle of 0°. At these angles, the length of the machine side and the length of the upper leg segments are such that the knees cannot touch. Now the dashed line in Figure 4.8 indicates the furthest range of each knee when viewed from above. If either foot is out of this range, its knee can clearly not extend beyond it. So it can be seen that the two feet can never be in a position where both knees are extended past them. The feet can be in the circular area defined by the knee of either leg, and when they are in these areas one knee can extend past its foot, but the feet cannot be in both areas simultaneously. This is because the two areas have been prevented from overlapping by the configuration shown in Figure 4.8. The feet can be in either knee area, or outside both, but never in both simultaneously.
4.16.2 Preventing contact during the rear leg swing

Contact of the two legs on either side was prevented when both feet were placed by use of the above described configuration. It remained then to ensure that the legs did not contact when one of them was in the swing phase. When the front leg started its swing, it was guaranteed by the configuration of Figure 4.8 not to be in a position that would produce leg contact. From that position it rotated forward quickly while the rear leg rotated backward somewhat more slowly. So the two legs, from the moment of lifting, moved away from each other. This ensured that front leg swings could never result in striking of the legs. This was not true of rear leg swings however. As the rear leg ended its swing, approaching the front leg, it rotated forward quickly while the front leg rotated backwards towards it. During this time, it was possible for the legs to contact, and it was preventing this that this heuristic was devoted to.

The heuristic started by checking that the front knee was extended past its foot. If the front knee was not extended past its foot, there was no contact problem. The nature of the Swing Leg Kinematics heuristic ensured that the rear leg did not swing past the front leg, so if the front knee was not extended past its foothold, the rear leg would simply butt up against the front foot. This would occur regardless of the position of the rear knee. There was no danger of the front knee beginning to extend past its foot when it had not done so previously, because as the leg continued rotating backward it got further away from this condition. The heuristic also checked to make sure that the two legs were actually close enough to strike. So both of the described checks made sure that there was at least of a possibility of leg contact. Then the heuristic performed the following calculations (see Figure 4.2):

\[
\begin{align*}
 h'_k &= L_U \cos \phi' \\
 l'_k &= L_U \sin \phi' \\
 f'_h &= L_U \cos \phi' + L_L \cos(\pi - \phi' - \epsilon') \\
 f'_v &= L_U \sin \phi' - L_L \sin(\pi - \phi' - \epsilon') \\
 k'_h &= L_U \cos \phi^r \\
 k'_v &= L_U \sin \phi^r \\
 f'_h &= L_U \cos \phi^r + L_L \cos(\pi - \phi^r - \epsilon^r) \\
 f'_v &= L_U \sin \phi^r - L_L \sin(\pi - \phi^r - \epsilon^r)
\end{align*}
\]
Where these were the horizontal and vertical distances to the knees and feet of the two legs, as indicated by the subscripts. These values and others are illustrated in Figure 4.9. Now the horizontal distance to where the extension of the front leg intersected the rear leg was calculated using the sine law:

\[ h = \text{distance between the shoulders} \cdot \frac{\sin(\frac{\pi}{2} + \theta_f)}{\sin(\theta_r - \theta_f)} \]  

(4.88)

Next \( i_v' \), the vertical height of the intersection, was determined. If the intersection occurred in the extension range of the rear leg, this value was determined by interpolation as follows:

\[ i_v' = k_v - L \sin(\phi + \epsilon) \frac{f_h - k_h}{h_h - k_h} \]  

(4.89)

Otherwise this value was simply set to the height of the rear foot:

\[ i_v' = f_v' \]  

(4.90)

Thus a value indicating the approximate height at which the two legs would contact, should the rear leg continue swinging forward at the same height, was calculated. This value was then used to determine the horizontal distance that the rear leg should be away from the front shoulder to prevent contact, \( i_h' \). This distance included the horizontal extension of the front leg to the vertical contact point and a safety margin. There were two cases considered. The first was when the height of contact was higher than the front knee, which was the farthest extension of the front leg. In this case:

\[ i_h' = k_h + \text{safety margin} \]  

(4.91)

The second case was when the height of contact was below the knee. This condition occurred as the leg dropped downwards until its foot reached the ground. In this case, \( i_h' \) was interpolated:

\[ i_h' = f_h' + (k_h - f_h') \frac{i_v' - f_v'}{k_v - f_v'} + \text{safety margin} \]  

(4.92)

The limiting distance, \( i_h' \), could be interpreted as a barrier that blocked the forward swing of the rear leg from approaching the front leg too closely. The barrier ran a fixed distance behind the lower segment of the front leg up to the height of the front knee. Then it maintained this distance and continued vertically up to infinity (or at least to the end of the atmosphere). A rear leg that contacted the barrier was held back in its forward progress and, as it dropped towards its foothold, “slid” down the barrier.
When the front leg stretched back far enough to eliminate any extension of its knee past its foot, the barrier effectively disappeared.

Finally, with $i_h$, the maximum forward horizontal angle of the rear leg was calculated:

$$\theta_{\text{max}} = \tan^{-1} \left( \frac{\text{distance between the shoulders} + i_h \sin \theta'}{i_h \cos \theta'} \right)$$  \hspace{1cm} (4.93)

This angle was then used to restrict the forward motion of the rear leg as it approached the front one. It should be noted that this angle was not a fixed one. As the front leg rotated backward, the barrier determining the angle tilted forward until it became vertical and disappeared. The effect of this heuristic then, was simply to block the forward progress of a rear leg that was swinging too aggressively. In the vast majority of cases, the Swing Leg Kinematics heuristic ensured that the Prevent Leg Collision heuristic was not needed. The Swing Leg Kinematics heuristic normally dropped the rear leg down towards its foothold long before it could make contact with the front leg. The purpose of this heuristic was just to ensure that this happened.
Figure 4.9: Rear Leg Swing Contact Prevention

\[ \gamma^\circ = \gamma - \left( \frac{\pi}{2} - \theta^- \right) - \left( \frac{\pi}{2} + \theta^r \right) = \theta^- - \theta^r \]
This chapter will discuss the factors that must be considered in a practical quadruped design. These factors include ones that result from general quadruped characteristics, and ones that result from the constraints placed on a machine that is to use the LOGHAQ heuristics. To illustrate the points presented, and to prepare for the chapter on the simulation of the LOGHAQ heuristics, the model used in the simulations will be developed throughout the chapter.

5.1 Machine Dimensions

As mentioned in Section 3.6, the actual dimensions of a quadruped machine are not very important in themselves. It is the access range of the feet that is the largest concern. The access range of the feet determines how long the stride can be, and thus, the speed at which the machine can operate. As a general rule, the greater the range of the feet the better. The freedom of operation of the machine, and the kinds of terrain conditions it can handle, depend on the extent of the access range. The access range is determined jointly by the dimensions of the machine, the length of the legs, and the angular limits on the movement of the legs. All of these factors can be traded against each other to produce a machine with the desired range of foot access. The considerations when these tradeoffs are made include the power of the actuators available, the useful range of these actuators, and the loads that the joints, legs, and machine platform will bear.

5.1.1 Leg dimensions

The leg dimensions are a function of the capabilities of the available actuators, and of the loading that will be placed on the legs. As a general rule, the legs of a walking machine should be made as long as possible relative to the body. This provides for the greatest foot access range. This is best illustrated by considering the limited range of a large body with very short legs, as compared to a small body with very long legs.
Once a leg length is chosen, it has to be decided how this length should be divided up between the two segments. Now a three degree of freedom leg could be constructed by having a two degree of freedom shoulder and a straight telescoping leg. The advantage that a leg with a knee has over this design is that obstacles between the shoulder and the foot are more easily cleared. The top segment of the leg can pass over them roughly horizontally, and the lower segment can run vertically down to the foot. To take best advantage of this, the two segments should be about the same length. As shown in Figure 5.1, if the upper segment is very short compared to the lower one, the knee will be close to the shoulder, and the lower leg segment will run directly from that close knee to the foot in the manner of a telescoping leg design. As shown in Figure 5.1, if a short lower segment is used, a similar condition occurs. Further, the extendibility range of legs with uneven segments is less than those with even segments. So the legs should be made as long as possible and have equal segment lengths.

5.1.2 Platform dimensions

In general, a longitudinally oriented walking machine should be longer than it is wide. To prevent the legs on each side from striking, there must be enough distance between their shoulders to keep them apart in their range of access. Section 4.16 details how the minimum side length should be determined from given leg lengths and horizontal angular limits to prevent leg contact. This section also states that
Chapter 5. Practicalities of Quadruped Design

the legs should be close enough so that the two feet can move together throughout a wide range. So preferably the shoulders should be placed no further apart than the calculated minimum side length. Thus the desired length of a machine is easily determined once the leg design has been made. The desired width is not as well defined. In fact, as the legs do not enter each other's range across the front and back of the machine, the width is arbitrary. As explained in Section 3.6, increasing the width of a machine does somewhat improve its maximum speed potential, but not significantly. Increasing the width also improves the lateral stability, but the laterally stability margin will normally be much greater than the margin in the direction of travel anyway. In fact, as the legs add to the effective width of the machine stance, even a narrow platform has a very wide stance. Considering that a narrower stance would be of benefit in traversing terrain with obstacles that have to be avoided (Song, 1989), the machine should be made as narrow as is feasible.

5.1.3 Model dimensions

To simulate the LOGHAQ heuristics, a model was developed that was based on the Kaiser Spyder X5M (see Figure 5.2), a commercially available legged machine. The standard Kaiser does not have three degree of freedom legs. It has two degree of freedom legs that are somewhat restricted in their range. The rear legs are rather short and have powered wheels at the end of them. The front legs are somewhat longer with simple foot pads. In the model developed from this machine, four uniform length, three degree of freedom legs were used to replace those of the standard design. The modelled legs had two segments, each of length 1.2m. This length was based on the length of the standard front legs so that it would remain in a practical realm. In addition, this length was short enough to prevent leg contact with the horizontal angular limits chosen (to be described in the next section). The commercial machine has a base that is wider than it is long. In addition the legs rotate horizontally along the front and back edges, not along the sides. So in the model, the base was turned about 90°. This made the length greater than the width, and moved the horizontal leg movement range to the sides. The resulting model was 1.78m long and 1.16m wide.
Figure 5.2: The Kaiser Spyder X5M
5.2 Angular Range Limitations

5.2.1 Factors in the angular range limitation

To provide the greatest foot access range, the angular range of the joints should be made as large as possible. In a practical machine design however, limitations have to be placed on the range of the joint angles to ensure that the machine can be supported properly. Loading on the joints determines the angles through which they can move. The joint angles are also limited by the functional range of the actuator systems that are powering the joints. The forces exerted on both the joints and the actuators are a direct result of the length of the legs, and the machine loading. So the angular range of the legs is determined by the total leg design, and the desired range should thus be considered when the design is made.

As explained in Section 4.16, the horizontal angular limits affect whether or not the legs can strike each other as the machine moves. These angular limits should be determined in conjunction with the length of the machine, and the length of the upper leg segments.

5.2.2 Angular range limitations of the model

The model used for simulation had horizontal angular limits of 45° in front of a line extending laterally from the machine, and 45° behind it (θ in Figure 3.1). In reference to the same lateral line, the vertical angular limits were 45° above it, and 45° below it (ϕ in Figure 3.1). The extension angle (ε in Figure 3.1) was limited to a minimum of 40° and a maximum of 155°. These limitations were chosen to provide a reasonable foot access range, without being so large that they were mechanically impractical. The total extension angle range was greater than that of the other two angles. This was because substantial leg extension capability is required for a reasonable foot access range, and because the knee can have a greater range of motion with less mechanical difficulty. The latter reason is due to the fact that the knee is a one degree of freedom joint, and the fact that while the machine platform is likely to be crowded with equipment, the extension mechanics can be situated by themselves on the upper leg segment.

The vertical angle that the legs assumed when they could not access their footholds and were hovering over them was set to 36° in the simulation. This provided reasonable ground clearance without being too close to the maximum vertical angle.
5.3 Maximum Rates of Change

5.3.1 Maximum joint angle change rates

The angular rate of change that can occur in any joint of a practical machine is entirely dependent on the linkage and actuator system used. As shown in Section 3.6, the maximum horizontal angle rate of change in the forward swing direction is the main factor that limits machine speed. Thus it is critical that the machine be designed to drive the horizontal angle forward as quickly as possible. The fact that the legs are not loaded during the forward swing could be used to advantage.

If dynamic performance is not considered, the absolute rate of angular change in the machine joints is not important to a movement planner. Instead, the angular change that can occur in one sample time is what must be considered. During simulation of the LOGHAQ heuristics, the angular changes that could occur per sample time could be set arbitrarily small. To allow for accurate control and good graphic output resolution, the maximum angular change rates were set to 2.5°/sample time for all of the joints. This rate provided the needed resolution and did not result in excessive computational effort during long term simulations.

5.3.2 Maximum body rotational change rates

To ensure that the accelerative forces, both translational and rotational, that a practical walking machine is exposed to are not too severe, extensive dynamic testing should be performed. Such testing should consider loads experienced by the machine platform, the legs of the machine, and any onboard operator. To prevent excessive forces in any machine which might be controlled by them, the LOGHAQ heuristics placed limits on the rotational change rates allowed about three axes running through the center of the machine. These were the longitudinal axis, the lateral axis, and the vertical axis. The change rates allowed about each of these axes would normally be determined by knowledge of machine dynamic performance and the desired platform movement limitations. As this information was not available, the rotational change rate allowed about each of these axes was set to 2.5°/sample time.

5.3.3 Maximum acceleration

Again to prevent excessive forces in any machine which might be controlled by them, the LOGHAQ heuristics placed a limit on the maximum acceleration allowed at the center of the machine. While
determining this also depended on knowledge of machine dynamic performance and the desired platform movement limitations, the effects of different acceleration limits could be more easily seen on the graphic display of the simulations. After empirical observation of the machine response, the maximum acceleration was set to 0.001m/sample time\(^2\). This resulted in performance that was conservative, but not sluggish.

When the machine was forced to halt, the acceleration limit was not used. The velocity of the machine fell to zero immediately. In a practical machine application, this could not be allowed. The machine would not be able to halt that quickly, and it would be subjected to excessive forces in its attempt to do so. The dynamic performance of a practical machine should be analyzed, and some sort of plan developed to anticipate sudden halts and initiate negative acceleration before hand. The stability margin could also be increased so that if the machine does overshoot its stability region, it will still remain stable. Such strategies are meaningless though, unless dynamic performance knowledge is available. It should further be noted that the maximum allowed acceleration during a sudden halt should not be determined in any way by considerations of desired platform movements. It is far more important to stop a machine before it collapses than to prevent extreme platform accelerations. While an onboard operator may be thrown against the front of his cab, that is far less of a concern than having him involved in the destruction of a very expensive machine.

5.4 Machine Stance

5.4.1 Width of stance

Once the width of a machine platform has been determined, the width of the total machine stance is determined by the distance that the footholds are placed laterally relative to the machine. As stated in Section 3.6, a wider stance improves the maximum speed of a machine slightly, and, as stated in Section 5.1, a wider stance also increases the lateral stability, but neither of these are primary considerations of the stance determination. What is more important is how the lateral foot placement distance affects the access range of the feet. If the feet are placed far out, near the end of the leg range, the total foot movement for a given horizontal angular change is increased (see Equation 3.17). This means that longer strides can be taken, and the speed of the machine can be consequently increased. In exchange, the torques felt at the shoulders are also increased, and the foot access beyond the lateral foothold distance is reduced. The latter effect increases the chance of leg over-extensions occurring if the footholds should be
placed further away from the machine than expected. As detailed in Section 4.14, this could significantly affect the long term continuation of machine movement. Conversely, if the lateral placement distance is made too close to the machine, the likelihood of leg under-extensions occurring increases. The optimal lateral foot placement distance to provide the greatest freedom of leg movement is one that is half way between the near and far limits of the leg range. Empirical tests of the simulated model indicated that having the lateral placement distance set to 1.3 times the upper leg segment length best balanced the need for maximum leg freedom of movement, and the desire to have the feet extended as far out as possible. The stance width then, including the machine width, was 4.28m.

5.4.2 Machine height

The height of a walking machine has important consequences on its walking pattern when it is on sloped terrain. If the terrain had a roll to it, as shown in Figure 4.3, the LOGHAQ heuristics followed a foot placement strategy that kept the horizontally projected machine center of gravity roughly half way between the horizontally projected footholds on either side of the machine (see Section 4.9). This resulted in the effective lateral placement distance on one side of the machine being larger than the other. The legs on that side were then correspondingly extended. Effectively, the platform of the machine was offset from the pattern defined by the feet. Similarly, if the terrain had a tilt to it, the platform was offset forward if the machine is going uphill, and backward if it was going down. The former condition caused the legs to be extended backward more than usual, and the latter caused them to be extended forward more than usual. So any terrain slope at all resulted in changes in the normal pattern of the leg extensions. As the LOGHAQ heuristics were designed to operate around fairly well maintained normal machine states, these effects were not desirable. To make them as small as possible, the effective platform offset had to be reduced, and this could only be accomplished by decreasing the height of the machine.

