

**STABLE RUNNING FOR A QUADRUPED ROBOT  
WITH COMPLIANT LEGS**

**by**

**Didier Papadopoulos**

**Department of Mechanical Engineering  
McGill University**

**April 2000**

**A Thesis submitted to the Faculty of Graduate Studies and Research  
in partial fulfilment of the requirements for the degree of  
Master of Engineering**

**© Didier Papadopoulos, 2000**



**National Library  
of Canada**  
**Acquisitions and  
Bibliographic Services**  
385 Wellington Street  
Ottawa ON K1A 0N4  
Canada

**Bibliothèque nationale  
du Canada**  
**Acquisitions et  
services bibliographiques**  
385, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*

*Our file Notre référence*

**The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.**

**The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.**

**L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.**

**L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.**

**0-612-64241-0**

**Canadä**

**To my family,  
and Sandra**

# Abstract

Robots with four legs offer a good tradeoff between stability, load carrying capacity, and mechanical complexity when compared to bipeds and hexapods. In order to achieve the best mobility, speed, and energy efficiency, dynamic walking and running operation is preferable to static gaits. This requires leg compliance to reduce impact forces and energy consumption.

At the Ambulatory Robotics Laboratory, (ARL), we have pursued an agenda of low mechanical complexity in our Scout I and II robots, in order to decrease cost and increase reliability. Research previously undertaken at the ARL group has accomplished walking and stair climbing with Scout I and walking with stiff legs with Scout II. In this thesis, we demonstrate that Scout II, with only an additional compliant prismatic joint per leg, is able to bound. We show that dynamic running is possible with very simple control strategies. Open loop control, where switching torque values at the hip during support or flight phases results in a stable bounding gait. We also investigate more elaborate controllers that control forward speeds. The bound controllers were first developed and validated in simulation. These strategies were then implemented on the Scout II robot yielding successful running at speeds of up to 1.2 m/s.

# Résumé

**Les robots quadrupèdes offrent un bon compromis entre la stabilité, la capacité à transporter des charges et la complexité mécanique vis-à-vis les bipèdes et hexapodes.** Dans le but d'atteindre la meilleure mobilité, vitesse et performance énergétique, la marche ainsi que la course dynamique sont préférables aux démarches statiques. Ceci nécessite au robot des jambes flexibles afin de réduire les forces à l'impact ainsi que la consommation énergétique.

A L'ARL (Ambulatory Robotics Laboratory), nous avons recherché une conception mécanique simple dans la réalisation de nos robots Scout I et Scout II, dans le but de réduire leur coût et d'augmenter leur fiabilité. Des travaux antécédents ont porté sur la réalisation du processus de marche sur une surface plane et de l'ascension d'escaliers avec Scout I, ainsi que la marche avec Scout II. Le sujet de cette thèse est de démontrer que Scout II, avec seulement un degré de mobilité additionnel jambe, sera capable de courir en bondissant. Nous montrons que courir dynamiquement est possible tout en utilisant des démarches simples pour le contrôle du robot. Rien qu'en alternant le couple moteur pendant la phase de support ou la phase aérienne du cycle, on accomplit une démarche stable. Dans un temps ultérieur, nous étudions une stratégie plus élaborée afin de contrôler la vitesse de croisière du robot. Un modèle semblable au robot fut développé et simulé sur logiciel dans le but de vérifier les différentes stratégies de bond. Ces derniers sont plus tard implantés avec succès sur Scout II donnant une démarche en bond avec une vitesse avoisinant 1.2 m/s.

# Acknowledgements

Lots of people have helped during my stay at CIM. I would like to thank them for their support during my research:

- First and foremost, I would like to thank Professor Martin Buehler for his guidance and for being there whenever needed. He was the one who first introduced me to legged locomotion research and then helped me understand it in more detail while pursuing my master's research.
- Sami Obaid introduced me to CIM, and specifically the Ambulatory Robotics Lab. He developed the laser and gyro sensing on Scout II. Sami later became one of my closest friends. Thanks a lot Sami for the support and the long hardware debugging sessions on Scout II.
- Geoff Hawker was always there whenever I needed someone to consult. He was very helpful in discussing with me the different aspects of my work. Geoff maintained the ARL PC's and setup Scout II's PC.
- Thanks to Liana Mitrea for her continuous moral support as well as the drinks and cookies she used to bring to the lab.
- Thanks to Martin de Lasa for the tremendous help in the initial experimental phases of the running.
- Thanks a lot to Shervin Talebinejad for assisting me with the Scout II modeling and analysis as well as for his Scout II 3D model.
- Other people like Don Campbell and Dave McMordie helped a lot in debugging the hardware problems on Scout II.
- Robert Battaglia built Scout II, without his work none of my research would have been possible.
- I would also like to thank the CIM administrative staff, Marlene Gray, Kathleen VanderNoot, and Ornella Cavaliere, for ensuring that things ran smoothly.
- The greatest help, support and understanding came from my fiancée Sandra Bernotti. Her moral contribution to this research is inestimable.

# Table of Contents

<b>CHAPTER 1.....</b>	<b>1</b>
1.1 MOTIVATION.....	1
1.2 BACKGROUND.....	3
1.3 PREVIOUS WORK AT ARL.....	5
1.4 SCOUT II .....	8
1.5 AUTHOR'S CONTRIBUTION.....	10
1.6 THESIS OVERVIEW.....	10
<b>CHAPTER 2.....</b>	<b>11</b>
2.1 INTRODUCTION.....	11
2.2 NOTATIONS AND ASSUMPTIONS .....	14
2.3 DERIVATION OF THE EQUATIONS OF MOTION .....	15
2.3.1 <i>Back Stance Phase</i> .....	16
2.3.2 <i>Flight Phase</i> .....	21
2.4 WORKING MODEL 2D® SIMULATION SOFTWARE.....	25
2.4.1 <i>Working Model 2D® Script</i> .....	26
2.4.2 <i>Robot Model</i> .....	27
2.4.3 <i>Robot Constraints</i> .....	30
2.5 RUNNING ALGORITHMS .....	42
2.5.1 <i>Open Loop Controller</i> .....	42

2.5.2	<i>Closed Loop Controller with Velocity Feedback</i> .....	46
2.5.3	<i>Control of Forward Speed by Touchdown Angle</i> .....	54
2.5.4	<i>Torque Velocity Feedback Controller</i> .....	56
2.5.5	<i>Jumping Over Obstacles</i> .....	58
2.5.6	<i>Closed Loop Controller with Pitch Control</i> .....	60
2.5.7	<i>Turning</i> .....	63
2.6	CONCLUSION .....	65
<b>CHAPTER 3.....</b>		<b>66</b>
3.1	INTRODUCTION.....	66
3.2	RUNNING SETUP .....	66
3.3	RUNNING IMPLEMENTATIONS.....	69
3.3.1	<i>Bounding Gait</i> .....	69
3.3.2	<i>Pronking Gait</i> .....	75
3.3.3	<i>Closed loop Controller with Velocity Feedback</i> .....	78
3.4	TURNING .....	79
3.5	OTHER BEHAVIORS.....	80
3.6	CONCLUSION .....	84
<b>CHAPTER 4 .....</b>		<b>85</b>
4.1	SUMMARY .....	85
4.2	FUTURE RECOMMENDATIONS.....	86
<b>BIBLIOGRAPHY .....</b>		<b>88</b>

**APPENDIX.....92**

**A.1 BACK LEGS SUPPORT PHASE.....92**

**A.2 FRONT LEGS SUPPORT PHASE .....95**

# List of Figures

FIGURE 1-1: ROBOTRAC BY IFR .....	2
FIGURE 1-2: URBAN ROBOT BY IS ROBOTICS.....	2
FIGURE 1-3: ANDROS ROBOT BY REMOTEC .....	2
FIGURE 1-4: PATRUSH QUADRUPED ROBOT BY KIMURA ET AL.....	4
FIGURE 1-5: SCAMPER QUADRUPED ROBOT BY FURUSHO ET AL.....	4
FIGURE 1-6: CARL .....	6
FIGURE 1-7: ARL MONOPOD I.....	7
FIGURE 1-8: ARL MONOPOD II. ....	7
FIGURE 1-9: ARL SCOUT I. ....	8
FIGURE 1-10: SCOUT II WITH THE TWO LEG CONFIGURATIONS .....	9
FIGURE 2-1: PHASE RELATIONSHIPS FOR DIFFERENT RUNNING GAITS.....	13
FIGURE 2-2: THE FOUR RUNNING STATES. ....	13
FIGURE 2-3: DRAWING OF SCOUT II.....	14
FIGURE 2-4: BACK STANCE LEG LENGTH COMPARISON. ....	19
FIGURE 2-5: BACK STANCE BODY PITCH COMPARISON.....	20
FIGURE 2-6: BACK STANCE LEG ANGLE COMPARISON. ....	20
FIGURE 2-7: FLIGHT AFTER BACK BODY PITCH COMPARISON.....	22
FIGURE 2-8: FLIGHT AFTER BACK HORIZONTAL POSITION COMPARISON.....	23
FIGURE 2-9: FLIGHT AFTER BACK HEIGHT COMPARISON.....	23
FIGURE 2-10: SCOUT II LEG DESIGN.....	28

FIGURE 2-11: SCOUT II MODEL BUILT WITH WORKING MODEL 2D®.....	29
FIGURE 2-12: MOTOR TORQUE SPEED MODEL.....	32
FIGURE 2-13 EXPERIMENTAL TORQUE-SPEED CURVE.....	33
FIGURE 2-14: FOOT FORCES UPON CONTACT WITH THE GROUND. ....	35
FIGURE 2-15: FREE BODY DIAGRAM FOR THE TOE CONTACT FORCES. ....	36
FIGURE 2-16: SIMULATION RESULTS FOR THE THREE SLIP FUNCTIONS. ....	41
FIGURE 2-17: OPEN LOOP CONTROLLER (LEG): BODY PITCH AND FORWARD VELOCITY. ....	45
FIGURE 2-18: OPEN LOOP CONTROLLER (HIP): BODY PITCH AND FORWARD VELOCITY. ....	46
FIGURE 2-19: LEG TRAJECTORY DURING THE STANCE PHASE. ....	48
FIGURE 2-20: : CLOSED LOOP CONTROLLER WITHOUT SLIP: BODY PITCH AND FORWARD SPEED.....	50
FIGURE 2-21: CLOSED LOOP CONTROLLER WITH SLIP: BODY PITCH AND FORWARD SPEED	51
FIGURE 2-22: CLOSED LOOP CONTROLLER WITH SLIP PREVENTION: BODY PITCHING AND BODY FORWARD SPEED .....	52
FIGURE 2-23: TORQUE PLOT DURING THE BACK STANCE PHASE.....	53
FIGURE 2-24: COMPARISON OF LEG TRACKING WITH AND WITHOUT MOTOR MODEL.....	53
FIGURE 2-25: COMPARISON OF TORQUE-SPEED CURVE OPERATION FOR THE FOUR QUADRANTS WITH AND WITHOUT THE MOTOR MODEL. ....	54
FIGURE 2-26: BODY PITCH AND ROBOT FORWARD VELOCITY VALUES FOR CHANGES IN DESIRED TOUCHDOWN ANGLES.....	55
FIGURE 2-27: STEP CHANGES IN FORWARD VELOCITIES CONTROLLED BY THE HIP ACTUATOR TORQUE. ....	57

<b>FIGURE 2-28: TORQUE LIMITATIONS ON THE MOTOR DURING THE RUNNING CYCLE .....</b>	<b>58</b>
<b>FIGURE 2-29: OBSTACLE CROSSING: BODY PITCH, FORWARD SPEED AND HOPPING HEIGHT</b>	<b>59</b>
<b>FIGURE 2-30: VELOCITY AND PITCH CONTROL: BODY PITCH AND FORWARD SPEED. ....</b>	<b>62</b>
<b>FIGURE 2-31: FLOW CHART OF THE RUNNING CYCLE .....</b>	<b>62</b>
<b>FIGURE 2-32: BODY YAW DURING THE TURNING ALGORITHMS.....</b>	<b>64</b>
<b>FIGURE 2-33: TOP VIEW OF THE TURNING IN 3D SIMULATION .....</b>	<b>64</b>
<b>FIGURE 3-1: SCOUT II.....</b>	<b>67</b>
<b>FIGURE 3-2: EXPERIMENTAL OPEN LOOP RUNNING CONTROL CYCLE .....</b>	<b>70</b>
<b>FIGURE 3-3: BACK LEGS AND FRONT LEGS STATES FOR THE OPEN LOOP CONTROL.....</b>	<b>70</b>
<b>FIGURE 3-4: COMPARISON BETWEEN FORWARD RUNNING SPEEDS IN SIMULATION AND EXPERIMENTATION. ....</b>	<b>71</b>
<b>FIGURE 3-5: COMPARISON BETWEEN BODY PITCHING IN SIMULATION AND EXPERIMENTATION. ....</b>	<b>72</b>
<b>FIGURE 3-6: COMPARISON BETWEEN LEG ANGULAR DISPLACEMENT IN SIMULATION AND EXPERIMENTATION. ....</b>	<b>73</b>
<b>FIGURE 3-7: BACK LEGS TORQUE FOR SIMULATION AND EXPERIMENTATION.....</b>	<b>74</b>
<b>FIGURE 3-8: CLOSE UP OF TORQUE CHANGE DURING ONE CYCLE .....</b>	<b>75</b>
<b>FIGURE 3-9: BACK AND FRONT LEGS STATES FOR THE PRONKING. ....</b>	<b>76</b>
<b>FIGURE 3-10: BODY PITCHING THROUGHOUT THE PRONKING GAIT. ....</b>	<b>77</b>
<b>FIGURE 3-11: LEG ANGULAR CHANGES FOR THE PRONKING GAIT. ....</b>	<b>77</b>
<b>FIGURE 3-12: BODY PITCHING AND FILTERED FORWARD VELOCITY DURING THE RUNNING EXPERIMENT WITH THE CLOSED LOOP CONTROLLER. ....</b>	<b>78</b>

FIGURE 3-13: NINETY DEGREES TURNING EXPERIMENT .....	79
FIGURE 3-14: STANDING UP SEQUENCE.....	81
FIGURE 3-15: SEQUENCE OF EVENTS FOR SITTING DOWN. ....	82
FIGURE 3-16: SEQUENCE OF EVENTS FOR SITTING DOWN. ....	83
FIGURE A-0-1: FRONT STANCE LEG LENGTH COMPARISON.....	100
FIGURE A-0-2: FRONT STANCE BODY PITCHING COMPARISON.....	100
FIGURE A-0-3: FRONT STANCE LEG ANGLE COMPARISON.....	101

## List of Tables

TABLE 2.1: NOTATION FOR SCOUT II STATES .....	15
TABLE 2.2: NOTATION FOR SCOUT II PARAMETERS .....	15
TABLE 2.3: SCOUT II MODEL PARAMETERS .....	29
TABLE 2.4: HIP ACTUATOR SPECIFICATIONS .....	31
TABLE 2.5: ROBOT OVERALL RUNNING STATE FOR THE CLOSED LOOP RUNNING CONTROLLER .....	61

# **Chapter 1**

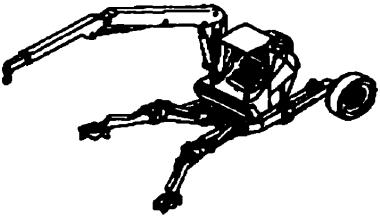
---

## **Introduction**

### **1.1 Motivation**

Most creatures on earth use legs for locomotion on solid ground. Legs provide a unique tradeoff between efficient locomotion on level ground, and the ability to traverse uneven or difficult terrain. Other advantages of legged locomotion are numerous. Legged robots can travel with minor ground-robot contact as compared to wheeled or tracked vehicles, which require a continuous path of support. This is a major issue in the case of plantations, for example, where crop damage must be minimized. On the other hand, tracked vehicles can inflict serious damage to the supporting surface. On flat terrain, wheeled locomotion is faster and more efficient than legged locomotion but fails to function adequately in areas where the terrain is uneven. Legged locomotion has the advantage of reaching places that wheeled robots cannot. In order to overcome some of the limitations of simple wheeled or tracked vehicles, a number of hybrid vehicles, combining legs and wheels [15][31][36] (see Figure 1-1) and articulated tracked vehicles [18][33] have been developed (see Figure 1-2, Figure 1-3). Such devices achieve greatly improved mobility and are increasingly moving into applications in the areas of bomb disposal, construction, excavation and forestry in rough terrain, military tasks, and others.

## Chapter 1. Introduction



**Figure 1-1: Robotrac by IFR [17].**



**Figure 1-2: Urban Robot by IS Robotics.**



**Figure 1-3: ANDROS robot by REMOTEC.**

While hybrid and articulated tracked vehicles can be ideal solutions when engineered for particular urban or outdoor environments, it is unlikely that they can compete with the mobility and versatility of multi-legged platforms in unstructured settings, on badly fractured terrain, or when a large variety of terrain needs to be handled. For legged robots to achieve practical utility, they must become faster, more robust, more efficient, more autonomous and less expensive than contemporary prototypes. On that account dynamically stable machines are the best alternative. Static machines, which are statically stable throughout their entire motion, are constrained by limited achievable speeds and must have a high number of legs and actuators, rendering them expensive and complex to control. Statically stable robot must have at least four legs to maintain static stability, but typically have six or eight to also provide sufficient mobility over rough terrain. Dynamically stable robots, on the other hand, can operate with fewer legs, even only one [1][2][32]. This not only makes the design simpler but also permits higher speeds and a wider number of behaviors. To better understand these complex behaviors, the first step is to study the control of simple robotic platforms. This has been achieved to some extent in the Ambulatory Robotics Laboratory where a class of robot with a simple design combined with a very wide range of behaviors has been introduced. The Scout class of robots are quadrupeds with only one actuated degree of freedom per leg. Scout I and Scout II have already demonstrated steady state walking behaviors. Initial investigations into passive running had been performed, but had not yielded successful implementation. Passive dynamics is defined as the unforced response of a dynamic system. Passive motion has the advantage of being smoother, more natural and more efficient than non-

passive ones. For any mobile robot, it is imperative to be power autonomous, and therefore energy efficiency is of prime importance. Efficient robots will have a higher autonomy and a lower energy supply load to be carried on board [2], permitting some new applications, such as high speed chasing situations, hazardous disposal or surveillance of such dangerous environments as nuclear power plants or volcanoes.

## 1.2 Background

In recent years, a great deal of research has been conducted in the area of dynamically stable robots. The first dynamically stable robot was the BIPER, which was built by Miura and Shimoyama [28]. This was a biped robot controlled by three motors that could achieve dynamically stable walking. The research on dynamically stable robots expanded with Raibert [32] when he began working on simple controllers for his pneumatic monopod. He introduced simple running control concepts such as the three part running controller - one each for hopping height, forward speed and body pitch - that were simple yet effective. He later expanded his work to bipeds and quadrupeds. Later Schwind et al. [35] studied Raibert's planar hopper in order to analytically verify the stability of the forward velocity controller. They then suggested another forward velocity controller that used coupled feedback that takes into account the dynamic structure of the robot. Although this new velocity controller gives better regulation than Raibert's decoupled feedback controller, Schwind and Koditschek suggest that the price - both sensing and computation are dramatically increased – for more complicated controllers might not be worthwhile.

Further research into running investigated the advantages of passive elements in the robots. McGeer [25] has studied the concept of passive walking and running with biped robots. He investigated passive-dynamic running with legged robots having passive hip actuators as well as linear springs along the leg length. Through analytical work and simulation, McGeer showed that running can be achieved with no forcing required to generate the gait.

## Chapter 1. Introduction

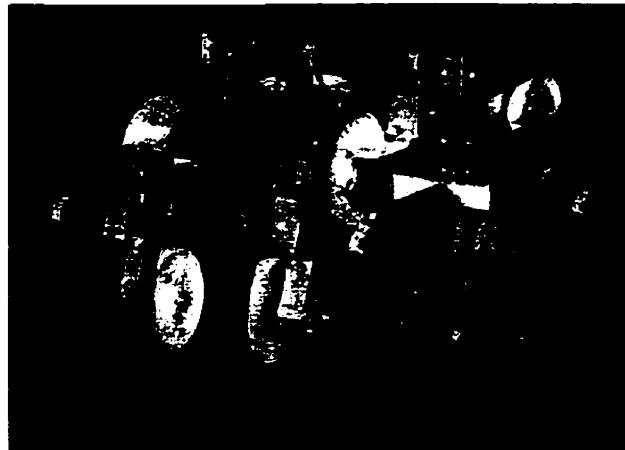
More work on legged robots with actuated and unactuated degrees of freedom such as Scout II includes investigation into ankle compliance. Control strategies for such a mechanism have been attempted by Keon [19]. He proposed a controller for a biped with a two degrees of freedom actuated hip and a two degrees of freedom compliance in the ankle. His simulation results proved that such a mechanism works effectively.

Kimura et al [20] introduced a quadruped running robot that has actuated hip and knee joints as well as a passive spring mechanism for each leg (see Figure 1-4). The running controller is based on a neural oscillator network, a stretch reflex and a flexor reflex mechanism. Dynamic running and walking on flat terrain was successfully achieved with this approach.

Furusho et al. [11] implemented a bounding gait on the SCAMPER robot. Even though the robot was not designed with explicit mechanical compliance, the compliance of the feet, legs, and the belt transmission, and the effective compliance of the PD joint servo loops are likely significant. The controller divided one complete running cycle into eight states and switched the two joints per leg between free rotation, position control and velocity control.



**Figure 1-4: PATRUSH quadruped robot by Kimura et al.**



**Figure 1-5: SCAMPER quadruped robot by Furusho et al.**

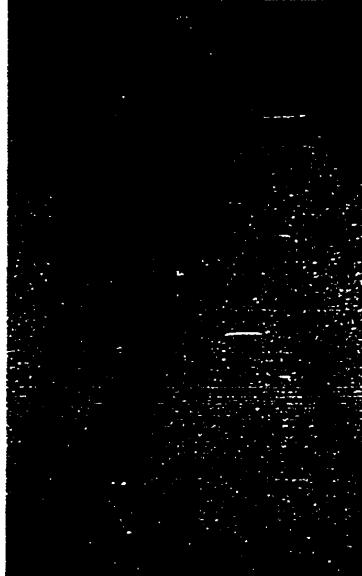
Berkemeier [5] performed an analytical study on a simplified quadrupedal running robot. Although expressions for exact maps could not be obtained, approximate maps for bounding and pronking were derived and used to predict the behavior of the different running parameters. The simple model predicted that, for a given set of parameters, the bound offers greater acceleration since its period is shorter, and so the legs contact the ground more frequently. This also allows a greater maneuverability since it gives more opportunities to steer around obstacles. The bounding gait is the main type of running that is addressed in this research.

### 1.3 Previous Work at ARL

Work at the Ambulatory Robotics Laboratory involves the study and the implementation of legged robotic systems. This includes designing and building legged robots with minimal complexity and cost. Our concern is to minimize the complexity while maintaining high performance in terms of locomotion speeds, agility, and reduced energy consumption.

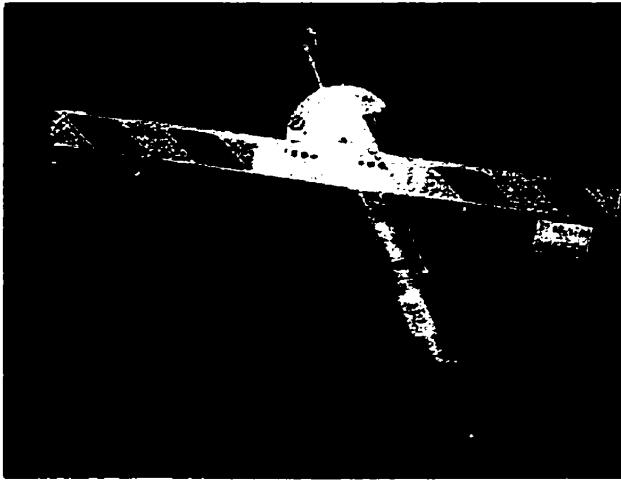
Initial work done under this mandate was the design and construction of CARL (Compliant Articulated Robot Leg). This leg design was aimed at achieving high mobility and simplicity, reduced friction loss and a substantial weight saving via the use of a novel transmission design called ATLAS (AnTagonistic LADD Actuation System, LADD stands for Linear to Angular Displacement Device) [26][27].

## Chapter 1. Introduction

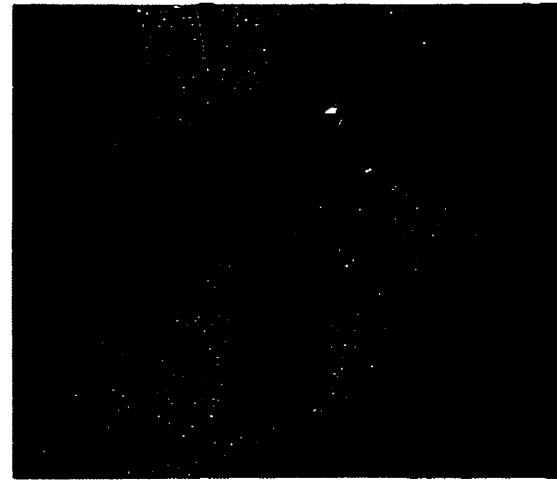


**Figure 1-6: CARL, Compliant Articulated Robot Leg with Antagonistic LADD Actuation.**

Another generation of robots investigated at ARL were the Monopod I and II. Driven by the conviction that simple robots could achieve a high degree of mobility with minimum complexity, the Monopod I is a one legged robot with a rotary degree actuation at the hip and a prismatic passive degree of freedom in the leg [13]. The Monopod I, with a running speed of 1.2 m/s and 125 W average mechanical power consumption, was the fastest and most efficient electrically actuated legged robot of its time. With an additional passive hip oscillation, the Monopod II was able to surpass its predecessor by achieving a running speed up to 1.25 m/s with a total mechanical power expenditure of only 48 W. This had the highest efficiency among all actively controlled legged robots [2].

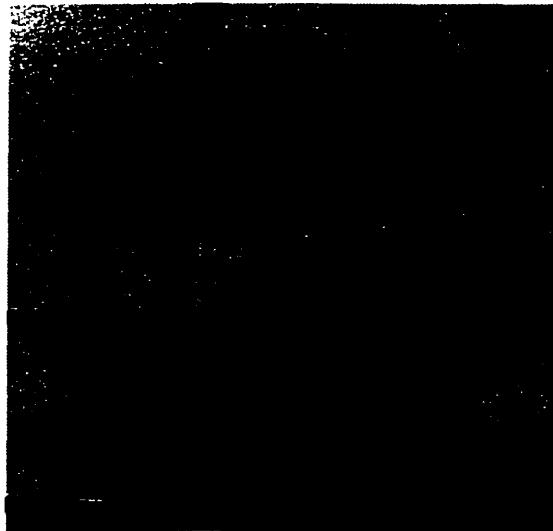


**Figure 1-7: ARL Monopod I.**



**Figure 1-8: ARL Monopod II.**

The results obtained in previous work on the Monopod I and II prompted the need to develop quadruped robots that could use similar principles of locomotion in order to move towards more practical applications in a 3D world. The first of the Scout class of robots was Scout I. Scout I is a quadruped robot with only one degree of freedom per leg located at the hip. With minimal sensing, Scout I is capable of achieving a stable open loop walking gait. It also achieves step climbing on steps with heights up to 45 % of its leg length. Other additional behaviors such as turning, side stepping and sitting down have also been successfully implemented [7][8][38]. Given the range of achievable behaviors on Scout I, it was decided to develop a larger prototype, Scout II, that could accomplish the same range of behaviors, but at higher speeds and efficiencies because of additional passive elements.



**Figure 1-9: ARL Scout I.**

## 1.4 Scout II

Current research in ARL (Ambulatory Robotics Lab) focuses on controller development for the Scout II robot. Built at ARL [4], Scout II is a quadruped robot with two degrees of freedom per leg. One degree of freedom is situated at each hip level and is controlled via a 90 Watt DC motor. Depending on its configuration, Scout II can be equipped with a telescoping leg where the second degree of freedom is a prismatic unactuated joint attached to a spring damper mechanism. In another configuration, the second degree of freedom is an unactuated rotary knee joint, attached to each leg [14]. Figure 1-10 shows Scout II with both leg configurations.



**Figure 1-10: Scout II with the two leg configurations. The picture on the left has the legs equipped with a prismatic joint. The picture on the right has Scout II with the legs equipped with a rotary joint [14].**

The legs equipped with knees enable Scout II of accomplishing walking behaviors with trotting gaits. In its current configuration, Scout II weighs 27 kg. It is 35 cm high, 60 cm long and 45 cm wide. It has been designed to be simple yet complex enough to achieve a multitude of tasks such as walking, stair climbing and running. Scout II is completely autonomous with no external power source or communication lines attached to it. It is equipped with an on-board PC for all necessary computations, and a set of two 12 V batteries. Tele-operation is possible with a wireless link. Current work on Scout II is focused on developing new controllers for a wide variety of behaviors as well as incorporating a new line of sensors to increase Scout II's navigational capability.

This thesis presents control algorithms designed for a bound running gait. Analysis and simulation investigate possible running controllers which were later implemented on Scout II. Although running simulations are first done in the sagittal plane, 3D simulations are later used to test turning controllers. At the end of this study Scout II is able to run and turn as well as perform some other interesting behaviors such as standing up or sitting down.

### 1.5 Author's Contribution

- The equations of motion of the system have been derived for the different running phases.
- The equations of motion of running have been validated using Working Model 2D®.
- Running simulations have been developed and tested on Working Model 2D ®.
- Turning simulations have been developed and tested on Working Model 3D ®.
- Stable bounding and pronking gaits have been implemented on Scout II.
- Turning and other complementary behaviors have been implemented on Scout II.

### 1.6 Thesis Overview

In Chapter 2 the equations of motion of Scout II during running are derived. Then they are integrated for the different running phases. Data plots obtained for the robot body pitch, body pitch speed, leg angles, leg angular speeds, leg length and leg length speed are shown. Obtained values are compared with the data from Working Model 2D ® [21] simulations in order to verify the Scout II mathematical model. The Working Model 2D® program is then presented and the organization of the running simulations is explained. The chapter also explains the motor model, as well as, the toe-ground slip prevention model used in the modeling of the robot. Finally the running controllers and the simulation results are presented starting from the simplest controller to the more elaborate one.

In Chapter 3 the different controllers studied are integrated into the Scout II software. Results and analysis obtained from the running experiments are presented. Other complementary behaviors are also studied.

Chapter 4 summarizes major conclusions and finding, and outlines a set of recommendations for future work on Scout II.

# **Chapter 2**

---

## **Analysis and Simulations**

### **2.1 Introduction**

The aim of this chapter is to derive the equations of motion for Scout II and compare a numerical integration of these equations to the results from a commercial simulation package, Working Model 2D®. This will allow the cross validation of the simulation model and the mathematical model. This comparison will give insight into how well the mathematical model can be used to study the Scout II robot and develop controllers. The equations of motion are a good means of formulating the problem, and conveying it to the other interested research parties. Furthermore they can be later on used for computational controller development using simulated annealing or genetic algorithms. The running controllers can then be tested and adjusted using simulations.