Another reason that the height of a machine should be minimized is that doing so may increase the foot access range. If a machine design includes considerable leg movement capability above the shoulder height, this area is best utilized by getting the shoulder as close to the ground as possible. The value of doing so depends on the actual machine design. The model used in the simulations of the LOGHAQ heuristics had equal vertical angle movement limits above and below the shoulders, and it was thus advantageous in this case to reduce the machine height.

As stated in the previous subsection, the lateral extension of the machine legs should be made as large as possible to increase the distance covered as the legs rotate. This lateral extension should not be
increased to the point where there is danger of leg over-extension however. If the height of a machine is reduced, the same total leg extension can reach further laterally.

Having a lower machine height has another advantage. As described in Messuri (1985), the “energy stability margin” of a walking machine is increased as the height of the machine is lessed. The energy stability margin is a measure of the effort required to tip over a machine when it is in a statically stable state. A machine with a high energy stability margin is more impervious to external disturbances.

While the height of a walking machine should be reduced for many reasons, it must always be kept high enough to provide adequate ground clearance. The ability to cross normally impassable terrain is the main advantage of using walking machines over other forms of ground transportation. This ability should not be sacrificed. The height of a machine must then be made as low as possible, but not so low that its ground clearance is affected. The height of the simulation model, after empirical tests to determine this point, was set to 0.8m.

5.4.3 Length of stride

Section 3.6 described in detail why longer stride lengths are necessary to improve the speed of a machine. If a machine stride is made too long however, the chances of its legs over-extending increases. The potentially disastrous results of this were described in detail in Section 4.14. Now assuming that the feet of a walking machine will normally move along its sides at the lateral foot placement distance (see Section 4.9), the maximum forward distance that the feet of the machine can travel is:

\[ \text{lateral foot placement distance} \cdot \tan \theta_{\text{max}} \]  

(5.1)

Where \( \theta_{\text{max}} \) is the maximum forward horizontal angle of the legs of the machine. To ensure that over-extensions are avoided, the stride length should be made much less than this amount. After empirical observation of performance, the stride length of the simulation model was set to:

\[ 0.6 \cdot \text{lateral foot placement distance} \cdot \tan \theta_{\text{max}} = 0.936 \text{m} \]  

(5.2)
5.5 Stability Controlling Distances

5.5.1 Stability margin

The stability margin chosen for a practical machine depends on the dynamic behavior of the machine when operating under the conditions produced by the LOGHAQ heuristics. The stability margin size determines the radius of a vertical cylinder that encloses the center of gravity of the machine. When the machine is subjected to dynamic forces, its effective center of gravity moves around its nominal one. As long as this effective center of gravity remains in the vertical cylinder, the LOGHAQ heuristics ensure that the machine will remain stable. Thus the stability margin has to be chosen based on expected machine behavior. The less affect dynamic forces have on a machine, the smaller the stability margin need be.

Some effort should be made to minimize the size of the stability margin. As described in Section 3.6, the larger the stability margin is, the longer the creeping periods are, and the lower the maximum possible speed is. In addition, as will be described at the end of this section, the larger the stability margin is, the lower the maximum turning rate is. So the stability margin should be made as small as possible within the range of expected machine dynamic behavior.

The stability margin was set to 0.15m in the simulations of the LOGHAQ heuristics. This seemed to produce a reasonably sized vertical stability cylinder. Such a judgement was difficult to make however with no knowledge of the dynamic behavior characteristics of the model.

5.5.2 Buffer distance

The buffer distance determined, in the LOGHAQ heuristics, how far behind the planned center of gravity the machine should move before each half-cycle ended. As stated in Section 4.11, the machine could not move right up to the planned center of gravity, as this location was an unstable one. The planned center of gravity was used as a trajectory specifying point only.

For ideal performance, the buffer distance should be set to exactly the stability margin distance. This would allow the machine to move as close to the front edge of the stability triangle as possible. Then, if the LOGHAQ heuristics calculated everything perfectly, a speed would be determined that brought the machine to this point just as the front foot touched down. A new stability triangle would be formed by the footfall, and the machine could continue on without stopping. The planning accuracy required to do this would be formidable however, especially if the machine exhibited significant dynamic response
Chapter 5. Practicalities of Quadruped Design

characteristics. To prevent the machine from arriving at the termination point defined by the buffer distance, and having to halt suddenly because the next stability triangle has not yet been formed by the front footfall, the buffer distance should be made larger than the stability margin. The buffer distance should not be made too large though, as the further the machine is behind the planned center of gravity at the end of the half-cycle, the further it must creep to reach the next stability triangle.

After empirically testing different buffer distances, a distance of 0.4m was chosen for the LOGHAQ simulations. This distance, although somewhat conservative, prevented the machine from having to halt suddenly in all but the most unusual conditions.

5.5.3 Following distance

The following distance determined, in the LOGHAQ heuristics, how far behind the front foot the rear foot was placed. As mentioned in Section 3.3, this distance should be made as small as possible to prevent excessive backward shifting of the stability triangle front edge when the rear foot takes over support, but it should be made large enough to prevent the legs from striking. It was set to 0.12m in the simulations, one tenth of the length of the leg segments.

5.5.4 Maximum turning rate

For a given stride length and stance width, the maximum possible turning rate of the machine is largely defined by the following distance and the stability margin. It is affected by many other factors as well, such as the terrain slope, but these two values have the greatest overall effect on it. If $2d_1$ is set to the machine stance width, $w$, and $d_\infty$ is set to twice the sum of the stability margin and the following distance, Equation 4.48 provides an approximation of the maximum turning rate when the machine is on flat terrain. The maximum turning rate for the simulation model, using this approximation, was 5.14° per half-cycle.

5.6 The Settling Factors

5.6.1 Movement, tilt, and roll settling factors

In the LOGHAQ heuristics, the settling factors determined how quickly the machine adapted the terrain orientation indicated by its foot positions. The machine, when it entered a new half-cycle, had a trajectory of motion defined by the conditions it had previously encountered. The settling factors determined
how much of this old trajectory was maintained, and how much of the new one was adopted. This is discussed in detail in Section 4.7. After extensive empirical testing of the LOGHAQ heuristics, the movement (translational), tilt, and roll settling factors were all set to 0.6. This settling rate provided quick response to new terrain conditions, but maintained enough of the old trajectory to prevent exaggerated response to transient changes.

5.6.2 Rotational settling factor

The rotational settling factor was set to 0.1 in the simulations of the LOGHAQ heuristics. This is because it is the operator of the machine who should specify any rotational change. If the movement plan developed from a half-cycle foothold configuration indicates that a certain tilt or roll should be adopted, this is most likely the case. These footholds reflect the attitude of the terrain over which the machine is moving. Further, the operator has no control over the tilt and roll of the machine. This is not the case with the rotation. The terrain cannot indicate which way the machine should turn, that is entirely the prerogative of the operator. The rotational settling factor was set to 0.1 to allow only a small amount of accommodation to the foothold locations. Not doing so to some degree could have resulted in the over-extension of legs whose feet were far away from their expected locations.

5.6.3 Front placement angle learning rate

The front placement angle learning rate determined, in the LOGHAQ heuristics, how much each new front horizontal placement angle was averaged in to the running estimate of this angle. The placement angle estimate was, as stated in Section 4.6, maintained separately for each side of the machine. The learning rate was set to 0.4 in the simulations of the LOGHAQ heuristics. This rate responded well to change, but was conservative enough to prevent extreme cases from effecting the estimate excessively.
In this chapter the simulations performed to evaluate the LOGHAQ heuristics will be described. The simulations were based on the model walking machine developed in the last chapter. To fully investigate the capabilities of the heuristics, a wide variety of terrain conditions were simulated. Each terrain condition will be described in the following sections, as well as the simulation response to it, and an analysis of that response. To begin with however, the general simulation format, the performance measures collected, and the manner in which the simulated terrain was constructed will be described.

6.1 The Simulation Format

The basic format of the simulations was to start the machine in a standard position, allow it to proceed forward until it had failed or completed a specified number of steps, and then repeat the process a fixed number of times. The standard position in which the machine started was based on the original attitude of the terrain upon which it sat. The terrain attitude was specified, in terms of tilt and roll, before the simulation began. The attitude of the machine was set to match that of the terrain, and its center of gravity was placed at height $h$ above it. The location of the feet of the machine were determined by locating them relative to the point on the terrain directly underneath its center of gravity (Figure 4.3 illustrates the effects of locating relative to this point when the terrain has a roll to it).

\begin{align*}
  P_1 &= -h\vec{k} + 1.5s\vec{A} - \frac{w}{2}\vec{S} \\
  P_2 &= -h\vec{k} + 0.5s\vec{A} + \frac{w}{2}\vec{S} \\
  P_3 &= -h\vec{k} - 0.5s\vec{A} - \frac{w}{2}\vec{S} \\
  P_4 &= -h\vec{k} - 1.5s\vec{A} + \frac{w}{2}\vec{S}
\end{align*}

(6.1)  
(6.2)  
(6.3)  
(6.4)

Where $h$ and $w$ are the height and width of the machine stance respectively, $s$ is the stride length, $\vec{A}$ and $\vec{S}$ are the orientation vectors (see Section 4.3), and $\vec{k}$ is a unit vector in the vertical direction. The original foot locations were similar to those of machine position $A$ in Figure 3.2.
Chapter 6. Heuristic Simulation

From the original position, the simulated machine advanced forward using a movement plan generated by the LOGHAQ heuristics. The heuristics determined everything from the overall half-cycle movement plan to the joint angle changes required to follow it. The desired joint angles that were determined every sample time were immediately adopted by the simulated joints. Thus, perfect following of the movement plan occurred. Movement of the machine continued until it had failed, or a specified maximum number of steps (usually 100) had been made. Then the machine was reset to its original state and a new trial was begun. This continued until a specified number of trials (usually 100) had been completed.

6.2 The Performance Measures Collected

Many performance measures were recorded throughout the trials. In reference to the overall success of the trials, the following was output:

- The number of successfully completed trials.
- The number of failed trials, including a summary of the cause of the failures.
- The average number of steps before the successes, and the standard deviation of this average.
- The average number of steps before the failures, and the standard deviation of this average.

For those trials that completed, the following values were output:

- The average and peak distance travelled during the trials.
- The average and peak total rotation that occurred during the trials, where this rotation was measured relative to the initial orientation of the machine.
- The average and peak total lateral movement that occurred during the trials, where this total lateral movement was the sum of all movements of the machine to either side of its longitudinal axis.
- The average and peak number of attitude adjustments made to reach footholds.
- The average and peak number of rotational adjustments made to reach footholds.
- The average and peak number of sudden halts that occurred during the trials. The average speed of the machine immediately before these halts occurred was also determined.
For all of the trials, the number of attitude adjustments, rotational adjustments, and sudden halts were divided by the total number of steps taken to determine:

- The probability per step of an attitude adjustment occurring.
- The probability per step of a rotational adjustment occurring.
- The probability per step of a sudden halt occurring.

Statistics were also gathered on the performance of the simulated machine during individual sample times. These values indicated the finer performance of the LOGHAQ heuristics. For each of the measures below, average, peak, and standard deviation values were output:

- The speed.
- The absolute instantaneous (one sample time) lateral movement.
- The absolute vertical error (the difference between the actual height of the machine above the terrain and the nominal stance height, \( h \)).
- The absolute tilt error (relative to the terrain tilt).
- The absolute roll error (relative to the terrain roll).
- The absolute instantaneous (one sample time) rotational change.
- The absolute rotational error per half-cycle.

It was found during the simulations that some of the data collected did not provide any meaningful information about the simulated machine performance. For example, the simulated machine followed the plan developed by the LOGHAQ heuristics precisely, so the measure of the absolute rotational error per half-cycle was always very small. This value would be of more interest in a dynamic performance analysis. Many of the other statistics collected each sample time were of little interest for the same reason. They would simply be better used as measures of dynamic response characteristics.

Another problem that was discovered was that many of the performance measures were meaningless in certain simulation conditions. The absolute instantaneous rotational change measure, for example, was meaningless when the simulated machine had been instructed to turn. While the machine was turning, the instantaneous rotational change that should have been occurring was impossible to predict,
so a measure of it could not be compared to anything. This value was only meaningful when the machine had been instructed to move straight ahead, in which case it indicated an error in performance.

So in the analysis that follows, most of the performance measures collected during the simulations will not be detailed. Only those that were meaningful and that best illustrated the performance of a machine operating under control of the LOGHAQ heuristics. Far too much data was collected for all of it to be explicitly relayed.

6.3 The Simulated Terrain

The highly unconstrained nature of walking machine movement provides room for researchers to adopt a wide variety of testing strategies when evaluating the performance of their machines. There are simply too many variables involved in the behavior of each machine for a standard testing strategy to be adopted. In addition, this researcher found very few structured walking machine tests, complete with detailed performance analyses, described in the literature. This was true in reference to both simulated walking machines and actual mechanical devices. Due to both of these reasons, a standard testing strategy was not available with which the LOGHAQ heuristics could be evaluated. As a result, a testing strategy had to be developed that would allow a meaningful performance analysis.

6.3.1 The planar structure of the simulated terrain specification

The terrain was modelled in the simulations by simply defining a flat plane over which the machine moved. The tilt and roll of this terrain was specified before the simulation began. Now as stated in Section 3.4, the LOGHAQ heuristics were concerned only with the footholds presented to them, they were not concerned with any other aspects of the terrain. So the only time that the actual terrain structure was considered during the simulations was when the Foothold Determination heuristic (see Section 4.9) requested a new foothold. When this occurred, the simulator within which the LOGHAQ heuristics were running calculated the foothold location on the flat plane that best corresponded with the placement guidelines presented by the Foothold Determination heuristic.

6.3.2 The “Roughness” of the simulated terrain specification

Once the best foothold location on the terrain plane had been determined, this location was perturbed in space by a variable distance. The direction of the perturbation was determined, as shown in Figure
6.1, by randomly choosing the two angles of a polar coordinate system. These angles were restricted so that the direction of the perturbation would only be in the half sphere facing the walking machine. The other half of the sphere was not used because it only contained foothold locations that were further away from the machine than the maximum distance specified by the Foothold Determination heuristic. Once the exact direction of the perturbation was chosen, the magnitude of it was selected randomly to be within a defined distance. This maximum distance was referred to as the “roughness”. All of the random choices were made with a uniform probability distribution.

So the simulated terrain consisted of flat planes only, and the foothold selections were perturbed off these planes by random distances that were constrained to be within a specified roughness range. This seemed to be intuitively reasonable. When one observes a section of real terrain, the overall slope of that terrain is considered at first, and then the roughness of the slope surface. The random perturbations made in the simulations modelled the presence of surface variations due to things like dips and small rocks. For the purposes of structured analysis, the roughness of the simulated terrain used throughout this study had a consistent distribution, and a maximum range.
6.3.3 Comparative analysis of this terrain simulation method with others

Hirose et. al. (1984a and 1984b) state that simulated terrain can be composed entirely of “holes” and variable height “poles”. These holes and poles are discrete components of a tightly packed simulated terrain grid. Holes represent locations that cannot be chosen as footholds, and the tops of the poles constitute the “solid” ground that the footholds can be made on. The grid within which holes and poles are contained is horizontally oriented. So an upward sloping terrain would consist of rows of poles that gradually increase in height.

It is true that, if the size of the holes and poles are made small enough, holes and poles can model any given terrain, but there is some question as to whether they would do so in a useful manner. Certainly it does not seem that the best way to represent a mountain flank would be with poles that vary in height from zero to 2000 meters. Further, adding roughness with a known statistical distribution to such a simulated terrain would be awkward. This is because roughness would have to be implemented by varying the pole heights about some underlying terrain plane specification. The height of the poles is all that can be varied, and the direction of this variation is always along the length of the poles, so the distribution of the roughness will bear some correlation with the direction of the length of the poles. The statistical distribution of the roughness in directions other than this would be difficult to ascertain. Obviously it would be affected by the cross-sectional size of the poles, but many other characteristics of the hole and pole environment could affect it, including the density of the holes. It would be even more difficult to ensure a uniform roughness distribution in all directions. It may be desirable to have a terrain roughness distribution that is oriented in the direction of the poles when these poles are perpendicular to the underlying terrain plane (ie.: when the terrain is horizontal), but when terrain with an underlying slope is desired, it is unlikely that roughness that has a distribution that is oriented at some arbitrary angle relative to the slope would be permissible.

It should also be noted that an underlying terrain plane would have to be made use of, as described above, to provide for any control over the roughness distribution at all. Such an underlying terrain plane was used in the terrain simulation method of this researcher, but the method of adding roughness to this plane was far simpler. Using a terrain plane in combination with holes and poles would seem to unnecessarily complicate the problem of constructing simulated terrain with roughness of a known distribution.

Hirose et. al. (1984a and 1984b) probably used this method of terrain modelling because it was well
suited to how their walking machines operated. These machines maintained a horizontal attitude, and performed an intensive search for available footholds. So having simulated terrain that was primarily horizontal and had a large number of holes in it was entirely appropriate. They could easily define the percentage of the terrain that consisted of holes, and then test the ability of their machine to cross this terrain. The hole and pole terrain simulation method was thus adequate for their purposes, but this researcher feels that it cannot be readily extended to general terrain modelling applications.

Work done at Ohio State University (McGhee, 1984 and 1979) also made use of simulated terrain defined by holes and poles. Other work done there (McGhee, 1984) involved simulated terrain that was constructed of planar sections. These sections were defined in a random fashion. As a result, the roughness that the simulated machine encountered was not consistent, nor was its statistical distribution known. More strict terrain modelling methods may have been used in the work done at O.S.U., but such efforts were not found in the literature.

The basic difference between the simulated terrain description methods used by other researchers, and the one adopted in this study is that the simulated terrain of most other researchers was completely defined before their simulations began. Only the underlying terrain structure, in the form of a simple plane, was predefined in the simulated terrain used in this study. Then, during the course of the simulation, the amount that each foothold was perturbed off this plane was determined. Thus the terrain roughness conditions that the machine encountered could be completely and independently controlled. Further, only the position of the current foothold locations were stored in the simulating computer at any given time. The terrain simulation methods used by other researchers required a complete description of the surface of the simulation region. Thus the method used by this researcher required less computer memory to maintain the simulated terrain, and less computational effort to track the relevant terrain locations as the machine moved relative to them. As well, there were no limits placed on where the machine could go. It could not encounter the edge of terrain defined by a limited database. The plane over which it walked was infinite. This allowed for more comprehensive testing of the LOGHAQ heuristics, as the simulated machine could be permitted to walk long distances.

6.3.4 Terrain changes

To allow for more complex simulated terrains, another feature was incorporated in the simulator used to test the LOGHAQ heuristics. Different planar sections could be joined together to compose the underlying terrain structure. Each of these planes could then have any desired attitude. The junctions
between the planes, or "terrain breaks", consisted of vertical transition planes that were defined as being located in front of the machine. At a terrain break, one terrain plane ended where it intersected with the vertical transition plane, and another terrain plane began on the other side of the transition plane. Specifying terrain transitions in this manner facilitated the simulation of terrain changes between terrain planes that did not intersect at the location desired for the transition. The angle with which the machine encountered the terrain breaks, when viewed from above, was specified as desired. Normally it was set to 0°, which resulted in the terrain break being perpendicular to the forward direction of the machine. The relationship in space between the two terrain planes on either side of a terrain break was defined by the height difference between them directly in front of the machine.

Multiple terrain breaks could be simulated in this manner. The distance across the plane between each successive pair of terrain breaks was specified in the forward direction of the machine. When a series of terrain breaks had been defined, the trial step number in which these terrain breaks would appear in front of the machine was also defined. When that step occurred, all of the terrain changes were constructed as specified. The distance from the machine to the first break in a series was either specified, or could be chosen randomly. The random selection feature facilitated unpredictable encounters with the new terrain conditions.

Specifying simulation terrain changes in this manner maintained the solid underlying terrain structure needed for consistent performance analysis, yet provided the potential for a wide variety of terrain conditions to be simulated. These could include ditches, randomly scattered logs, rivers (a very deep trench could be constructed so that no footholds were available across a horizontal section), stairs, or simple folds in the terrain.

6.4 The Effects of Terrain Roughness

Now that the method used to simulate the LOGHAQ heuristics has been described, the results of a number of simulations will be presented. Each of the simulations performed was designed to test some aspect of performance of a machine that is being controlled with the LOGHAQ heuristics. The results that will be presented in the following sections have been compiled from the large quantity of data produced to best illustrate the various aspects of performance.
6.4.1 A benchmark: perfectly smooth and flat terrain

The first terrain that was simulated was perfectly smooth and flat terrain (no roughness). This is the least challenging terrain that can be presented to a walking machine, and the response to it was thus used as a benchmark against which the response to other terrain simulations was compared. During the simulation, the machine was made to take 100 steps and it travelled 43.0m during these steps. Due to the unchallenging terrain conditions, this distance was greater than that travelled in all of the other simulations (excluding one in which the stride length was increased). Correspondingly, the average speed, at 0.0085m/sample time, was also the highest. The standard deviation of the average speed was very low at 0.0005m/sample time. This indicated a high degree of speed consistency that was due to the smoothness of the simulated walking conditions. The peak speed attained, 0.0098m/sample time, was very close to the average speed, again revealing very consistent speed maintenance. This indicated that the techniques used in the LOGHAQ heuristics to maintain consistent speeds were effective. During the simulation the machine did not halt, nor did it make any rotational or attitude adjustments. All of these performance measures are summarized in Figure 6.2.

During the simulation, the machine rotated a total of 0.08° away from its initial orientation, and moved laterally a total of 0.028m. These values were the lowest encountered throughout the simulations. This was again due to the unchallenging terrain conditions. The average value of the tilt and roll error measurements were also low at 1.28° and 0.27° respectively.

6.4.2 The limits of smooth terrain performance

A number of simulations were performed in order to determine the limits of terrain attitude variation under which the simulated machine could operate when the terrain was perfectly smooth (no roughness).
When the terrain tilt was increased in 1° increments, with the terrain roll held constant, the simulated machine was able to continue walking until the tilt reached 50°. When the terrain tilt was decreased, a limit of −55° was reached. When the terrain roll was increased, with the terrain tilt held constant, a limit of 43° was reached. All of these performance limits represent the theoretical response of the simulated machine to perfectly smooth terrain.

6.4.3 Varying the roughness

After the perfectly smooth and flat terrain had been simulated, further simulations were performed on flat terrain with varying degrees of roughness. The roughness was varied from zero to 1.0m in increments of 0.1m. The simulations consisted of 100 trials of 100 steps at each roughness level. Figure 6.3 shows the trial completion rates that occurred in each of these simulations. It can be seen that as the roughness increased towards 0.6m the machine was able to complete most of the trials. Beyond 0.6m, the roughness was simply too large relative to the 1.78m long machine, and failure rates increased. These failures were mainly due to the machine being unable to reach its front footholds, which was a result of the high variability of these foothold locations. Other failures were due to the vertical angles of the legs of the machine being forced too far upward when extremely high footholds were encountered.

It will be shown, in Section 6.9, that the response of the simulated machine to increasing roughness levels was similar to the response that occurred in this simulation. That is, regardless of the underlying terrain attitude, a roughness level was reached beyond which a rapid decrease in the trial success rate occurred. The roughness level at which this took place depended on the terrain attitude.

Figure 6.4 shows how the average distance travelled by the machine (during successful trials only) decreased as the roughness increased. This was expected because when the roughness increased, the footholds were often perturbed into positions behind where they would ideally have been located. This delayed forward progress. For the same reason, the average speed decreased as the roughness increased. Figure 6.5 illustrates this. Figure 6.5 also includes lines on both sides of the average speed line that indicate one standard deviation of the average speed. This can be seen to grow as the roughness increased and consistent speeds were more difficult to maintain.

Figure 6.6 shows the average and peak, tilt and roll errors (relative to the terrain over which the machine was moving) as the roughness increased. It can be seen that both of these error measures grew as the roughness increased. This was due to the fact that at higher roughness levels, the variability of the foothold locations made the underlying terrain attitude more difficult to determine. Also, as the
Figure 6.3: The Effects on Completion Rate of Varying the Terrain Roughness
Figure 6.4: The Effects on Distance Traveled of Varying the Terrain Roughness
machine found movement more difficult, it had to make attitude adjustments more often, resulting in larger tilt and roll errors.

None of the trials completed in the simulation with a roughness of 1.0m. The simulated conditions were simply too harsh. It is illustrative to consider the effects of these extreme conditions however. They resulted in a probability of attitude adjustment occurring of 0.053 per step, and a probability of rotational adjustment occurring of 0.002 per step. This was expected because, as explained in Section 4.14, attitude adjustments to reach inaccessible front footholds were more likely to occur than rotational adjustments to reach rear footholds. Machine halting had to occur when either type of adjustment was made, but it also occurred at other times. The probability of the machine halting was 0.066 per step. The average machine speed immediately before halting occurred was 0.0056m/sample time. This speed before halting was typical of that which occurred throughout the simulations. The peak speed that
Figure 6.6: The Effects on the Tilt and Roll Errors of Varying the Terrain Roughness
occurred in this 1.0m roughness simulation was 0.0250m/sample time, and the standard deviation of the average speed was 0.0033m/sample time. Compared to the average speed of 0.0046m/sample time, these values indicated that the speed was very inconsistent when the machine was operating under such rough conditions. The average tilt and roll error measurements (relative to the terrain over which the machine was moving) were 8.50° and 4.38° respectively. These values were roughly an order of magnitude larger than those that occurred during the simulation of the terrain with no roughness.

6.4.4 Roughness selection for further simulations

Based on the results described above, a standard roughness of 0.5m was used throughout the rest of the simulations. These simulations considered aspects of machine response other than the effects of different roughness levels, and this variable thus had to be fixed. It was apparent that a roughness of 0.5m was challenging when the simulated machine walked on flat terrain, but that this roughness level did not severely impair performance. Thus, it was felt that 0.5m roughness would promote interesting machine responses in simulations designed to reveal other aspects of performance, but that its effects would not be the dominating ones.

6.5 The Effects of Differing Terrain Attitudes

A number of simulations were performed to determine the performance of the LOGHAQ heuristics when the machine being controlled by them was crossing terrain of varying attitudes. A roughness of 0.5m was used, and 100 trials of 100 steps were made for each terrain condition.

6.5.1 Varying the tilt

Figure 6.7 shows the trial completion rates that resulted from varying the terrain tilt from $-45^\circ$ to $45^\circ$ in increments of $15^\circ$. The roll during these simulations was kept at $0^\circ$. It can be seen how extreme terrain tilt conditions greatly reduced the trial success rate. The failures on the downward sloping terrain were mainly due to the machine being unable to reach down to new front footholds. The failures on the upward sloping terrain were mainly due to the legs of the machine becoming over-extended as the machine moved forward toward new stability triangles. Both types of failures were expected from the way in which the terrain slope affected the relationship between the machine center of gravity and the supporting foothold locations (see Section 5.4).
Figure 6.7: The Effects on Completion Rate of Varying the Terrain Tilt
Figure 6.8 shows the probability per step of attitude adjustments, rotational adjustments, and machine halts occurring as the tilt was varied. This figure indicates that, while both extremes of terrain tilt resulted in a high probability per step of these machine responses, the downward sloping terrain was more likely to do so. This was because it was more likely, as the simulated machine crossed the upward sloping terrain, that one of its legs would over-extend before it got to the edge of the current stability triangle, than for it to have to make attitude adjustments to reach an inaccessible front foothold once it had arrived at the front edge (see Section 5.4).

6.5.2 Varying the roll

Figure 6.9 shows the trial completion rates that resulted from varying the terrain roll from zero to 30° in increments of 15°. At rolls greater than 30° the leg positioning became too difficult for the machine.
The tilt during these simulations was kept at 0°. Negative rolls were not simulated, as the machine would simply have responded to them in a manner that was a mirror image of its positive roll response. It can be seen in Figure 6.9 how extreme terrain roll conditions greatly reduced the trial success rate. The failures were mainly a result of the machine being unable to make front foot placements. This was because the legs of the machine, under extreme roll conditions, were highly over-extended on one side of the machine and under-extended on the other (see Section 5.4). This resulted in new front footholds often being inaccessible to legs that were already at the extent of their movement range. Leg over-extensions during movement were also a common cause of failure.

Figure 6.10 shows the probability per step of attitude adjustments, rotational adjustments, and machine halts occurring as the roll was varied. As expected, the probability of each of these occurrences increased with the roll.
Figure 6.10: The Probability of Adjustments and Halts for Different Terrain Rolls
6.5.3 Varying the tilt and roll together

To determine how a machine controlled by the LOGHAQ heuristics performed under extreme terrain slope conditions, the tilt and roll were varied together. They were varied from $-30^\circ$ to $30^\circ$ in increments of $15^\circ$. Figure 6.11 shows the trial completions rates for simulations subjected to these slope variations. It can be seen that the success rate of the LOGHAQ heuristics dropped off rapidly when the terrain varied in either direction away from flat. The cause of the failures paralleled that of the simulations involving only terrain tilt variations. When the machine was travelling down a slope, it usually failed because it could not make front foot placements. When it was travelling up a slope, the failures normally resulted from leg over-extensions. In general, the failures that occurred during these simulations were similar to those that occurred under terrain tilt conditions, but they were further aggravated by the terrain roll.
Figure 6.12: The Probability of Adjustments and Halts for Different Terrain Tilts and Rolls

Figure 6.12 shows the probability per step of attitude adjustments, rotational adjustments, and machine halts occurring as the tilt and roll were varied. These curves also parallel those generated during the terrain tilt variation simulations, as shown in Figure 6.8.

Figure 6.13 shows the average speed of the simulated machine as it was subjected to the combined tilt and roll variations. As expected, the highest average speed occurred on the flat terrain. More challenging terrain conditions made maintenance of the speed more difficult. Figure 6.13 also includes lines on both sides of the average speed line that indicate one standard deviation of the average speed. It can be seen that the standard deviation remained quite constant. This was due to the fact that while the tilt and roll of the terrain were varied, the roughness of it was not.
Figure 6.13: The Effects on Speed of Varying the Terrain Tilt and Roll Together
6.6 Turning

The simulations described in this section were performed to determine the performance of a machine controlled by the LOGHAQ heuristics when turning was desired. A roughness of 0.5m was used for all of the simulations in this section.

6.6.1 Varying the desired turning rate

A number of simulations were performed to determine the response of a machine controlled by the LOGHAQ heuristics to different desired turning rates. The turning rates requested of the simulated machine were varied from zero to 15° per half-cycle in increments of 2.5° per half-cycle. These simulations were made on flat terrain and each consisted of 100 trials. Each of these trials ended when the machine had completed a total rotation of 90° or had failed.

When the turning rate was 10° per half-cycle or less, the majority of the trials completed successfully. When extreme turning rates were requested (those above 10° per half-cycle), there were a wide variety of reasons for the failures that resulted. Typically the failures were due to the legs on the outside of the machine turn becoming over-extended in some manner.

Figure 6.14 shows how many steps were required, on average, for the machine to complete a 90° rotation with the various turning rates. Now each half-cycle consists of two steps, so it would be expected that, with a 2.5° per half-cycle turning rate, it would take a machine 72 steps to complete a 90° rotation. This is very close to the average number of 75.9 that resulted during the simulations. The small discrepancy is due to the roughness of the terrain over which the machine was turning.

At a 5° per half-cycle turning rate, 36 steps should be necessary to complete the 90° rotation. This is far less than 54.4 steps that were required on average for this turning rate during the simulations. This discrepancy is due to the fact that, with the model machine dimensions used, the maximum theoretical turning rate was 5.14° per half-cycle (see Section 5.5). As explained in Section 4.9, when the turning rate resulted in foothold locations being desired that were not far enough forward to produce properly sized stability triangles, the desired foothold locations were moved forward to an appropriate position, and the turning rate was temporarily adjusted in the next half-cycle to reflect the changes. This is what caused the 90° rotation to take more steps than expected at the 5° per half-cycle turning rate. While this turning rate was theoretically marginally acceptable, it was not conservative enough when the machine was subjected to terrain roughness. Thus, it was affected by the portion of the Foothold Determination
heuristic that prevented stability triangles from being formed that were too small. Turning rates beyond 5° per half-cycle were affected by this even more. When turning rates were attempted that were higher than 5° per half-cycle, the performance of the machine degraded, as seen in Figure 6.14. This was because the machine could not turn at these rates, and in attempting to do so, counter-productive movement decisions were made by the LOGHAQ heuristics. These counter-productive decisions consisted mainly of having the machine change its rotational orientation faster than the turning rate that was possible with the foot placement limitations (see Section 4.9). This resulted in troublesome inconsistencies between the orientation of the machine platform and the general orientation of its supporting feet. As the turning rate was increased beyond 5° per half-cycle, the probability of failure increased as a result of these counter-productive decisions.