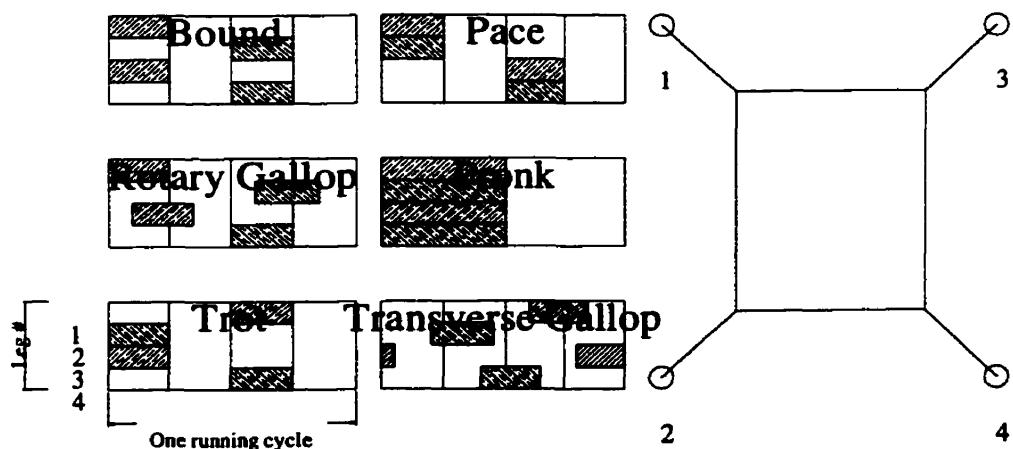
First, Section 2.2 of this chapter presents the nomenclature and assumptions used to describe the Scout II model. In, section 2.3 the equations of motion governing the different phases of the running cycle are derived. In order to justify the assumptions made in the derivation of the equations of motion, these equations are then compared to a more complete simulation model. Section 2.4 introduces the Working Model 2D® package and describes its applications in this thesis. The robot model used in simulation is discussed, with emphasis on modeling constraints. Section 2.5 presents the different

## Chapter 2. Analysis and Simulations

controllers simulated for running, jumping, and turning in addition to their results.

Finally, section 2.6 summarizes the outcomes from the simulations.

Running is defined by a sequence of dynamically stable events whereby the robot alternates between stance and flight phases. Stance occurs when any of the robot's legs are in contact with the ground, while flight occurs when all of the legs are in the air. In order for the robot to get around from one location to another, repetitive cycles of stance and flight states must take place. To achieve running, a quadruped can use a number of leg sequences. Each set of sequences is called a gait. A set of running gaits observed most frequently in nature is presented in Figure 2-1. Two types of running gaits will be shown in this research; the pronk and the bound gaits. In pronking, all legs move together during the stance phase, leaving no phase difference between them. When compared to bounding, the pronking gait has greater ground clearance and lower speeds [5]. Animals such as deer use the pronk gait. In the bounding gait, the front legs move in unison, and so do the hind legs. There is however a phase shift of about 180 degrees between the hind legs and the front legs motions. According to Berkemeier, the bounding gait "... has the shortest gait period, thus allowing for more frequent interactions of the legs with the ground, to avoid obstacles and provide acceleration" [5]. Certain animals such as mice use the bounding type of gaits.

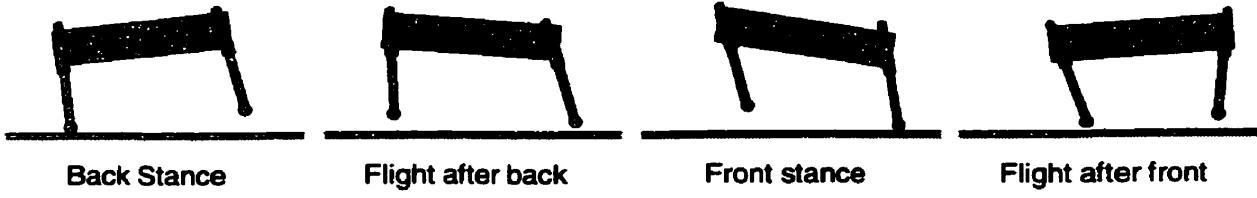


## Chapter 2. Analysis and Simulations

**Figure 2-1: Phase relationships for different running gaits throughout a complete running cycle [3][5][29]. A shaded area represents a leg that is on the ground. A blank area represents a leg being in the air. The figure on the left shows the leg numbering convention used in this research.**

Although the pronking gait is addressed in Chapter 3, the running gait that will be mostly considered is the bound. One complete cycle of bounding can be divided into four states, as shown in Figure 2-2. The state of the robot in each of the four events is as follows:

- ***Back stance:*** This is the configuration where the robot is supported on the ground by its back legs.
- ***Flight after back:*** This is the configuration where none of the robot legs are touching the ground and the robot has just left the ground after a back stance.
- ***Front stance:*** This is the configuration where the robot is supported on the ground by its front legs.
- ***Flight after front:*** This is the configuration where none of the robot legs are touching the ground and the robot has just left the ground after a front stance.

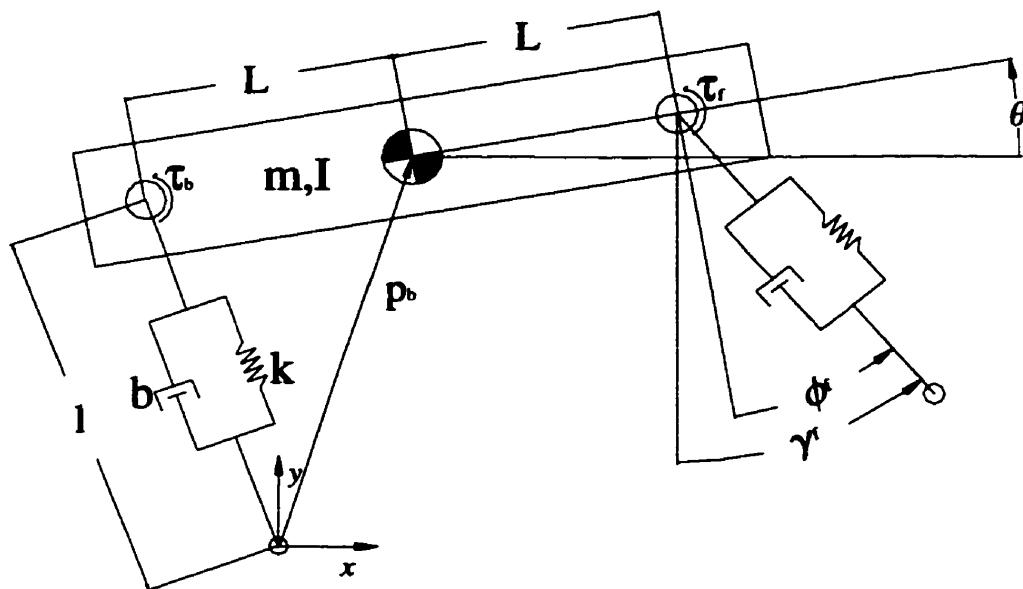


**Figure 2-2: The four running states. A bounding gate Requires a sequence of running states from left to right.**

There are also two transitions used to switch from one phase to the other; the *touchdown* transition happens at the instant when the robot switches from a flight phase to a stance phase, and the *lift-off* transition takes place when the robot is switching from a stance phase to a flight phase. Depending on which controller is used, different tasks are given to each pair of legs in any one of the four states.

## 2.2 Notations and Assumptions

The robot model used in the derivation of the equations of motion in section 2.3 is illustrated in Figure 2-3. Table 2.1 lists the different symbols used to describe various states of Scout II. The notation used for the robot dimensions and physical properties is shown in Table 2.2. The Scout model was derived with the assumption of massless legs.



**Figure 2-3:** Drawing of Scout II as considered in the derivation of the equations of motion.

Symbol	Description
$X_b$	X value of the back leg
$X_f$	X value of the front leg
$X_{lo}$	Value at lift-off
$X_{td}$	Value at touchdown
$X_{fl}$	Value during the flight phase

$X_s$	Value during the stance phase
-------	-------------------------------

Table 2.1: Notation for Scout II states

Symbol	Description
$l_0$	Free leg length (zero spring force)
$L$	Half the distance between the hip joints
$m$	Body mass
$I$	Body moment of inertia about the center of mass ( $I=mr^2$ )
$r$	Body radius of gyration
$k$	Spring stiffness
$b$	Damping coefficient
$\theta$	Body angle w.r.t horizontal
$l_i$	Leg length
$\phi_i$	Hip angle
$\gamma_i$	Leg angle
$x_i$	Body Cartesian coordinates from the supporting toe (i)
$\tau_i$	Hip Torque
$T$	Flight time

Table 2.2: Notation for Scout II parameters

## 2.3 Derivation of the Equations of Motion

The purpose of this section is to find the equations of motion that describe the behavior of the system in any of the running phases and assess the results obtained by

## Chapter 2. Analysis and Simulations

comparison to a simulation software package. If this model is valid, further analysis into running can thus be achieved using the derived equations of motion. First the equations of motion for the back stance phase are derived using the Lagrange method. Then, the equations for the flight after back phases are derived using the Newtonian approach. In both cases the equations are integrated and the results obtained are compared with simulation data.

### 2.3.1 Back Stance Phase

The following is the derivation of the governing equations of motion for the case where the robot is supported on the ground by its back legs (Figure 2-2). The same analysis for the case where the robot is on the front set of legs is presented in section A.2 of the Appendix. The generalized coordinate vector  $\mathbf{q}$  for this system is chosen to be  $\mathbf{q} = [\phi_b, \theta, l_b]^T$ . From the generalized coordinates and Figure 2-3, the body Cartesian coordinates can be expressed at the contact point of the back leg's toe and the ground as follows,

$$\mathbf{p}_b = \begin{bmatrix} -l_b \sin(\theta + \phi_b) + L \cos \theta \\ l_b \cos(\theta + \phi_b) + L \sin \theta \end{bmatrix}. \quad (2-1)$$

These can be differentiated with respect to time, in order to give the body velocities in the x- and y- directions,

$$\dot{\mathbf{p}}_b = \begin{bmatrix} -l_b (\dot{\theta} + \dot{\phi}_b) \cos(\theta + \phi_b) - l_b \sin(\theta + \phi_b) \dot{\theta} - L \dot{\theta} \sin \theta \\ l_b \cos(\theta + \phi_b) - l_b (\dot{\theta} + \dot{\phi}_b) \sin(\theta + \phi_b) + L \dot{\theta} \cos \theta \end{bmatrix}. \quad (2-2)$$

## Chapter 2. Analysis and Simulations

Next, the total body kinetic energy, potential energy and energy loss due to friction can be derived. It is then possible to find the Lagrangian function,  $\mathcal{L}$ , by subtracting the potential energy from the kinetic energy. The Lagrange equation of motion is then,

$$\frac{d}{dt} \left( \frac{D\mathcal{L}}{D\dot{q}} \right) - \frac{D\mathcal{L}}{Dq} + \frac{DB}{D\dot{q}} = F, \quad (2-3)$$

where  $B$  and  $F$  represent the dissipation energy function of the system and the torque moment applied to the system respectively.

The complete derivation of the Lagrange equations of motion is found in section A.1 of the Appendix of this report. By direct substitution, the expanded form for the back legs support case becomes,

$$\begin{cases} m[gL\cos(\theta) - gl\sin(\phi_b + \theta) + [-2L\sin(\phi_b) + 2l]\dot{\phi}_b \dot{l}_b - Ll_b \cos(\phi_b)\dot{\phi}_b^2 \\ + [-2L\sin(\phi_b) + 2l]\ddot{l}_b \theta - 2Ll_b \cos(\phi_b)\dot{\phi}_b \dot{\theta} + L\cos(\phi_b)\ddot{l}_b \\ + [-Ll_b \sin(\phi_b) + l_b^2]\ddot{\phi}_b + [L^2 + r^2 + l_b^2 - 2Ll_b \sin(\phi_b)\ddot{\theta}] \\ ml_b[-g\sin(\phi_b + \theta) + 2l_b\dot{\phi}_b + 2l_b\dot{\theta} + L\cos(\phi_b)\dot{\theta}^2 + l_b\ddot{\phi}_b + [-L\sin(\phi_b) + l_b]\ddot{\theta}] \\ k(l_b - lo) + gmc\cos(\phi_b + \theta) + bl_b - ml_b\dot{\phi}_b^2 - 2ml_b\dot{\phi}_b\dot{\theta} + [Lms\sin(\phi_b) - ml_b]\dot{\theta}^2 \\ + ml_b\ddot{\theta} + Lm\cos(\phi_b)\ddot{\theta} \end{cases} = \begin{cases} 0 \\ \tau_b \\ 0 \end{cases}. \quad (2-4)$$

The above equations of motion are reduced to the following form,

$$A_b(q)\ddot{q} + B_b(q, \dot{q}) = \begin{cases} 0 \\ \tau_b \\ 0 \end{cases}, \quad (2-5)$$

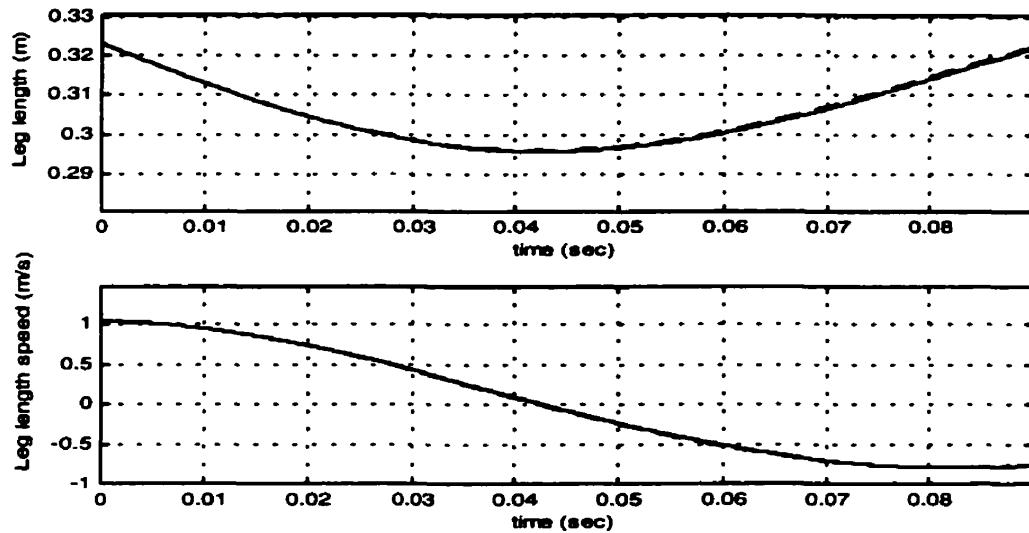
## Chapter 2. Analysis and Simulations

where  $A_b(q)$  is the  $3 \times 3$  inertia matrix,  $B_b(q, \dot{q})$  is the gravity and centrifugal force vector.

Let  $x = [l_b, i_b, \theta, \dot{\theta}, \phi_b, \dot{\phi}_b]^T$  be the state vector. Then the above set of equations is written in the following state space form,

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{m} [k[l_o - x_1] - bx_2 + m[x_1(x_4 + x_6)^2 - g \cos(x_3 + x_5) + Lx_4^2 \sin(x_5)] \\ \quad - L \cos(x_5)[Lx_1[k[x_1 - l_o] + bx_2] \cos(x_5) + \tau_b[L \sin(x_5) - x_1]]] \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{Lx_1[k[x_1 - l_o] + bx_2] \cos(x_5) + \tau_b[L \sin(x_5) - x_1]}{mr^2 x_1} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = \frac{1}{mr^2 x_1^2} [\tau_b[L^2 + 2r^2 + 2x_1^2] - 4mr^2 x_1 x_2 [x_4 + x_6] + L[-2x_1[kx_1[x_1 - l_o] \\ \quad + bx_2 x_1 + mr^2 x_4^2] \cos(x_5) - L\tau_b[\cos(2x_5) - 4x_1 \sin(x_5)] + Lx_1[k[x_1 - l_o] + bx_2] \\ \quad \sin(2x_5)] + 2gmr^2 x_1 \sin(x_3 + x_5)]] \end{array} \right\}. \quad (2-6)$$

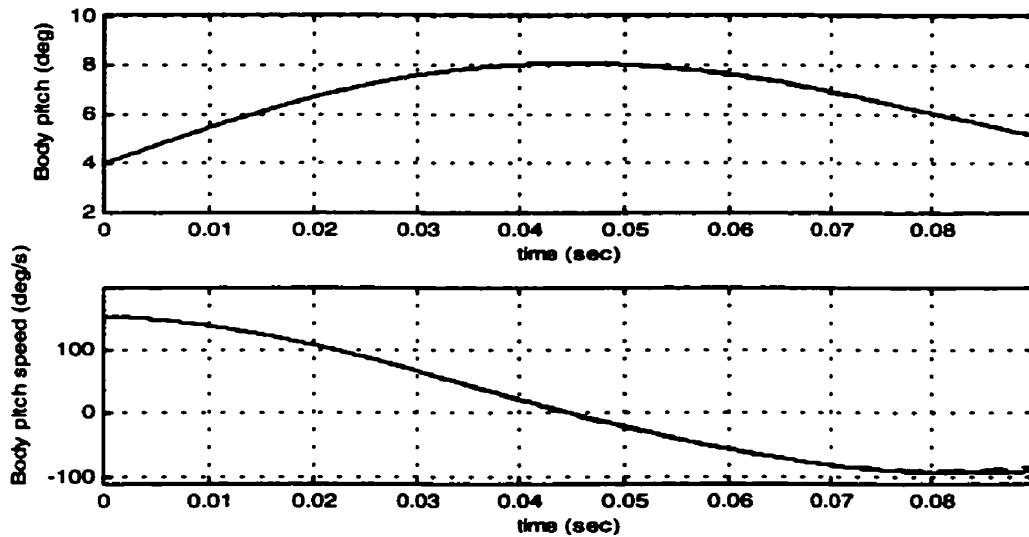
The previously obtained equations of motion are next compared to the simulation package used to test the running controllers. The simulation program is called Working Model 2D®. A complete description of this simulation package is presented in section 2.4. Given a set of initial conditions, the state space form equation is integrated with respect to time, in order to identify the behavior of the state variables. This integration is executed by the Mathematica® [37] package. Concurrently, the same behavior is simulated via the Working Model 2D® simulation software. The outcome of the equations is then compared to the simulation data.



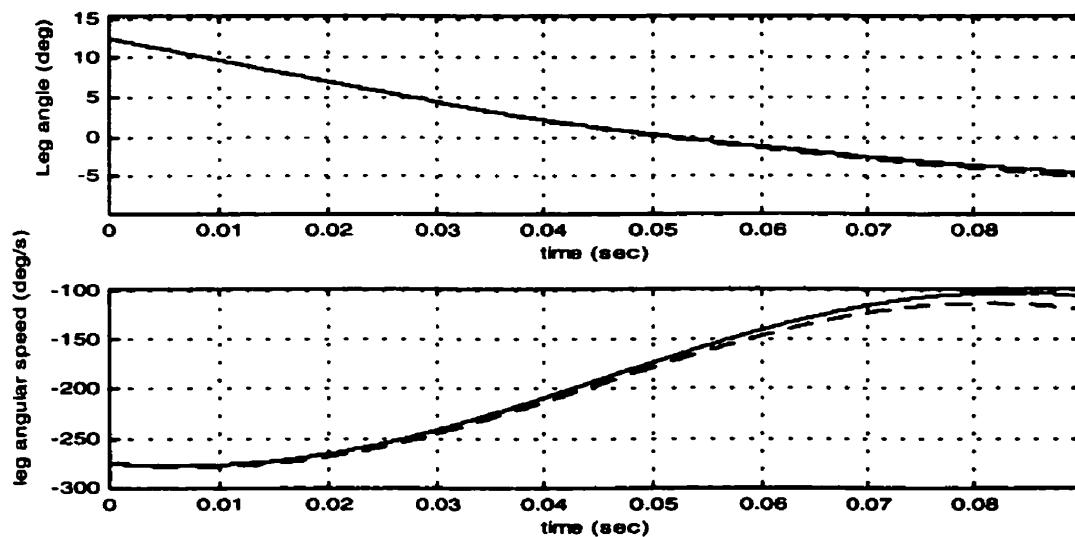
**Figure 2-4: Comparison between Working Model 2D® and Mathematica® plots for back leg length variations throughout the back stance phase. The maximum leg length error is 0.52% of the 10 cm full scale leg displacement. The maximum error in leg length speed is 0.65% of the 2 m/s full scale leg speed.**

Figures 2.4 – 2.6 are Matlab® plots comparing the simulation and mathematical model data. The parameters compared in the plots are the body pitch, body pitch speed, leg length, leg length speed, leg angle, and leg angular speed. Both sets of data are obtained for the same initial conditions. The time duration used in the comparison is dictated by the stance time. The continuous lines represent the Working Model 2D® results, while the dashed lines represent the data obtained from the model derived. The following figures illustrate how closely the mathematical integration resembles the simulation. The discrepancies are due to the assumption of massless legs in the derived equations of motion when these masses are included in the Working Model 2D® simulation. One could predict that these errors can vary with varying conditions, nevertheless, the errors are relatively small and hence the massless leg assumption is valid for the back stance phase.

## Chapter 2. Analysis and Simulations



**Figure 2-5: Comparison between Working Model 2D® and Mathematica® data for body pitch during the back stance phase. The maximum body pitch error is 0.9% with respect to the 10 deg full scale body pitch amplitude. The maximum body pitch speed error is 2.4% with respect to the 300 deg/s full scale body pitch amplitude.**



**Figure 2-6: Comparison between Working Model 2D® data and Mathematica® data for the back leg angle during the back legs support phase. The leg angle error is 2.1% of the 20 deg full scale leg amplitude. The leg angle error is 6.8% of the 200 deg/s full scale leg speed amplitude.**

### 2.3.2 Flight Phase

Since the running gait includes a ballistic state in between each of the stance phases, it is important to look into the robot behavior during this interval of time. The running cycle is analyzed in this section for the phase where the robot is in flight after having been supported by the back legs (Figure 2-2). First, the equations of motion for the flight phase are derived. Then they are compared to the simulation values to assess the mathematical model used.

Throughout the flight phase the body behaves as a projectile, with initial velocities taken at lift-off from the last stance phase. The World Cartesian coordinates and their derivatives at any time,  $t$ , for the body center of gravity are written in the following manner,

$$\mathbf{p} = \begin{bmatrix} x_{lo} + \dot{x}_{lo} t \\ -\frac{1}{2} gt^2 + \dot{y}_{lo} t + y_{lo} \end{bmatrix}. \quad (2-7)$$

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x}_{lo} \\ -gt + \dot{y}_{lo} \end{bmatrix}. \quad (2-8)$$

The body angular position and speeds are as follows,

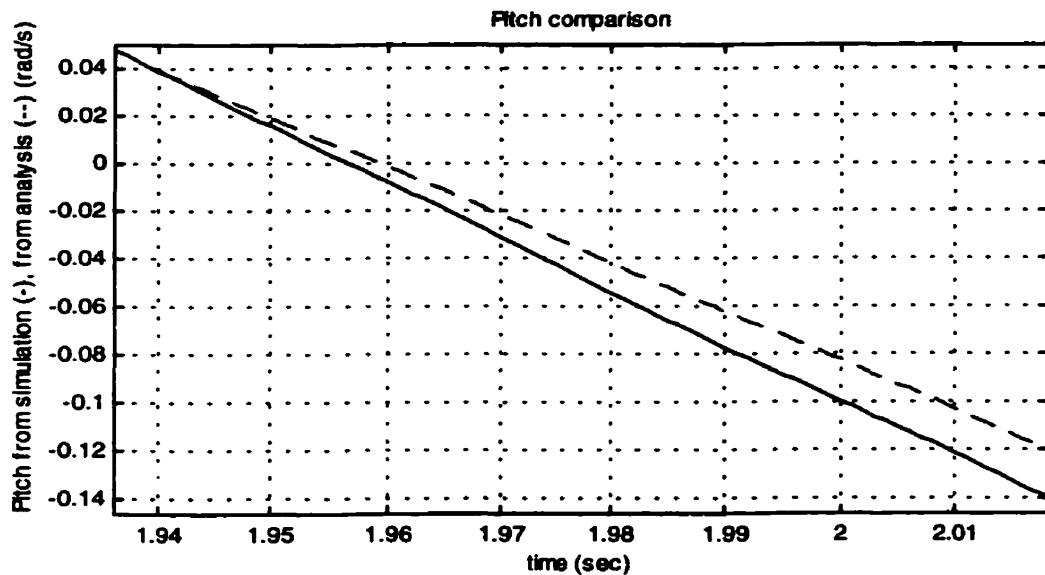
$$\theta = \theta_{lo} + \dot{\theta}_{lo} t. \quad (2-9)$$

$$\dot{\theta} = \ddot{\theta}_{lo}. \quad (2-10)$$

Given that the flight time is available, the body configuration and speeds can be calculated using the flight phase equations. The values for the pitch, body height, and

## Chapter 2. Analysis and Simulations

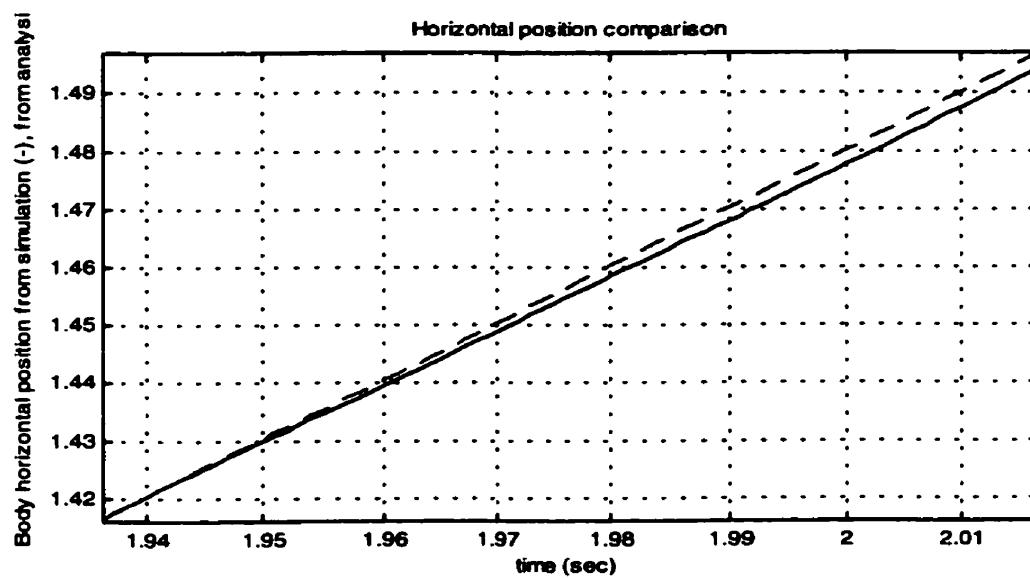
vertical speed during the flight phase are recorded from the Working Model 2D® simulation. These values are then weighed against those obtained from the analytical solution. Figures 2.7 – 2.9 represent the comparisons of the body pitch, body forward position, and vertical position.



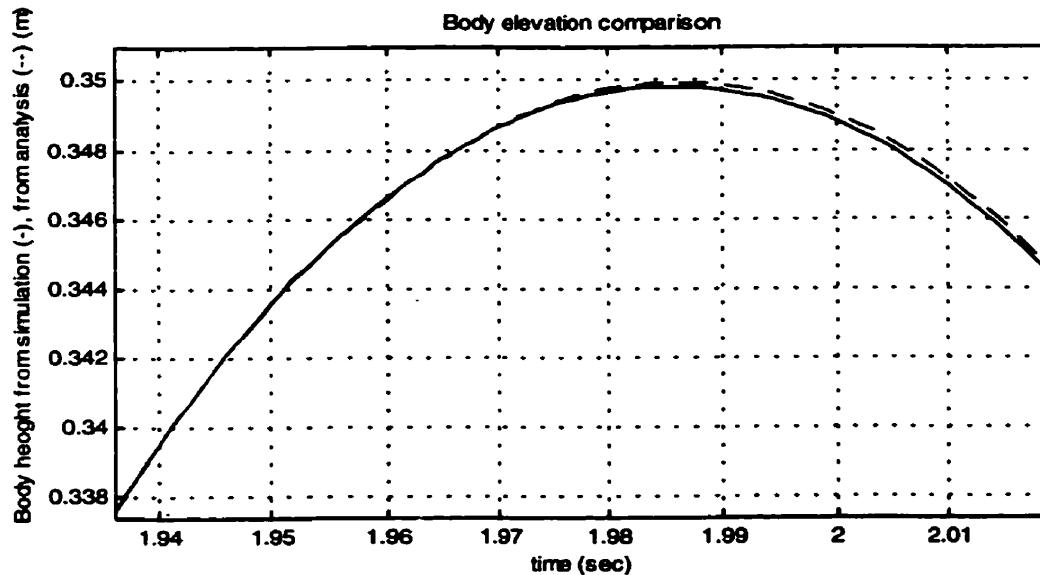
**Figure 2-7: Body pitch during the flight phase. The maximum error in body pitch is 10.52% of the 0.2 rad full scale body pitch amplitude.**

We notice in Figure 2-7 that the pitch is changing in simulation faster than what is seen from the flight equations. The discrepancy here comes from the fact that the robot does not exactly behave as a projectile during the flight phase. That can be explained by the fact that the legs move during the flight phase, in order to position themselves at a specified angle in preparation for landing. Given that the legs in simulation are not massless, the body angular velocity accelerates during flight, as a direct consequence of the conservation of angular momentum. The data obtained for the body horizontal position and forward position does not present major errors.

## Chapter 2. Analysis and Simulations



**Figure 2-8: Comparison of body horizontal position between simulation and analysis during the flight phase. The maximum error in leg length body horizontal position is 3.75% of the 0.08m full scale horizontal displacement.**



**Figure 2-9: Comparison of body height between simulation and analytical equations. The maximum error in hopping height 2.08% of the 0.012 m full scale hopping height.**

## Chapter 2. Analysis and Simulations

The equations of motion were derived in the above sections for all running phases. Following this, they were validated by comparison to simulation results. The comparison results show that the equations of motion produce small errors when compared to the simulation data, except during the flight phase, where the effect of the leg rotation introduces significant error.

If combined together, the above equations of motion could be used to prescribe the desired robot states at discrete times, once per complete running cycle. For example, the robot dynamics can be represented by the step-to-step map

$$S : \mathbf{x}_{td,n+1} = S(\mathbf{x}_{td,n}, \tau_f(t), \tau_b(t)), \quad (2-11)$$

which maps the robot's touchdown states at one front leg impact to the touchdown states at the next front leg impact, as a function of the front and back hip torques between the two front leg impacts. Since the leg position can be freely controlled during flight, the inputs in (2-11) can be reformulated as

$$S : \mathbf{x}_{td,n+1} = S(\mathbf{x}_{td,n}, \phi_{f,td}, \phi_{b,td}, \tau_f(t), \tau_b(t)), \quad (2-12)$$

where  $\phi_{f,td}$ , and  $\phi_{b,td}$  are the front and back hip touchdown angles, and  $\tau_f(t)$ ,  $\tau_b(t)$  are the front and back leg torques during the respective legs' stance phases.