Figure 6.15 shows the average speed of the simulated machine that resulted from the various turning
Figure 6.15: The Effects on Speed of Turning

The average speed tended to decrease as the turning rate was increased. The reason for this was that as the turning rate increased, the size of the stability triangles on the inside of the turn decreased. This reduced the distance that could be travelled while crossing these stability triangles, and thus reduced the overall speed of movement. Figure 6.15 also includes lines on both side of the average speed line that indicate one standard deviation of the average speed. For the same reason, the standard deviation increased as the turning rate increased. As the stability triangles on the inside of the turn became smaller, there was less consistency between the speed of the machine when it crossed them and when it crossed the fixed size stability triangles on the outside of the turn (see Section 4.9). Both of these machine speed effects level off somewhat at turning rates above 5° per half-cycle. This was because the size of the inside stability triangles was not permitted to fall below the minimum size necessary for continued machine movement. This was discussed briefly above, and in detail in Section 4.9.
6.6.2 Turning on slopes

To determine how well a machine being controlled by the LOGHAQ heuristics could turn on sloped terrain, a number of turning simulations were performed on terrain of varying attitudes. The overall deviation of the terrain attitude away from horizontal was varied from zero to $30^\circ$ in increments of $10^\circ$. The limit of $30^\circ$ was chosen because this was the maximum terrain roll that the simulated machine could handle. For each terrain attitude the machine was made to turn through four quadrants of terrain slope. For example, on the terrain sloped at $10^\circ$, the machine was started in a position where the terrain had a tilt of $10^\circ$ and a roll of $0^\circ$. Then it was made to turn $90^\circ$ until the terrain had a tilt of $0^\circ$ and a roll of $-10^\circ$. Next it was made to turn $90^\circ$ again until the terrain had a tilt of $-10^\circ$ and a roll of $0^\circ$. Again it was made to turn $90^\circ$ until the terrain had a tilt of $0^\circ$ and a roll of $10^\circ$. Finally, it was made to turn $90^\circ$ again until it returned to its initial position. At the beginning of each of these $90^\circ$ turns the machine was placed in the starting configuration detailed in Section 6.1. Each of these quadrants was simulated separately in 100 trials. The final results of these 400 trials were combined, but breaking the simulations up into four parts prevented repeated failures in one quadrant of the slope from preventing the collection of results from the others.

As it was found, both theoretically and practically, that $5^\circ$ of turning per half-cycle was the highest practical turning rate for the model machine, this turning rate was adopted in these simulations.

Figure 6.16 shows the trial completion rates for the various slope attitudes. As expected, the simulated machine had more difficulty completing the $90^\circ$ turns as the slope attitude was increased. When the machine failed in quadrants of downward tilting terrain, the main reason was that the front foot placements could not be made. This was consistent with the simulation results for downward sloping terrain when the simulated machine did not have to turn (see Section 6.5). The machine also failed in these quadrants because the legs on the outside of the turn became over-extended. When the machine failed in quadrants of upward sloping terrain, the main reason was that the legs became over-extended. This was again consistent with the simulation findings for when the machine did not have to turn. The turning aggravated the over-extension problem however, as the over-extensions occurred far more often in the legs on the outside of the machine.

Even though the simulated machine maintained the same $5^\circ$ per half-cycle turning rate throughout these simulations, the number of steps required to complete the $90^\circ$ turns increased with the slope attitude. This was due to the difficulties that the harsher slope attitudes presented to the LOGHAQ...
Figure 6.16: Response to Varying Slope Attitudes While Turning at 5° per Half-Cycle
heuristics. Correspondingly, the probability of attitude and rotational adjustments, as well as the probability of the machine halts, increased with the slope attitude.

6.7 Performance During Terrain Changes

After a thorough analysis had been made of the performance of the LOGHAQ heuristics in guiding a simulated machine over planar terrain, simulations were performed to investigate the ability of the heuristics to guide a machine over changes in terrain. An endless variety of terrain changes could have been simulated, and in fact many were tried to determine what the heuristics were capable of, but for the purposes of comprehensive analysis, only a limited set of simple changes were simulated extensively.

For each terrain change simulated, 100 trials of 16 steps were made. The changes were presented to the simulated machine after its fourth step at random distances in front of it. The four steps were made so that the roughness (set to 0.5m as usual) of the terrain before the changes would move the machine out of its fixed initial configuration and into a more random one. Empirical observation revealed that a total of 16 steps were needed to ensure that the machine had fully adopted the attitude of the terrain after the changes. The movement of the machine both before and after the terrain changes was made as small as possible to prevent machine failures that had nothing to do with the terrain changes from excessively affecting the results.

6.7.1 Changes in terrain tilt

Two types of changes were made to the tilt of the terrain. These were to have the terrain initially flat and then change it to tilted, and to have it initially tilted and then change it to flat. In either case, the tilted portion was varied from $-45^\circ$ to $45^\circ$ in increments of $15^\circ$. The trial completion rates that resulted from these changes are shown in Figure 6.17. This figure indicates that the simulated machine could handle difficult terrain tilt changes with few machine failures. The only change that was not handled with a great deal of success was the change from flat terrain to terrain that tilted downward steeply. The failures that this terrain change caused were mainly due to the machine being unable to reach its new front footholds. This was expected, as these front footholds, after the terrain change, were significantly lower than those the machine had previously encountered. Thus they were difficult to reach. When the terrain tilted upward after the terrain change, the footholds beyond the change were actually brought closer to the machine. So in this case the terrain change actually aided the machine somewhat.
Figure 6.17: The Effects on Completion Rate of Various Terrain Tilt Changes
Figures 6.18 and 6.19 show the average number of attitude and rotational adjustments, and the average number of machine halts that occurred as a result of the terrain tilt changes. Figure 6.18 shows the response when the terrain was initially flat, and Figure 6.19 when it was initially tilted. These figures indicate that the simulated machine made the most attitude adjustments when it encountered new terrain attitudes that resulted in its front footholds being located higher up relative to its body. This was because the machine had to lift up its front end to make these placements.

6.7.2 Changes in terrain roll

Two types of changes were made to the roll of the terrain. These were to have the terrain initially flat and then change it to rolled, and to have it initially rolled and then change it to flat. In either case, the rolled portion was varied from zero to 30° in increments of 10°. Negative rolls were not simulated as the
Figure 6.19: The Average Number of Adjustments and Halts During Tilted to Flat Changes
machine would simply have responded to them in a manner that was a mirror image of its positive roll response. The trial completion rates that resulted from these changes are shown in Figure 6.20. This figure indicates that the simulated machine had more difficulty handling roll changes than it did tilt changes. This was because roll changes resulted in distinct breaks in the foothold levels. The terrain before the change stopped suddenly, and the terrain that continued on afterwards was rotated relative to it. Tilt changes only resulted in simple folds of the simulated terrain.

Figure 6.20 indicates that changes from flat terrain to rolled terrain were more difficult for the simulated machine than changes from rolled terrain to flat terrain. The failures that occurred during roll changes were mainly due to the machine being unable to reach new front footholds.

Figures 6.21 and 6.22 show the average number of attitude and rotational adjustments, and the average number of machine halts that occurred as a result of the terrain roll changes. Figure 6.21
Figure 6.21: The Average Number of Adjustments and Halts During Flat to Rolled Changes

shows the response when the terrain was initially flat, and Figure 6.22 when it was initially rolled. These figures further indicate that changes from flat terrain to rolled terrain were more difficult for the simulated machine than changes from rolled terrain to flat terrain.

8.8 The Effects of Varying the Stance of the Machine

In order to test the conclusions drawn in Section 5.4 regarding how machine stance affects performance, a number of simulations were performed in which variations were made to different aspects of the stance. To best determine the effects of the variations, it was decided to select an earlier simulation whose conditions had challenged the abilities of the LOGHAQ heuristics. The simulation conditions that were selected were ones where the terrain had a tilt of 30° and no roll. A roughness of 0.5m was used. These conditions had previously resulted in a trial completion rate of roughly 75%. The simulated machine
Figure 6.22: The Average Number of Adjustments and Halts During Rolled to Flat Changes
was not made to turn, and no terrain changes were made.

It should be noted that using one set of simulation conditions in this manner could have biased the types of failures that resulted. The failures that normally resulted under these particular conditions would be the ones that most commonly occurred. This effect was observed for, and the analysis was adjusted accordingly.

Once the simulation conditions had been selected, the width and height of the machine stance, as well as the stride length, were altered one at a time in a number of simulations. All of the simulations consisted of 100 trials of 100 steps.

6.8.1 Varying the stride

Equation 5.2 was used, in Section 5.4, to determine the model machine stride length from the lateral foot placement distance desired and the maximum forward horizontal angle of the legs of the machine. For a given lateral foot placement distance, the maximum forward horizontal angle, \( \theta_{\text{max}} \), determines the maximum forward distance that the feet of a machine can travel as follows (Equation 5.1):

\[
lateral\ foot\ placement\ distance \cdot \tan \theta_{\text{max}} \tag{6.5}
\]

Now as explained in Section 5.4, this maximum distance was multiplied by 0.6 to determine the stride length used throughout the simulations. The factor of 0.6 was empirically observed to produce reliable machine performance in preliminary simulations. To further investigate the simulated machine response to this factor, it was varied from 0.4 to 0.7 in increments of 0.1. The lower limit was set to 0.4 because when the stride length was made shorter than the length defined by 0.4, the stability triangles were too small to facilitate walking. This problem was similar in nature to the one that resulted from excessive turning rates, as described in Section 4.9. The upper limit was set to 0.7 because when strides longer than the length defined by 0.7 were used, the legs over-extended almost immediately.

The stride lengths generated by the factors from 0.4 to 0.7 were simulated in the standard conditions. The trial completion rates for the various factors are shown in Figure 6.24. These results were somewhat unexpected. It was thought that the shorter stride lengths, due to the fact that they were more conservative, would result in more successful completions than the longer stride lengths. The reason that they did not was that when the stride length was shorter, the foot in front of a rear leg that was about to swing had to travel further backward to facilitate that swing. Figure 6.23 shows why this was the case. The extra distance that the front feet had to travel increased the chances that they would over-extend.
Such over-extensions were the cause of the majority of the failures that occurred with the shorter stride lengths.

When the factor of 0.7 was used, the majority of the failures were due to leg over-extensions. Section 4.14 explains why overly long stride lengths result in this form of failure. As stated above, factors above 0.7 resulted in almost immediate leg over-extensions. The factor of 0.6 was thus found to be the one that produced the best simulated machine performance.

Figure 6.25 shows the effects on the average speed of the simulated machine of varying the stride length. The speed increased with the stride as predicted in Section 3.6. Correspondingly, the distance travelled by the machine during the trials increased from an average of 31.0m at a factor of 0.4, to 44.2m at a factor of 0.7. Figure 6.25 also includes lines on both sides of the average speed line that indicate one standard deviation of the average speed. The standard deviation did not vary much, as the parameters that could substantially alter the difficulty of the movement planning, especially the roughness, were held constant throughout the simulations.

6.8.2 Varying the stance width

To determine the effects of altering the width of the simulated machine stance, the lateral distance that the feet were placed relative to the machine was varied in a number of simulations. As described in Section 5.4, this lateral foot placement distance was normally set to 1.3 times the upper leg segment length. This was empirically observed to produce reliable machine performance in preliminary simulations. To verify the validity of this choice, the lateral foot placement distance was varied from 1.1 times the upper leg segment length to 1.5 times the upper leg segment length in increments of 0.1. The trial completion rates that resulted from the various lateral foot placement distances are shown in Figure 6.26. When the lateral foot placement distance was made large, the failures that resulted were mainly due to leg over-extensions. When it was made small, the access range of the feet became too limited, and the failures resulted mainly from the front legs over-extending as they moved backward to facilitate rear leg swings. Lateral foot placements distances outside the range used in these simulations resulted in almost immediate failures due to these effects. These results corresponded exactly with the predictions of performance made in Section 5.4.
Figure 6.23: The Relationship Between Front Foot Travel and Stride Length
Figure 6.24: The Effects on Completion Rate of Varying the Stride Length
Figure 6.25: The Effects on Speed of Varying the Stride Length
Chapter 6. Heuristic Simulation

Figure 6.26: The Effects on Completion Rate of Varying the Stance Width
6.8.3 Varying the stance height

To determine the effects of altering the height of the simulated machine stance, it was varied, in a number of simulations, from 0.4m to 1.2m in increments of 0.2m. The trial completion rates that resulted from these simulations are shown in Figure 6.27. When the height was made large, the failures that resulted were mainly due to leg over-extensions. When the height was made small, the failures that resulted were mainly due to the machine being unable to access new footholds that were too high relative to it. They were also due to the vertical angle of the legs being rotated too far upward. Stance heights outside the range used in these simulations resulted in almost immediate failures from these effects. While a height of 0.4m resulted in poor performance of the simulated machine, Figure 6.27 shows that, in general, smaller heights improved performance. Section 5.4 discusses many of the reasons for this. The main conclusion of Section 5.4 is that the height should be made as small as possible. The simulation results bear this out.

Figure 6.27 indicates that a height of 0.6m resulted in better performance than a height of 0.8m. The latter height was the one used in all of the other simulations of the LOGHAQ heuristics. The difference between the two responses was not great however, and the larger height had the advantage of additional ground clearance. Further, the simulation conditions used included a terrain tilt of 30°. It is on sloped terrain that smaller heights improve performance (see Section 5.4), and the simulated machine was not always placed on sloped terrain, so it was felt that the height of 0.8m had been the best to use.

6.9 The Effects of Roughness in Varying Terrain Conditions

To determine the general effect of roughness in varying terrain conditions, a final set of simulations were performed. While the first simulations performed investigated the effects of roughness on a simulated machine crossing flat terrain, it was felt that another analysis should be made into the general effects of roughness. So a number of simulations were performed with different combinations of terrain attitude and roughness. The attitude was varied from a terrain tilt of zero to 45° in increments of 15°. The terrain roughness was varied from zero to 1.0m in increments of 0.1m. The terrain roll was kept constant at zero. The trial completion rates that resulted from these simulations, in which 100 trials of 100 steps were made, are shown in Figure 6.28. This figure shows that at each of the terrain tilts, the simulated machine was able to perform with complete success up to a certain roughness. Then, as the roughness was increased beyond that point, the performance at that terrain tilt decayed rapidly until no trials
Figure 6.27: The Effects on Completion Rate of Varying the Stance Height
could be completed. Further, the roughness at which this decay began occurred sooner the greater the terrain tilt was. This indicates that, for any given underlying terrain attitude, there is a roughness level that should not be exceeded if failure free machine response is desired. So for a given machine design, the response of that machine is determined by the overall characteristics of the terrain it is crossing, where these characteristics are a combination of the underlying terrain attitude and the terrain roughness.

It is also apparent in Figure 6.28 that the responses to the flat and 15° tilt terrain were not significantly different in terms of completion rates. Further, increasing the terrain tilt from 30° to 45° had a much greater effect on the decaying of the machine response than increasing the terrain tilt from 15° to 30°. These two observations indicate that, to the LOGHAQ heuristics, the effective difficulty of the terrain may increase in some exponential manner with the increase in the underlying terrain attitude.
Chapter 7

Conclusions and Recommendations for Further Work

7.1 Summary of the Study

This investigation was concerned with the development of a longitudinally oriented movement planner for quadruped walking machines. In the initial stages of the investigation, walking machine research of the past was considered, and it was determined that a movement planner of this nature would be valuable. Most of the recent work on walking machines has involved hexapod designs, and the work that has involved quadrupeds was concerned with machines that could not handle rough terrain, or with machines that crossed rough terrain without following its contours. It was felt that movement of the latter type did not provide the best performance capability on sloped terrain. Thus, a quadruped movement planner that made use of the longitudinally oriented, terrain following strategies of hexapod machines was felt to be of value. The motivation for extending the work in the field of quadrupeds, when this is actually the less popular design, was that quadrupeds may be more suited to quasi-dynamic operation. Quasi-dynamic operation greatly increases the movement speeds possible for a walking machine. This mode of operation is difficult to achieve however, and much remains to be learned about it. Investigations such as this one, in the realm of statically stable quadruped walking machines, help prepare the background for this potential mode of operation.

After it had been decided that a longitudinally oriented movement planner for a quadruped would be useful, investigations were made into the general characteristics of quadruped ambulation. These investigations started with previous work in the field, and lead to the development of new conceptions of the quadruped walking problem. The final result was a general movement planning strategy for quadrupeds. This strategy was further developed, in conjunction with extensive computer simulations, to produce a complete forward movement planning package that consisted of a number of heuristics, the LOGHAQ heuristics. These heuristics generated a continuous forward movement plan that followed the terrain. The heuristics did not provide dynamic control, instead they determined a series of joint angles settings that a dynamic controller could follow. The movement plans were not generated in a manner
that was oblivious to the actual response of the machine however. Every sample time the plans were recalculated entirely using only the current machine state and an overall movement plan. The latter consisted of general movement guidelines that were determined after each foot had completed its forward transition, and was not affected by dynamic response of the machine.

When the LOGHAQ heuristics had been completely developed, consideration was made of how a machine should be designed to best respond to them. It was also determined how the heuristic controlling parameters should be adjusted to result in the best machine performance. This analysis resulted in the development of a number of general design and operation guidelines that applied not only to quadrupeds operating under control of the LOGHAQ heuristics, but to all quadrupeds. In the process of developing these guidelines, a model machine was generated that was subsequently subjected to simulated testing.

Finally, extensive simulations were performed to test the capabilities of the LOGHAQ heuristics, and to determine the validity of the general guidelines that had been developed. These simulations were carefully constructed to ensure thorough, controlled testing of the heuristics. A wide variety of terrain conditions were simulated, and the machine response to them during a large number of crossings was recorded. These tests revealed that the model machine controlled by the LOGHAQ heuristics could handle, with few crossing failures, a wide variety of terrain attitudes, as well as changes from one attitude to another. The tests also confirmed all of the concepts related to the general characteristics of quadruped walking behaviour that were developed during the course of the study. In addition, a few more characteristics of quadruped ambulation were discovered.

7.2 Results of the Simulations

During the simulations of the LOGHAQ heuristics, a number of observations were made. In general, it was found that the heuristics were able to guide the simulated machine over a wide variety of terrain conditions with a considerable degree of success. On perfectly smooth terrain (no roughness), the simulated machine was able to operate on upward terrain tilts of up to 50°, on downward terrain tilts of up to −55°, and on terrain rolls of up to 43°. These values indicate the performance limits of the simulated machine on perfectly smooth terrain. When the terrain roughness was set to 0.5m (which was 28% of the model machine length and 42% of the length of each leg segment), the machine successfully completed 100 steps 74% of the time when the terrain was tilted either upward or downward 45°. Again at a roughness of 0.5m, the simulated machine was successful 68% of the time when the terrain was rolled
15°, and 46% of the time when the terrain was both tilted and rolled 15° (see Section 6.5). Terrain roll conditions affected the performance of the machine more drastically because terrain roll greatly increased the probability of leg over-extensions occurring (see Section 5.4). The overall simulation performance indicated that, for a given walking machine design, the LOGHAQ heuristics should be able to guide that machine over a useful range of terrain attitudes. The exact performance of the heuristics would depend on the actual machine dimensions, and on the roughness of the terrain.

It should be repeated (from Section 4.14) that all of the movement failures that occurred during the simulations could have been recovered from. Doing so would have required shuffling of the feet of the machine however, which was not within the capabilities of the LOGHAQ heuristics. These heuristics facilitated only continuous forward movement.

The simulations revealed that the response of the simulated machine to differing terrain roughness levels followed a pattern. For different terrain attitudes, the same general response was observed as the roughness was increased. The machine completed all of its 100 step trials until the roughness reached a certain level. Then the trial completion rate dropped off rapidly until no trials were completed. The roughness level at which this decay in performance occurred depended on the underlying terrain attitude (see Section 6.9 and Figure 6.28). It would thus be feasible, given a specific machine design, to determine operational limits for a set of expected general terrain attitudes. At each terrain attitude, the maximum roughness level at which the machine will operate with complete success could be determined.

It was found during the simulations that numerous aspects of the machine performance degraded as the terrain conditions were made more difficult. These terrain conditions consisted of both the underlying terrain attitude and its roughness. The distance travelled by the machine over a given number of steps decreased as the difficulties of the terrain prevented selection of foothold locations that would best match the requirements of the LOGHAQ heuristics. For similar reasons, the average speed of the machine decreased, and the standard deviation of this average speed increased. The errors in the tilt and roll of the machine platform relative to the actual terrain attitude also increased as the terrain conditions became more challenging (see Sections 6.4 and 6.5). These general degradations in performance also increased as the turning rate of the machine was increased. This resulted because increased turning rates made movement planning more difficult (Section 6.6). All of these effects were expected.

One of the main goals during the development of the LOGHAQ heuristics was to design the heuristics so that they would result in consistent speeds being maintained as the machine moved forward. It was
found during the simulations, by considering the standard deviation of the average speed, that consistent speeds were in fact maintained under most simulation conditions. Extreme terrain slopes, high levels of terrain roughness, and high turning rates resulted in more inconsistent speeds however. This was an expected response.

The simulated machine performed with fairly high levels of success when it crossed from terrain of one underlying attitude to terrain of another (Section 6.7). Its response was highly dependent on the nature of the terrain change and the severity of the difference between the two terrain attitudes however.

In addition to these general observations, a number of specific expectations of performance were borne out:

- The speed was found to increase as the stride length was increased, as predicted in Section 3.6 (Section 6.8). Excessive stride lengths resulted in leg over-extension failures however, as predicted in Section 4.14.

- An optimal lateral foot placement distance relative to the machine was found, as predicted in Section 5.4. This distance, 1.3 times the upper leg segment length, was close enough to the machine to prevent the legs from having to stretch too far outwards, and far enough away from the machine to prevent excessive shortening of the foot access range (Section 6.8).

- It was found, as predicted in Section 5.4, that smaller machine stance heights resulted in more successful machine response (Section 6.8).

- Equation 4.48 (Section 4.9) accurately predicted the maximum feasible turning rate that the simulated machine could adopt (Section 6.6).

- When the machine was travelling across upward sloping terrain, its failures were mainly due to leg over-extensions (Section 6.5). When it was travelling across downward sloping terrain, its failures were mainly due to inaccessible front footholds (Section 6.5). Both of these effects were expected and were due to the relationship between the center of gravity of the machine and its supporting foothold locations on the sloped terrain (see Section 5.4).

- The machine made attitude adjustments to reach front footholds far more often than rotational adjustments to reach rear footholds, as predicted in Section 4.14 (Section 6.4).
It was discovered during the simulations that short stride lengths, instead of resulting in slow but unfailing movement as predicted in Section 3.6, actually resulted in high levels of machine failure. The reasons for this are explained in Section 6.8.

7.3 Contributions of This Work

This study resulted in the compilation and extension of a wide range of knowledge about movement planning for longitudinally oriented quadruped walking machines. Such a compilation, especially in regard to quadrupeds that can handle rough terrain, has not, as far as this researcher knows, been made to date. In addition to this, a number of specific contributions were made to the field of quadruped ambulation during the course of developing the LOGHAQ heuristics. These included:

- The concept of linked stability triangle support patterns was developed, as well as how these patterns can be used to determine an overall movement plan (see Section 3.4).
- The understanding was extended about how various aspects of quadruped walking machines, especially their stride length, affect the maximum possible speed of these machines (see Section 3.6).
- A complete strategy was developed for determining how the placement of new footholds should be guided to best facilitate smooth turning (see Section 4.9).
- Extensive developments were made regarding how to maintain consistent machine speeds throughout all phases of movement (see Section 4.11).
- A thorough investigation was made into what conditions force quadrupeds to halt their forward motion. Techniques to facilitate continued movement, once this had occurred, were developed (see Section 4.14).
- A complete strategy for avoiding leg collisions was developed (see Section 4.16).
- The practical considerations of designing a quadruped walking machine were fully discussed (see Chapter 5).
- An improved method of specifying simulated terrain was developed (see Section 6.3).
Chapter 7. Conclusions and Recommendations for Further Work

7.4 Future Work Possibilities

A wide range of work could be performed to extend that done in this study. Such work could include simple improvement of the LOGHAQ heuristics, inclusion of them in a complete movement control package, and extension of this, and other quadruped research to provide quasi-dynamic walking capabilities.

7.4.1 Dynamic application of the LOGHAQ heuristics

The LOGHAQ heuristics were never tried on a machine, either simulated or real, that had dynamic response characteristics. It was felt that the recalculation of the movement plan every sample time would ensure realistic response of the planner to dynamic machine behavior, but this was not proven. Dynamic application of the heuristics may reveal practical weaknesses, and will definitely highlight aspects of the heuristics that have to be “tweaked” to provide reasonable performance. For example, as a machine controlled by the LOGHAQ heuristics moves, it cycles through a number of states. During the simulations performed in this study, the transition between these states was determined by machine responses that, having no dynamic disturbances affecting them, were always progressive. In a real application, such smooth progression cannot be ensured. The legs of a machine with dynamic response will bounce when they hit the ground, and waver when they are in the air. To prevent such behavior from causing the LOGHAQ heuristics to oscillate between states, some hysteresis would have to be included in the state transition determinations.

It may also be found that the LOGHAQ heuristics have to be adapted separately to every machine upon which they are used. This would particularly be the case if the general design of a machine that was using the heuristics differed from the design outlined in this paper. A different leg design could be used for example, but some modifications to the heuristics would be required.

7.4.2 Improvements to the LOGHAQ heuristics

The most obvious improvement that could be made to the LOGHAQ heuristics would be to modify them so that the machine being controlled by them never has to halt its forward motion. This would be very difficult to accomplish however. This is because the footholds that are presented as available to a walking machine are not selected by the machine itself, they are chosen by a scanning system (possibly a human) that selects each foothold out of those that are accessible on the terrain. It is the footholds that determine the long term behavior of a walking machine. If a machine that is being controlled by
Chapter 7. Conclusions and Recommendations for Further Work

The LOGHAQ heuristics can place its feet wherever the heuristics dictate, the machine will never have to halt. This was shown in simulations over a wide range of terrain conditions (see Chapter 6). So the failures that occur when a machine is being controlled by the LOGHAQ heuristics are due to footholds being chosen that are not at the locations desired by the heuristics. The control over the foothold selection could be improved however. The interaction between the LOGHAQ heuristics and the foothold scanning system could be increased to facilitate better selections being made. Further, the long term effects of each foothold choice could be determined by projecting the future movements of the machine. Then footholds that will lead to the machine halting can be rejected. This technique was used by both Hirose et al. (1984a and 1984b) and McGhee et al. (1984, or Ozguner, 1984). Both of these solutions, especially the latter, would require extensive computational effort. Neither solution guarantees that a suitable foothold will be found, they just ensure that if there is one, it will be located.

The LOGHAQ heuristics could be modified so that it will be known before-hand when the machine being controlled by them is going to halt. Then the heuristics could slow down the machine so that an abrupt halt does not result. This was discussed further in Section 5.3.

When a machine being controlled by the LOGHAQ heuristics is travelling slowly, there is little danger of it suddenly reaching the front edge of the current stability triangle. Further, even if it should reach the front edge, its low speed would not result in an abrupt halt taking place. The buffer distance is what is used to prevent both of these problems, but it also affects the speed of the movement (see Section 4.11). The larger the buffer distance is, the lower the machine speed. So when the machine is moving slowly, the buffer distance could be reduced, with little ill effect, to improve the speed.

Finally, as stated in Section 5.4 and shown in Figure 4.3, when a machine operating under LOGHAQ heuristic control is on terrain with a roll to it, the legs on one side of the machine become over-extended and the legs on the other side become under-extended. This occurs because the horizontal projection of the center of the machine is made to lie half way between the horizontal projection of the feet on both sides of the machine. To reduce the extension problem, the machine could be allowed to move laterally in the direction of the lower feet. Such an action would affect how the stability triangles are crossed however, and would require some changes in the planning of overall movements (see Section 4.7).

7.4.3 A complete movement control package

The LOGHAQ heuristics develop a plan that facilitates continuous forward movement of a longitudinally oriented quadruped. They do not however, allow for any other type of movement. A complete movement
package could be developed that incorporates the LOGHAQ heuristics, and also includes:

- In-place turning to provide higher turning rates than those possible with only forward movement.
- Shuffling of the feet to allow a machine to continue moving after its footholds have forced a halt (see Section 4.14).
- Side-stepping.
- Backing up (likely a simple reverse application of the LOGHAQ heuristics).
- Obstacle crossing strategies (Choi, 1988).
- Operator adjustment of machine attitude and height (Waldron, 1986).
Appendix A

Glossary of LOGHAQ Terms

**Access range** the region, determined by the joint angle limits of its legs, that the foot of a walking machine can move within.

**Buffer distance** the distance back from the front edge of the stability triangles that the center of gravity of a machine moves toward.

**Creep** the phase of quadruped motion when the machine is moving forward with all four feet supporting.

**Follow the leader** a walking technique where rear feet are placed in, or near, footholds used by the front feet (see Section 3.3).

**Following distance** the short distance behind the front feet that the rear feet are placed in a follow the leader gait.

**Ground clearance angle** the arbitrarily defined vertical angle that the leg of a machine adopts to avoid striking the ground as it swings forward.

**Half-cycle** the period in which a single stability triangle is crossed, and within which the two legs on one side of a machine are swung forward.

**Hovering** the positioning of a leg in the air above a foothold that it cannot reach (see Section 4.12).

**Knee** the one degree of freedom joint that connects the two segments of a leg.

**Lateral foot placement distance** the distance from the side of the machine that it is preferred that the footholds be made with.

**LOGHAQ** longitudinally oriented gait heuristics for ambulating quadrupeds.

**Orientation vectors** the three orthogonal vectors that define the attitude of a walking machine platform (see Section 4.3).

**Over-extension** movement of a leg beyond the reach defined by the angular limitations of its joints.

144
Appendix A. Glossary of LOGHAQ Terms

Quadruped crawl the most stable low speed quadruped gait (see Section 3.2).

Roughness the maximum distance, in a terrain simulation, that a chosen foothold can be perturbed from its preferred location (see Section 6.3).

Settling factors parameters that determine how much of the old trajectory and attitude of a machine is used in determining the planned trajectory and attitude changes in a new half-cycle of movement.

Shoulders the two degree of freedom joint where the legs of a walking machine connect to its platform.

Stability margin the distance that must always be maintained between the machine center of gravity and the edge of the stability region.

Stability region the region defined by the horizontal projection of the supporting feet of a walking machine that the horizontal projection of its center of gravity must remain within.

Stability triangles the triangular stability regions defined by the three supporting feet of a quadruped (see Section 3.4).

Stride length the spacing distance between the front feet in the forward direction of a machine (this differs from traditional definitions).

Terrain following walking machine movement where the attitude of the machine is adjusted to match the perceived attitude of the terrain.

Under-extension movement of a leg too close to its shoulders so that it exceeds the angular limitations of its joints.
Appendix B

The LOGHAQ Code

The following is the "C" code implementation of the LOGHAQ heuristics. This code is fully documented in the paper that precedes this Appendix.
/***************************************************************/

/* The following are the global variables used by the LOGHAQ heuristics. */

/* machine position variables */
float xl, yl, zl, x2, y2, z2, x3, y3, z3, x4, y4, z4;
float afl, af2, af3, af4, au1, au2, au3, au4, ao1, ao2, ao3, ao4;

/* heuristic variables */
float afld, af2d, af3d, af4d, auld, au2d, au3d, au4d, aold, ao2d, ao3d, ao4d;
float tilt, tiltt, tiltl, roll, rollt, rolll, turnt, turnl;
float iss, jss, kss, ias, kas, ius, jus, kus, si, sj, sk;
float xf, yf, zf, xt, yt, zt;
float typ1, typ2, audrec, tradeg;

/* operator set variables */
float setlm, setlt, setlh, setlr, degrees, strid, height, wide;

/* message control variables */
int falt_num, falt_cas, messfl[20];
Appendix B. The LOGHAQ Code

main()
{
    int failcnt, new_leg, lift_leg, lifted, halted, trydown;
    int hung, rem, change, down_check(), stab_check();
    float ispeed, speed;

    /* these procedures reset the machine to simulation conditions */
    init_parameters();
    init_machine(&speed, &lifted, &halted, &trydown, &lift_leg);

    failcnt = 0;
    while (failcnt <= 5) {
        assess_move(lifted, lift_leg, speed);
        calc_pred_orient(lift_leg, rem);
        new_leg = FALSE;
        if (trydown) new_leg = down_check();
        if (new_leg) {
            halted = lifted = FALSE;
            update_swing_leg(&lift_leg);
            if (lift_leg == 3 || lift_leg == 4) hc_move_plan(lift_leg);
            calc_tar_ang(lift_leg);
            calc_pred_orient(lift_leg, 0);
            foot_deter(lift_leg);
            trydown = FALSE;
        }
        if (!halted) {
            if (lifted) halted = !stab_check(lift_leg, speed);
            else lifted = stab_check(lift_leg, -1.0);
            if (!halted) halted = !stab_check(0, speed);
        }
        if (!halted) ispeed = 0.0;
        else speed_calc(lift_leg, lifted, speed, ftispeed);
        hung = FALSE;
        if (lifted)
            swg_leg_kine(new_leg, lift_leg, ispeed, &rem, &trydown, &hung);
        else approx_remain(ispeed, &rem);
        if (hung && halted) adjust_attitude(lift_leg, &rem);
        body_kine(lift_leg, lifted, rem, &halted, &change, ispeed, speed);
        if (hung && halted && !change) abort_adjust();
        if (lifted && (lift_leg == 3 || lift_leg == 4)) prev_leg_col(lift_leg);
        if ((hung || !lifted) && halted && !change) failcnt += 1;
        else failcnt = 0;
        reset_messfl();
    }
    describe_failure(lift_leg, lifted);
/*****************************^  
assess_move(lifted, lift_leg, speed)

int lifted, lift_leg;
float *speed;
{
  float turnest, xdisest, ydisest, zdisest, ang;
  float xl0, yl0, zl0, x2o, y2o, z2o, x3o, y3o, z3o, x4o, y4o, z4o;
  float ja, ka, is, js, ks, iu, ju, ku, df, ds, du, arctan(), mag3();
  int ignore;
  double sin(), cos();

  extern float afl, af2, af3, af4, au1, au2, au3, au4;
  extern float ao1, ao2, ao3, ao4, tilt, roll;