The control problem can be formulated as finding hip touchdown angles  $\phi_{f,td}$ ,  $\phi_{b,td}$  and hip stance torques,  $\tau_f(t)$ ,  $\tau_b(t)$  which make a desired set of touchdown states  $\mathbf{x}_{td}$  a stable fixed point of the discrete dynamical system (2-12).

To our knowledge, there are no existing simple control synthesis methods for robots like Scout II, described by  $S$  (2-12) because it is an intermittent dynamical system, that is, the equations of motion change abruptly from flight to stance phase (also called Variable Structure System). In our case this means that  $S$  cannot be computed analytically. In

## Chapter 2. Analysis and Simulations

addition the system is characterized by other input constraints on the  $\tau_f$ , and  $\tau_b$ : The actual motors on Scout II have torque speed limitations that should be incorporated in the analysis. Another constraint is that the contact point between the robot's leg and the ground cannot be considered as a pin joint connection unless interactions between the leg' toe and the ground is modeled and a torque profile that ensures no slip is generated. Because of the above conditions and the errors obtained in validating the model for the flight phase, the Scout II model developed is only adequate for parametric studies in individual phases of running. We therefore use an alternative heuristic approach to derive running controllers.

In order to develop and test heuristically derived controllers, we use simulations. Even though the physical robot is its best model, deriving controllers on the actual robot can inflict too much wear and tear, and costly breakdowns. This is why a good simulation model is useful. It permits to test the running in dangerous situations and allow one to share with other researchers who want to develop controllers, but do not have the physical robot. As it was done in sections 2.3.1 and 2.3.2, simulation can also be used to compare simplified versions of the model with approximate or simplified analytical models to justify their validity.

### 2.4 Working Model 2D® Simulation Software

Working Model 2D® is a program that can simulate the dynamic motion of bodies under a variety of constraints. Using the basic geometric shapes, one can create a variety of bodies, which are further assembled into one complete model. Each body can be assigned a set of physical properties including mass, inertia, material, kinetic friction factor and electric charge. Once all of the bodies are built, they can be connected together via a set of constraints; a constraint applies a force (or torque) to the bodies at one or more points. Simulations run under the effect of external physical parameters called World Parameters, like gravity, static charge in the air, impact with other bodies and others. After the bodies are assembled and the world parameters are set, We begin the

## Chapter 2. Analysis and Simulations

simulation. This simulation package does not require the equations of motion of the system. Instead, it integrates the forces and moments acting on each body over a finite period of time in order to find the resulting accelerations, velocities, positions and orientations. The accuracy of this approach can be modified by setting the desired integrator type, integrator error, integration step and animation step. Two integrators are available in Working Model 2D®: Euler and Kutta-Merson. The integration error corresponds to the absolute acceptable error in the integration. The integration time step can be fixed by the user or it can be kept variable where Working Model 2D® automatically adjusts the integration time step throughout the simulation to optimize the computational performance. In both cases, the integration step should be smaller than the animation step, which is the time between frames of the animation updated on the screen.

In Working Model 2D®, meters enable the user to collect any desired data in a numerical or graphical form for subsequent analysis or implementation in a feedback algorithm. Simulation data can also be recorded and imported to other mathematical packages such as Matlab® or Mathematica® for further studies.

Working Model 2D® provides the user with two different methods of building, animating and analyzing an event. The first is through a user friendly interface where one can drag and drop objects and constraints onto the working area from a menu. The second method, which was used in this study, is through the Working Model 2D® Basic programming language. The latter is a coding utility that enables the user to construct models and set up different controllers while using scripting language based on visual basic. This method allows for more control and flexibility over the simulation.

In this section, a computer model of Scout II is created. The model is then tested under different running control strategies through simulation. Finally the data is recorded and analyzed using the Matlab® package.

### 2.4.1 Working Model 2D® Script

The Working Model 2D® simulation script was divided into three parts: The first part of the script states the sequence of events required for building the robot components

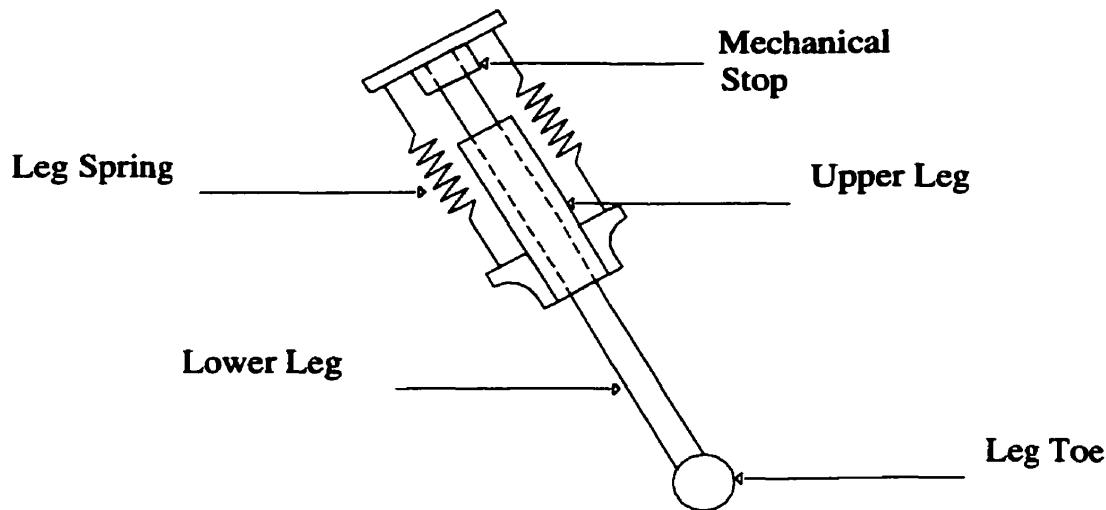
## Chapter 2. Analysis and Simulations

and attaching them together. The resulting model will carry a close resemblance to Scout II, where the dimensions and material properties are selected to match those of the real-life robot. Figure 2-11 is a reproduction of the model obtained in Working Model 2D®. The second part of the script contains functions that are added to the simulation. Two functions were included to make the running as realistic as possible. The first function restricts the motor operation to the characteristics specified by the manufacturer. The second is a slip prevention function; it checks the robot for toe-ground slip and corrects the torque at the hips to prevent slip term occurring. The last script consists of the running algorithms that were tested on the robot. In this section, different control strategies were used for the four phases of the running. These phases were sensed using the two states of the robot legs: The flight and stance phases. These were defined as follows:

- *Stance*: The corresponding leg is touching the ground. This state is sensed whenever the leg spring is greater than the rest length plus the pretension.
- *Flight*: The corresponding leg is in the air. This state is sensed whenever the spring length is equal to the rest length plus the pretension.

### 2.4.2 Robot Model

The robot model used in the simulations is shown in Figure 2-11. It consists of the robot torso, or body, which is connected to the upper legs through the motor shaft. The lower and upper legs form a prismatic joint. This sliding motion is opposed by a tension spring attached to both parts of the leg; the spring can be pre-tensioned as desired. As shown in Figure 2-10, a mechanical stop is added on Scout II to restrict the lower leg from sliding outside the upper leg. A stiff rope is used to represent the mechanical stop in the simulation model. In order to have a realistic sliding motion at the joint, a damper is attached in parallel with the springs. At the lower end of the legs a rubber disk, or toe, is rigidly fastened.



**Figure 2-10: Scout II leg design.**

All robot dimensions and material properties including elasticity, kinetic friction and others were adjusted to the actual values measured on Scout II. These dimensions and properties are shown in Table 2.3. An experiment was performed to determine the friction factor between the toes and the floor in the laboratory. The experiment consisted in applying a horizontal force onto a known mass resting on the floor. The known mass had the same material used for Scout II's toes. The magnitude of the force that displaced the mass horizontally was recorded using a force sensor. The experiment was performed with different masses and resulted in an average friction factor 0.45.

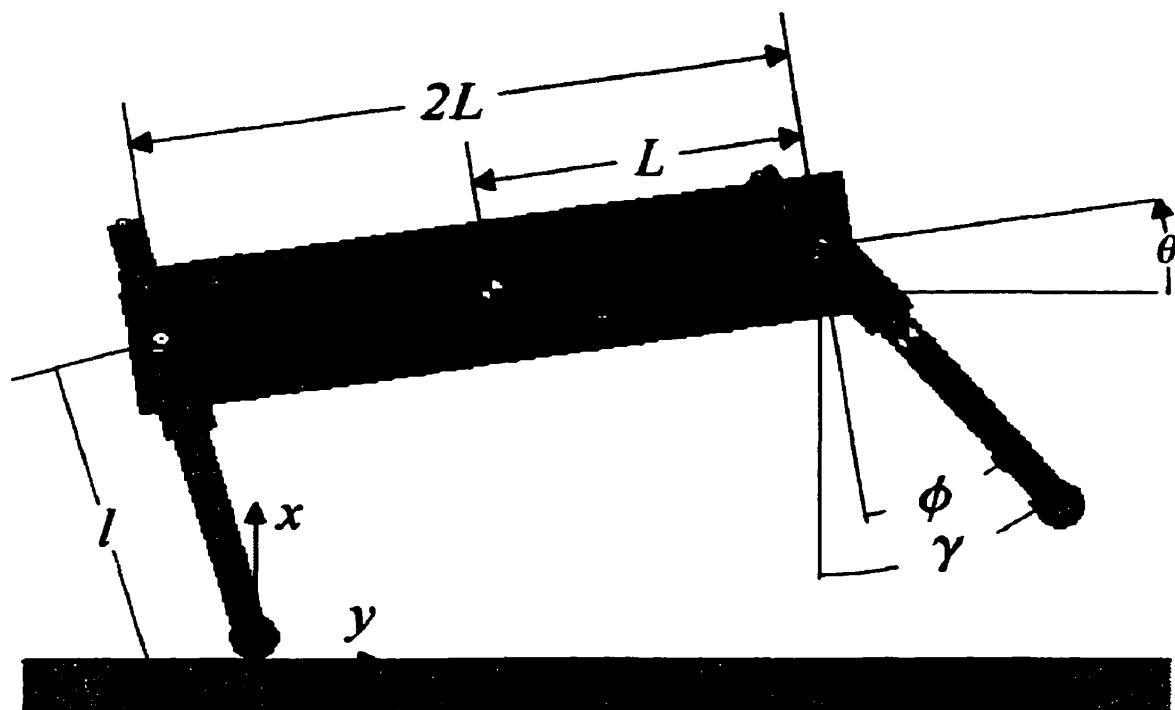


Figure 2-11: Scout II Model built with Working Model 2D°.

Parameter	Symbol	Value
Body Mass	M	23 Kg
Leg Mass	$m_l$	0.82 Kg
Body Inertia	I	1.091 Kg m <sup>2</sup>
Leg Inertia	$I_l$	0.019 Kg m <sup>2</sup>
Hip Length	2L	0.6 m
Body Height	H	0.126 m
Leg Length	l	0.323 m
Leg Spring Constant	K	3600 N/m
Leg Damping Constant	b	30 Ns/m
Toe Elasticity	e	0.8

Table 2.3: Scout II model parameters.

## Chapter 2. Analysis and Simulations

The integrator type used was the Kutta-Merson, the integration error was set to  $10^{-5}$ , and the integration step was variable in order to optimize the results obtained from the simulator. The animation step, which controls the refresh rate on the screen, was set to 1 ms. The latter is also the time step used in the running controller loop. The control input and outputs are thus updated at the animation step rate. This also matched the control time step on the actual Scout II robot.

Because the package is a two-dimensional simulator, the front legs were considered as one leg, and the same is true for the back legs. Since the running considered in this study is mainly in the sagittal plane, the 2D assumption for the robot is valid.

### 2.4.3 Robot Constraints

For a simulation to be worthwhile, it must accurately model the system of interest, interacting in the same way with the environment. Two such models that should be taken into account are the motors operating characteristics and the toe-to-ground interactions during running.

#### Motor Model

As stated by the manufacturer, the 90 W DC motors used on Scout II have a specific operating region [16]. Figure 2-12 is a representation of the torque-speed curve for the Maxon 118777 motor, the Maxon 110404 gearhead, and the sprocket and belt combination attached to the gearhead output shaft [4]. This operating range was obtained taking into account the gearhead maximum rated efficiency of 68%, and the 48/28 sprocket and belt combination efficiency of 96%.

The area below the 24 V operating line represents the operating range for Scout II's current configuration; that line is set by the motor mechanical and electrical setup, and the following form,

## Chapter 2. Analysis and Simulations

$$\omega = \omega_{n_l} + m\tau, \quad (2-13)$$

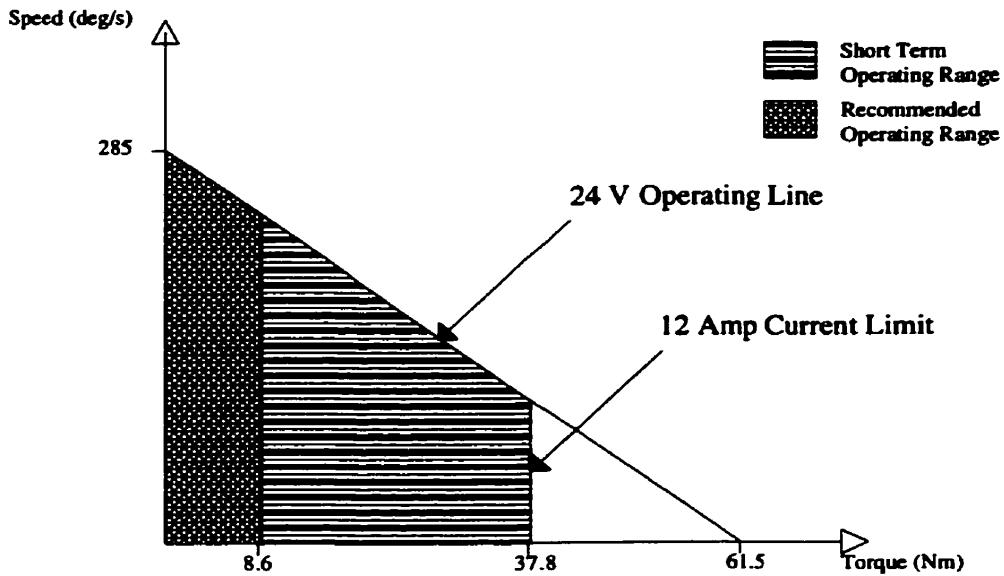
where  $\omega$  ( $s^{-1}$ ), is the angular speed,  $\omega_{n_l}$  ( $s^{-1}$ ) is the no load angular speed,  $m$  (( $sNm$ ) $^{-1}$ ) is the drop, and  $\tau$  (Nm) is the torque.

Regarding the mechanical setup, attaching a gear assembly to the motor shaft can modify the speed torque gradient, by either reducing speed and increasing torque, or increasing speed and reducing torque. As for the electrical set-up, the motor amplifier limits the maximum motor current to 12 A. This restriction appears as an abrupt torque limit at 37.8 Nm, obtained from the specifications in Table 2.4 and the following equation  $\tau_{stall} = I_{max} K_t N_g N_s \eta_g \eta_b$ . As for the operating voltage, it does not modify the torque-speed gradient but alters the operating range area by raising or lowering the voltage operating line. Regardless, the on-board batteries provide 24 V to the actual robot, so this value was used in the simulation.

Operating Specifications	
Maximum current limit ( $I_{max}$ )	12 A
Torque constant ( $K_t$ )	0.0389 Nm/A
Gearhead gear ratio ( $N_g$ )	72.38:1
Sprocket gear ratio ( $N_s$ )	48/28
Gearhead efficiency ( $\eta_g$ )	68%
Belt-sprocket combination efficiency ( $\eta_b$ )	96%

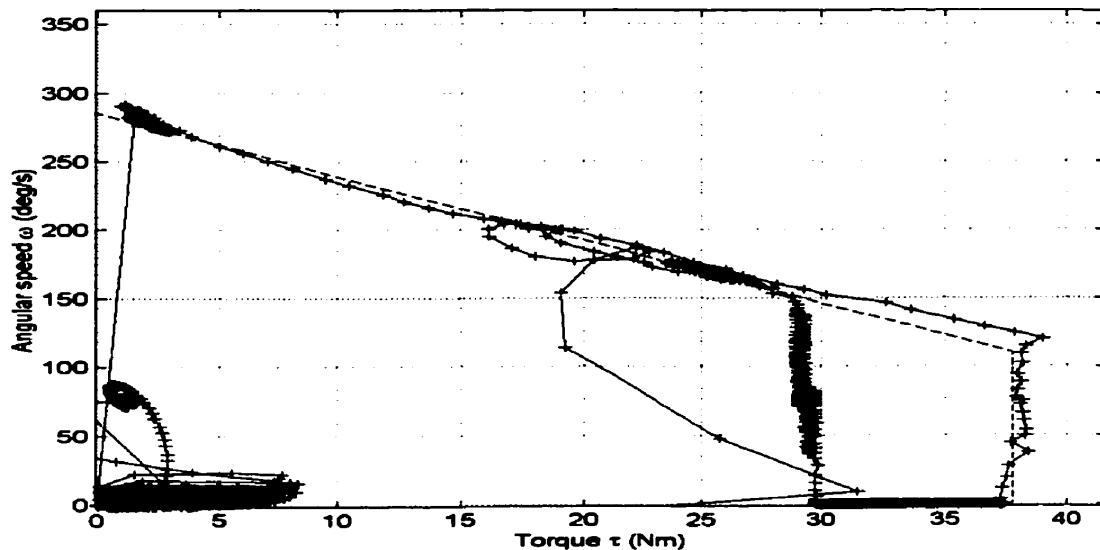
Table 2.4: Hip actuator specifications

## Chapter 2. Analysis and Simulations



**Figure 2-12: Replica of the motor torque-speed model provided by Maxon after gear mounting.**

A validation test for the Maxon torque-speed model is performed on the robot hip actuators. The test consists of commanding the maximum torque to the motors while forcing the legs by hand to move at different speeds. The torque values based on the current sensing feedback and corresponding speed data are then collected and plotted. The plot obtained in Figure 2-13 resembles the graph shown in Figure 2-12. The upper limit, which approximates a line with negative slope, is the expected 24 V Operating Line. The vertical line drawn at the 37.8 Nm reading is the 12 A Current Limit Line. All other points in the figure stand for data recorded inside the operating region. We can therefore rely on the data to be implemented in simulation.



**Figure 2-13: Experimental torque-speed curve in the first quadrant for the motor on Scout II. Data points at the extreme ends form the boundary of the motor operating region. The dashed line represents the boundaries of the motor model obtained from Figure 2-12.**

The above torque-speed curve is implemented in the simulation in order to limit the angular speeds and corresponding torques. The motor model is introduced to the simulation in a torque-speed check function. The function draws the torque-speed curve for the motor in the four operating quadrants. Each time a torque is commanded, the function checks whether that torque lies inside the operating range at that particular motor speed. If so, the torque is applied. Otherwise, the function brings the torque down to the limit of the allowable region, which is at the intersection of the 24 V operating Line and the current motor speed. The results and changes observed in the simulations upon the introduction of the torque-speed check function are discussed later in section 2.5.2.

### Ground Slip

One important aspect of legged locomotion is the interaction between the toe grip on the ground and the amount of torque that can be applied before the occurrence of slip.

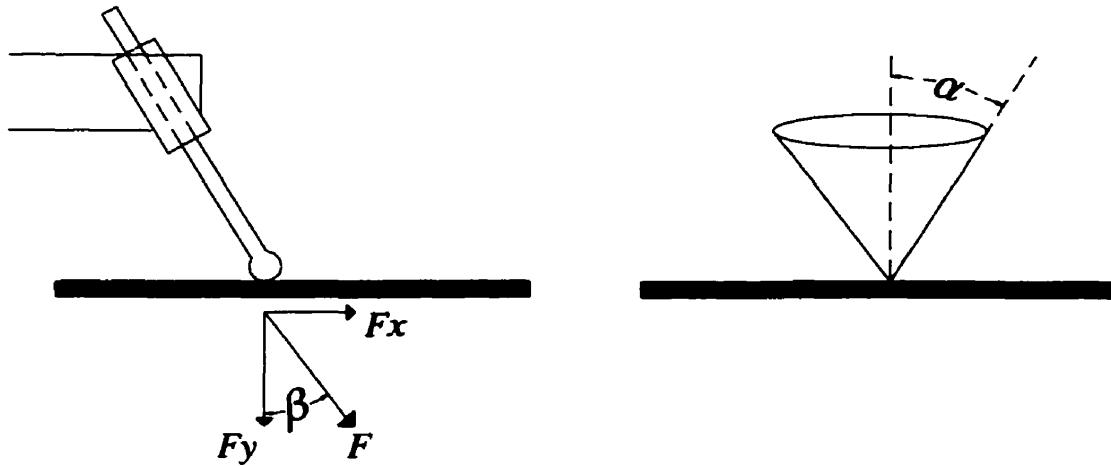
## Chapter 2. Analysis and Simulations

Failure to consider this effect can lead to the robot crashing as seen in section 2.5.2. This section first presents the equations used to analyze the force interactions between the robot's toe and the ground. Such an analysis was conducted in order to approximate the maximum torque that could be applied at the hip without causing toe slip. Then, a method of detecting slip is presented. Finally, a recovery scheme is explained. The three processes of slip prevention, detection, and recovery are then verified using the Working Model 2D® package. It should be noted that while many approaches to deal with slip occurrence can be proposed, the aim here is to find a technique that uses the minimal sensing possible thereby increasing the ease of implementation and the reliability of the method.

### Slip prevention

Assuming there is only one leg on the ground, slip may occur under two conditions. The first one occurs when the friction force on the leg is in the negative x-direction (see Figure 2-14), in other words when, slip tends to occur in the forward direction. The second case takes place when the friction force on the toe is in the positive x-direction, meaning that slip tends to occur in the backwards direction. The first usually happens immediately following touchdown, while the second one arises at the end of the stance phase.

The force analysis is quasi-static and assumes massless legs, point contact with the ground, and no accelerations. Forces in the x- and y- directions are considered at the point of toe-ground contact. In the derived equations  $F_b$  is the friction force in the leg,  $F_s$  is the force due to the spring, T is the reaction to the hip actuator torque, and  $N$  is the reaction of the ground.



**Figure 2-14:** The figure on the left shows the foot forces upon contact with the ground. The right figure shows the friction cone, which limits the foot force angle before slip occurrence.

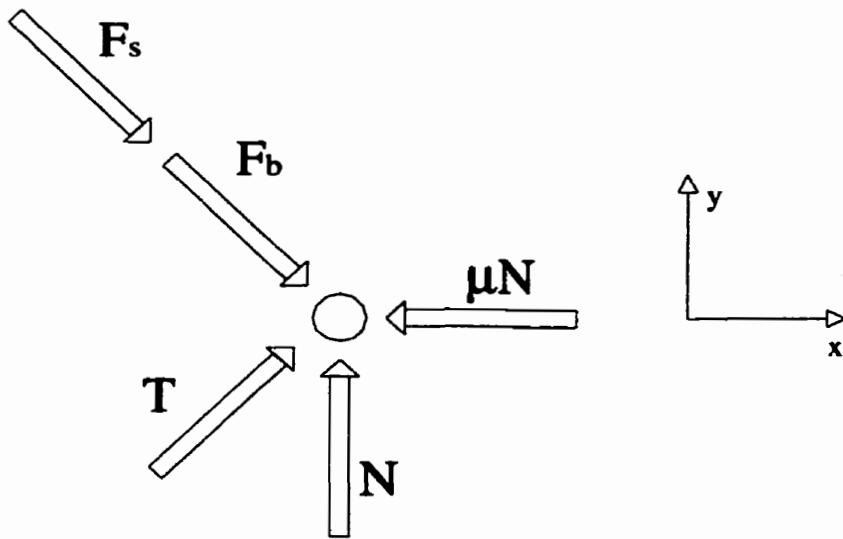
Figure 2-14 shows the forces applied by the toe on the ground. The toe will only slip if  $F_x > \mu F_y = \mu N$ , that is, the force is outside the friction cone. The friction cone represents the limiting case where  $F_x = \mu F_y = \mu N$ . If we substitute  $F_x = F \sin \alpha$  and  $F_y = F \cos \alpha$  then the friction cone is represented by,  $\alpha = \tan^{-1} \mu$ .

The Newton equation of forces applied at the limit of the friction cone can thus be written in the following manner. At first, the equations for the event of forward slip are described below according to Figure 2-15, where,

$$\sum F_x = F_b \sin(\gamma) + F_s \sin(\gamma) + T \sin(\gamma) - \mu N = m\ddot{x} \equiv 0, \quad (2-14)$$

$$\sum F_y = N - F_b \cos(\gamma) - F_s \cos(\gamma) + T \sin(\gamma) = m\ddot{y} \equiv 0, \quad (2-15)$$

$$\sum T = -\tau + Tl \equiv 0. \quad (2-16)$$



**Figure 2-15: Free body diagram for the contact forces between toe and ground.**

If we substitute the forces with the Scout II parameters (see Figure 2-11), the equations become,

$$k(l - l_0) \sin(\phi + \theta) + \frac{\tau}{l} \cos(\phi + \theta) + bl \sin(\phi + \theta) - \mu N = 0, \quad (2-17)$$

$$k(l_0 - l) \cos(\phi + \theta) + \frac{\tau}{l} \sin(\phi + \theta) - bl \cos(\phi + \theta) + N = 0. \quad (2-18)$$

The above equations are solved in order to find the torque at the limit of the friction cone,

$$\tau_{Limit,F} = \frac{-l[\sin(\theta + \phi) - \mu \cos(\theta + \phi)][k(l_0 - l) + b\dot{x}]}{\cos(\theta + \phi) + \mu \sin(\theta + \phi)}. \quad (2-19)$$

## Chapter 2. Analysis and Simulations

If the same approach is used for the event where slip occurs in the backward direction, the following is obtained,

$$\sum F_x = F_b \sin(\gamma) + F_s \sin(\gamma) + T \sin(\gamma) + \mu N = m\ddot{x} \equiv 0, \quad (2-20)$$

$$\sum F_y = N - F_b \cos(\gamma) - F_s \cos(\gamma) + T \sin(\gamma) = m\ddot{y} \equiv 0, \quad (2-21)$$

$$\sum T = -\tau + Tl \equiv 0. \quad (2-22)$$

Once again by substitution,

$$k(l - l_0) \sin(\phi + \theta) + \frac{\tau}{l} \cos(\phi + \theta) + bl \dot{s} \sin(\phi + \theta) + \mu N = 0 \quad (2-23)$$

$$k(l_0 - l) \cos(\phi + \theta) + \frac{\tau}{l} \sin(\phi + \theta) - bl \dot{s} \cos(\phi + \theta) + N = 0. \quad (2-24)$$

The above equations are again solved to find the torque at the limit of the friction cone,

$$\tau_{Limit,B} = \frac{-l[\sin(\theta + \phi) + \mu \cos(\theta + \phi)][k(l - l_0) + b\dot{x}]}{\cos(\theta + \phi) - \mu \sin(\theta + \phi)}. \quad (2-25)$$

Accordingly, the simulation now includes a function that continuously checks the actual and the desired torques and compares them to the maximum allowable torque before slip. If slip is predicted, the desired torque is clipped to the maximum allowable one while allowing for a safety factor of 0.9. That safety factor is accounted for as to take care of the assumptions considered at the beginning of this section. The results and changes in the simulations are discussed later in section 2.5.2.

## Chapter 2. Analysis and Simulations

### Slip detection

The slip detection function constantly monitors the body acceleration. If at any time, the body acceleration exceeds a certain threshold, the recovery function is called.

Two methods to detect the beginning of slip will be presented. The first was implemented on Scout II due to its simplicity and reliability. The second method is more sensitive to the effectiveness of the sensing on Scout II and thus less reliable. It was therefore only investigated in simulation.

The first method is to constantly monitor the hip angular speed. If at any time during the stance phase this angular speed changes instantly, slip is indicated. The amount of instant change in speed used to detect slip was tuned in simulation and on the experimental robot.

The second method is accomplished by comparing two methods of calculating the acceleration of the robot. Whenever these two methods give different results, the robot is considered to be slipping. According to Figure 2-11, it is possible to find the body acceleration geometrically as follows,

$$x = -l \sin(\theta + \phi) + \frac{L}{2} \cos(\theta + \phi), \quad (2-26)$$

$$\dot{x} = -l \sin(\theta + \phi) - l(\dot{\theta} + \dot{\phi}) \cos(\theta + \phi) - \frac{L}{2} (\dot{\theta} + \dot{\phi}) \sin(\theta + \phi), \quad (2-27)$$

$$\begin{aligned} \ddot{x} = & -l \ddot{\sin}(\theta + \phi) - l(\ddot{\theta} + \ddot{\phi}) \cos(\theta + \phi) - 2l(\dot{\theta} + \dot{\phi}) \cos(\theta + \phi) \\ & + l(\dot{\theta} + \dot{\phi})^2 \sin(\theta + \phi) - \frac{L}{2} (\ddot{\theta} + \ddot{\phi}) \sin(\theta + \phi) - \frac{L}{2} (\dot{\theta} + \dot{\phi})^2 \cos(\theta + \phi) \end{aligned} \quad (2-28)$$

The above acceleration can be calculated from the sensing available on Scout II. It can be compared with the forward acceleration obtained using the Newton equations. This acceleration can be found as follows,

$$-\frac{\tau}{l} \cos(\theta + \phi) - F_b \sin(\theta + \phi) - F_s \cos(\theta + \phi) = M\ddot{x}, \quad (2-29)$$

$$\ddot{x} = \frac{1}{M} [-\frac{\tau}{l} \cos(\theta + \phi) - k(l_0 - l) \sin(\theta + \phi) - b\dot{l} \cos(\theta + \phi)]. \quad (2-30)$$

We now have two values for the center of mass acceleration. The value obtained from equation (2-28) will be called  $\ddot{x}_a$  and the one obtained from equation (2-30) will be  $\ddot{x}_e$ . The first one is also the acceleration of the toe referenced to the body center of mass position. The latter is the body acceleration when the toe is pinned to the ground. When the toe slips on the ground, the sensed acceleration,  $\ddot{x}_e$ , becomes considerably different from the acceleration,  $\ddot{x}_a$ , which is the expected value based on the Newton equations for no slip.

### Slip recovery

If toe slip is detected, the torque should be appropriately reduced in order to stop it. The recovery function therefore interferes in order to stop the leg from slipping.

There are different ways to recover from slipping. Boone et al [6] proposed several reflexive responses to a slip during dynamic locomotion. Their most successful approach consists of lifting the slipping foot during stance and repositioning it for another attempt. In some cases, more than one leg drops down to the ground to ensure maximum grip. Although successful, the above approach assumes controllable leg lengths. On Scout II the prismatic joint in the leg is completely passive. Other than the above presented methods, a slip recovery function based on ground speed matching can be implemented. Whenever the robot slips on the ground, the hip actuator will control the leg angular speed to match the robot forward velocity. This will therefore properly fix the toe back onto the ground. Unfortunately this requires a sensing device that can measure the

## Chapter 2. Analysis and Simulations

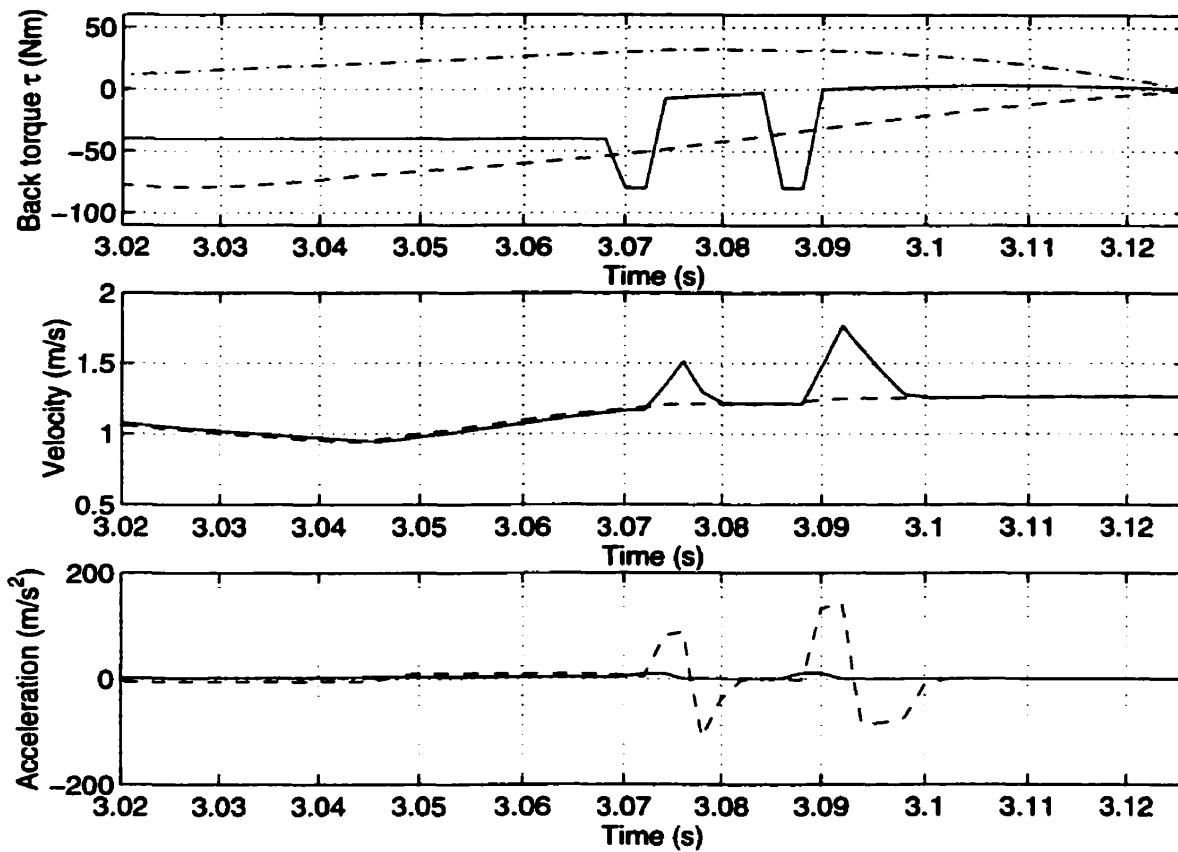
forward velocity of the robot without having to refer to the leg angle and body pitch. This is yet not possible with the sensory devices on Scout II.

The method used in simulation and experiment to recover from slipping will make use of the same equations implemented for preventing slip. However, in this case the friction factor,  $\mu_d$ , will be smaller, to take into account that the toe is not fixed onto the ground. Using  $\mu = 2\mu_d$  was suggested by Nagel [30], who used the same approach to recover from slipping in his simulation work on slip with legged robots. If this is the case, the torque value applied at the instant of slip detection will be,

$$\tau_{Limit} = \frac{-l[\sin(\theta + \phi) + \mu_d \cos(\theta + \phi)][k(l - l_0) + b\dot{x}]}{\cos(\theta + \phi) - \mu \sin(\theta + \phi)}. \quad (2-31)$$

It was previously implied that running with constant torque during the stance phases required constant monitoring of the applied torque by the slip prevention function. In this section, slip during the stance phase is intentionally induced. The experiment goes as follows: At a particular time during the robot's running cycle, the slip prevention controller is deactivated and a relatively high torque is applied by the hip actuators, inducing slip. At this instant, the leg angular speed will accelerate. This will induce a difference between the calculated acceleration,  $\ddot{x}_a$ , and the expected acceleration,  $\ddot{x}_e$ , of the body. The slip detection function will hence detect the slip and call on the recovery procedure. The recovery function calculates the torque required to resume the running motion with minimal slip. As the calculated torque is applied, the toe grips back onto the ground and the robot continues the running.

## Chapter 2. Analysis and Simulations



**Figure 2-16: Simulation results for the three slip functions.** Dotted lines in the top figure represent the torque limits obtained in the slip prevention function. The continuous line is the torque applied at the motors. In the middle graph, the dotted line is the actual forward velocity, while the continuous line is the forward velocity calculated from the geometry. In the bottom figure, the continuous line is the acceleration expected from the Newton equations, while the dotted line is the acceleration obtain from geometry.

The results of the simulation experiment are illustrated in Figure 2-16. The top plot shows on two occasions when the hip actuator torque is commanded to  $-80$  Nm for the pair of legs on the ground, thereby exceeding the torque limit set by the slip prevention function. Every time the torque limit is exceeded, the slip recovery function then reduces the torque in order to stop the toe from slipping. The plot in the middle is a comparison between the forward velocity given by Working Model 2D<sup>®</sup> and the forward velocity

## Chapter 2. Analysis and Simulations

calculated geometrically or using equation ( 2-27 ). The plot shows that each time the torque exceeds the limit, the geometrically calculated torque and the one obtained by Working Model 2D® differ. This only happens when the leg slips on the ground. Afterwards, as the slip recovery function interferes, both velocities match again. The bottom plot shows how both methods of calculating the body acceleration differ at the instant of slip. Thus, this validates the approach used by the slip detection function.

## 2.5 Running Algorithms

The purpose of this section is to present different running algorithms applied in simulation. First, the simplest algorithms will be explained and tested, more elaborate controllers will then follow. The open loop controller that uses only feedback of the leg state is first presented in section 2.5.1. Then a closed loop controller that uses velocity feedback is presented in section 2.5.2. Sections 2.5.3, 2.5.4, 2.5.5, and 2.5.6 are modifications of the closed loop controller that introduce control over the forward speed, hopping height and body pitching of the robot during running. Finally, section 2.5.7 discusses simulations in a 3 D simulation package and investigates running with turning.

### 2.5.1 Open Loop Controller

Amazingly, an open loop controller was found to stabilize dynamic walking in Scout I [38]. While there is no a priori indication that such an open loop controller would stabilize a highly unstable system like Scout II, we still wanted to try it to get some initial insight.

The Open Loop Controller combines independent control sequences for the front and back legs respectively. Each pair of legs has two states, Stance and Flight. The independent control sequences command the leg pairs as a function of their state. As such the controllers do not make any assumptions on a bounding sequence of states (see figure 2-1, 2-1). In this manner, a leg pair does not require any sensing or feedback of what the other leg pair is doing or any robot state parameters such as body attitude or forward

## Chapter 2. Analysis and Simulations

velocity. The sensing required for this controller are the leg angles low-level servo control and the leg spring length for detecting leg states. The body pitch, spring length, and spring length speed are only needed for the slip prevention function.

While in flight, the leg is commanded to a constant leg angle in preparation for touchdown, via a PD controller. As observed by Raibert [32], for each forward velocity there is a unique touchdown angle that results in zero net acceleration. By commanding the appropriate angle when the leg contacts the ground, the robot's forward speed and hopping height can be maintained. This will ensure that the robot legs always have enough clearance during the flight phase to sweep the legs once again to their intended touchdown position. To overcome the energy losses, the hip actuators will apply torque to the legs during the stance phase.

There is therefore an energy cycle that goes as follows: As the robot falls onto the ground, the leg angle at touchdown causes the transfer of some of the forward energy into the springs, in order to maintain a constant hopping height. In the first part of the stance phase, the springs store potential energy as the robot's vertical and horizontal speeds are decreasing. The energy in the springs is then released, and the robot lifts off the ground. All the while, energy is added into the system by the motor, in order to overcome the losses due to the legs.

The heuristic approach used here relies on descriptions in previous work done on dynamic robots [1][10][32][38]. Previous work and initial simulation trials have demonstrated that increasing leg angles at touchdown will decrease the forward speed of the robot, increase the hopping height, increase body pitching, and decrease the flight after stance phase. The latter will thus lead to the vanishing of one flight phase of the four phases of the bound. Following these guidelines, the leg angles at touchdown are modified to get a balance between forward speed which is necessary to move the robot forward and hopping height which is necessary to clear the legs for the return phase. The motor torques at the hips have to be adjusted to ensure that the energy loss in the legs is recovered. There are three types of energy loss in the legs. The first one is due to the friction in the prismatic joint. Besides the energy loss due to friction, there are two other kinds of losses that are present: those that occur at touchdown and those that take place

## Chapter 2. Analysis and Simulations

during lift-off [32]. During touchdown the leg dissipates its kinetic energy to the ground damping when it is suddenly brought to rest. The mechanical stops also dissipate a fraction of the robot's kinetic energy at lift-off when they push the legs off the ground. It is also necessary to properly balance the torques between the front hip actuators and the back hip actuators. If the front legs were on the ground, applying a negative torque would lead to a backward pitching of the robot. In other words the robot's body would rotate counter-clockwise, hence decreasing the vertical distance between the back toes and the ground. This unfortunately increases the risk of toe stubbing. In the event of the back legs supporting the robot, applying a negative torque would rotate the body in a counter-clockwise direction. This, on the other hand, will increase the clearance between the front toes and the ground, generating a situation that is favorable to the robot's running. Considering the previously mentioned situation, it is preferable to apply most of the torque in the back hips instead of having an equal partition between back and front. This in turn will increase toe-to-ground clearance for both stance phases.

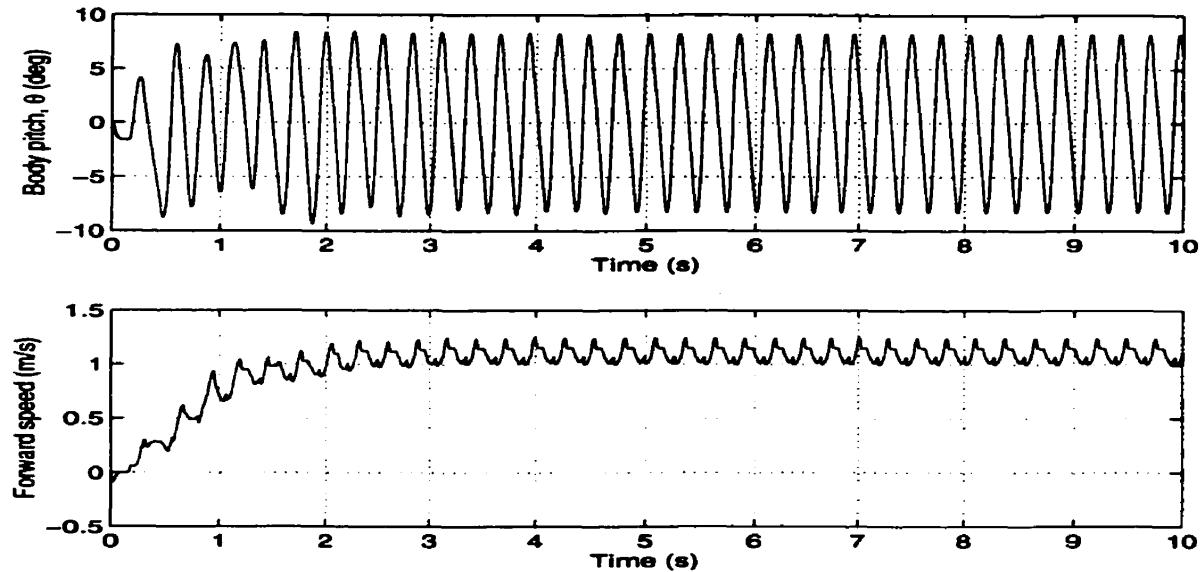
$$C_1 = \begin{cases} \gamma_{f,td} = 22^\circ \\ \gamma_{b,td} = 18^\circ \\ \tau_{f,s} = -10 \text{ Nm} \\ \tau_{b,s} = -40 \text{ Nm} \end{cases}$$

After consideration of the above remarks, the open loop controller was successfully simulated with the values shown in  $C_1$ . As shown in Figure 2-17, the simulated robot has a forward speed of 1.2 m/s. Its body oscillation has an amplitude of 6.5 degrees and a period of 0.29 seconds.

It is clear from the results obtained that compliant quadruped running control is possible in its simplest form, without any explicit feedback control forward speed or stance time. This open loop controller results in a robust stable periodic behavior. A

## Chapter 2. Analysis and Simulations

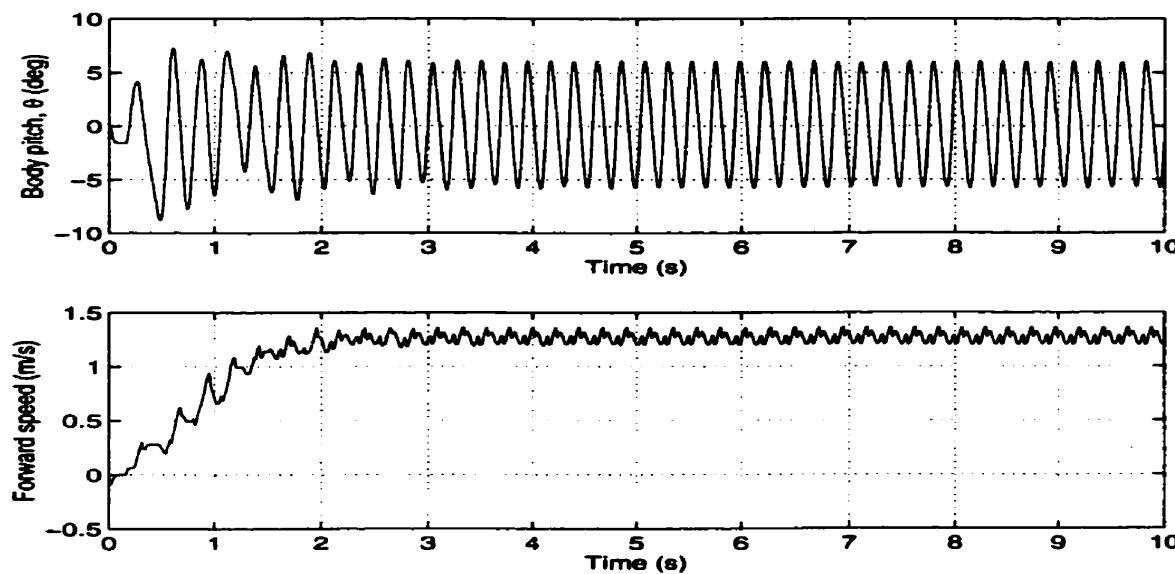
slight variation of the leg angle does not crash the robot but results in a stable running at slightly different speeds. Further variation of the leg angles will lead to the vanishing of one of the flight phases.



**Figure 2-17: Body pitch and forward velocity during running for the open loop controller with fixed legs touchdown angles.**

A variation of the above controller could incorporate the control of the hip angle during the stance phase. This change will not require feedback of the body pitching to calculate the leg angle at touchdown. Figure 2-18 represents the data obtained from such an open loop controller with  $\phi_{f,id} = 25^\circ$ , and  $\phi_{b,id} = 20^\circ$ .

The disadvantage of this controller is that it only runs at one specific forward speed, given a set of touchdown angles. If another speed is desired, the appropriate touchdown angles should be derived using simulation. Although this is possible, it was decided to investigate a closed-loop controller that takes advantage of the robot velocity feedback.



**Figure 2-18: Body pitch and forward velocity during running for the open loop controller with fixed hip touchdown angles.**

### 2.5.2 Closed Loop Controller with Velocity Feedback

This closed loop controller is similar to the open loop controller in that there are separate states for the hind and the front legs. Control of the locomotion relies on the state of each pair of legs, and is independent from the overall robot state. However, in this controller, feedback of the forward velocity of the robot was utilized as a control input.

During the flight phase, the corresponding leg is commanded to a set point angle. This angle is continuously updated until touchdown. In contrast to the open loop controller where the hip angle was controlled, this controller commands the leg angles. The commanded angle depends on the body pitch and the previous forward velocity, the stance time, and the desired hopping height. The controller has the following arguments,

## Chapter 2. Analysis and Simulations

$$\left\{ \begin{array}{l} XCG_d = \frac{\dot{x}_T}{2} + x_{offset} \\ \gamma_d = \arctan \left( \frac{XCG_d}{\sqrt{l^2 - XCG_d^2}} \right) \\ \phi_d = \gamma_d - \theta \end{array} \right. \quad (2-32)$$

The idea behind this controller is taken from Raibert who did previous work on dynamically running robots. Raibert states that at any running speed, there exists one leg angle at touchdown that will maintain the forward running speed constant. To calculate the position of the leg angle at touchdown, the control system estimates the locus of points over which the center of gravity will travel during the next stance phase. This distance is thus approximated by the product of the forward speed and the duration of the stance phase. As seen in Figure 2-19, to place the foot forward, the distance to the front

of the hip is,  $\dot{x} \frac{T}{2}$ . Because a spring mass system oscillates with a period that is

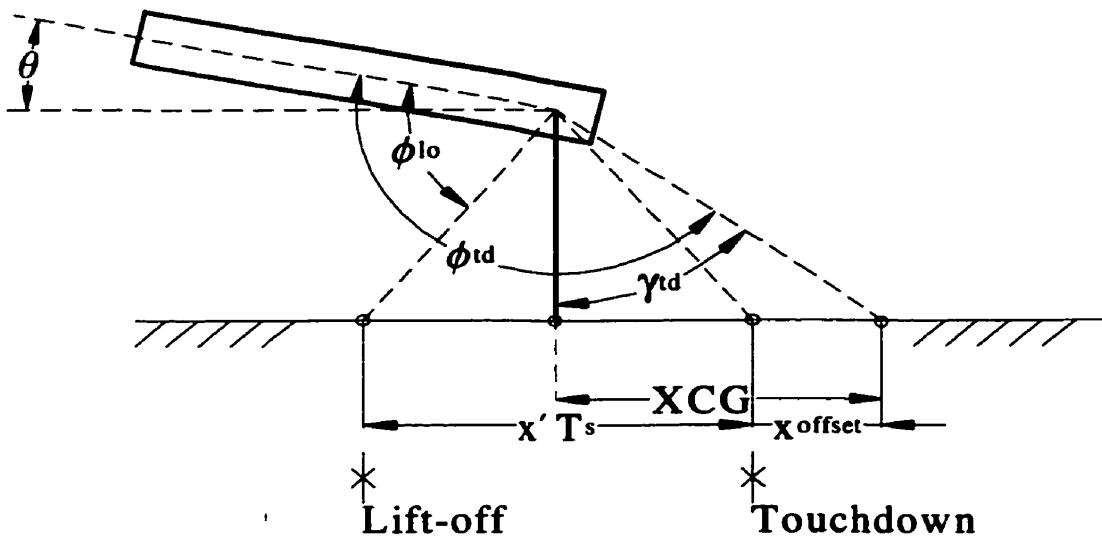
independent of amplitude, the duration of the stance phase is nearly constant for a given leg stiffness. The control system hence uses the duration of the previous stance phase as the expected duration for the next stance phase. Thus the touchdown angle is obtained only from the feedback of the forward velocity and the stance duration in the previous cycle. In order to increase or decrease the forward velocity one would need to decrease or increase the leg angle at touch down by adding an offset distance which is included in  $x_{offset}$ . Therefore the horizontal forces acting on the body throughout the stance phase

assume a non zero value thus accelerating or decelerating the robot. This observation was first introduced by Raibert [32] in the form of a proportional controller, based on the forward velocity error,

## Chapter 2. Analysis and Simulations

$$x_{\text{offset}} = k_x (\dot{x} - \dot{x}_d), \quad (2-33)$$

where,  $\dot{x}$  is the forward speed,  $\dot{x}_d$  is the desired forward speed and  $k_x$  is a feedback gain.



**Figure 2-19: Leg trajectory during the stance phase.**

In addition to the hip actuator, Raibert's robots included an articulated prismatic joint in the leg; thus he had a means of injecting energy into the system during the stance phase to maintain the hopping height constant. Since in our robot, the aim was to decrease cost by keeping the amount of actuators to a minimum, we are unable to control the vertical hopping height via a leg actuator in an independent fashion. Instead we had to transfer some forward energy to the vertical via  $x_{\text{offset}}$ . We were able to inject energy into the system to maintain constant hopping height by adding an extra term in  $x_{\text{offset}}$ . This term increases the leg touchdown angle in order to store some of the forward kinetic energy in the spring and releasing it later in the stance phase in order to maintain a certain hopping

## Chapter 2. Analysis and Simulations

height. An extra term,  $a$ , is now added to the equation to account for the energy exchange,

$$x_{\text{offset}} = k_x (\dot{x} - \dot{x}_d) + a. \quad (2-34)$$

The second line of equation (2-32) is a kinematic transformation to find the required hip angle with respect to the vertical axis. The third line of (2-32) transforms the commanded angle into a hip angle referenced to the body pitch. The final hip angle is regulated using a PD controller.

It should be noted that the body attitude and dynamics at touchdown for either the front legs or hind legs are different. Therefore, the back and front offset distances do not have the same magnitude of effect on the forward speed and hopping height. As a result the front legs offset distance is greater than that of the hind legs. This could also be seen in the first controller where the front legs touchdown angle was considerably larger than that of the back legs.

$$C_2 = \begin{cases} x_{\text{offset},f} = 0.07m \\ x_{\text{offset},b} = 0.04m \\ \tau_{b,s} = -40Nm \\ \tau_{f,s} = -10Nm \end{cases}$$

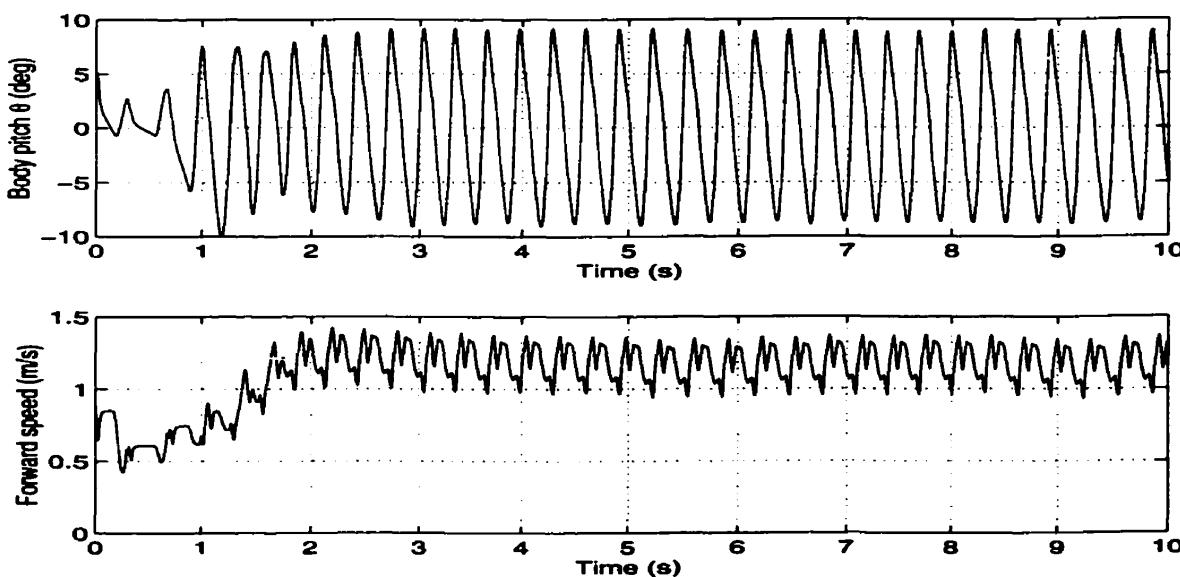
The closed loop controller was simulated with the values shown in  $C_2$ . The feedback of the forward velocity was done continuously in the cycle. The results of the closed loop controller simulation were very satisfactory and the running was steady with smooth transitions between flight and stance. The four phases of the running cycle were distinct and very clear. Figure 2-20 shows the body pitch and forward speed data

## Chapter 2. Analysis and Simulations

recorded from the simulation. The forward velocity obtained averaged 1.2 m/s and the body pitching amplitude was  $8^{\circ}$ .

The running turned out to be successful, but there were two flaws in the initial simulation. The first was the no slip assumption, which was found to be far from true in practice. The other factor that was not considered is the motor torque-speed model. While the simulation results in Figure 2-20 were satisfying to the eye, a closer look at the data in Figure 2-25 revealed motor speeds that reached 1000deg/sec in the hip speeds. This speed is about four times the maximum speed of the motor. These observations instigated the investigation of running simulations with the motor model and the toe-ground interaction model.

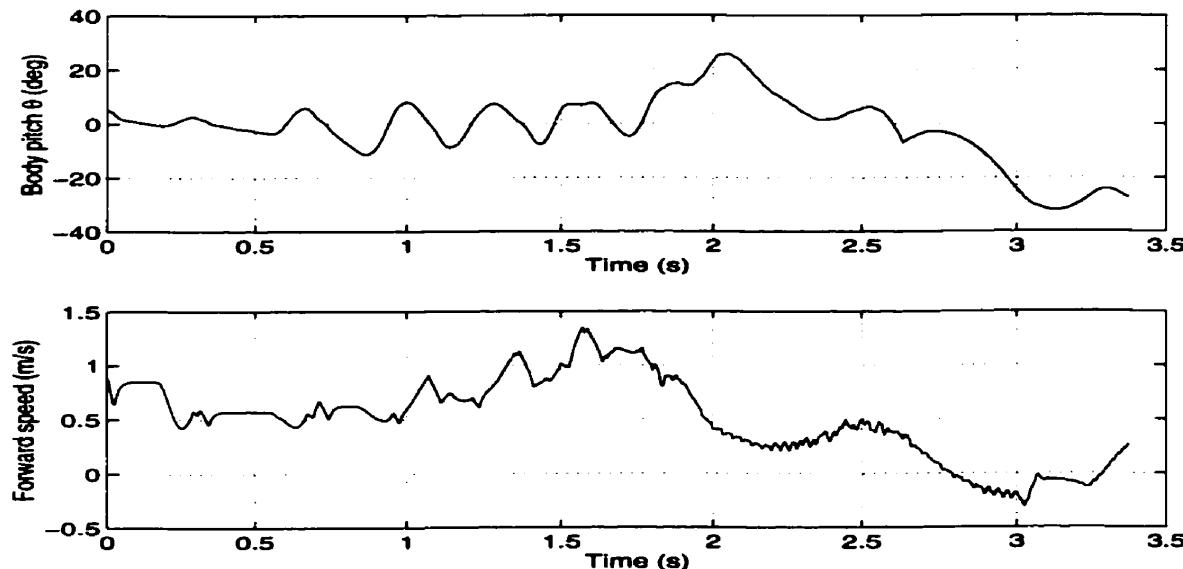
Please note that while these conditions are mentioned in this section, the same events were also observed at the initial time when the open loop controller was being investigated. These remarks are presented here to study the effect of simulations with unmodeled motor characteristics and toe-to-ground interaction. The open loop simulation previously shown incorporated the motor model and the slip functions.



**Figure 2-20: Body pitch and forward velocity during running for a closed loop controller with no restrictions on the environment and the model.**

## Chapter 2. Analysis and Simulations

The next step was to include the actual friction factors measured from the ground in the lab. An experiment set up in the lab revealed a friction factor of about 0.45. When the friction factor in the simulation was set to 0.45, the robot crashed only after a few running steps. Figure 2-21 testifies for the results obtained. The robot speed decreases and the robot crashes on the ground because of the legs slipping backward and losing the energy stored in the springs and that provided by the motor. Due to the decrease in hopping height, the legs do not have enough clearance in order to sweep back inside and prepare for the next landing, thus causing the toes to stub.

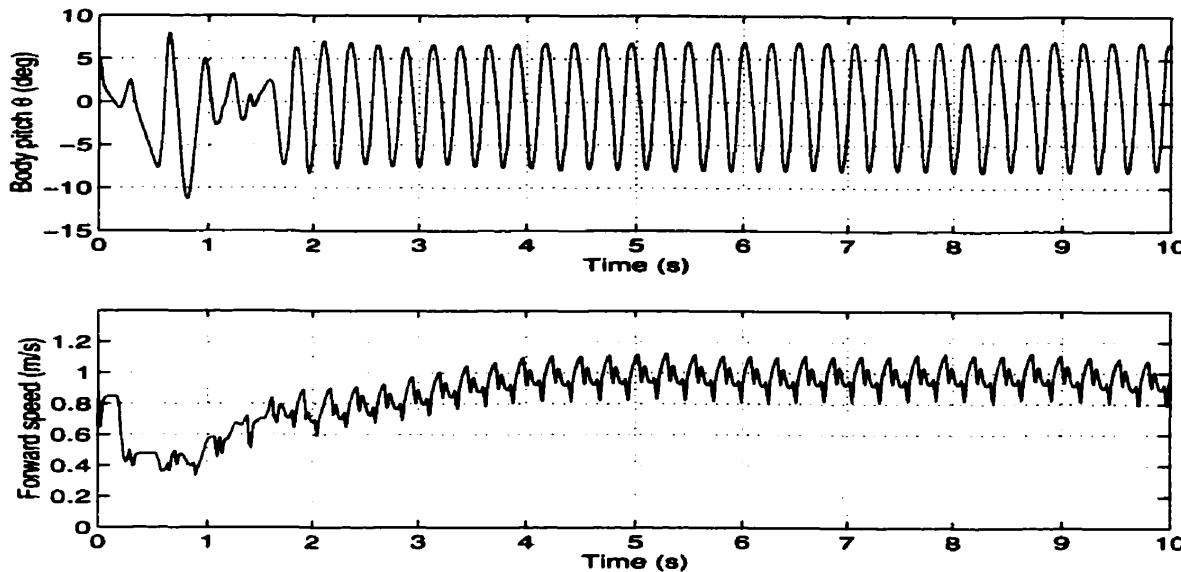


**Figure 2-21: Body pitch and forward velocity during running for a closed loop controller with experimental friction factor.**

The next step was to call for the slip prevention function. Figure 2-22 below represents the data recorded from the simulation for the body pitch and forward speed with the slip prevention function. It is clear from the plot that the robot now achieves a steady state running gait. The forward velocity and body pitch go through consistent and repetitive cyclic motions similar to the first case where no slip was present. The difference that can be noticed from when no slip was considered is in the magnitude of the body forward velocity. The velocity is less than the previous one by 0.2 m/s. This is

## Chapter 2. Analysis and Simulations

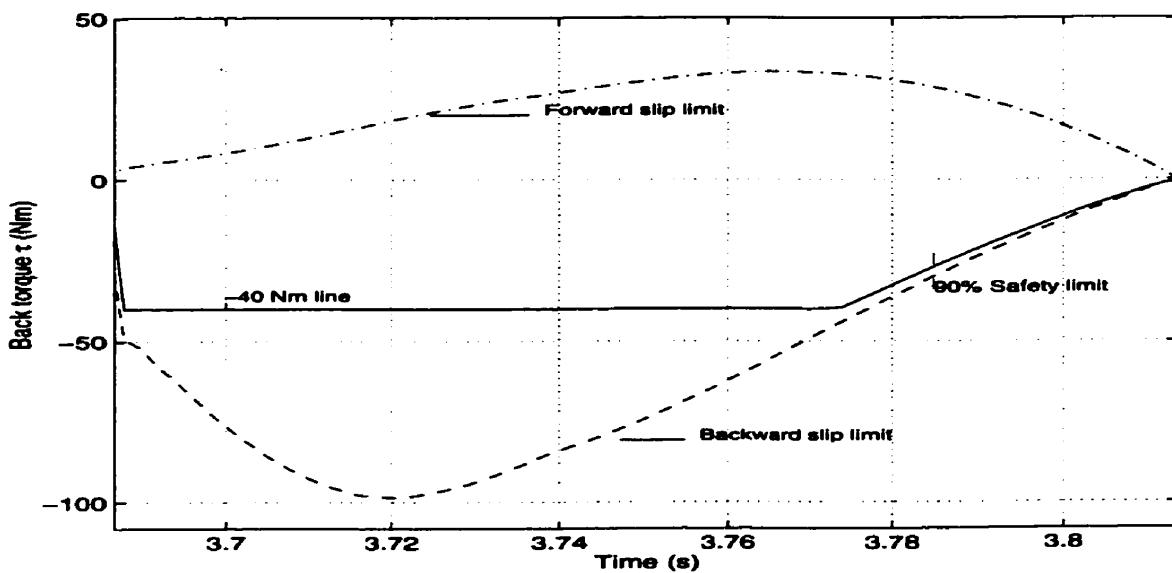
due to the torque reduction applied during stance and hence less energy input to the system. Figure 2-23 shows how the torque applied to the motor is clipped at certain times and hence it is less than the desired torque. The desired back torque being equal to -40 Nm in all previous simulations.