  /* changed */
  extern float xl, yl, zl, x2, y2, z2, x3, y3, z3, x4, y4, z4;
  extern float xf, yf, zf, xt, yt, zt, turnt, turnl;

  xlo = xl; ylo = yl; zlo = zl;
  x2o = x2; y2o = y2; z2o = z2;
  x3o = x3; y3o = y3; z3o = z3;
  x4o = x4; y4o = y4; z4o = z4;

  if (lifted == FALSE) {
    switch(lift_leg) {
      case 1: ignore = 3; break;
      case 2: ignore = 4; break;
      case 3: ignore = 2; break;
      case 4: ignore = 1;}}
  else ignore = lift_leg;

  ja = cos(tilt);
  ka = sin(tilt);
  is = cos(roll);
  iu = sin(roll);
  js = iu * ka;
  ks = -iu * ja;
  ju = -is * ka;
  ku = is * ja;
get_distances(af1, au1, ao1, &df, &ds, &du);
x1 = -is * (WIDTH + ds) + iu * du;
y1 = ja * (BASE + df) - js * (WIDTH + ds) + ju * du;
z1 = ka * (BASE + df) - ks * (WIDTH + ds) + ku * du;

get_distances(af2, au2, ao2, &df, &ds, &du);
x2 = is * (WIDTH + ds) + iu * du;
y2 = ja * (BASE + df) + js * (WIDTH + ds) + ju * du;
z2 = ka * (BASE + df) + ks * (WIDTH + ds) + ku * du;

get_distances(af3, au3, ao3, &df, &ds, &du);
x3 = -is * (WIDTH + ds) + iu * du;
y3 = -ja * (BASE - df) - js * (WIDTH + ds) + ju * du;
z3 = -ka * (BASE - df) - ks * (WIDTH + ds) + ku * du;

get_distances(af4, au4, ao4, &df, &ds, &du);
x4 = is * (WIDTH + ds) + iu * du;
y4 = -ja * (BASE - df) + js * (WIDTH + ds) + ju * du;
z4 = -ka * (BASE - df) + ks * (WIDTH + ds) + ku * du;

turnest = 0.0;

if (ignore != 1 && ignore != 2) {
  ang = arctan(y1 - y2, x1 - x2) - arctan(ylo - y2o, xio - x2o);
  if (ang > PI) ang -= 2.0 * PI;
  if (ang < -PI) ang += 2.0 * PI;
  turnest += ang;
}

if (ignore != 3 && ignore != 1) {
  ang = arctan(y3 - y1, x3 - x1) - arctan(y3o - ylo, x3o - xio);
  if (ang > PI) ang -= 2.0 * PI;
  if (ang < -PI) ang += 2.0 * PI;
  turnest += ang;
}

if (ignore != 4 && ignore != 3) {
  ang = arctan(y4 - y3, x4 - x3) - arctan(y4o - y3o, x4o - x3o);
  if (ang > PI) ang -= 2.0 * PI;
  if (ang < -PI) ang += 2.0 * PI;
  turnest += ang;
}

if (ignore != 2 && ignore != 4) {
  ang = arctan(y2 - y4, x2 - x4) - arctan(y2o - y4o, x2o - x4o);
  if (ang > PI) ang -= 2.0 * PI;
  if (ang < -PI) ang += 2.0 * PI;
  turnest += ang;
}

turnest = turnest / 3.0;
turnt += turnest;
turnl += turnest;
Appendix B. The LOGHAQ Code

rotate(turnest, &xt, &yt);
rotate(turnest, &xf, &yf);
rotate(turnest, &x1o, &y1o);
rotate(turnest, &x2o, &y2o);
rotate(turnest, &x3o, &y3o);
rotate(turnest, &x4o, &y4o);

xdisest = ydisest = zdisest = 0.0;

if (ignore != 1) {
    xdisest += x1 - x1o;
    ydisest += y1 - y1o;
    zdisest += z1 - z1o;
}
if (ignore != 2) {
    xdisest += x2 - x2o;
    ydisest += y2 - y2o;
    zdisest += z2 - z2o;
}
if (ignore != 3) {
    xdisest += x3 - x3o;
    ydisest += y3 - y3o;
    zdisest += z3 - z3o;
}
if (ignore != 4) {
    xdisest += x4 - x4o;
    ydisest += y4 - y4o;
    zdisest += z4 - z4o;
}

xdisest = xdisest / 3.0;
ydisest = ydisest / 3.0;
zdisest = zdisest / 3.0;
*speed = mag3(xdisest, ydisest, zdisest);

xf += xdisest;  yf += ydisest;  zf += zdisest;
xt += xdisest;  yt += ydisest;  zt += zdisest;

******************************************************************************
calc_pred_orient(lift_leg, rem)

int lift_leg, rem;
{
    float tiltr, rollr, turnr;
    double cos(), sin();

    extern float tilt, tiltl, roll, rolll, turnl;

    /* changed */
    extern float ias, jas, kas, iss, jss, kss, ius, jus, kus, si, sj, sk;

    tiltr = tiltl; rollr = rolll; turnr = turnl;

    if (rem != 0) {
        if (tiltr > (tilt + rem * MAXTIL)) tiltr = tilt + rem * MAXTIL;
        if (tiltr < (tilt - rem * MAXTIL)) tiltr = tilt - rem * MAXTIL;
        if (rollr > (roll + rem * MAXROL)) rollr = roll + rem * MAXROL;
        if (rollr < (roll - rem * MAXROL)) rollr = roll - rem * MAXROL;
        if (turnr > (rem * MAXTUR)) turnr = rem * MAXTUR;
        if (turnr < (-rem * MAXTUR)) turnr = -rem * MAXTUR;
    }
Appendix B. The LOGHAQ Code

ias = 0.0;
jas = cos(tiltr);
kas = sin(tiltr);
iss = cos(rollr);
ius = sin(rollr);
jss = ius * kas;
kss = -ius * jas;
jus = -iss * kas;
kus = iss * jas;
rotate(turnr, &ias, &jas);
rotate(turnr, &iss, &jss);
rotate(turnr, &ius, &jus);

switch(lift_leg) {
case 1: si = BASE * ias - WIDTH * iss;
sj = BASE * jas - WIDTH * jss;
sk = BASE * kas - WIDTH * kss;
break;
case 2: si = BASE * ias + WIDTH * iss;
sj = BASE * jas + WIDTH * jss;
sk = BASE * kas + WIDTH * kss;
break;
case 3: si = -BASE * ias - WIDTH * iss;
sj = -BASE * jas - WIDTH * jss;
sk = -BASE * kas - WIDTH * kss;
break;
case 4: si = -BASE * ias + WIDTH * iss;
sj = -BASE * jas + WIDTH * jss;
sk = -BASE * kas + WIDTH * kss;
}

/******************************
update_swing_leg(lift_leg)
int *lift_leg;

{ extern float af1, af2;
  /* changed */
extern float typ1, typ2;

  switch(*lift_leg) {
    case 1: *lift_leg = 4;
      typ1 = typ1 + (af1 - typ1) * 0.4;
      break;
    case 2: *lift_leg = 3;
      typ2 = typ2 + (af2 - typ2) * 0.4;
      break;
    case 3: *lift_leg = 1;
      break;
    case 4: *lift_leg = 2;}}
Appendix B. The LOGHAQ Code

hc_move_plan(lift_leg)

int lift_leg;
{
  float ib, jb, kb, in, jn, kn, ia, ja, ka, is, js, ks, iu, ju, ku;
  float iap, jap, kap, iup, jup, kup, rollp, tiltp, turnp;
  float dis, xp, yp, zp, mag3();
  double sin(), cos();

  extern float xl, yl, zl, x2, y2, z2, x3, y3, z3, x4, y4, z4, tilt, roll;
  extern float height, setlm, setlt, setlh, setlr, tradeg; /* operator */

  /* changed */
  extern float turnt, tiltt, rollt, xt, yt, zt;

  xp = (x2 + xl) / 2.0;
  yp = (y2 + yl) / 2.0;
  zp = (z2 + zl) / 2.0 + height;

  dis = mag3(xp, yp, zp);
  ia = 0.0;
  ja = cos(tilt);
  ka = sin(tilt);
  is = cos(roll);
  iu = sin(roll);
  js = iu * ka;
  ks = -iu * ja;
  ju = -is * ka;
  ku = is * ja;
  plane_rotate(tradeg, &is, &js, &ks, &ia, &ja, &ka);

  xt = dis * ia + (xp - dis * ia) * setlm;
  yt = dis * ja + (yp - dis * ja) * setlm;
  zt = dis * ka + (zp - dis * ka) * setlm;

  if (lift_leg == 4) {
    ib = x2 - x3;
    jb = y2 - y3;
    kb = z2 - z3;
    iap = x1 - x3;
    jap = y1 - y3;
    kap = z1 - z3;
  }
  else {
    ib = x4 - x1;
    jb = y4 - y1;
    kb = z4 - z1;
    iap = x2 - x4;
    jap = y2 - y4;
    kap = z2 - z4;
  }

  iup = jb * kap - kb * jap;
  jup = kb * iap - ib * kap;
  kup = ib * jap - jb * iap;
Appendix B. The LOGHAQ Code

get_attitude(ia, ja, ka, iu, ju, ku, &turnt, &tiltt, &rollt);
get_attitude(iap, jap, kap, iup, jup, kup, &turnp, &tiltp, &rollp);

turnt += (turnp - turnt) * setlt;
tilt += (tiltp - tiltt) * setlh;
rollt += (rollp - rollt) * setlr;

******************************************************************************
calc_tar_ang(lift_leg)

int lift_leg;
{
    extern float turnl, tiltl, rolll;
    /* changed */
    extern float turnl, tiltl, rolll;

    if (lift_leg == 1 || lift_leg == 2) {
        turnl = turnt;
        tiltl = tiltt;
        rolll = rollt;
    }
    else {
        turnl = turnt / 2.0;
        tiltl = (tiltt + tilt) / 2.0;
        rolll = (rollt + roll) / 2.0;
    }

******************************************************************************
foot_deter(lift_leg)

int lift_leg;
{
    float is, js, ks, ia, ja, ka, w, cstride, oxf, oyf, ozf, bckstr, mstr;
    float m, t, mag2(), mag3(), arctan();
    double fabs(), tan(), cos();

    extern float xl, yl, zl, x2, y2, z2, x3, y3, z3, x4, y4, z4;
    extern float ias, jas, kas, iss, jss, kss;
    extern float strid, tilt, degrees, height; /* operator */

    /* changed */
    extern float xf, yf, zf, tradeg;

    if (lift_leg == 3) {
        xf = xl - ias * FTCLR;
        yf = yl - jas * FTCLR;
        zf = zl - kas * FTCLR;
    }
    if (lift_leg == 4) {
        xf = x2 - ias * FTCLR;
        yf = y2 - jas * FTCLR;
        zf = z2 - kas * FTCLR;
if (lift_leg == 1 || lift_leg == 2) {
  is = iss; js = jss; ks = kss;
  ia = ias; ja = jas; ka = kas;
  plane_rotate(degrees, &is, &js, &ks, &ia, &ja, &ka);
  tradeg = degrees;

  if (lift_leg == 1) {
    w = (x2 - xt) * is + (y2 - yt) * js + (z2 - zt + height) * ks + wide;
    xf = x2 - w * is;
    yf = y2 - w * js;
    zf = z2 - w * ks;
    m = (y2 - y3) / (x2 - x3);
    t = (xf * m - x3 * m - yf + y3) / (ja - ia * m);
    mstr = 4.0 * (MARGIN + FTCLR) / mag2(ia, ja) + t;
    xf += mstr * ia; yf += mstr * ja; zf += mstr * ka;

    if (degrees > 0.0) {
      cstride = w * tan(arctan(strid, 2.0 * wide) - degrees) - mstr;
      if (cstride < 0.0) {
        mess(15, degrees, 0, FALSE);
        cstride = 0.0;
        tradeg = arctan(strid, 2.0 * wide) - arctan(mstr, w);}
    } else {
      cstride = strid - mstr;
      bckstr = (xf - x3) * ia + (yf - y3) * ja + (zf - z3) * ka;
      if (((bckstr + cstride) > (2.0 * strid))
      cstride = 2.0 * strid - bckstr;
      if (cstride < 0.0) cstride = 0.0;}
  }

  else {
    w = (xt - xl) * is + (yt - yl) * js + (zt - height - zl) * ks + wide;
    xf = xl + w * is;
    yf = yl + w * js;
    zf = zl + w * ks;
    m = (yl - y4) / (xl - x4);
    t = (xf * m - x4 * m - yf + y4) / (ja - ia * m);
    mstr = 4.0 * (MARGIN + FTCLR) / mag2(ia, ja) + t;
    xf += mstr * ia; yf += mstr * ja; zf += mstr * ka;

    if (degrees < 0.0) {
      cstride = w * tan(arctan(strid, 2.0 * wide) + degrees) - mstr;
      if (cstride < 0.0) {
        mess(15, degrees, 0, FALSE);
        cstride = 0.0;
        tradeg = arctan(mstr, w) - arctan(strid, 2.0 * wide);}
    } else {
      cstride = strid - mstr;
      bckstr = (xf - x4) * ia + (yf - y4) * ja + (zf - z4) * ka;
      if (((bckstr + cstride) > (2.0 * strid))
      cstride = 2.0 * strid - bckstr;
      if (cstride < 0.0) cstride = 0.0;}
  }

  oxf = xf; oyf = yf; ozf = zf;

  get_foothold(cstride, is, js, ks, &xf, &yf, &zf);
if (mag3(xf - oxf, yf - oyi, zf - ozf) < (cstride / 2.0))
    mess(12, 0.0, 0, FALSE);
if (mag3(xf - oxf, yf - oyi, zf - ozf) > (cstride * 1.25))
    mess(13, 0.0, 0, FALSE);}

int stab_check(lift_leg, spd)
int lift_leg;
float spd;
{
    float iv, jv, margin, mag2();
    int balcd, report;
    extern float xl, yl, x2, y2, x3, y3, x4, y4;
    balcd = TRUE;
    if (spd == -1.0) report = FALSE;
    else report = TRUE;
    if (lift_leg != 1 & lift_leg != 2) {
        iv = xl - x2; jv = yl - y2;
        margin = (jv * x2 - iv * y2) / mag2(iv, jv);
        if (margin < (MARGIN + FTCLR)) {
            balcd = FALSE;
            if (report == TRUE) mess(14, spd, 0, TRUE);
            report = FALSE;}
    }
    if (lift_leg != 3 & lift_leg != 1) {
        iv = x3 - xl; jv = y3 - yl;
        margin = (jv * xl - iv * yl) / mag2(iv, jv);
        if (margin < MARGIN) balcd = FALSE;}
    if (lift_leg != 4 & lift_leg != 3) {
        iv = x4 - x3; jv = y4 - y3;
        margin = (jv * x3 - iv * y3) / mag2(iv, jv);
        if (margin < MARGIN) balcd = FALSE;}
    if (lift_leg != 2 & lift_leg != 4) {
        iv = x2 - x4; jv = y2 - y4;
        margin = (jv * x4 - iv * y4) / mag2(iv, jv);
        if (margin < MARGIN) balcd = FALSE;}
    if (lift_leg != 1 & lift_leg != 4 & lift_leg != 0) {
        iv = xl - x4; jv = yl - y4;
        margin = (jv * x4 - iv * y4) / mag2(iv, jv);
        if (lift_leg == 2 & margin < MARGIN) balcd = FALSE;
        if (lift_leg == 3 & margin < MARGIN) balcd = FALSE;}
    if (lift_leg != 2 & lift_leg != 3 & lift_leg != 0) {
        iv = x2 - x3; jv = y2 - y3;
        margin = (jv * x3 - iv * y3) / mag2(iv, jv);
        if (lift_leg == 4 & margin < MARGIN) balcd = FALSE;
        if (lift_leg == 1 & -margin < MARGIN) balcd = FALSE;}
    if (balcd == FALSE & report == TRUE) mess(0, spd, 0, TRUE);
    return(balcd);}
Appendix B. The LOGHAQ Code