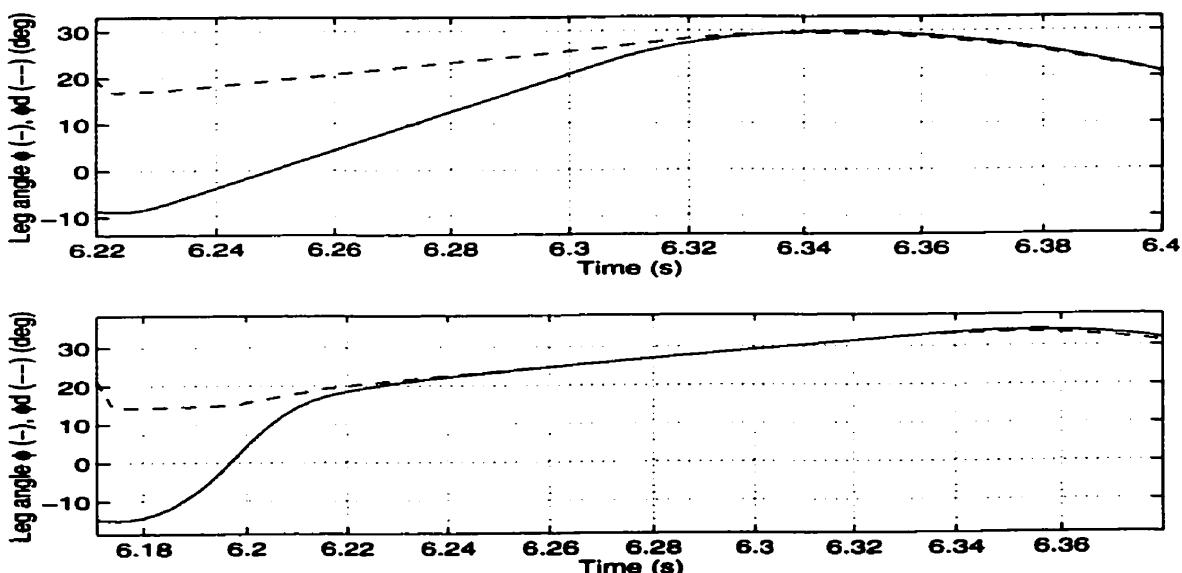
**Figure 2-22: Body pitching and body forward velocity with the slip prevention model implemented.**

The motor model was then implemented to ensure that the commanded torques lie within the motor operating range. Upon implementing the motor model, the leg tracking the set point during the flight phase was analyzed. The recorded data were plotted in Figure 2-24. The plots show that with the motor model, the leg response was slower and hence the rise time was twice as long as with no motor model. However, the final outcome of positioning the leg at the desired angle before touchdown was achieved in both models.

## Chapter 2. Analysis and Simulations



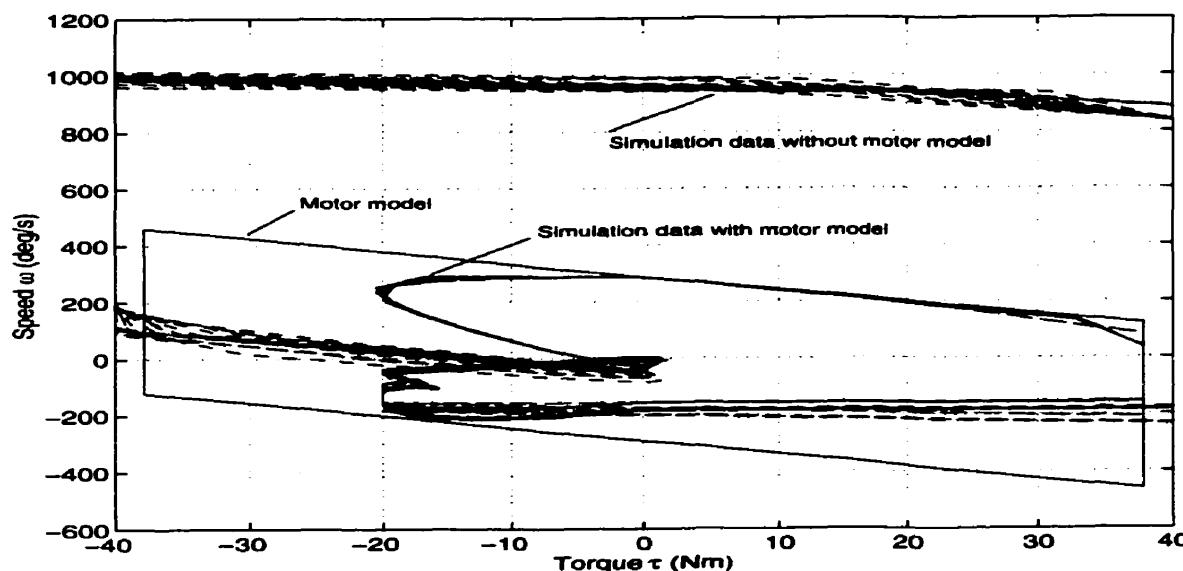
**Figure 2-23:** Torque plot during the back leg stance case, the continuous line represents the applied torque, the dashed lower line represents the backward slip torque limit and the dashed upper line represents the forward slip torque limit. A safety factor of 0.9 was used.



**Figure 2-24:** Top: leg angle tracking in simulation with no motor model implemented. Bottom: leg angle tracking in simulation with motor model implemented.

## Chapter 2. Analysis and Simulations

The other change was that the robot speed decreased by another 0.1 m/s from the 1.2 m/s desired velocity. The torque-speed operating data were recorded for both simulations with and without motor model. It can be seen in Figure 2-25 how the two operations are different especially during the flight phases, which is when the angular speeds are positive. While the angular speeds for the simulation with the motor model implemented are within the operating range shown in the figure by the parallelogram, those without the motor model are outside the allowable area for most of the flight phase.



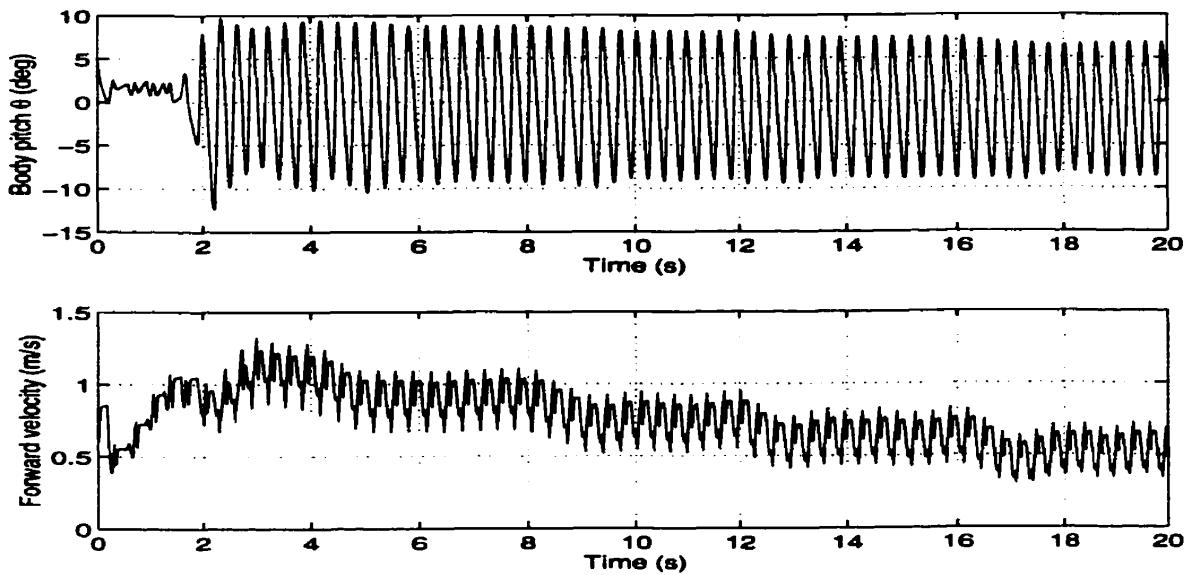
**Figure 2-25:** Torque-speed curve operation for the four quadrants, simulation with motor model (-), simulation with no motor model (—). The parallelogram is the motor operating range for the four quadrants. The torque-speed data from the running with the motor model is within the operating range. The data for the running without the motor model crosses the motor operating range.

### 2.5.3 Control of Forward Speed by Touchdown Angle

To improve control over the robot, the next step was to simulate the running with a controlled forward velocity. Since forward speed is constant in the flight phase, any acceleration must occur during the stance phase. It was explained in equation ( 2-32 ) that

## Chapter 2. Analysis and Simulations

an increase in the touchdown angle of the legs would reduce the forward velocity of the robot. This hypothesis is tested by having the previous simulation run with increasing leg angle offsets. Note that the energy put into the system is almost constant, the leg torque was tested with constant 40 Nm in the back and 10 Nm in the front. The step changes in the forward leg touchdown angle resulted in rapid changes in forward velocities, still maintaining a steady state run. The changes in forward velocities and body attitude are shown in Figure 2-26.



**Figure 2-26: Body pitch and robot forward velocity values for changes in desired touchdown angles.**

The figure shows how the robot velocity was controlled in steps of 1.1, 0.9, 0.7, 0.6 and 0.5 m/s. The body pitching oscillation remained stable, however the amplitude decreased with increasing offsets, and oscillation became asymmetric around the zero degrees pitching value.

The above demonstration can also serve to show that the controller can work under outside disturbances that could disable the legs from reaching the right touchdown angle. Variations in leg angle due to unexpected conditions can be rectified. The system is therefore robust enough to handle minor disturbances.

## Chapter 2. Analysis and Simulations

Another method of controlling the forward velocity is by adding more or less energy into the system during the stance period. This is done by controlling the desired forward velocity using the hip actuator torque.

### 2.5.4 Torque Velocity Feedback Controller

The closed loop controller with velocity feedback was modified to accommodate a torque controlled velocity during the stance phase. Instead of giving a constant torque during the stance phase, the hip actuators apply enough torque to keep the robot at a certain commanded speed. The torque controller is a simple proportional gain controller based on the current speed and the desired speed,

$$\tau = KP(\dot{x} - \dot{x}_{desired}) . \quad ( 2-35 )$$

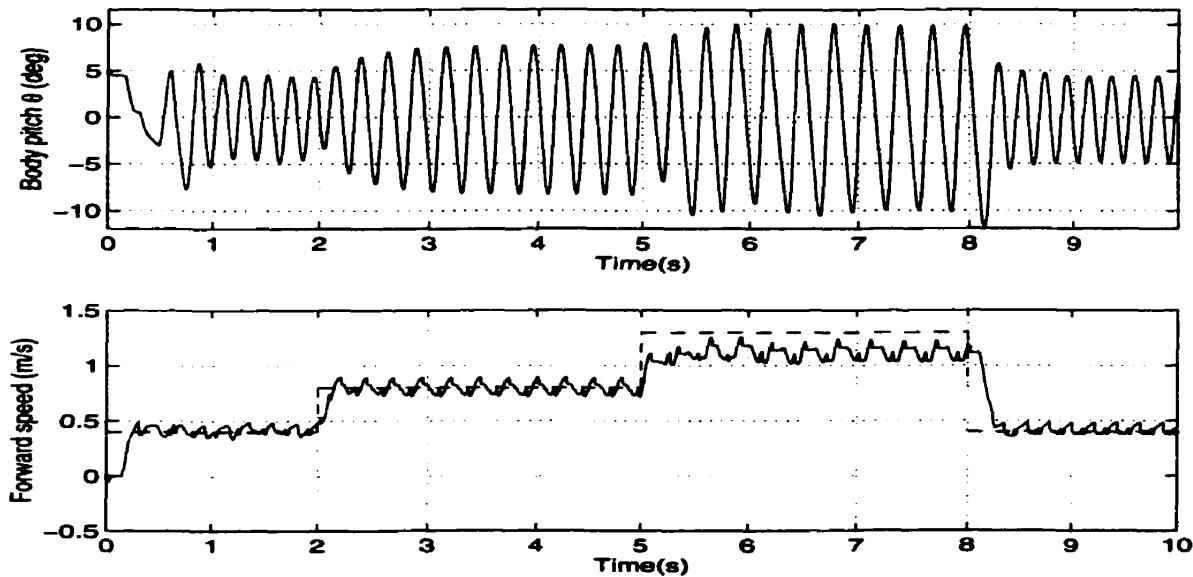
The simulation was tested with step changes in desired forward speed. Step changes of 0.4, 0.8, 1.3 and 0.4 were commanded at 0, 2, 5 and 8 seconds respectively. Figure 2-27 illustrates the results. The step changes were achieved while affecting the body pitching amplitude in the same way observed in section 2.5.3. Compared to the velocity controller based on leg touchdown angle, this controller has less fluctuation in forward velocity during one cycle. The torque velocity feedback controller also continues to achieve symmetric body pitching.

A closer look at the pitching frequency shows that it changes as the forward speed does. It can be seen that the stance time decreases with increasing forward velocity. This result was earlier studied by Ahmadi et al. [1]. In fact, if the robot is considered as a spring damper mass system, the stance period can be approximated by,

$$T_s \approx \frac{\pi}{w_l} , \quad ( 2-36 )$$

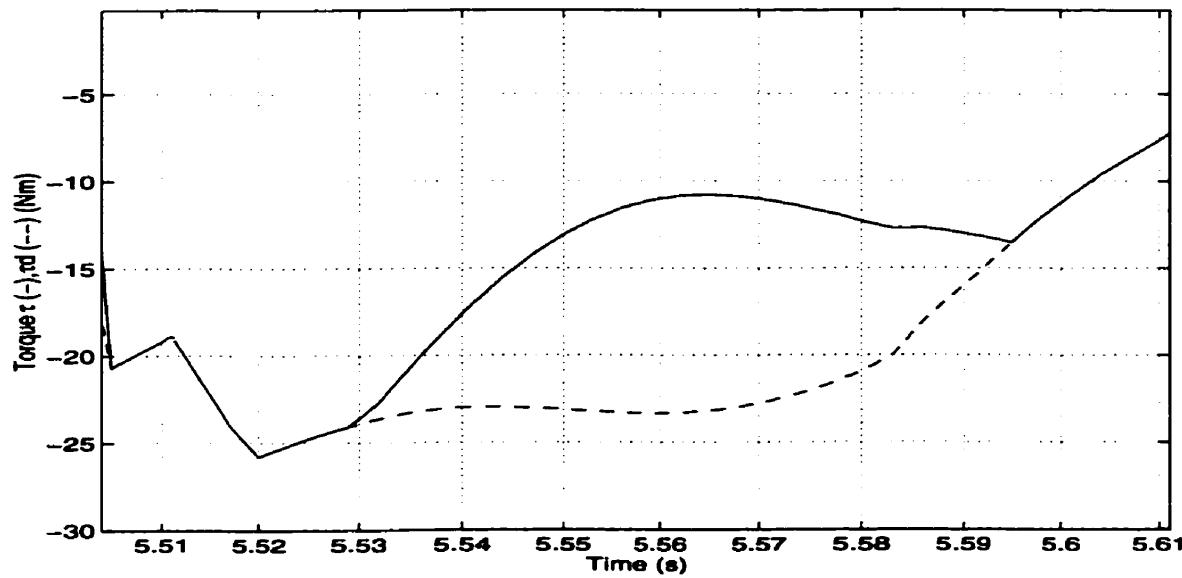
## Chapter 2. Analysis and Simulations

where  $w_i$  is the system vertical natural frequency. This however is only acceptable for vertical operations with zero forward speed. In this simulation, the robot has a stance period of 0.185 s at zero speed. This value drops to 0.137 s at a speed of 0.4 m/s, then to 0.128 s at 0.9 m/s, and 0.123 s at 1.1m/s.



**Figure 2-27: Step changes in forward velocities controlled by the hip actuator torque.**

Another result worth investigating is how the velocity tracking error increased as the desired robot velocity increased. When the desired velocity was 1.3 m/s it was noticed that the nominal velocity achieved was 1.1 m/s. The reason for that is the motor operating characteristics. As the robot desired forward speed increases, the proportional controller commands increasing torques. Unfortunately, as the motor's desired speed is increased the motor achievable torque is decreased as illustrated in the motor torque-speed curve. Hence the motor reaches the limit of the torque-speed curve, causing saturation. In Figure 2-28, the dotted line represents the torque calculated by the velocity controller, while the continuous line is the maximum motor achievable torque. There is therefore a limit to the achievable speeds dictated mainly by the motor characteristics.



**Figure 2-28: Torque limitations on the motor during the running cycle. The dashed line is the torque obtained from the proportional controller. The continuous line represents the torque achievable by the motor**

### 2.5.5 Jumping Over Obstacles

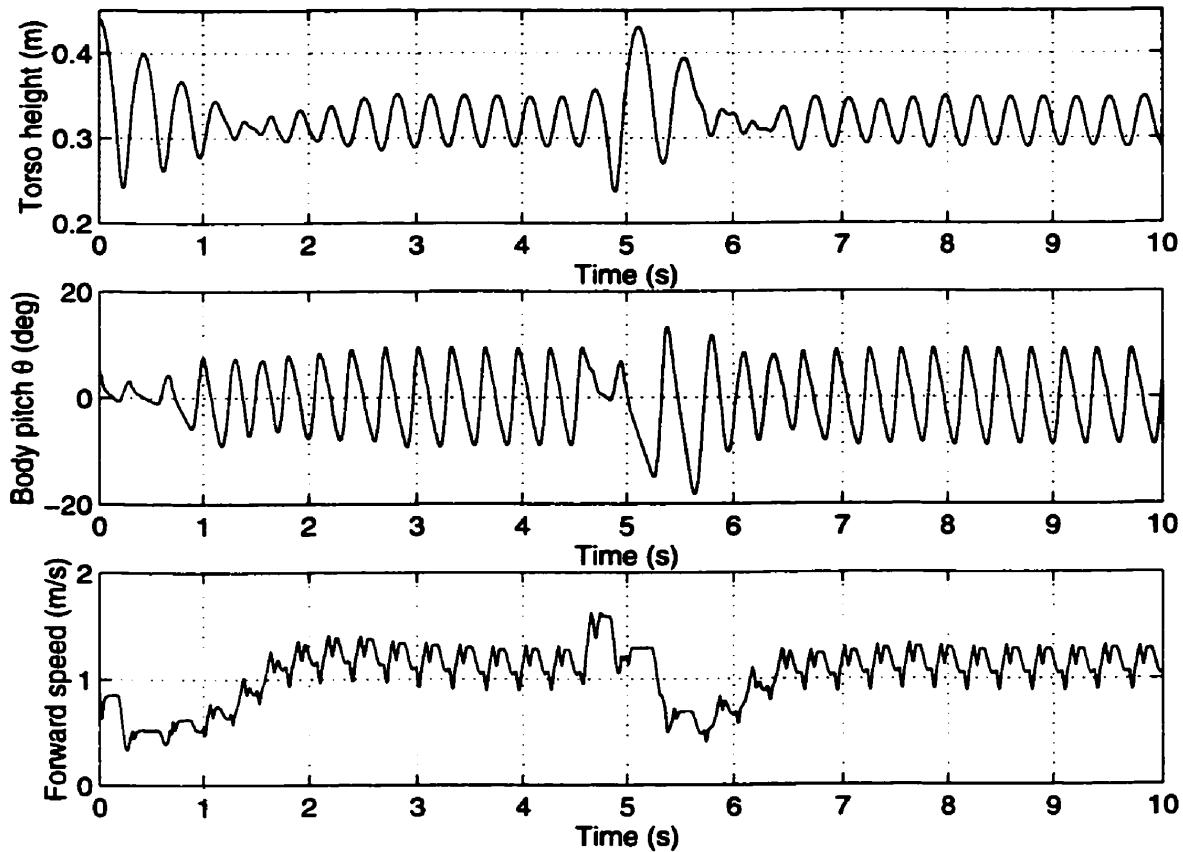
Because the leg angles are responsible for transferring the forward kinetic energy into a vertical energy, one can use them to have the robot cross obstacles. By increasing the touchdown angle the robot will store more energy in the springs. This energy can then lead to higher hopping heights.

$$C_{2,n-1} = \begin{cases} x_{\text{offset},f} = 0.09m \\ x_{\text{offset},b} = 0.05m \\ \tau_{f,s} = -20Nm \\ \tau_{b,s} = -50Nm \end{cases}$$

The jumping controller investigated here, is set so that in the running cycle just before a jump is desired, the leg offset angle and the stance torque are increased. This

## Chapter 2. Analysis and Simulations

simultaneously injects more energy in the system and stores a bigger portion of the forward kinetic energy in the springs. At the next cycle the robot will thus have a jump higher than the usual running height. Following the jump, the offset angle is returned to the initial value thus bringing the robot back to its steady state running. The values used during the cycle before the jump are those in  $C_{2,n-1}$ .



**Figure 2-29:** Running data for a simulation where the robot hopping height was increased to simulate obstacle crossing. The first figure represents the height of the center of mass of the robot. At the time of five second the jumping controller is used and the robot height increases by nine cm in the next step. The robot then converges to the steady state hopping height after five steps.

## Chapter 2. Analysis and Simulations

Figure 2-29 shows how the robot height was increased by 9 cm after the jumping controller was used. Although the forward velocity and body pitching are momentarily disturbed, these go back to their previous values within five running cycles. This also demonstrates that the controller can deal with large perturbations.

### 2.5.6 Closed Loop Controller with Pitch Control

The last controller implemented was a closed loop controller with intermittent velocity control and pitch control. Instead of using independent leg states to derive the controller, this controller uses the overall robot state. It is in some sense a one way controller in that the state switching is unidirectional. In other words, if the robot switched from back stance to flight after back (see Figure 2.2), then the next state expected is the front stance, if the robot goes to any other state then the controller will not recognize the state and hence the robot will crash.

In addition to the four robot states shown in Figure 2-2, two extra states were introduced. Each of the stance phases was divided into two parts, the first one is loading and the second one is unloading. The two states refer to the robot's leg behavior. During the loading phase, the leg shortens while the robot vertical speed slows down. In the second phase of stance, the robot leg extends and the robot accelerates.

In the loading phase of running, the motors control the forward velocity of the robot with a proportional gain controller similar to ( 2-35). During the unloading phase, the motors first control the robot forward velocity and secondly the robot body pitching attitude. As shown in (2-37), the body pitching attitude is made to follow a cosine wave with a desired amplitude  $A$ , and using a cycle time  $T_{sf}$  , obtained from the previous stance time. The total portion of the stance phase dedicated to the forward velocity control is here denoted by X. This was an attempt to introduce some pitch control to reduce the amplitude of the oscillatory motion. Table 2.5 below summarizes the robot running states for this controller.

## Chapter 2. Analysis and Simulations

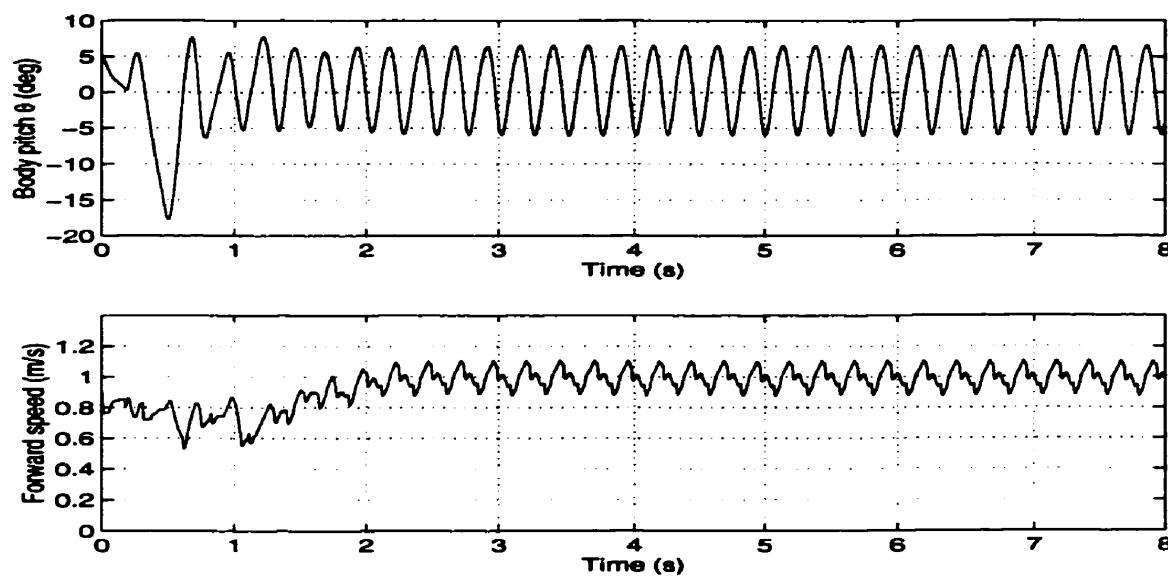
$$\left\{ \begin{array}{l} \theta_d = A \cos(w_{lf}t - \frac{\pi}{2}) \\ w_{lf} = \frac{\pi}{2T_{sf}} \end{array} \right. \quad (2-37)$$

State	Condition	Action
Back to front flight	Back leg lift-off	Front leg: position leg for touchdown Back leg: zero torque
Front leg loading	Front leg touchdown	Front leg: velocity control Back leg: position leg for touchdown
Front leg unloading	Front leg decompressing	Front leg: velocity control for X% of state, pitch control for (1-X)% of the state Back leg: Position leg for touchdown
Front to back flight	Front leg lift-off	Front leg: zero torque Back leg: Position leg for touchdown
Back leg loading	Back leg touchdown	Back leg: velocity control Front leg: position leg for touchdown
Back leg unloading	Back leg decompressing	Back leg: velocity control for X% of state, pitch control for (1-X)% of the state Front leg: Position leg for touchdown

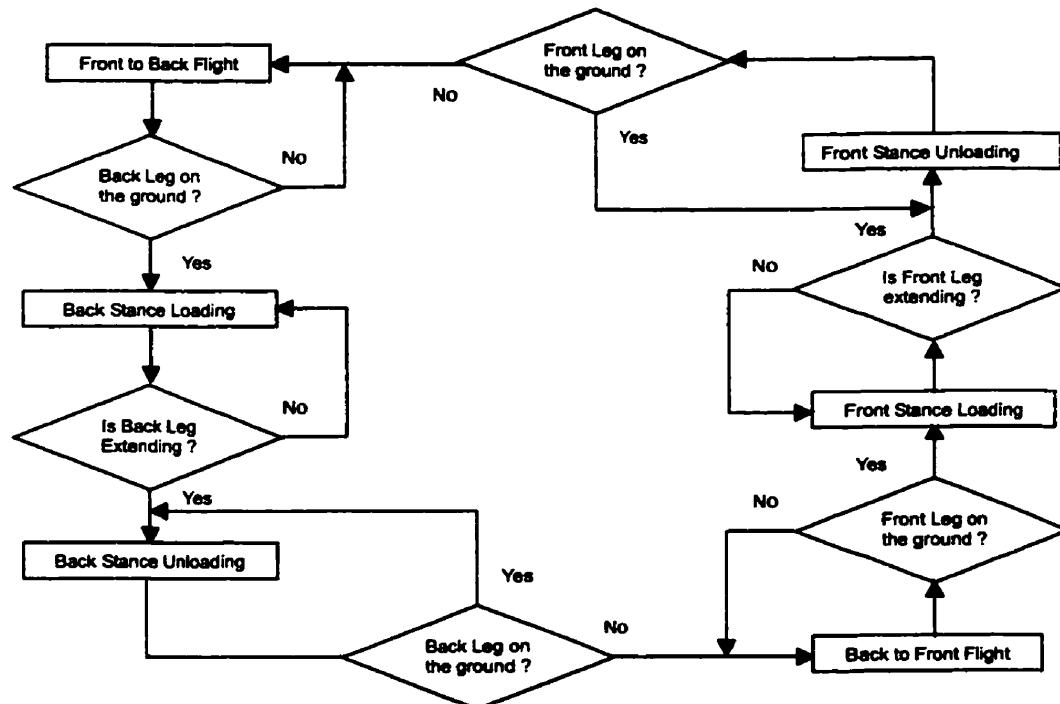
**Table 2.5: Robot overall running state for the closed loop running controller.**

The running simulation was run with a desired forward velocity of 1.2 m/s and a maximum pitching amplitude of 5°. Figure 2-30 below illustrates the running behavior with X=0.7. The body pitching was close to the desired amplitudes while the body forward velocity had a nominal error of 0.25 m/s. Body pitch control thus has the disadvantage of poor forward velocity tracking when compared to the closed loop controller with velocity control only.

## Chapter 2. Analysis and Simulations



**Figure 2-30: Body pitch and forward velocity for closed loop running controller with speed and pitch control.**



**Figure 2-31: Flow chart of the running cycle**

### 2.5.7 Turning

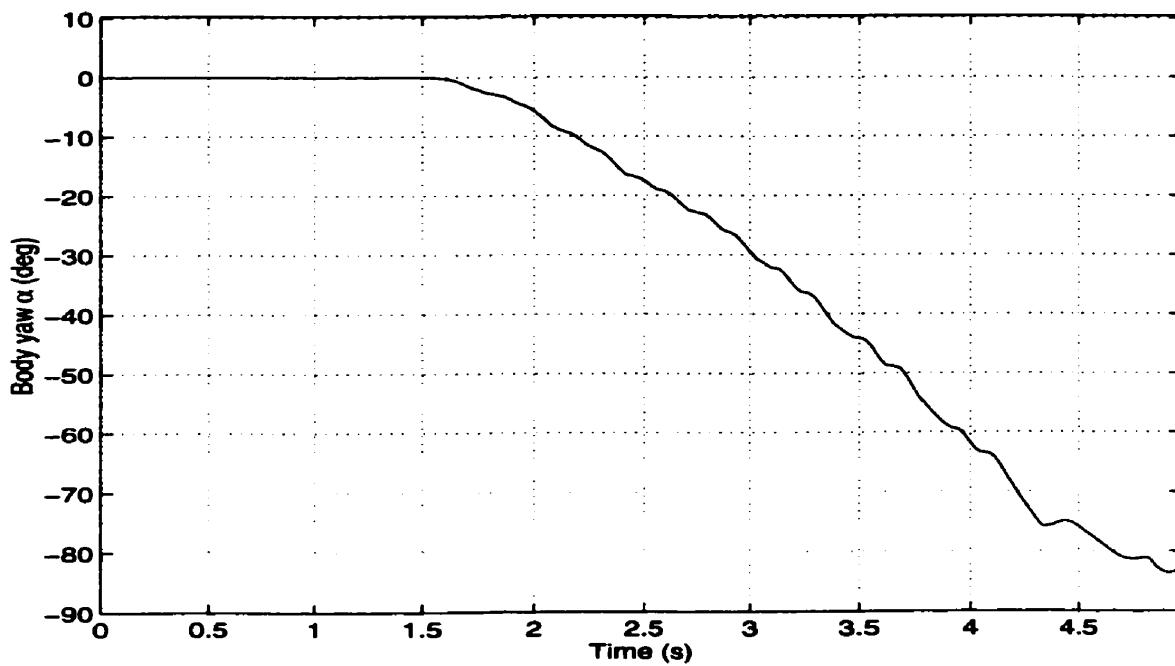
The running simulations discussed in the previous sections constrained the motion to the sagittal plane. For the robot to function in a wider range of settings it must be capable of changing its direction.

In the case of running in the sagittal plane, a two dimensional simulation software is adequate. To study turning behaviors, a full three dimensional software package had to be used. Working Model 3D® [22] was used for this purpose. Working Model 3D® is similar to the 2D version in that simulations can be controlled via a script written in Visual Basic. The Scout II model is built using the interactive Working Model 3D® tools while the running script was written in the Microsoft Visual Basic Excel Editor[23].

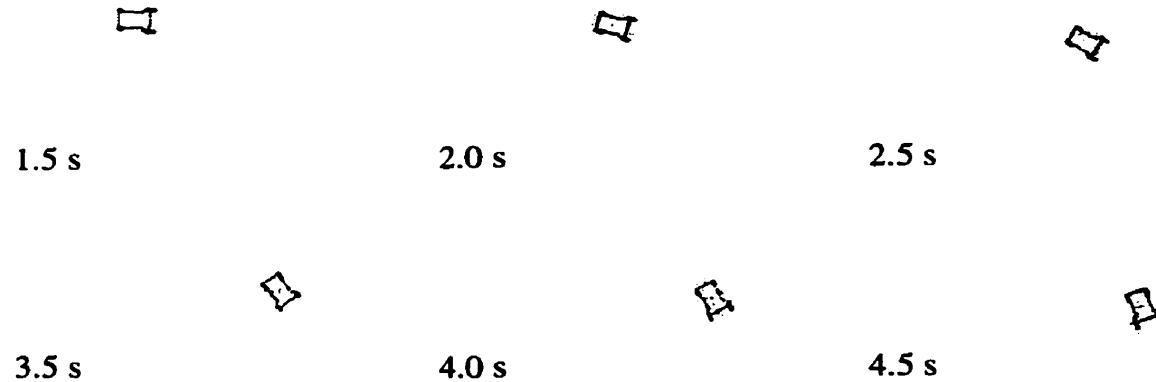
The turning is a simple modification to the open loop controller. The idea is to apply differential torques to the left and right sides of the legs during the stance phases. This causes a moment in the vertical direction, forcing the robot to turn. If the robot is to turn in the left direction, the torque on the right set of legs is made larger than the left one by a certain factor greater than one.

A simulation is run where the torques during the stance phase are increased by 50% for the left side legs. As shown in Figure 2-32 and Figure 2-33, the robot made an approximate ninety degree turn after which the body rolling oscillation became large enough to destabilize the bound. Nevertheless the simulation showed that it is possible to make the robot turn using this simple algorithm.

## Chapter 2. Analysis and Simulations



**Figure 2-32: Body yaw during the turning algorithms. The running direction is turning to the left.**



**Figure 2-33: Top view of the turning during the three-dimensional running simulation. Figure is read from left to right.**

### 2.6 Conclusion

In this chapter, the equations of motion of the system were derived for the flight and stance phases of running. These were then compared to the simulation to assess the mathematical model.

A simulation model of the motor operation was then developed and implemented in Working Model 2D®. The interaction between the robot's toes and the ground was then addressed, and different techniques for preventing, detecting and recovering from slip were proposed and tested.

Then, various running controllers were developed and successfully simulated. An open loop controller with fixed hip angles at touchdown and fixed motor torques during the stance phase was first investigated. This controller has the advantage of using the minimum sensing for the control algorithm. However it lacks explicit control over running parameters such as forward speed and body pitching. Further more elaborate closed loop controllers based on the center of mass travel were implemented and tested. The control over the running speed of the robot was achieved with different approaches. In addition, the possibility of jumping over obstacles was addressed. The advantage of the closed loop controller is the effective control over the robot forward velocity, which does not require explicit control over the leg angle at touchdown. The controller alone calculates the appropriate touchdown angle based on the forward speed of the robot. This controller also recovers from considerable disturbance within a few steps as seen in the jumping attempt. The closed loop controller however requires sensing of a few parameters such as body pitch, forward speed, and stance time, which makes it more difficult to implement. An attempt to build a closed loop controller based on the overall robot state was done with some combined body pitch and forward velocity control. This controller was able to reduce the body pitching during running but on the expense of a loss in forward velocity. Finally, 3D running with turning behavior was simulated in Working Model 3D®.

# **Chapter 3**

---

## **Experimental Results**

### **3.1 Introduction**

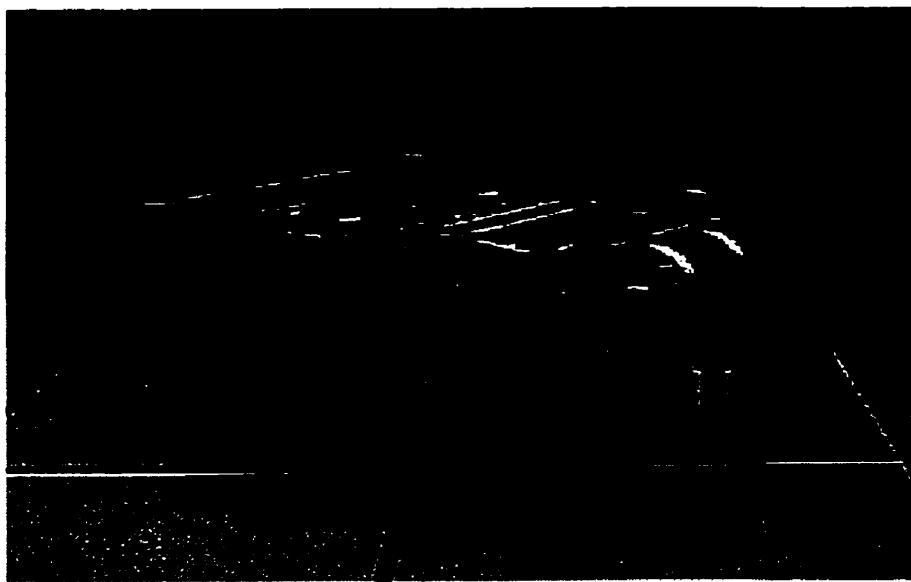
This chapter presents the results obtained upon the implementation of the running controller on the Scout II robot. Section 3.2 describes the experimental setup including the sensing and data collection method on Scout II. Section 3.3, analyses the results obtained upon the implementation of the open loop controller, the closed loop controller and a pronking gait on Scout II. Section 3.4 presents the results for the turning. Finally, the auxiliary controllers for lying down and getting up are discussed and presented in section 3.5.

### **3.2 Running Setup**

The Scout II robot is equipped with a suite of sensors. Two laser range sensors attached to the front and back of the robot allow measurement of distances to the ground and hence body pitch attitude and elevation of the body's center of mass. A MURATA® solid state gyroscope is mounted as a backup sensor, in the event that the laser sensors go out of range at a maximum sensing distance of 44 cm. Torque measurement can be calculated from the current readings from the motor amplifiers. To measure leg lengths, linear potentiometers are mounted inside each leg. These allow measurement of up to 10

### Chapter 3. Experimental Results

cm in displacement. They are also used for the sensing of the leg flight and stance states. The leg is considered in flight whenever the potentiometer measures the leg length to be equal to the rest length. This can also differentiate between the loading and unloading parts of the stance phase. An optical encoder is attached to each motor to read the angle of the leg. The hip angular speeds and accelerations are obtained by differentiating the hip angles. Since the encoders are incremental, the hip angles are automatically calibrated at start-up by moving the legs past a known angle, marked by Hall effect sensors.



**Figure 3-1: Scout II**

The SPP/SPI system mounted on Scout II was used to collect the sensory data during running and send it to the on-board PC. This data is then used for the necessary computations and also stored in the computer's memory. At the end of an experiment, the stored data is retrieved to analyze the experimental results.

The robot is completely autonomous, powered by two 12 V batteries, located at the front and the back of the robot to maximize the body's inertia. During the experiments, no power or communication cables were attached to the robot.

## Chapter 3. Experimental Results

A difference between the actual robot and the simulation model is that in the simulation, the front and back leg pairs were treated as one leg. In the real-life situation, the legs had to be synchronized to operate as one. A function that synchronizes the pair of back legs and the pair of front legs was introduced. The synchronization function applies a PD controller to the left and right legs so that they always work together. If one leg is moving faster than the other, less torque is applied to it. This prevents the fast leg from surpassing the other. The proportional and derivative gains have been tuned during the preliminary experiments. The amount of torque reduction is applied to the "faster" leg via ,

$$\Delta\tau = K_p \Delta\phi + K_d \Delta\dot{\phi} , \quad (3-1)$$

where  $\Delta\phi$  and  $\Delta\dot{\phi}$  are the differences in left and right hip angles and angular speeds,

Initial running implementation attempts as well as previous work done on Scout II pointed out the presence of toe slip during stance phases. Implementation work also demonstrated that toe slip had negative results on the running behavior such as the loss of most of the body angular momentum [10]. For the purpose of slip prevention, the same slip torque approximation used in section 2.4.3 is employed. In this function, the torque commanded to the legs is continuously checked to an approximate torque limit according to equation ( 2-19 ) or ( 2-25 ). If this torque limit is exceeded the robot leg is expected to lose grip on the ground. Thus, the function reduces the commanded value to within the safe region, whenever the desired torque exceeds the limit

Because of the approximation approach used in the slip prevention and the changes in ground properties, it is impossible to anticipate slippage in all cases. For that reason a slip detection and correction function was implemented. In our controller, the hip angular acceleration is monitored during the stance phase. If at any point a rapid change in leg angular velocity is detected by the encoders, slip is recognized and the function reduces the torque momentarily. The method used to detect slip is thus equivalent to the first technique introduced in section 2.4.3. Because of the noise obtained after two

differentiations of the hip angular speeds, the second method, which relies heavily on sensing many of the robot parameters, was not used. This torque reduction slows down the leg and reduces the horizontal force of the foot on the ground to a level that is within the friction cone.

### 3.3 Running Implementations

In this section, the results obtained from the experiments with the open loop controller and closed loop controller are presented. Two types of running were observed in experiment: bounding and pronking. The bound is compared to the simulation results obtained in section 2.5.

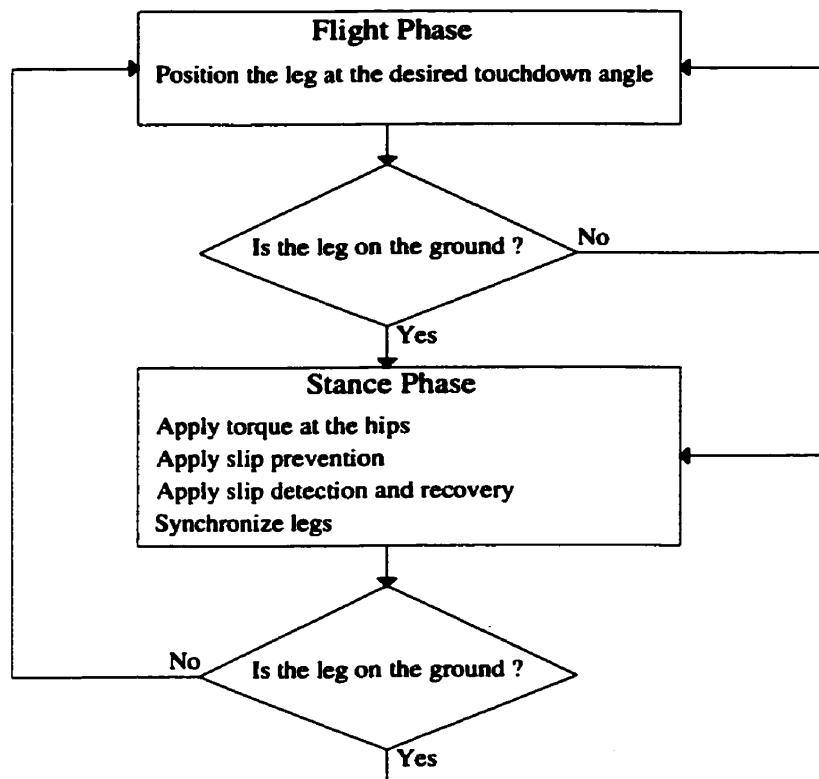
#### 3.3.1 Bounding Gait

As suggested by Working Model 2D® running simulations, it is possible to achieve a steady bounding gait by choosing appropriate set of constant motor torques and leg touchdown angles. The open loop running controller was implemented on the Scout II robot. A back torque of 35 Nm per leg and a front torque of 10 Nm per leg was used before the 60 % average motor efficiency is considered [4] [16]. After efficiency is taken into account the torque should match the values used in the simulation. A touchdown leg angle of twenty-two degrees for the front legs and eighteen degrees for the back legs was commanded for flight phases.

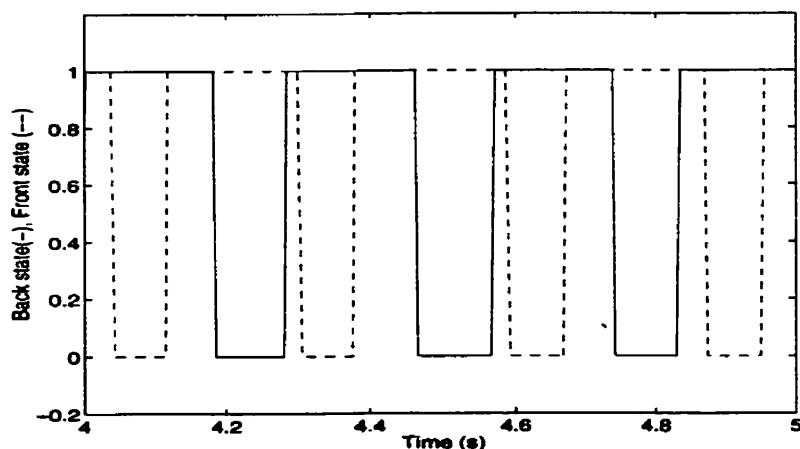
The slip prevention of section 2.4.3 was implemented on both simulation and experimental data. The only difference in the experimental slip approximation function is that it dealt with each of the front legs or back legs independently.

The architecture of the controller implemented on Scout II is summarized in Figure 3-2. It resulted in a steady state running gait with parameter values comparable to those observed in simulations.

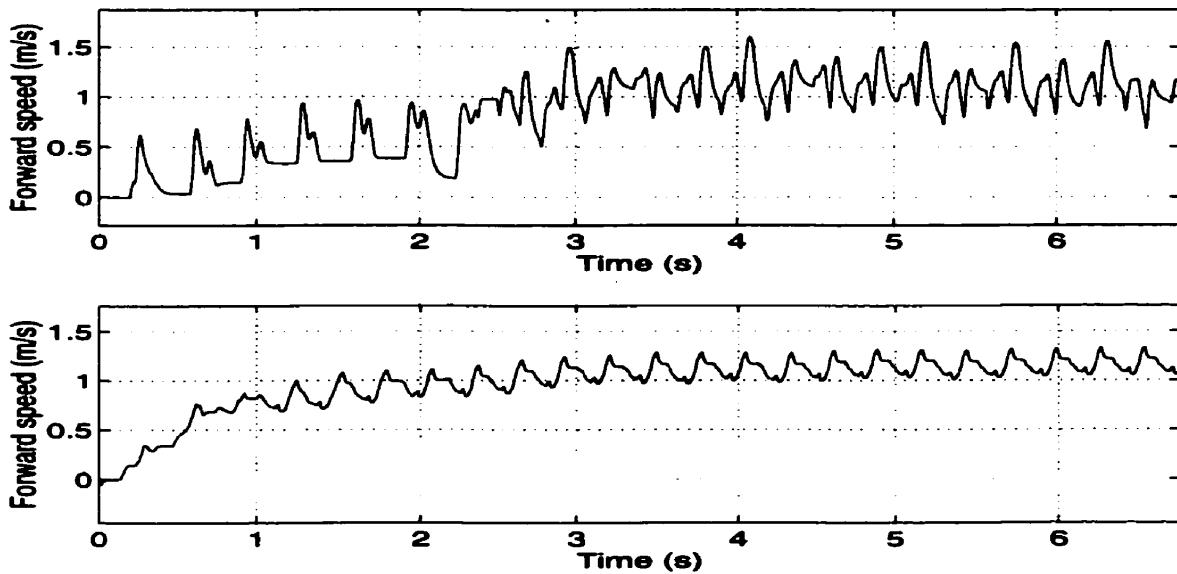
## Chapter 3. Experimental Results



**Figure 3-2: Experimental open loop running control cycle implemented on each set of legs.**



**Figure 3-3: Back legs and front legs states for the open loop control. The continuous line is equivalent to the back state. The dashed line is equivalent to the front state. A value of one represents the flight phase, and a value of zero represents the stance phase.**

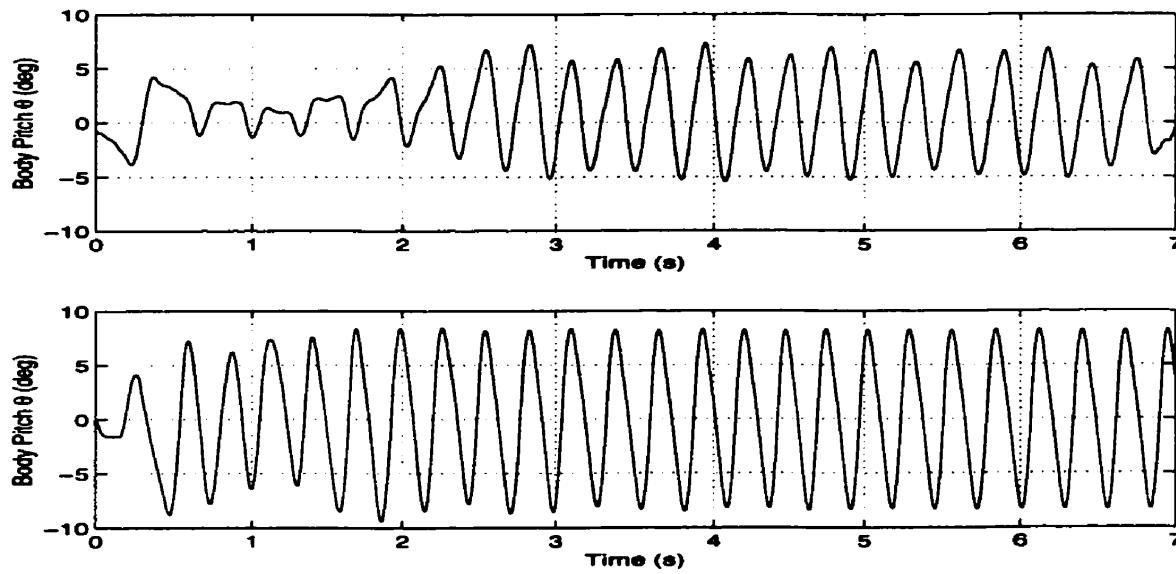


**Figure 3-4: Comparison between forward running speeds in simulation and experimentation.**

Both simulation and experimental runs shown in Figure 3-4 started at zero initial speed and accelerated until steady state speeds were achieved. The simulation and experimental data show a response time of 3.0 s. Both speeds reach a steady value of about 1.2 m/s. Fluctuation in the forward speed of the simulation is smaller. The reason for this discrepancy is due to the poor forward velocity sensing in the experiment.

The backlash in the motor, the gearbox, and the belt transmission was of the order of several degrees, which made the forward velocity data obtained from equation ( 2-27 ) quite unreliable. In addition, the values of  $\dot{l}$ ,  $\dot{\theta}$ , and  $\dot{\phi}$  were obtained by differentiating  $l$ ,  $\theta$ , and  $\phi$  in real time. The differentiation of these terms and the electrical noise in the system added to the errors in calculating the forward velocity. Although a second order low pass filter was implemented to diminish the errors, it could not be used extensively due to the sensitivity of the subsequent controllers to the delay created by the filtering. Note that this controller does not use feedback of the forward velocity, it is therefore not affected by the poor velocity signal. However the closed loop controller described in section 2.5.2 relies heavily on the velocity sensing.

## Chapter 3. Experimental Results



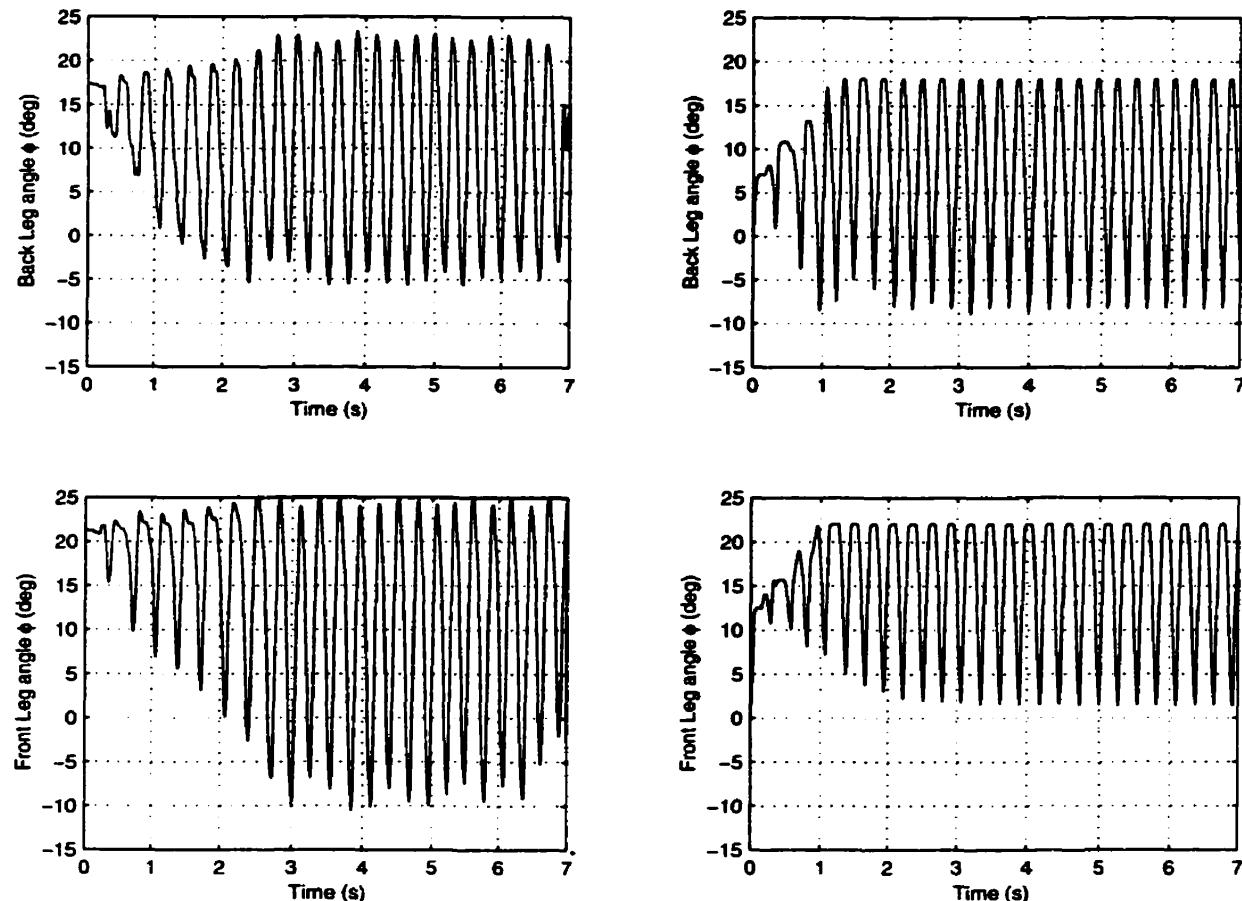
**Figure 3-5: Comparison between body pitching in simulation and experimentation.**

The two plots in Figure 3-5 show a comparison of body pitch for the experimental run and the simulation. After steady state is reached it is noticed that body oscillation is of the same magnitude. Simulation amplitude is  $8^{\circ}$  and symmetric about the horizontal line, while the experimental pitching oscillation changes between  $7^{\circ}$  and  $-5^{\circ}$  degrees. The cycle time for body oscillation is similar in both graphs. It has a value of 0.29 s per oscillation cycle.

The leg angles for both simulation and experiment resulted in comparable values during the flight phase. The experimental leg angles overshoot the desired value by approximately four degrees before touchdown occurs. This can be seen in the left graphs of Figure 3-6, for both back and front leg angles. This outcome is due to the leg positioning controller used during the flight phase. Note that simulation runs didn't consider friction in the hip joints or variation in the motor efficiency, which could have contributed to this slight positioning error. The plots in Figure 3-6 show that the back legs travel an angular distance of  $28^{\circ}$  in both simulation and experiment during the

### Chapter 3. Experimental Results

stance phase, but the limits are different. The front legs angular displacement however differs by about  $10^{\circ}$  more for the experimental results.

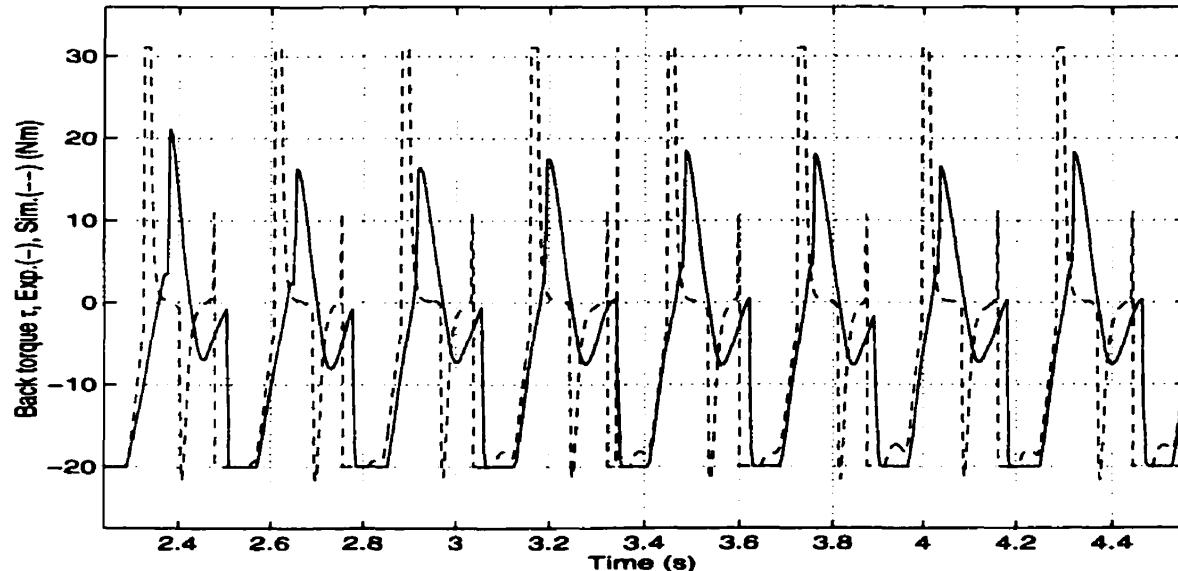


**Figure 3-6: Comparison between leg angular displacement in simulation and experimentation. Top figures are the back leg angles, and bottom figures are the front leg angles. The figures on the right are the simulation data, and those on the left are the experimental data.**

Figure 3-7 shows the torques for the back legs of the simulation and the experiment. This plot shows the nice repeatability of the data in both sets of results. A closer look at the torque changes during one cycle is shown in Figure 3-8. It shows that during the stance phase the torque plots are similar in shape. The square wave in the lower section of the graph represents the leg states. A low value represents stance and a

## Chapter 3. Experimental Results

high value represents flight. During the flight phases both simulation and experimental torques also have comparable shapes. Two torque peaks observed in the flight phase differ in simulation and experiment, which could be related to the PD leg positioning gains. As previously stated, the feedback of the hip angles and angular speed on Scout II included slight errors due to backlash in the system and due to differentiation.



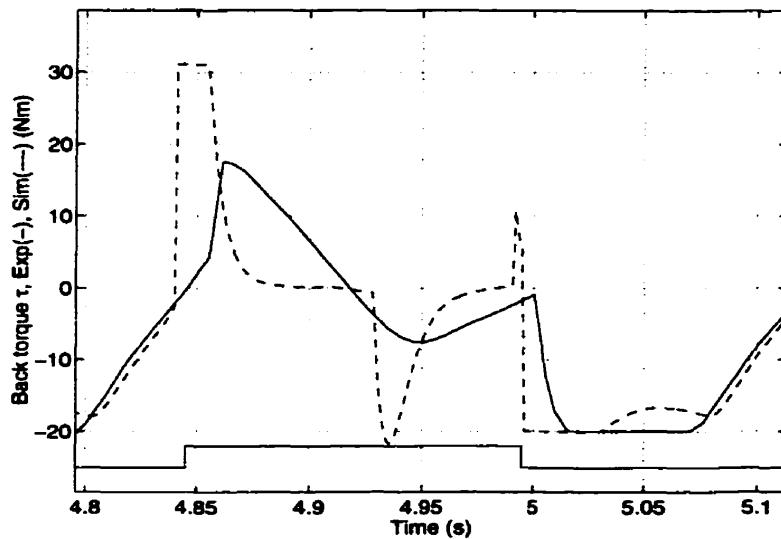
**Figure 3-7: Plot of back legs torque for simulation and experimentation.**

The continuous line represents the torque from the experimental run. The discontinuous line is the torque in simulation.