speed_calc(lift_leg, lifted, speed, *ispeed)

int lift_leg, lifted;
float speed, *ispeed;
{
    float upper, lower, df, ds, ca, ma, cs, ms, afp, try, por, typsw;
    float xb, yb, zb, xo, yo, zo, t, dis, diso, dcel, arctan(), mag3(), mag2();
    double fabs(), cos();
    int i;

    extern float xt, yt, zt, af1, af2, af3, af4, xf, yf, zf, si, sj, sk;
    extern float las, jas, kas, iss, jss, kss, typ1, typ2, tilt;

    dis = mag3(xt, yt, zt) - BUFFER / cos(tilt);
    por = dis / mag3(xt, yt, zt);
    xb = xt * por; yb = yt * por; zb = zt * por;

    if (lifted == 1 || lift_leg == 2) {
        if (lift_leg == 1) afp = af1;
        else afp = af2;
        ds = (x - si - xb) * las + (y - sj - yb) * jas + (z - sk - zb) * kas;
        df = (x - si - xb) * iss + (y - sj - yb) * jss + (z - sk - zb) * kss;
        if (lift_leg == 1) ds *= -1.0;
        por = fabs(arctan(df, ds) - afp) / MAXANG;
        if (por < 1.0 || dis < 0.0) *ispeed = speed;
        else *ispeed = dis / por;
    }
    else {
        xo = yo = zo = 0.0;
        if (lift_leg == 3) {
            afp = af3;
            typsw = typ1;
        }
        else {
            afp = af4;
            typsw = typ2;
        }

        if (lifted == FALSE && lift_leg == 3) {
            t = (x4 - x3) * yl - (y4 - y3) * xl;
            t = t / ((x4 - x3) * yb - (y4 - y3) * xb) + MARGIN / mag2(xb, yb);
            xo = xb * t; yo = yb * t; zo = zb * t;
            df = las * (x3 - xo - si) + jas * (y3 - yo - sj) + kas * (z3 - zo - sk);
            ds = iss * (x3 - xo - si) + jss * (y3 - yo - sj) + kss * (z3 - zo - sk);
            afp = arctan(df, -ds);
        }
        else {
            t = (x4 - x3) * y2 - (y4 - y2) * x2;
            t = t / ((x4 - x3) * yb - (y4 - y2) * xb) + MARGIN / mag2(xb, yb);
            xo = xb * t; yo = yb * t; zo = zb * t;
            df = las * (x4 - xo - si) + jas * (y4 - yo - sj) + kas * (z4 - zo - sk);
            ds = iss * (x4 - xo - si) + jss * (y4 - yo - sj) + kss * (z4 - zo - sk);
            afp = arctan(df, ds);
        }

        ca = las * (x - xo - si) + jas * (y - yo - sj) + kas * (z - zo - sk);
        ma = las * (x - xo) + jas * (y - yo) + kas * (z - zo);
        cs = iss * (x - xo - si) + jss * (y - yo - sj) + kss * (z - zo - sk);
        ms = iss * (x - xo) + jss * (y - yo) + kss * (z - zo);

        if (lift_leg == 3) {
            cs = -cs; ms = -ms;
        }
Appendix B. The LOGHAQ Code

upper = 1.0; lower = 0.0;
for (i = 1; i <= 20; i++) {
    try = (upper + lower) / 2.0;
    df = ca - try * ma;
    ds = cs - try * ms;
    por = fabs(arctan(df, ds) - afp);
    por = por / (por + typsw + arctan(2.0 * BASE - df, ds));
    if (por < try) upper = try;
    else lower = try;
}
por = fabs(arctan(df, ds) - afp) / MAXANG;
if (por < 1.0) *ispeed = speed;
else *ispeed = try * mag3(xb - xo, yb - yo, zb - zo) / por;
}
diso = mag3(xo, yo, zo);
if (lifted == FALSE & & *ispeed < speed & & diso > 0.0) {
    dcel = (speed * speed - *ispeed * *ispeed) / 2.0 / diso;
    if (dcel < (MAXACL * 0.9)) *ispeed = speed;
}
if (lifted == FALSE & & *ispeed < (3.0 * MAXACL)) *ispeed = 3.0 * MAXACL;
if (*ispeed < (speed - MAXACL)) *ispeed = speed - MAXACL;
if (*ispeed > (speed + MAXACL)) {
    *ispeed = speed + MAXACL;
    if (speed < 0.0001) mess(2, 0.0, 0, FALSE);
}
if (*ispeed < MAXACL) *ispeed = MAXACL;
if (*ispeed < (MAXACL * 1.5) & & speed >= (MAXACL * 1.5))
mess(3, 0.0, 0, FALSE);
*
*******************************************************************************/

swg_leg_kine(new_leg, lift_leg, ispeed, rem, trydown, failed)

int new_leg, lift_leg, *rem, *trydown, *failed;
float ispeed;
{
    float xc, yc, zc, aff, aaf, aof, df, ds, du, afd, aod, aft;
    float ca, cs, ma, ms, cu, mu, reme, arctan();
    extern float xt, yt, zt, xf, yf, zf, si, sj, sk, typ1, typ2;
    extern float ias, jas, kas, iis, jss, kss, ius, jus, kus;
    extern float af1, af2, af3, af4, au1, au2, au3, au4, ao1, ao2, ao3, ao4;
    /* changed */
    extern float afd1, af2d, af3d, af4d, au1d, au2d, au3d, au4d;
    extern float aod1, aod2, aod3, aod4, audrec;
    xc = xt; yc = yt; zc = zt;
    normalize(&xc, &yc, &zc);
    xc *= ispeed; yc *= ispeed; zc *= ispeed;
    ca = ias * (xf - si) + jas * (yf - sj) + kas * (zf - sk);
    ma = ias * xc + jas * yc + kas * zc;
    cs = iss * (xf - si) + jss * (yf - sj) + kss * (zf - sk);
    ms = iss * xc + jss * yc + kss * zc;
    cu = ius * (xf - si) + jus * (yf - sj) + kus * (zf - sk);
    mu = ius * xc + jus * yc + kus * zc;
switch(lift_leg) {
    case 1: afd = af1; aud = au1; aod = ao1;
        cs = -cs; ms = -ms;
        break;
    case 2: afd = af2; aud = au2; aod = ao2;
        break;
    case 3: afd = af3; aud = au3; aod = ao3;
        cs = -cs; ms = -ms;
        break;
    case 4: afd = af4; aud = au4; aod = ao4;
}

afd = afd;
*failed = TRUE;

if (arctan(ca, cs) > afd) {
    for (*rem = 1; (aft <= (MAXFA + 2.0 * MAXANG)) && *failed; *rem += 1) {
        df = ca - *rem * ma;
        ds = cs - *rem * ms;
        du = cu - *rem * mu;
        get_angles(df, ds, du, &aff, &auf, &aof, failed, 0, 0);
        aft += MAXANG;
        if (aft > aff) *failed = TRUE;
    }
}
else {
    for (*rem = 1; (aft >= (MAXRA - 2.0 * MAXANG)) && *failed; *rem += 1) {
        df = ca - *rem * ma;
        ds = cs - *rem * ms;
        du = cu - *rem * mu;
        get_angles(df, ds, du, &aff, &auf, &aof, failed, 0, 0);
        aft -= MAXANG;
        if (aft > aff) *failed = TRUE;
    }
*rem -= 1;

    if (*failed == TRUE) {
        mess(4, 0.0, 0, FALSE);
        if (lift_leg == 1) aff = typ1;
        if (lift_leg == 2) aff = typ2;
        if (lift_leg == 3 || lift_leg == 4) aff = arctan(ca, cs);
        if (aff > MAXFA) aff = MAXFA;
        if (aff > MAXRA) aff = MAXRA;

        if (afd < (aff + MAXANG) && afd > (aff - MAXANG)) afd = aff;
        if (afd < aff) afd += MAXANG;
        if (afd > aff) afd -= MAXANG;
        if (aud < UPANG) aud += MAXANG;
    }
Appendix B. The LOGHAQ Code

```c
else {
    if (new_leg == TRUE) aud += MAXANG;
    else {
        reme = fabs(aof - aod) / MAXANG;
        if (reme > *rem) {
            mess(5, 0.0, 0, FALSE);
            *rem = reme + 1;
        }

        reme = fabs(auf - aud) / MAXANG;
        if (reme > *rem) *rem = reme + 1;

        if (aod < (aof + MAXANG) && aod > (aof - MAXANG)) aod = aof;
        if (aod < aof) aod += MAXANG;
        if (aod > aof) aod -= MAXANG;

        if (*rem == 1 || *trydown == TRUE) {
            afd = aff;
            aud -= MAXANG;
            *trydown = TRUE;
        } else {
            if (afd > (aff + MAXANG) && afd < (aff - MAXANG)) afd = aff;
            if (afd < aff) afd += MAXANG;
            if (afd > aff) afd -= MAXANG;

            if (aud > auf && ((aud - auf) / MAXANG) >= (*rem - 2.5))
                aud += (audrec - MAXANG) / 2.0;
            else {
                if (aud < UPANG || aud < (auf + PI / 4.0))
                    aud += (audrec + MAXANG) / 2.0;
            }

            if (aud > MAXUA) aud = MAXUA;
            if (aud < MAXDA) aud = MAXDA;
        }
    }
}
```

```c
switch(lift_leg) {
    case 1: af1d = afd; audrec = aud - au1; au1d = aud; aod1 = aod; break;
    case 2: af2d = afd; audrec = aud - au2; au2d = aud; aod2 = aod; break;
    case 3: af3d = afd; audrec = aud - au3; au3d = aud; aod3 = aod; break;
    case 4: af4d = afd; audrec = aud - au4; au4d = aud; aod4 = aod;}
```

```c
/approx_remain(ispeed, rem)
float ispeed;
int *rem;
{
    float dis, mag3();
    double cos();

    extern float tilt, xt, yt, zt;

    if (ispeed > 0.0) {
        dis = (mag3(xt, yt, zt) - BUFFER / cos(tilt)) / 2.0;
        if (dis < 0.0) *rem = 1;
        else *rem = dis / ispeed + 1;
    } else *rem = 1;
```
/*******************************
adjust_attitude(lift_leg, rem)

int lift_leg, *rem;
{
    float ia, ja, ka, is, js, ks, iu, ju, ku, rolla, turna, arctan();
    float xw, yw, zw;
    double fabs(), cos(), sin();

    extern float xf, yf, zf, tilt, roll;

    /* changed */
    extern float tiltl, rolll, turnl;

    if (lift_leg == 1 || lift_leg == 2) {
        mess(10, 0.0, 0, FALSE);
        *rem = 1;
        xw = xf + height * sin(roll);
        zw = zf + height * cos(roll) * cos(tilt);

        rolla = arctan(zw, fabs(xw));
        if (xf > 0.0) rolla *= -1.0;
        rolla -= roll;

        rolll = roll + rolla;
        if (rolla > (MAXROL / 3.0)) rolll = roll + MAXROL / 3.0;
        if (rolla < (-MAXROL / 3.0)) rolll = roll - MAXROL / 3.0;

        ja = cos(tilt);
        ka = sin(tilt);
        is = cos(rolll);
        iu = sin(rolll);
        js = iu * ka;
        ks = -iu * ja;
        ju = -is * ka;
        ku = is * ja;

        xw = xf + height * iu;
        yw = yf + height * ju;
        zw = zf + height * ku;

        iu = js * zw - ks * yw;
        ju = ks * xw - is * zw;
        ku = is * yw - js * xw;

        ia = ju * ks - ku * js;
        ja = ku * is - iu * ks;
        ka = iu * js - ju * is;

        get_attitude(ia, ja, ka, iu, ju, ku, &turna, &tiltl, &rolla);
        if ((tiltl - tilt) > (MAXTIL * 0.9)) tiltl = tilt + (MAXTIL * 0.9);
        if ((tilt - tiltl) > (MAXTIL * 0.9)) tiltl = tilt - (MAXTIL * 0.9);}
}
else {
    mess(16, 0.0, 0, FALSE);
    *rem = 1;
    if (lift_leg == 3) turnl = -MAXTUR / 2.0;
    else turnl = MAXTUR / 2.0;}}

/body_kine(lift_leg, lifted, rem, halted, change, ispeed, speed)/

int lift_leg, lifted, rem, *halted, *change;
float ispeed, speed;
{
    float xd, yd, zd, ia, ja, ka, is, ks, js, iu, ju, ku;
    float ih, jh, kh, df, ds, du, turnd, tiltd, rolld;
    double cos(), sin(), fabs();
    int failed, rst;
    extern float xt, yt, zt, tilt, tiltl, roll, rolll, turnl;
    extern float x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4;
    /* changed */
    extern float af1d, af2d, af3d, af4d, au1d, au2d, au3d, au4d;
    extern float a01d, a02d, a03d, a04d;
    xd = xt; yd = yt; zd = zt;
    normalize(&xd, &yd, &zd);
    xd *= ispeed; yd *= ispeed; zd *= ispeed;
    turnd = turnl / rem;
    tiltd = tilt + (tiltl - tilt) / rem;
    rolld = roll + (rolll - roll) / rem;
    if (tiltd > (tilt + MAXTIL)) tiltd = tilt + MAXTIL;
    if (tiltd < (tilt - MAXTIL)) tiltd = tilt - MAXTIL;
    if (rolld > (roll + MAXROL)) rolld = roll + MAXROL;
    if (rolld < (roll - MAXROL)) rolld = roll - MAXROL;
    if (turnd > MAXTUR) turnd = MAXTUR;
    if (-turnd > MAXTUR) turnd = -MAXTUR;

    *change = FALSE;
    if (fabs(tiltd - tilt) > (MAXTIL / 4.0)) *change = TRUE;
    if (fabs(rolld - roll) > (MAXROL / 4.0)) *change = TRUE;
    if (fabs(turnd) > (MAXTUR / 4.0)) *change = TRUE;
    ia = 0.0;
    ja = cos(tiltd);
    ka = sin(tiltd);
    is = cos(rolld);
    iu = sin(rolld);
    js = iu * ka;
    ks = -iu * ja;
    ju = -is * ka;
    ku = is * ja;
    rotate(turnd, &ia, &ja);
    rotate(turnd, &is, &js);
    rotate(turnd, &iu, &ju);
rst = FALSE;

if (lift_leg != 1 || lifted == FALSE) {
    ih = xi - (xd + BASE * ia - WIDTH * is);
    jh = yj - (yd + BASE * ja - WIDTH * js);
    kh = zk - (zd + BASE * ka - WIDTH * ks);
    df = ih * ia + jh * ja + kh * ka;
    ds = -ih * is - jh * js - kh * ks;
    du = ih * iu + jh * ju + kh * ku;
    get_angles(df, ds, du, &afid, &auid, &ao1d, &failed, 1, *halted);
    if (failed == TRUE) rst = TRUE;
}

if (lift_leg != 2 || lifted == FALSE) {
    ih = x2 - (xd + BASE * ia + WIDTH * is);
    jh = y2 - (yd + BASE * ja + WIDTH * js);
    kh = z2 - (zd + BASE * ka + WIDTH * ks);
    df = ih * ia + jh * ja + kh * ka;
    ds = ih * is + jh * js + kh * ks;
    du = ih * iu + jh * ju + kh * ku;
    get_angles(df, ds, du, &af2d, &au2d, &ao2d, &failed, 2, *halted);
    if (failed == TRUE) rst = TRUE;
}

if (lift_leg != 3 || lifted == FALSE) {
    ih = x3 - (xd - BASE * ia - WIDTH * is);
    jh = y3 - (yd - BASE * ja - WIDTH * js);
    kh = z3 - (zd - BASE * ka - WIDTH * ks);
    df = ih * ia + jh * ja + kh * ka;
    ds = -ih * is - jh * js - kh * ks;
    du = ih * iu + jh * ju + kh * ku;
    get_angles(df, ds, du, &af3d, &au3d, &ao3d, &failed, 3, *halted);
    if (failed == TRUE) rst = TRUE;
}

if (lift_leg != 4 || lifted == FALSE) {
    ih = x4 - (xd - BASE * ia + WIDTH * is);
    jh = y4 - (yd - BASE * ja + WIDTH * js);
    kh = z4 - (zd - BASE * ka + WIDTH * ks);
    df = ih * ia + jh * ja + kh * ka;
    ds = ih * is + jh * js + kh * ks;
    du = ih * iu + jh * ju + kh * ku;
    get_angles(df, ds, du, &af4d, &au4d, &ao4d, &failed, 4, *halted);
    if (failed == TRUE) rst = TRUE;
}

if (rst == TRUE) {
    move_rst(lift_leg, lifted);
    if (*halted == FALSE) mess(1, speed, 0, FALSE);
    *halted = TRUE;
    *change = FALSE;
}

/************************************************************/

abort_adjust()
{
    extern float tilt, roll;
    /* changed */
    extern float turnl, tiltl, rolll;

    turnl = 0.0;
    tiltl = tilt;
    rolll = roll;
/* Appendix B. The LOGHAQ Code */

prev_leg_col(lift_leg)

int lift_leg;
{
float aff, auf, aof, afr, aur, aor, afro, fkh, ffh, ffi, rkh, rfh, rfi;
float iv, fkv, ffv, rms, aflm, arctan();
double sin(), cos();

extern float af1d, af2d, au1d, au2d, au3d, au4d, ao1d, ao2d, ao3d, ao4d;
extern float af3, af4;

/* changed */
extern float af3d, af4d;

if (lift_leg == 3) {
    aff = afld; auf = aus; aof = ao1;
    afr = af3d; aur = au3d; aor = ao3d;
    afro = af3;
} else {
    aff = af2d; auf = au2d; aof = ao2d;
    afr = af4d; aur = au4d; aor = ao4d;
    afro = af4;
}

if (afr == aff) afr += PI / 45.0;

fkh = UPLL * cos(auf);
ffh = fkh - LOLL * cos(auf + aof);
ffi = BASE * 2.0 / sin(afr - aff) * sin(PI / 2.0 - afr);

if (fkh > ffh && afr > (aff + PI / 10.0)) {
    rkh = UPLL * cos(aur);
    rfh = rkh - LOLL * cos(aur + aor);
    rfi = BASE * 2.0 / sin(afr - aff) * sin(PI / 2.0 + aff);

    if (rfi < rfh) {
        iv = UPLL * sin(aur);
        iv = LOLL * sin(aur + aor) / (rfh - rkh) * (rfi - rkh);
    } else iv = UPLL * sin(aur) - LOLL * sin(aur + aor);

    fkv = UPLL * sin(auf);

    if (iv > fkv) rms = fkh + FTCLR / 2.0;
    else {
        ffv = fkv - LOLL * sin(auf + aof);
        rms = ffh + (fkh - ffh) / (fkv - ffv) * (iv - ffv) + FTCLR / 2.0;
    }

    aflm = arctan(2.0 * BASE + rms * sin(aff), rms * cos(aff));

    if (afr > aflm) {
        mess(11, 0.0, 0, FALSE);
        if ((afro - MAXANG) > aflm) afro = afro - MAXANG;
        if (lift_leg == 3) af3d = aflm;
    else af4d = aflm;
    }}
}
/***/
/* These procedures are those called by LOGHAQ's main procedures.  */

move_rst(lift_leg, lifted)
int lift_leg, lifted;
{
    extern float af1, af2, af3, af4, au1, au2, au3, au4;
    extern float ao1, ao2, ao3, ao4;
    /* changed */
    extern float afld, af2d, af3d, af4d, au1d, au2d, au3d, au4d;
    extern float aold, ao2d, ao3d, ao4d;
    if (lifted == FALSE) lift_leg = 0;
    if (lift_leg != 1) {
        afld = afl; auld = aul; aold = aol;}
    if (lift_leg != 2) {
        af2d = af2; au2d = au2; ao2d = ao2;}
    if (lift_leg != 3) {
        af3d = af3; au3d = au3; ao3d = ao3;}
    if (lift_leg != 4) {
        af4d = af4; au4d = au4; ao4d = ao4;}}

get_attitude(ia, ja, ka, iu, ju, ku, *turn, *tilt, *roll)
float ia, ja, ka, iu, ju, ku, *turn, *tilt, *roll;
{
    float hp, vp, mag2(), mag3(), arctan();
    *turn = arctan(-ia, ja);
    *tilt = arctan(ka, mag2(ia, ja));
    hp = (iu * ja - ju * ia) / mag2(ja, ia);
    vp = -iu * ia * ka - ju * ja * ka + ku * (ia * ia + ja * ja);
    vp = vp / mag3(ia * ka, ja + iu * ia * ja);
    *roll = arctan(hp, vp);}
Appendix B. The LOGHAQ Code

get_angles(df, ds, du, af, au, ao, failed, leg, halted)

int *failed, leg, halted;
{
    float x, ll, arctan(), arccos(), mag2();
    double asin(), sin(), fabs();

    *failed = FALSE;
    *af = arctan(df, ds);
    if (ds < 0.0) {
        if (leg != 0) mess(5 + leg, 0.0, 1, !halted);
        *failed = TRUE;
    }

    if (*af > MAXFA) {
        if (leg != 0) mess(5 + leg, 0.0, 2, !halted);
        *failed = TRUE;
    }

    if (*af < MAXRA) {
        if (leg != 0) mess(5 + leg, 0.0, 3, !halted);
        *failed = TRUE;
    }

    if (*failed == FALSE) {
        ll = mag3(df, ds, du);
        if (ll > MAXLL) {
            if (leg != 0) mess(5 + leg, 0.0, 4, !halted);
            *failed = TRUE;
        } else {
            *ao = arccos((LC1 - 11 * 11) / LC2);
            if (*ao < MAXIA) {
                if (leg != 0) mess(5 + leg, 0.0, 5, !halted);
                *failed = TRUE;
            }

            x = LOLL * sin(*ao) / ll;
            if (fabs(x) > 1.0) x = x / fabs(x);
            *au = arctan(du, mag2(df, ds)) + asin(x);

            if (*au > MAXUA) {
                if (leg != 0) mess(5 + leg, 0.0, 6, !halted);
                *failed = TRUE;
            }

            if (*au < MAXDA) {
                if (leg != 0) mess(5 + leg, 0.0, 7, !halted);
                *failed = TRUE;
            }
        }
    }
}

get_distances(af, au, ao, df, ds, du)

float af, au, ao, *df, *ds, *du;
{
    float beta, rad;
    double cos(), sin();

    beta = PI - ao - au;
    rad = UPLL * cos(au) + LOLL * cos(beta);
    *ds = rad * cos(af);
    *df = rad * sin(af);
    *du = UPLL * sin(au) - LOLL * sin(beta);
}
plane_rotate(turn, is, js, ks, ia, ja, ka)
{
    float mag, ish, jsh, ksh;
    double tan(), sqrtO;
    ish = *is; jsh = *js; ksh = *ks;
    mag = tan(turn);
    *is += mag * *ia;
    *js += mag * *ja;
    *ks += mag * *ka;
    *ia -= mag * ish;
    *ja -= mag * jsh;
    *ka -= mag * ksh;
    mag = sqrt(*is * *is + *js * *js + *ks * *ks);
    if (mag != 0.0) {
        *is = *is / mag;
        *js = *js / mag;
        *ks = *ks / mag;
    }
    mag = sqrt(*ia * *ia + *ja * *ja + *ka * *ka);
    if (mag != 0.0) {
        *ia = *ia / mag;
        *ja = *ja / mag;
        *ka = *ka / mag;
    }
}
rotate(turn, x, y)
float turn, *x, *y;
{
    float rad, ang, arctanO;
    double sqrt(), cos(), sin();
    rad = sqrt(*x * *x + *y * *y);
    ang = arctan(*y, *x) + turn;
    *x = cos(ang) * rad;
    *y = sin(ang) * rad;
}
normalize(i, j, k)
float *i, *j, *k;
{
    float mag;
    double sqrtO;
    mag = sqrt(*i * *i + *j * *j + *k * *k);
    if (mag != 0.0) {
        *i = *i / mag;
        *j = *j / mag;
        *k = *k / mag;
    }
}
float mag3(x, y, z)
float x, y, z;
{
    double sqrt();
    return(sqrt(x * x + y * y + z * z));}

float mag2(x, y)
float x, y;
{
    double sqrt();
    return(sqrt(x * x + y * y));}

float arctan(y, x)
float y, x;
{
    float ang;
    double atan(), fabs();
    if (x == 0.0) ang = PI / 2.0;
    else ang = atan(fabs(y / x));
    if (x < 0.0) ang = PI - ang;
    if (y < 0.0) ang = -ang;
    return(ang);}

float arccos(x)
float x;
{
    double acos(), fabs();
    if (fabs(x) > 1.0) x = x / fabs(x);
    return(acos(x));}
Appendix B. The LOGHAQ Code

/***********************/
/* These procedures are ones called by the LOGHAQ heuristics that 
would normally be hardware operators. */

get_foothold(cstride, is, js, ks, xf, yf, zf)
float cstride, is, js, ks, *xf, *yf, *zf;

/* This procedure would normally return a permissible foothold 
location from a sensor system. This location is guided by 
the parameters passed to the sensor system. */
{}

int down_check()
/* This procedure would normally return hardware signals from the 
foot contact sensors that indicate touchdown. */
{}
/***********************************************************************/
/* These procedures are status indicating output drivers used by the LOGHAQ heuristics. */

mess(num, pflo, pint, halt_cause)

int num, pint, halt_cause;
float pflo;
{
    if (messflag[num] == 0) {
        printf("- ");
        switch(num) {
            case 0: printf("About to go off balance, halting from a ");
                printf("speed of %g,\n", pflo);
                break;
            case 1: printf("A leg has reached its limit, halting from a ");
                printf("speed of %g,\n", pflo);
                break;
            case 2: printf("Sudden acceleration occurring from a low speed,\n");
                break;
            case 3: printf("Moving very slowly,\n", pflo);
                break;
            case 4: printf("Swing leg positioning error - using fixed swing,\n");
                break;
            case 5: printf("Unusual event: knee movement is limiting factor,\n");
                break;
            case 6: printf("Leg 1 positioning error: ");
                break;
            case 7: printf("Leg 2 positioning error: ");
                break;
            case 8: printf("Leg 3 positioning error: ");
                break;
            case 9: printf("Leg 4 positioning error: ");
                break;
            case 10: printf("Adjusting attitude to reach foothold,\n");
                break;
            case 11: printf("Rear leg being restricted in forward swing,\n");
                break;
            case 12: printf("New forward foothold extremely close,\n");
                break;
            case 13: printf("New forward foothold beyond maximum stride,\n");
                break;
            case 14: printf("Halting to allow for back to front foot shuffle ");
                printf("from a speed of %g,\n", pflo);
                break;
            case 15: printf("Having difficulty turning at ");
                printf("%g degrees,\n", pflo * 180.0 / PI * 2.0);
                break;
            case 16: printf("Adjusting rotation to reach foothold,\n");
        }
        if (num >= 6 && num <= 9) {
            switch(pint) {
                case 1: printf("negative side distance,\n"); break;
                case 2: printf("extended too far forward,\n"); break;
                case 3: printf("extended too far backward,\n"); break;
                case 4: printf("over-extended,\n"); break;
                case 5: printf("under-extended,\n"); break;
                case 6: printf("extended too far upward,\n");break;
                case 7: printf("extended too far downward,\n");}
    }
Appendix B. The LOGHAQ Code

messfl[num] = 2;
if (halt_cause == TRUE) {
  falt_num = num;
  falt_cas = pint;}

reset_messfl()
{
  int i;
  for (i = 0; i < 20; i++) {
    messfl[i] -= 1;
    if (messfl[i] < 0) messfl[i] = 0;}

describe_failure(lift_leg, lifted)
int lift_leg, lifted;
{
  int found;
  found = FALSE;
  printf("- ");
  if (lifted == FALSE) {
    if ((falt_num == 8 && lift_leg == 3 && falt_cas == 3) ||
        (falt_num == 9 && lift_leg == 4 && falt_cas == 3)) {
      printf("Failure 1: rear leg rotated too far back during creeping.\n"),
      found = TRUE;}
    if ((falt_num == 8 && lift_leg == 3 && falt_cas == 4) ||
        (falt_num == 9 && lift_leg == 4 && falt_cas == 4)) {
      printf("Failure 2: rear leg over extended during creeping.\n"),
      found = TRUE;}
  }
  if (lifted == TRUE & (lift_leg == 1 || lift_leg == 2) &
      (falt_num == 0 || falt_num == 14)) {
    printf("Failure 3: end of stability triangle reached and ");
    printf("forward placement failed.\n"),
    found = TRUE;}
  if (found == FALSE) {
    printf("Unusual failure: ");
    if (lifted == TRUE) printf("leg %d placement failed, and ", lift_leg);
    else printf("creeping to swing leg %d, and ", lift_leg);
    if (falt_num == 0 || falt_num == 14)
      printf("reached end of stability region.\n"),
    else {
      if ((falt_num - 5) == lift_leg) printf("it ");
      else printf("leg %d ", falt_num - 5);
switch(falt_cas) {
    case 1: printf("had a negative side value.\n"); break;
    case 2: printf("extended too far forward.\n"); break;
    case 3: printf("extended too far backward.\n"); break;
    case 4: printf("over-extended.\n"); break;
    case 5: printf("under-extended.\n"); break;
    case 6: printf("extended too far upward.\n"); break;
    case 7: printf("extended too far downward.\n");
}
Appendix B. The LOGHAQ Code

/** These procedures were used to set the initial conditions during the simulations of the LOGHAQ heuristics. */

init_parameters()
{
  double tan();
  /* changed */
  extern float wide, strid, height, setlm, setlh, setlr, setlt;
  extern float tradeg, degrees;
  wide = 1.3 * UPLL + WIDTH;
  strid = (wide - WIDTH) * tan(MAXFA) * 0.6;
  height = LOLL * 2.0 / 3.0;
  setlm = setlh = setlr = 0.6;
  setlt = 0.1;
  tradeg = degrees = 0.0;}

init_machine(speed, lifted, halted, trydown, lift_leg)

float *speed;
int *lifted, *halted, *trydown, *lift_leg;
{
  float ja, ka, is, js, ks, iu, ju, ku, df, ds, du, ih, jh, kh, arctan();
  int fail, i;
  double cos(), sin();
  /* changed */
  extern float x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4;
  extern float af1, af2, af3, af4, au1, au2, au3, au4, ao1, ao2, ao3, ao4;
  extern float typ1, typ2, audrec, tilt, roll;
  *speed = 0.0;
  *halted = FALSE;
  *lifted = *trydown = TRUE;
  *lift_leg = 1;
  typ1 = typ2 = arctan(1.5 * strid - BASE, wide - WIDTH);
  audrec = 0.0;
  tilt = roll = 0.0;
  ja = cos(tilt);
  ka = sin(tilt);
  is = cos(roll);
  iu = sin(roll);
  js = iu * ka;
  ks = -iu * ja;
  ju = -is * ka;
  ku = is * ja;
Appendix B. The LOGHAQ Code

\begin{verbatim}
xl = -wide * is;
yl = -wide * js + 1.5 * strid * ja;
zl = -wide * ks + 1.5 * strid * ka - height;
x2 = wide * is;
y2 = wide * js + 0.5 * strid * ja;
z2 = wide * ks + 0.5 * strid * ka - height;
x3 = -wide * is;
y3 = -wide * js - 0.5 * strid * ja;
z3 = -wide * ks - 0.5 * strid * ka -
x4 = wide * is;
y4 = wide * js - 1.5 * strid * ja;
z4 = wide * ks - 1.5 * strid * ka - height;

ih = xl + WIDTH * is;
jh = yl - (BASE * ja - WIDTH * js);
k = zl - (BASE * ka - WIDTH * ks);
df = jh * ja + kh * ka;
ds = -ih * is - jh * js - kh * ks;
du = ih * iu + jh * ju + kh * ku;
get_angles(df, ds, du, ftall, &aul, &aol, ftfail, 1, FALSE);

ih = x2 - WIDTH * is;
jh = y2 - (BASE * ja + WIDTH * js);
k = z2 - (BASE * ka + WIDTH * ks);
df = jh * ja + kh * ka;
ds = ih * is + jh * js + kh * ks;
du = ih * iu + jh * ju + kh * ku;
get_angles(df, ds, du, &af2, &au2, fcao2, &fail, 2, FALSE);

ih = x3 + WIDTH * is;
jh = y3 - (-BASE * ja - WIDTH * js);
k = z3 - (-BASE * ka - WIDTH * ks);
df = jh * ja + kh * ka;
ds = -ih * is - jh * js - kh * ks;
du = ih * iu + jh * ju + kh * ku;
get_angles(df, ds, du, &af3, &au3, &ao3, fcfail, 3, FALSE);

ih = x4 - WIDTH * is;
jh = y4 - (-BASE * ja + WIDTH * js);
k = z4 - (-BASE * ka + WIDTH * ks);
df = jh * ja + kh * ka;
ds = ih * is + jh * js + kh * ks;
du = ih * iu + jh * ju + kh * ku;
get_angles(df, ds, du, &af4, &au4, &ao4, &fail, 4, FALSE);

falt_num = falt_cas = 0;
for (i = 0; i < 20; i++) messfl[i] = 0;
\end{verbatim}
References


References


References