Thus, tuning the proportional and differential gains in the experiment resulted in experimental gains that are different from the simulation. This difference accounts for the discrepancy in experimental and simulation torque values during the flight phase. Again, the experimental motor plots in this graph are averaged to 60% efficiency. Averaging the torque efficiency also introduces slight errors in the data. On Scout II, the torques supplied by the motors are assumed proportional to the current feedback reading. This assumption is of course imperfect since it doesn't take into account different inefficiencies at different motor speeds. It assumes that the commanded torque is achievable. In addition, this doesn't reflect that some of the torque, especially during

## Chapter 3. Experimental Results

transients, is a dynamic effect of torque being used to accelerate the motor rotor inertia, the gear inertia, and the leg inertia. This torque does not contribute as a propulsive force. Nevertheless, the torques in both simulation and experimental run have the same patterns. Due to the nature of this controller that does not restrict transitions from one state to another, the errors above mentioned contribute sometimes to variations in the duration of the states, sometimes even vanishing one of the flight phases.



**Figure 3-8: Close up of torque change during one cycle**

### 3.3.2 Pronking Gait

A running gait that is less commonly used than the bounding gait is the pronking gait. Animals such as deer use this gait. In this gait all four legs work in unison - they touch down and lift off simultaneously.

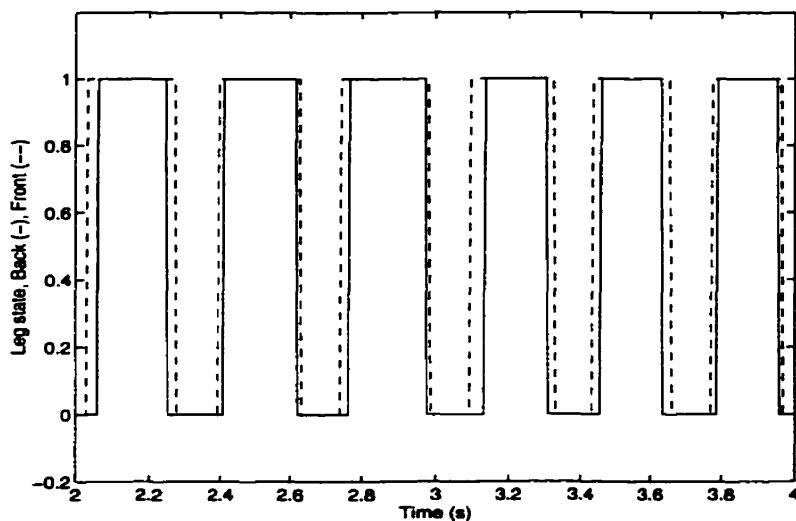
It was found while experimenting with the bounding gait that when the desired front leg angles were made smaller than the back leg angles at touchdown, the bounding gait converges toward a pronk gait. The Robot running gait becomes comparable to the monopod behavior in that all four legs operate in unison and effectively behave as one.

As seen in Figure 3-11, pronking was implemented with the front legs angles equal to  $16^{\circ}$  and the back legs angles equal to  $22^{\circ}$ . The back motors commanded torque was 35

### Chapter 3. Experimental Results

Nm for each motor while the front torque was 10 Nm each. These are the same fixed torque values as the ones used for bounding. As in the bounding case, the slip prevention function was implemented.

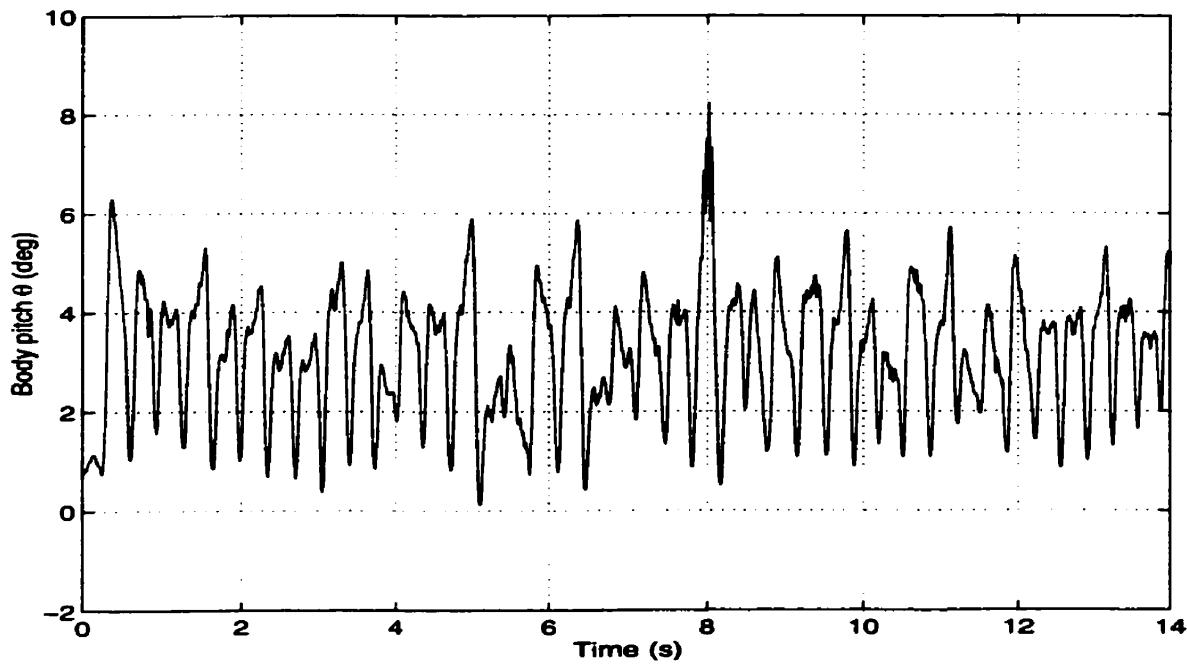
Figure 3-9 shows time intervals where both front and back legs are in flight. The stance periods are overlapping which implies that the running was a pronk gait and not a bounding gait.



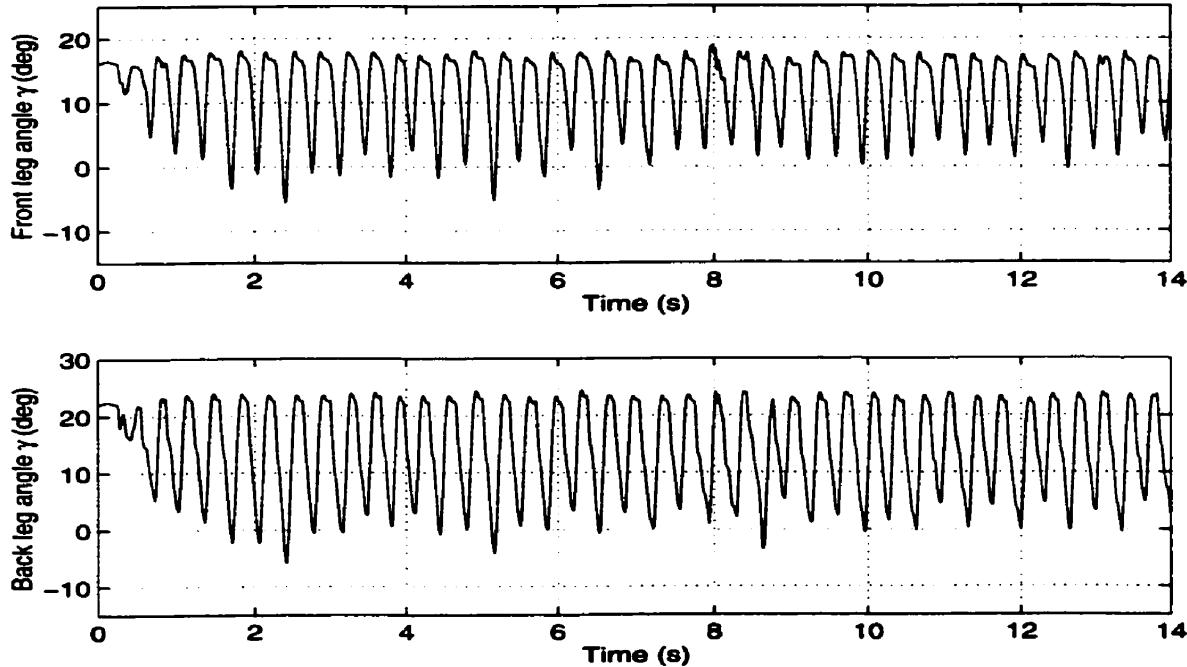
**Figure 3-9: Back and front legs states.** A reading of zero means the legs are in contact with the ground, while a value of one implies that the legs are in the air. The continuous line represents the back legs state. The dashed line represents the front legs state.

As shown in Figure 3-10, the body pitch angle is always positive. Pitching cycles are not periodic, as is the case in the bound. One cycle is now longer and equal to 0.33 seconds. The amplitude of oscillation is less than the one in the bounding gait. This type of running gait might prove more useful in situations where the robot is intended to transport objects sensitive to body pitching. Further study can be conducted in the pronk gait to find the optimum running configurations that will result in the smallest amount of pitching amplitude.

## Chapter 3. Experimental Results



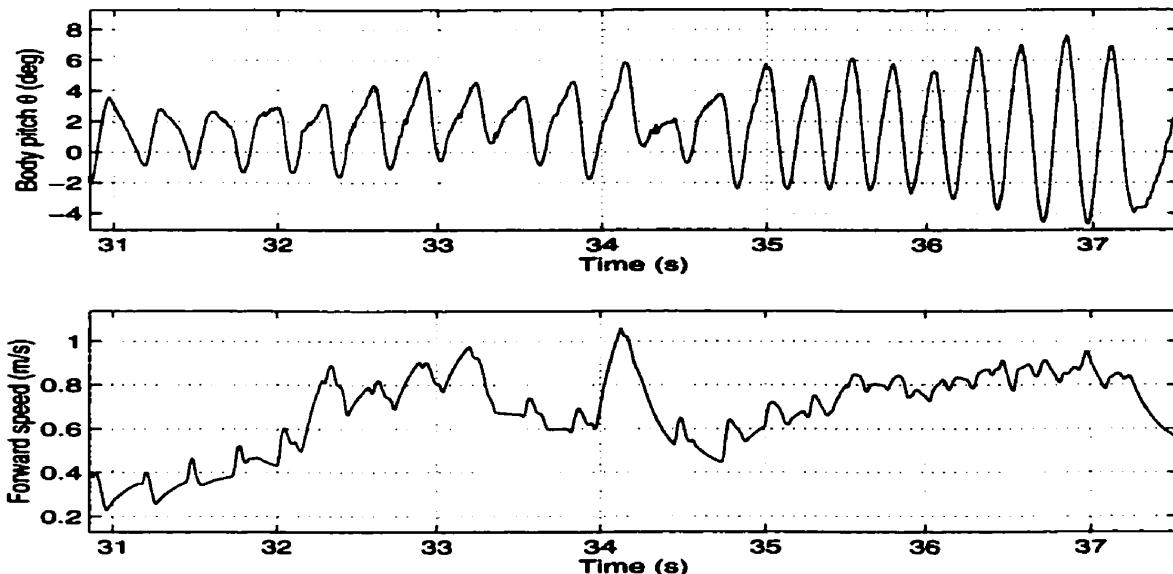
**Figure 3-10: Body pitching throughout the pronking gait.**



**Figure 3-11: Leg angular changes for the pronking gait. The upper graph has the trajectory of the front legs. The lower graph represents the trajectory of the back legs.**

### 3.3.3 Closed loop Controller with Velocity Feedback

The running controller with velocity feedback proved to be robust in simulation. The next step therefore was to test it on the actual robot. The robot applies a desired constant torque during the stance phase and controls the leg angles for touchdown according the relations shown in Figure 2-19 and equation ( 2-32 ).

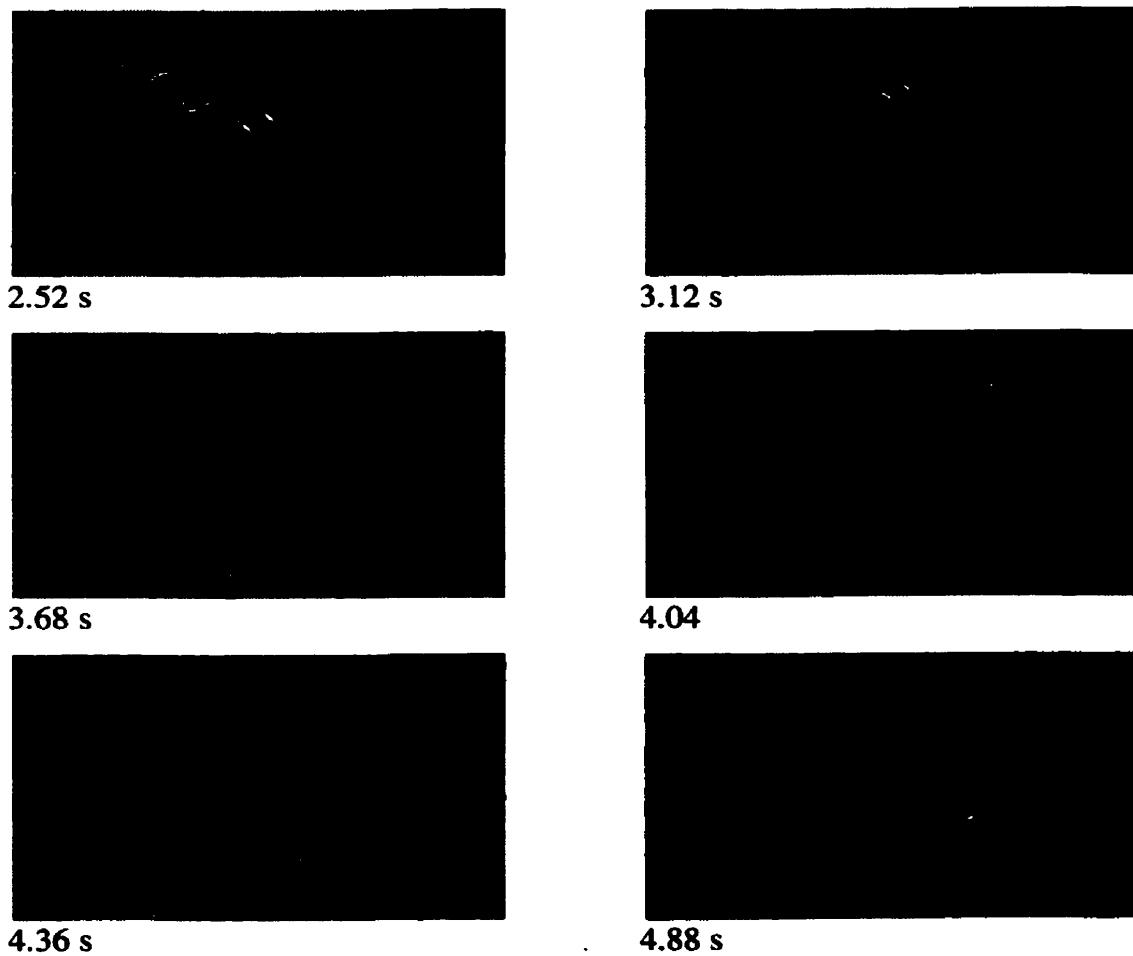


**Figure 3-12: Body pitching and filtered forward velocity during the running experiment with the closed loop controller.**

The controller resulted in the robot running continuously. However as shown in Figure 3-12, the results were not as steady as in simulation. This is due to the poor forward velocity sensing described in sections 3.2 and 3.3.1. Since the desired leg angles at touchdown are dependent on the velocity reading, this poor sensing had an adverse effect on the performance of this controller, sometimes leading to inappropriate touchdown. Nevertheless, the robot was able to recover from these disturbances and continue the running cycle. It is hoped that further improvement of the forward velocity sensing on Scout II will lead to a better running cycle.

### 3.4 Turning

In this section the results obtained from implementing the turning algorithm are presented. It was shown in simulation that by applying differential torques between the left and right side legs, the robot could turn in either left or right directions while bounding. Experimental results in Figure 3-13 show that it is possible to achieve a ninety degree turn in 11 steps. Moreover, the time necessary to achieve turning is comparable to the simulation result. The turning shown is done by applying 50 % more torque on the right side of the robot than on the left.



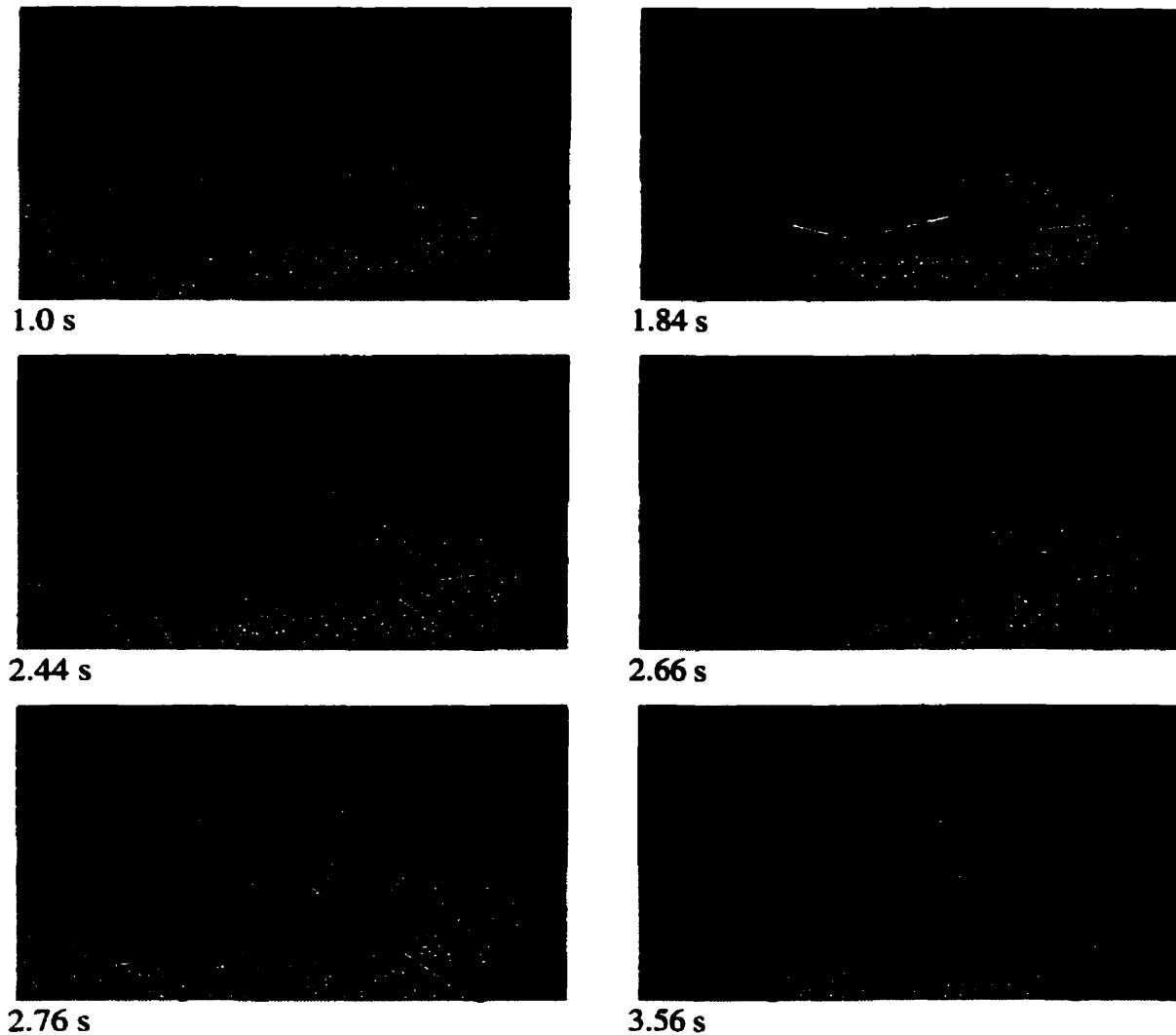
**Figure 3-13: Ninety degrees turning experiment. Figure is read from left to right.**

### 3.5 Other Behaviors

Other than running and turning the robot is also capable of accomplishing other behaviors. Experimental trials have been done on Scout II to implement sitting down and standing up behaviors. The following is a description of Scout II standing and sitting down.

In the standing up shown in Figure 3-14, Scout II starts by homing his legs. To home the legs, the hip actuators rotate the front legs in the clockwise direction and the back legs in the counter-clockwise direction until the Hall effect sensor for each of the legs is triggered. This corresponds to  $\phi_f = -90^\circ$  and  $\phi_b = 90^\circ$ . On the next step, both front and back legs push down applying pressure on the ground by rotating in the counter-clockwise and clockwise directions respectively. During this maneuver, the back legs are commanded to apply more torque than the front ones giving a positive pitching attitude to the robot. This clears the ground for the front legs as they are pointing down. The robot then lands on its front legs with all four legs vertical to the ground. Figure 3-14 illustrates the sequence of events for standing up.

### Chapter 3. Experimental Results

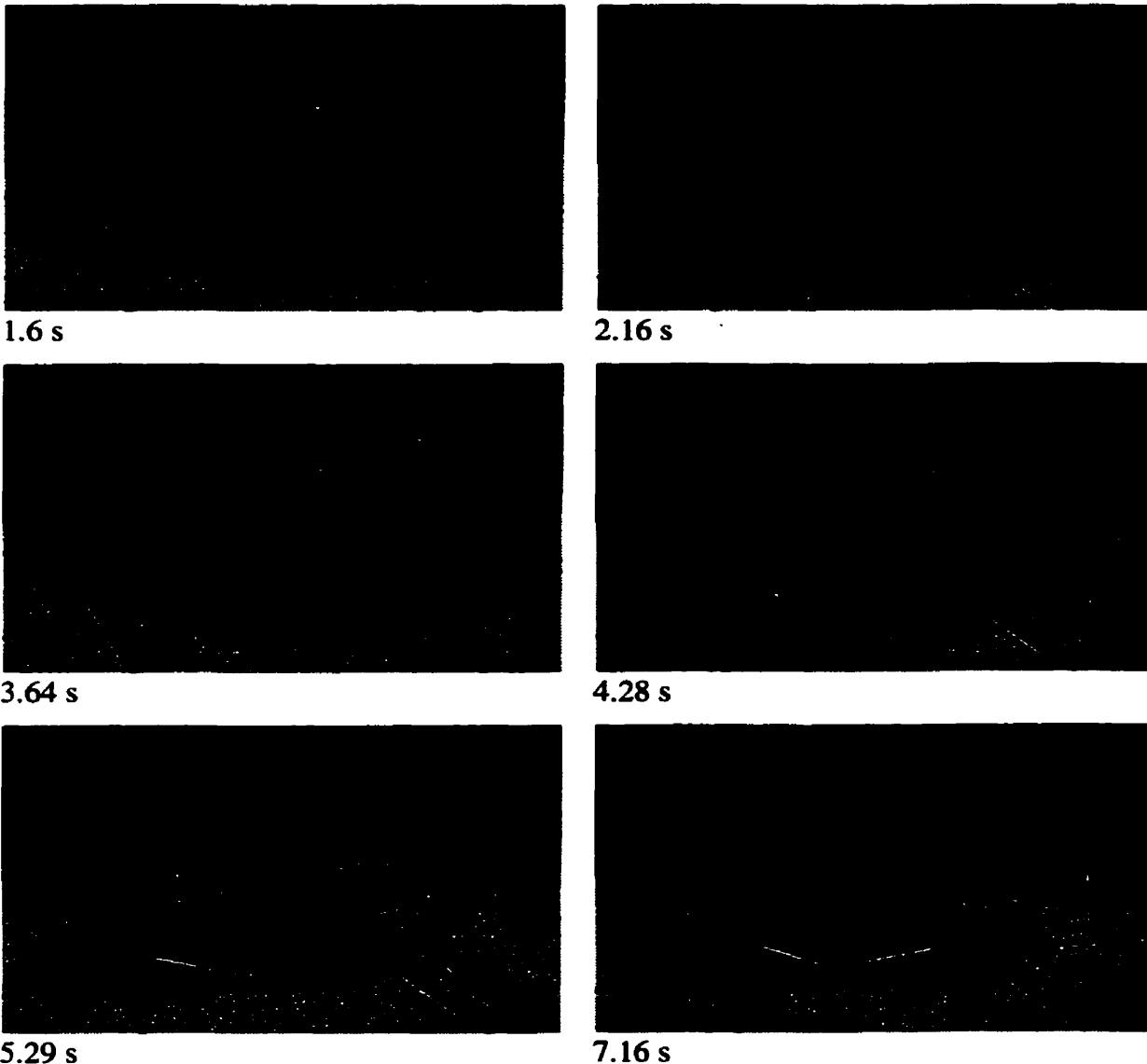


**Figure 3-14: Standing up sequence. Figure is read from left to right.**

Two sorts of sitting down were implemented on Scout II. The first one, which is shown in Figure 3-15, is almost the reverse process of the standing up. The robot starts by leaning down to one side with  $\phi_f = \phi_b = 15^\circ$ , thus enabling one pair of legs to slide to the inside. The robot then slides again to the other side enabling the front pair of legs to slide inside with  $\phi_f = -30^\circ$ , and  $\phi_b = 15^\circ$ . The back legs are then brought to an angle symmetric to the front ones with  $\phi_b = 30^\circ$ . Now that both pair of legs are pointing inside

### Chapter 3. Experimental Results

the robot, they move together inwards until they reach the horizontal position where  $\phi_b = 90^\circ$ , and  $\phi_f = -90^\circ$ . This is shown in Figure 3-15.



**Figure 3-15: Sequence of events for sitting down. Figures are read from left to right.**

The second way of sitting down, could be more appropriate than the first sitting sequence in some circumstances. It starts in the same manner as the first sitting method where the robot leans to one end leading to values of  $\phi_f$ , and  $\phi_b$  of  $15^\circ$ . Then the robot

### Chapter 3. Experimental Results

leans to the other end to achieve negative pitching. This is done by slowly incrementing the front legs angles while keeping the back pair of legs at a fixed angle. At the point where the robot body length and front legs are aligned ( $\phi_f = 90$ ) both legs start moving to the outside to get the robot back to the zero pitching position. The robot ends with its legs wide spread and its bottom touching the ground, with  $\phi_f = 90$  and  $\phi_b = -90$ . Figure 3-16 illustrates the sequence of events used to bring the robot down in the second method.

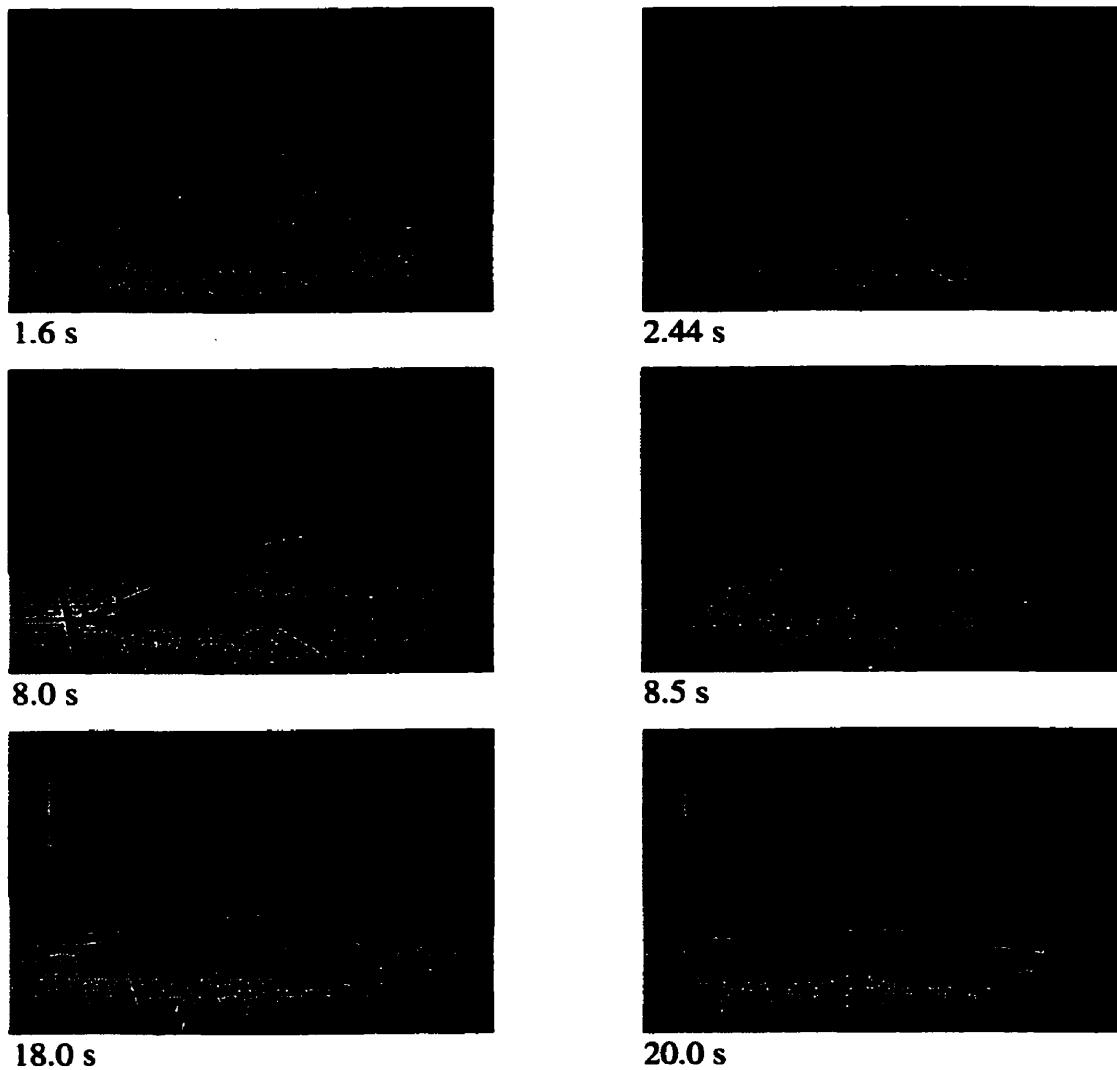


Figure 3-16: Sequence of events for sitting down. Figures are read from left to right.

### 3.6 Conclusion

This chapter presented the different experimental results obtained on Scout II. Open loop running was implemented and resulted in a stable bounding gait that is comparable to simulation A pronking gait which typically has less body pitching than the bounding was also implemented with the open loop controller. The closed loop running controller resulted in a running gait that was unsteady due to the poor velocity sensing. The differentiation on board, the backlash in the motor, the gearhead, and the belt transmission all contributed to significant errors that made it difficult to use the forward velocity sensing for the closed loop controller. Turning similar to the simulation one was implemented resulting in 90° turns with a simple algorithm that applies differential torques on the left and right side legs. Other behaviors such as standing up and sitting down were also successfully achieved.

# **Chapter 4**

---

## **Conclusion**

### **4.1 Summary**

This thesis presented running control algorithms for a quadruped robot with compliant legs. The four running phases were modeled in Chapter 2 and a comparison with the simulation software was completed. The analysis showed no major differences in the robot behavior in the stance phases. However, the results of the flight phases differed as consequence of the assumption of massless legs. The analytical model used to validate portions of the simulation can be used in the future for controller development.

The simulation results of the open loop running controller were successful. This controller demonstrated that running can be achieved with minimal sensing requirement. The robot ran steadily at a speed near 1.2 m/s. A modified version of Raibert's controller was successfully implemented in simulation. This controller used feedback of the forward velocity of the robot, and the stance time in order to estimate the appropriate leg angles at touchdown. Velocity control, pitch control, and jumping control were incorporated into the controller to increase the running behaviors. Modeling of the motor operation characteristics was also completed in order to draw attention to the limitations of the actual robot's mechanical system. A model of the interaction between the ground and the robot's toe was carried out to come up with a toe-slip prevention algorithm. The disadvantage of such a model is that it required knowledge of the ground properties. To

## Chapter 4. Conclusion

accommodate this limitation, a slip correction technique was introduced to permit recovery in the case of slip occurrence.

Experimental results in Chapter 3 show that Scout II is capable of running. Minor discrepancies between the experiments and the simulator existed for the open loop controller. The sensing used on Scout II and modeling of the toe and hip mechanical design affected the experimental results. The robot forward velocity sensing was not reliable enough to use in the closed loop velocity feedback controller. Further, the hip design included some backlash and compliance in the belt that were not modeled in simulation, and which had an adverse effect on the implementation of the running controllers. Other complementary behaviors such as turning, standing up and sitting down were also implemented on Scout II.

## 4.2 Future Recommendations

Although running proved successful on Scout II in many instances, it could be improved by implementing the following recommendation:

1. Installation of a gyroscope sensor for sensing yaw. This will allow more stable running, as it will include some aspect of yaw control. Turning will naturally be easier and longer turns will be achieved by proper control of the yawing motion.
2. The installation of a gyroscope sensor for the roll sensing will permit precise control of the turning angle of the robot. This will permit the implementation of a turning algorithm that relies on the desired roll angle.
3. An accelerometer or other sensing device to measure the exact forward velocity would be of great benefit, to improve feedback of the forward velocity of the robot. As seen in simulation, the closed loop controller can be modified to account for jumping and control of running speed. However these controllers rely on proper sensing of the forward velocity, which is still lacking on Scout II.

## Chapter 4. Conclusion

4. Modeling of the robot in the analysis could include the actual masses in the legs and consider the impact model during the phase transitions. The improved model could give a better understanding of the running cycle. It could also be used to simulate the running in a faster way.

To conclude, stable running was implemented in simulation and experiment, using simple running controllers. However, implementation results showed that increased control over the running behaviors comes at the expense of more elaborate controllers that require more sensing. Nevertheless this thesis demonstrates that running is possible with simple control techniques and suggests that further investigation can achieve a wider range of behavior.

# Bibliography

- [1] M. Ahmadi and M. Buehler. A Control Strategy for Stable Passive Running. In *Proc. IEEE Conf. Intelligent Systems and Robots*, pages 152-157, Pittsburgh, PA, Aug 1995.
- [2] M. Ahmadi and M. Buehler. The ARL Monopod II Running Robot: Control and Energetics. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1689-1694, May 1999.
- [3] R. M. Alexander. *Locomotion in Animals*. Blackie, 1982.
- [4] R.F. Battaglia. Design of the SCOUT II Quadruped with Preliminary Stair-Climbing. M. Eng. Thesis, McGill University, May 1999.
- [5] M.D. Berkemeier. Approximate Return Maps for Quadrupedal Running. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 805-810, Albuquerque, New Mexico, April 1997.
- [6] G.N. Boone and J.K. Hodgins. Reflexive Responses to Slipping in Bipedal Running Robots. In *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, vol. 3, pages 158 – 164, 1995.
- [7] M. Buehler, A. Cocosco, K. Yamazaki, R. Battaglia. Stable Open Loop Walking in Quadruped Robots with Stick Legs. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2348-2353, Detroit, Michigan, May 1999.
- [8] M. Buehler, R. Battaglia, A. Cocosco, G. Hawker, J. Sarkis and K. Yamazaki. Scout: A simple quadruped that walks, climbs and runs. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1707-1712, Leuven, Belgium, May 1998.
- [9] A. Cocosco and C. Dormanjian. Latex: Energy Through Compliance. Mechanical Engineering Laboratory final report, McGill University, Montreal, QC, 1996.

## Bibliography

- [10] A. Cocosco. Synthesis of a Quadruped Walking Robot. M. Eng. Thesis, McGill University, Montréal, QC, Canada, July 1998.
- [11] J. Furusho, A. Sano, M. Sakaguchi, and E. Koizumi. Realization of Bounce Gait in a Quadruped Robot with Articular-Joint-Type Legs. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 697-702, May 1995.
- [12] M. Garcia, A. Chatterjee, A. Ruina. Speed, efficiency, and stability of small-slope 2D passive dynamic bipedal walking. In *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, pages 2351-2356, 1998.
- [13] P. Gregorio, M. Ahmadi, M. Buehler. Design, Control, and Energetics of an Electrically Actuated Legged Robot. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 27, No. 4, pages 626-634, August 1997.
- [14] G. Hawker. Quadruped Totting with passive knees: Design, Control and Experiments. M. Eng. Thesis, McGill University, Montréal, QC, Canada, November 1999.
- [15] S. Hirose, H. Takeuchi. Study on Roller-Walk (Basic Characteristics and its Control). In *Proc. IEEE Int. Conf. Robotics and Automation*, vol.3, pages 2032-2037, 1996.
- [16] Interelectric AG, Sachseln, Switzerland. Maxon Motor Catalog, 1997/98, [www.maxon.com](http://www.maxon.com)
- [17] IFR, Institute of Robotics, [www.ifr.mavt.ethz.ch](http://www.ifr.mavt.ethz.ch)
- [18] IS Robotics, [www.isr.com/research](http://www.isr.com/research)
- [19] Y.Y. Keon. Locomotion of a biped robot with compliant ankle joints. In *Proc. IEEE Int. Conf. Robotics and Automation*, vol.1, pages 199-204 , 1997.
- [20] H. Kimura, K. Sakurama and S. Akiyama. Dynamic Walking and Running of the Quadruped Using Neural Oscillator. IROS 1998.

## Bibliography

- [21] Knowledge Revolution. *Working Model 2D® Version 4.0.1*. San Mateo, California, 1996. [www.krev.com](http://www.krev.com)
- [22] Knowledge Revolution. *Working Model 3D® Version 4.0*. San Mateo, California, 1998. [www.krev.com](http://www.krev.com)
- [23] Microsoft. *Microsoft Excel 97, Microsoft Visual Basic.* 1997. [www.microsoft.com/office](http://www.microsoft.com/office)
- [24] The MathWorks. *The Student Version of Matlab, Version 5.0®*. Natick, MA, 1998. [www.mathworks.com](http://www.mathworks.com)
- [25] T. McGeer. *Passive Bipedal Running*. Simon Fraser University, Burnaby, BC, April 1989.
- [26] G. Mennitto. Compliant Articulated Robot Leg with Antagonistic LADD Actuation. Master's Thesis, McGill University, Montréal, QC, Canada, February 1995.
- [27] G. Mennitto and M. Buehler. Experimental Validation of Compliance Models for LADD Transmission Kinematics. In *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, pages 385-390, 1995.
- [28] H. Miura and I. Shimoyama. Dynamic walk of a biped. *International Journal of Robotics Research*, 3(2), pages 60-74, Summer 1984.
- [29] E. Muybridge. Animals in motion. Dover Publications, Inc. New York, 1957.
- [30] J. Nagle. Anti Slip Control for a Legged Robot and Realistic Simulation of a Legged Creature. *United States Patent*. Patent number:5,644,204, July 1, 1997.
- [31] M. Hiller and T. Schmitz. (1991). Robotrac - an example of a mechatronic system. In *Conference on Mechatronics and Robotics*, Aachen, Germany. IOS Press. 1991.
- [32] M. H. Raibert. Legged Robots That Balance. MIT Press, Cambridge, MA, 1986.
- [33] Remotec, Northrop Grumman, Oak Ridge, TN, [www.remotec-andros.com](http://www.remotec-andros.com).
- [34] J. Sarkis. Control of running in a quadruped robot with actuation only at the hips. Master's thesis, McGill University, Montréal, QC, Canada, March. 1999.

## Bibliography

- [35] W.J. Schwind and D.E. Koditschek. Control of Forward Velocity for a Simplified Planar Hoping Robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 691-696, 1995.
- [36] A. Velimirović, M. Velimirović, V. Hugel, A. Iles, P. Blazevic. A New Architecture of Robot with "Wheels-With-Legs" (WWL). *IEEE*, 1990.
- [37] Wolfram Research, Mathematica® Version 3.0. 1996. [www.wolfram.com](http://www.wolfram.com)
- [38] K. S. Yamazaki. The Design and Control of Scout I, a Simple Quadruped Robot. Master's Thesis, McGill University, Montréal, QC, Canada, February 1999.

# APPENDIX

---

## Derivation of the Equations of Motion

### A.1 Back Legs Support Phase

The following is the derivation of the governing equations of motion for the case where the robot is supported on the ground by its hind legs. The generalized coordinates for this system are chosen to be  $l_b$ ,  $\theta$ , and  $\phi_b$ . From the generalized coordinates and

Figure 2-3, the body Cartesian coordinates can be expressed as,

$$\mathbf{p}_b = \begin{bmatrix} -l_b \sin(\theta + \phi_b) + L \cos \theta \\ l_b \cos(\theta + \phi_b) + L \sin \theta \end{bmatrix}.$$

These can be differentiated with respect to time, in order to give the body velocities in the x- and y- directions,

$$\dot{\mathbf{p}}_b = \begin{bmatrix} -l_b (\dot{\theta} + \dot{\phi}_b) \cos(\theta + \phi_b) - l_b \sin(\theta + \phi_b) \dot{\theta} - L \dot{\theta} \sin \theta \\ l_b \cos(\theta + \phi_b) - l_b (\dot{\theta} + \dot{\phi}_b) \sin(\theta + \phi_b) + L \dot{\theta} \cos \theta \end{bmatrix}.$$

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

The total body kinetic energy is the sum of the translational and rotational kinetic energies,

$$T = \frac{1}{2} m \dot{x}_b^T \dot{x}_b + \frac{1}{2} m r^2 \dot{\theta}^2,$$

$$T = \frac{m}{2} (\dot{l}_b^2 + l_b^2 \dot{\phi}_b^2 + 2L \cos(\phi_b) \dot{l}_b \dot{\theta} + [-2Ll_b \sin(\phi_b) \dot{l}_b + 2l_b^2] \dot{\phi}_b \dot{\theta} + [r^2 + L^2 - 2L \sin(\phi_b) l_b + l_b^2] \dot{\theta}^2).$$

The body potential energy is the sum of the potential energy due to elevation and the energy stored in the springs,

$$V = mg[l_b \cos(\theta + \phi_b) + L \sin \theta] + \frac{1}{2} k(l_b - l_0)^2.$$

The energy loss in the legs is approximated by a damper,

$$D = \frac{1}{2} b l^2.$$

The Lagrangian function,  $\mathcal{L}$ , is found by subtracting the potential energy from the kinetic energy:

$$\mathcal{L} = T - V$$

We now perform the differentiation required in the Lagrange equation,

$$\begin{aligned} \frac{d}{dt} \frac{D\mathcal{L}}{D\dot{\theta}} &= m[[l_b^2 + L^2 + r^2 - 2Ll_b \sin(\phi_b)]\ddot{\theta} + [-Ll_b \sin(\phi_b) + l_b^2]\ddot{\phi}_b + L \cos(\phi_b) \ddot{l}_b, \\ &- 2Ll_b \cos(\phi_b) \dot{\phi}_b \dot{\theta} [2l_b - 2Ll_b \sin(\phi_b)] \dot{\phi}_b \dot{l}_b [2l_b - 2Ll_b \sin(\phi_b)] \dot{l}_b \dot{\theta} - L \cos(\phi_b) \dot{\phi}_b^2] \end{aligned}$$

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

$$\frac{d}{dt} \frac{D\mathcal{L}}{D\dot{\phi}_b} = \frac{m}{2} [-2L\sin(\phi_b)\dot{l}_b\theta - 2Ll_b\cos(\phi_b)\dot{\phi}_b\dot{\theta} + 4l_b\dot{l}_b\dot{\phi}_b]$$

$$+ 4l_b\dot{l}_b\dot{\theta}[2l_b^2 - 2L\sin(\phi_b)]\ddot{\theta} + 2l_b^2\ddot{\phi}_b$$

$$\frac{d}{dt} \frac{D\mathcal{L}}{D\dot{l}_b} = m[-L\sin(\phi_b)\dot{\phi}_b\dot{\theta} + \ddot{l}_b + L\cos(\phi_b)\ddot{\theta}] ,$$

$$\frac{D\mathcal{L}}{D\theta} = gm[-L\cos(\theta) + l_b\sin(\phi_b + \theta)] ,$$

$$\frac{D\mathcal{L}}{Dl} = k(l_o - l_b) - gm\cos(\phi_b + \theta) + m(\dot{\phi}_b + \dot{\theta})(l_b\dot{\phi}_b + L\sin(\phi_b)\dot{\theta} + l_b\dot{\theta}) ,$$

$$\frac{D\mathcal{L}}{D\dot{\phi}_b} = m[gl_b\sin(\phi_b + \theta) - L\sin(\phi_b)\dot{l}_b\dot{\theta} - Ll\cos(\phi_b)\dot{\phi}_b\dot{\theta} - Ll_b\cos(\phi_b)\dot{\theta}^2] ,$$

The Lagrange equation of motion is,

$$\frac{d}{dt} \left( \frac{D\mathcal{L}}{D\dot{q}} \right) - \frac{D\mathcal{L}}{Dq} + \frac{DB}{D\dot{q}} = F .$$

By direct substitution, the expanded form for the back legs support case becomes,

$$\left\{ \begin{array}{l} m[gL\cos(\theta) - gl\sin(\phi_b + \theta) + [-2L\sin(\phi_b) + 2l]\dot{\phi}_b\dot{l}_b - Ll_b\cos(\phi_b)\dot{\phi}_b^2 \\ + [-2L\sin(\phi_b) + 2l]\dot{l}_b\dot{\theta} - 2Ll_b\cos(\phi_b)\dot{\phi}_b\dot{\theta} + L\cos(\phi_b)\ddot{l}_b \\ + [-Ll_b\sin(\phi_b) + l_b^2]\ddot{\phi}_b + [L^2 + r^2 + l_b^2 - 2Ll_b\sin(\phi_b)]\ddot{\theta}] \\ ml_b[-g\sin(\phi_b + \theta) + 2l\dot{\phi}_b + 2l\dot{\theta} + L\cos(\phi_b)\dot{\theta}^2 + l_b\ddot{\phi}_b + [-L\sin(\phi_b) + l_b]\ddot{\theta}] \\ k(l_b - l_o) + gm\cos(\phi_b + \theta) + bl_b - ml_b\dot{\phi}_b^2 - 2ml_b\dot{\phi}_b\dot{\theta} + [Lms\sin(\phi_b) - ml_b]\dot{\theta}^2 \\ + ml_b\ddot{\theta} + Lm\cos(\phi_b)\ddot{\theta} \end{array} \right\} = \begin{Bmatrix} 0 \\ \tau_b \\ 0 \end{Bmatrix} .$$

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

Let  $\mathbf{x} = [l_b, \dot{l}_b, \theta, \dot{\theta}, \phi_b, \dot{\phi}_b]^T$  be the state vector. Then the above set of equations can be written in the following state space form,

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{m} [k(l_o - x_1) - bx_2 + m[x_1(x_4 + x_6)^2 - g \cos(x_3 + x_5) + Lx_4^2 \sin(x_5)] \\ \quad - L \cos(x_5) [Lx_1(k[x_1 - l_o] + bx_2) \cos(x_5) + \tau_b(L \sin(x_5) - x_1)]] \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{Lx_1[k(x_1 - l_o) + bx_2] \cos(x_5) + \tau_b[L \sin(x_5) - x_1]}{mr^2 x_1} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = \frac{1}{mr^2 x_1^2} [\tau_b[L^2 + 2r^2 + 2x_1^2] - 4mr^2 x_1 x_2 [x_4 + x_6] + L[-2x_1(kx_1[x_1 - l_o] \\ \quad + bx_2 x_1 + mr^2 x_4^2) \cos(x_5) - L\tau_b(\cos(2x_5) - 4x_1 \sin(x_5))] + Lx_1[k(x_1 - l_o) + bx_2] \\ \quad \sin(2x_5)] + 2gmr^2 x_1 \sin(x_3 + x_5)] \end{array} \right\}.$$

### A.2 Front Legs Support Phase

The following is the complete derivation of the equations of motion and the comparison to Working Model 2D® of Scout II for the front stance phase. The equations of motion for the front legs support case are found using  $\mathbf{q} = [\phi_f, \theta, l_f]^T$ . The body Cartesian coordinates are expressed at the contact point of the front leg's toe and the ground as follows,

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

$$\dot{x}_f = \begin{bmatrix} -l_f \sin(\theta + \phi_f) - L \cos \theta \\ l_f \cos(\theta + \phi_f) - L \sin \theta \end{bmatrix}.$$

$$\ddot{x}_f = \begin{bmatrix} -\dot{l}_f \sin(\theta + \phi_f) - l_f \cos(\theta + \phi_f)(\dot{\theta} + \dot{\phi}_f) + L \dot{\theta} \sin(\theta) \\ \dot{l}_f \cos(\theta + \phi_f) - l_f \sin(\theta + \phi_f)(\dot{\theta} + \dot{\phi}) - L \dot{\theta} \cos(\theta) \end{bmatrix}.$$

The total body kinetic energy is the sum of the translational and rotational kinetic energies,

$$T = \frac{1}{2} m \dot{x}_f^T \dot{x}_f + \frac{1}{2} m r^2 \dot{\theta}^2,$$

$$T = \frac{m}{2} (\dot{l}_f^2 + l_f^2 \dot{\phi}_f^2 - 2L \cos(\phi_f) \dot{l}_f \dot{\theta} + [2L \sin(\phi_f) l_f + 2l_f^2] \dot{\phi}_b \dot{\theta} + [r^2 + L^2 + 2L \sin(\phi_f) l_f + l_f^2] \dot{\theta}^2)$$

Now the body potential energy is the sum of the potential energy due to elevation and the energy stored in the springs,

$$V = mg[l_f \cos(\theta + \phi_f) - L \sin \theta] + \frac{1}{2} k(l_f - l_0)^2.$$

The energy loss in the legs is approximated by a damper,

$$D = \frac{1}{2} b \dot{l}_f^2.$$

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

It is now possible to find the Lagrangian function,  $\mathcal{L}$ , by subtracting the potential energy from the kinetic energy,

$$\mathcal{L} = T - V .$$

We then perform the differentiation required in the Lagrange equation,

$$\frac{D\mathcal{L}}{D\dot{\theta}} = m[-L\cos(\phi_f)\dot{l}_f + [l_f^2 + Ll_f \sin(\phi_b)]\dot{\phi}_f + [L^2 + l_f^2 + r^2 + 2Ll_f \sin(\phi_b)]\dot{\theta}],$$

$$\frac{D\mathcal{L}}{D\dot{\phi}_f} = ml[l_f \dot{\phi}_f + [L\sin(\phi_f) + l_f]\dot{\theta}],$$

$$\frac{D\mathcal{L}}{D\dot{l}_f} = m[\dot{l}_f - L\cos(\phi_f)\dot{\theta}],$$

$$\begin{aligned} \frac{d}{dt} \frac{D\mathcal{L}}{D\dot{\theta}} &= m[[l_f^2 + L^2 + r^2 + 2Ll_f \sin(\phi_b)]\ddot{\theta} + [Ll_f \sin(\phi_b) + l_f^2]\ddot{\phi}_b - L\cos(\phi_f)\ddot{l}_f \\ &+ 2Ll_f \cos(\phi_f)\dot{\phi}_f \dot{\theta} + 2[l_f + L\sin(\phi_f)]\dot{l}_f \dot{\theta} + 2[l_f + L\sin(\phi_f)]\dot{l}_f \dot{\phi}_f + Ll_f \cos(\phi_f)\dot{\phi}_f^2], \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \frac{D\mathcal{L}}{D\dot{\phi}_f} &= m[2l_f \dot{l}_f \dot{\phi}_f + [2l_f + L\sin(\phi_f)]\dot{l}_f \dot{\theta} + Ll_f \cos(\phi_f)\dot{\phi}_f \dot{\theta} + l_f^2 \ddot{\phi}_b \\ &+ [Ll_f \sin(\phi_f) + l_f^2]\ddot{\theta}] \end{aligned}$$

$$\frac{d}{dt} \frac{D\mathcal{L}}{D\dot{l}_f} = m[L\sin(\phi_f)\dot{\phi}_f \dot{\theta} + \ddot{l}_f - L\cos(\phi_f)\ddot{\theta}],$$

$$\frac{D\mathcal{L}}{D\theta} = gm[L\cos(\theta) + l_f \sin(\phi_f + \theta)],$$

$$\frac{D\mathcal{L}}{D\phi_f} = m[gl_f \sin(\phi_f + \theta) + L\sin(\phi_f)\dot{l}_f \dot{\theta} + Ll_f \cos(\phi_f)\dot{\phi}_f \dot{\theta} + Ll_f \cos(\phi_f)\dot{\theta}^2],$$

$$\frac{D\mathcal{L}}{Dl_f} = k(l_0 - l_f) - gm\cos(\phi_f + \theta) + m(\dot{\phi}_f + \dot{\theta})(l_f \dot{\phi}_b + L\sin(\phi_b)\dot{\theta} + l_f \dot{\theta}),$$

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

The Lagrange equation of motion is,

$$\frac{d}{dt} \left( \frac{D\mathcal{L}}{D\dot{q}} \right) - \frac{D\mathcal{L}}{Dq} + \frac{DB}{D\dot{q}} = F .$$

By direct substitution, the expanded form for the front legs support case becomes,

$$\left\{ \begin{array}{l} m[gL\cos(\theta) - gl_f \sin(\phi_f + \theta) + [2L\sin(\phi_f) + 2l_f]\dot{\phi}_f \dot{l}_f + Ll_f \cos(\phi_f)\dot{\phi}_f^2 \\ + [+2L\sin(\phi_f) + 2l_f]\ddot{\phi} + 2Ll_f \cos(\phi_f)\dot{\phi}_f \ddot{\theta} - L\cos(\phi_f)\ddot{l}_f \\ + [+Ll_f \sin(\phi_f) + l_f^2]\ddot{\phi}_f + [l_f^2 + r^2 + l_f^2 + 2Ll_f \sin(\phi_f)]\ddot{\theta}] \\ ml_f [-g \sin(\phi_f + \theta) + 2l_f \dot{\phi}_f + 2l_f \dot{\theta} - L\cos(\phi_f)\dot{\theta}^2 + l_f \ddot{\phi}_f + [L\sin(\phi_f) + l_f]\ddot{\theta}] \\ k(l_f - lo) + gm\cos(\phi_f + \theta) + bl_f - 2ml_f \dot{\phi}_f \dot{\theta} + [-Lm\sin(\phi_f) - ml_f]\dot{\theta}^2 \\ + ml_f \ddot{\theta} - Lm\cos(\phi_f)\ddot{\theta} - ml_f \dot{\phi}_f^2 \end{array} \right\} = \begin{Bmatrix} 0 \\ \tau_f \\ 0 \end{Bmatrix}.$$

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

Let  $\mathbf{x} = [l_f, \dot{l}_f, \theta, \dot{\theta}, \phi_f, \dot{\phi}_f]^T$  be the state vector. Then the above set of equations can be written in state space form,

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{m} [k[lo - x_1] - bx_2 + m[x_1(x_4 + x_6)^2 - g \cos(x_3 + x_5) + Lx_4^2 \sin(x_5)] \\ \quad - L \cos(x_5)[Lx_1[k[x_1 - lo] + bx_2] \cos(x_5) + \tau_f[x_1 + L \sin(x_5)]]] \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = - \frac{Lx_1[k[x_1 - lo] + bx_2] \cos(x_5) + \tau_f[x_1 + L \sin(x_5)]}{mr^2 x_1} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = \frac{1}{mr^2 x_1^2} [\tau_f[L^2 + 2r^2 + 2x_1^2] - 4mr^2 x_1 x_2 [x_4 + x_6] + L[2x_1[kx_1[x_1 - lo] \\ \quad + bx_2 x_1 + mr^2 x_4^2] \cos(x_5) - L\tau_f[\cos(2x_5) + 4x_1 \sin(x_5)] + Lx_1[k[x_1 - lo] + bx_2] \\ \quad \sin(2x_5)] + 2gmr^2 x_1 \sin(x_3 + x_5)] \end{array} \right\}.$$

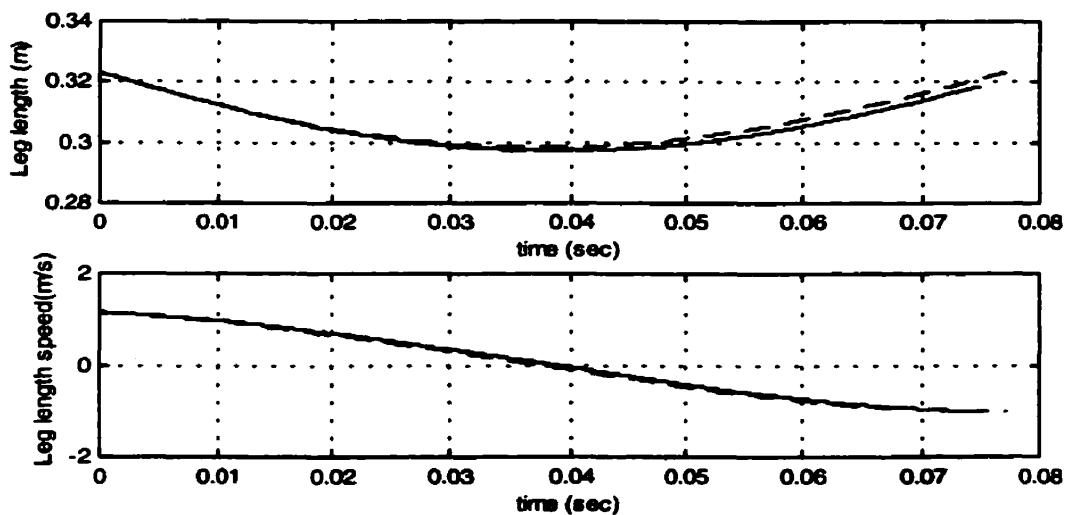
The above Lagrange equations of motion can be written as follows,

$$\mathbf{A}_f(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}_f(\mathbf{q}, \dot{\mathbf{q}}) = \begin{pmatrix} 0 \\ \tau_f \\ 0 \end{pmatrix}.$$

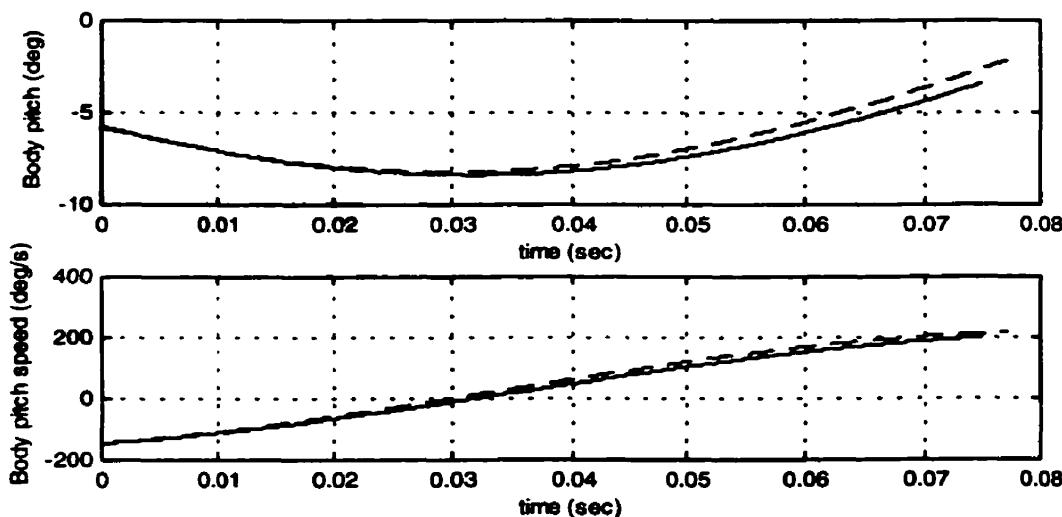
Where  $\mathbf{A}_f(\mathbf{q})$  is the  $3 \times 3$  inertia matrix,  $\mathbf{B}_f(\mathbf{q}, \dot{\mathbf{q}})$  is the gravity and centrifugal force vector.

As it is previously shown, the above equations of motion can be validated by comparison to the simulation results. The equations of motion for the front legs' stance phase are once again integrated using Mathematica®. The front legs' stance phase is also simulated using Working Model®. Next, a comparison similar to the one made for the back stance is performed.

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION

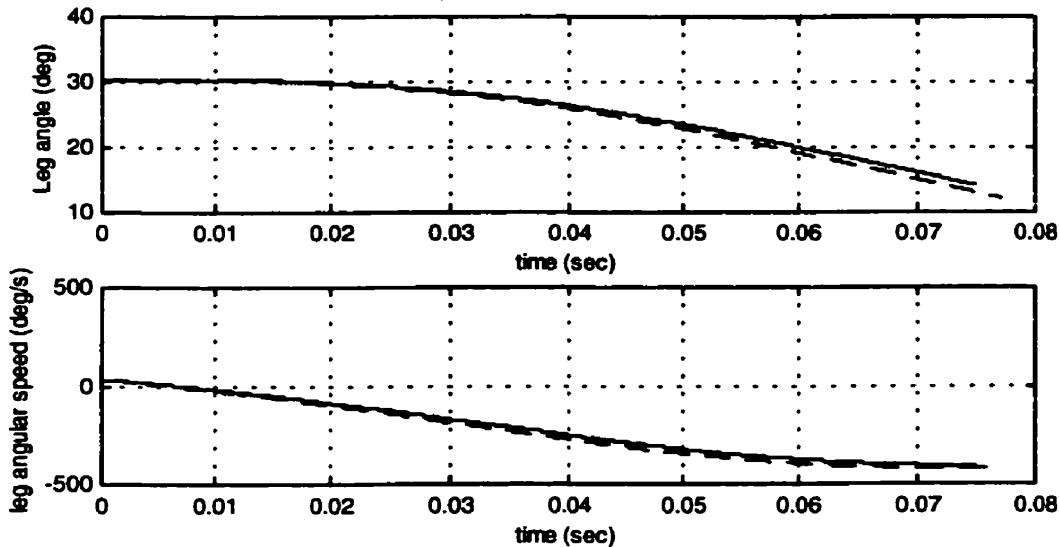


**Figure A-0-1:** Comparison between Working Model 2D® plots and Mathematica® plots for back leg length variations throughout the front legs support phase. The maximum leg length error is 2.62% of the 10 cm full scale leg displacement. The maximum error in leg length speed is 4.45% of the 2 m/s full scale leg speed.



**Figure A-0-2:** Comparison between Working Model 2D® plots and Mathematica® plots for body pitch values during the front legs support phase. The maximum body pitch error is 7.91% with respect to the 10 deg full scale body pitch amplitude. The maximum body pitch speed error is 8.53% with respect to the 300 deg/s full scale body pitch amplitude.

## APPENDIX. DERIVATION OF THE EQUATIONS OF MOTION



**Figure A-0-3: Comparison between Working Model 2D<sup>®</sup> plots and Mathematica<sup>®</sup> plots for the front leg angle during the front legs support phase. The leg angle error is 5.6% of the 20 deg full scale leg amplitude. The leg angle error is 16.8% of the 200 deg/s full scale leg speed amplitude.**