SIX-LEGGED WALKING MACHINE: THE ROBOT-EA308

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSTAFA SUPHİ ERDEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2006

Approval of the Graduate School of Natural and Applied Sciences

____

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

____

Prof. Dr. İsmet Erkmen
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

____

Prof. Dr. Kemal Leblebicioğlu
Supervisor

Examining Committee Members

Prof. Dr. Uğur Halıcı	(METU, EE)	____________________
Prof. Dr. Kemal Leblebicioğlu	(METU, EE)	____________________
Prof. Dr. Kemal Özgören	(METU, ME)	____________________
Asst. Prof. Dr. Afşar Saranlı	(METU, EE)	____________________
Asst. Prof. Dr. Uluç Saranlı	(Bilkent Unv., CENG)	____________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mustafa Suphi ERDEN

ABSTRACT

SIX-LEGGED WALKING MACHINE: THE ROBOT-EA308

Erden, Mustafa Suphi

Ph.D., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kemal Leblebicioğlu

July 2006, 157 pages

The work presented in this thesis aims to make contribution to the understanding and application of six-legged statically stable walking machines in both theoretical and practical levels. In this thesis five pieces of work, performed with and for the three-joint six-legged Robot-EA308, are presented: 1) Standard gaits, which include the well-known wave gaits, are defined and a stability analysis, in the sense of static stable walking, is performed on an analytical level. Various definitions are given; theorems are stated and proved. 2) A free gait generation algorithm with reinforcement learning is developed. Its facilities of stability improvement, smooth speed changes, and adaptation in case of a rear-leg deficiency with learning of five-legged walking are experimented in real-time on the Robot-EA308. 3) Trajectory optimization and controller design is performed for the protraction movement of a three-joint leg. The trajectory generated by the controller is demonstrated with the Robot-EA308. 4) The full kinematic-dynamic formulation of a three-joint six-legged robot is performed with the joint-torques being the primary variables. It is demonstrated that the proposed torque distribution scheme, rather than the conventional force distribution, results in an efficient distribution of required forces and moments to the supporting legs. 5) An analysis of energy efficiency is performed for wave gaits. The established strategies for determination of gait parameters for an efficient walk are justified using the Robot-EA308.

Keywords: Six-legged robot, walking, wave gaits, trajectory optimization, torque distribution, force distribution, reinforcement learning.

ÖZ

ALTI BACKLI YÜRÜYEN MAKİNA: ROBOT-EA308

Erden, Mustafa Suphi

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Kemal Leblebicioğlu

Temmuz 2006, 157 sayfa

Bu tez, altı-bacaklı durağan-kararlı olarak yürüyen makinalar üzerine, anlamaya ve uygulamaya yönelik olarak hem teorik hem de pratik düzeyde katkı sunmayı amaçlamaktadır. Tezde üç-eklemli altı-bacaklı Robot-EA308 için ve onu kullanarak gerçekleştirilen beş parça çalışma sunulmuştur: 1) Dalgalı yürüyüşleri de içeren standart yürüyüşler tanımlanmış ve analitik düzeyde, durağan-kararlı yürüyüş olmaları anlamında, kararlılık analizleri gerçekleştirilmiştir. Çeşitli tanımlamalar verilmiş, teoremler ifade edilip ispatlanmıştır. 2) Pekiştirmeli öğrenme içeren bir serbest yürüme algoritması geliştirilmiştir. Bu algoritma Robot-EA308 üzerinde gerçek zamanda uygulanmış, sağladığı kararlılık gelişimi, farklı hızlar arasında yumuşak geçiş yapabilme ve arka bacaklardan birinin arızalanması durumunda beş bacaklı yürümeyi öğrenerek uyum sağlayabilme özellikleri gözlenmiştir. 3) Üç-eklemli bir robot bacağının ileri atılımı için yörünge eniyilemesi ve kontrolcü tasarımı gerçekleştirilmiştir. Kontrolcünün ürettiği yörünge Robot-EA308 kullanılarak gösterilmiştir. 4) Üç-eklemli altı-bacaklı bir robotun tam kinematik ve dinamik formulasyonu, torklar birincil değişken olmak üzere, gerçekleştirilmiştir. Gerekli kuvvet ve momentleri destek sağlayan bacaklara dağıtmak için önerilen tork dağıtımının, yaygın olarak kullanılan kuvvet dağıtımı yaklaşımından daha verimli sonuç verdiği gösterilmiştir. 5) Dalgalı yürüyüşler için enerji verimliliği analizi gerçekleştirilmiştir. Yürüme parametrelerini verimli bir yürüyüş sağlayacak şekilde belirlemek üzere geliştirilen stratejiler gerçek Robot-EA308 üzerinde doğrulanmıştır.

Anahtar Kelimeler: Altı-bacaklı robot, yürüme, dalgalı yürüyüş, yörünge eniyilemesi, tork dağıtımı, kuvvet dağıtımı, pekiştirmeli öğrenme.

Bu çalışma büyükbabam Fevzi ERDEN'e, babaannem Kâmile ERDEN'e ve anneannem Döndü DAĞ'ın anısına adanmıştır.

This work is dedicated to my grandfather Fevzi ERDEN, my grandmother Kâmile ERDEN, and the memory of my grandmother Döndü DAĞ.

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to his supervisor Prof. Dr. Kemal Leblebicioğlu for his guidance and insight throughout the research.

The author thanks to his parents Gülseren Erden and İsmail Erden, and to his sister Zeynep Erden for their support, supervision, and encouragement throughout the years of his education.

The hours spent for this thesis would not be so enjoyable in any other place than the "Computer Vision and Intelligent Systems Research Laboratory" at EA308 in Electrical and Electronics Engineering Department. The author thanks to his colleagues for the friendly atmosphere and solidarity in their laboratory.

This study was partially supported by the research fund of Middle East Technical University as a scientific research project: BAP – 2002 – 03 – 01 – 06.

TABLE OF CONTENTS

PLAGIARISM .... **iii**

ABSTRACT .... **iv**

ÖZ .... **v**

ACKNOWLEDGEMENTS .... **vii**

TABLE OF CONTENTS .... **viii**

LIST OF TABLES .... **xi**

LIST OF FIGURES .... **xii**

CHAPTER

1. INTRODUCTION **1**

1.1. A Comparison of Wheeled, Tracked, and Legged Locomotion **1**

1.2. Biological Inspirations for Legged Locomotion Systems **4**

1.3. Multi Legged Walking **10**

1.4. Content of the Thesis **12**

2. STABILITY ANALYSIS OF "STANDARD GAITS" **15**

2.1. Introduction **15**

2.2. Definitions and Theorems for Stability Analysis of Periodic Gaits **16**

2.2.1. Definitions 17

2.2.2. Theorems 21

2.3. Conclusion **33**

3. FREE GAIT GENERATION WITH REINFORCEMENT LEARNING **35**

3.1. Introduction **35**

3.2. Discrete Model of Stepping and Explanation of States for the Learning Task **38**

3.3. Free Gait Generation (FGG) **41**

3.3.1. The Rule of Neighborhood 41

3.3.2. Free State Generation Algorithm 44

3.4. Free Gait Generation with Reinforcement Learning (FGGRL) **45**

3.4.1. Updating the utilities of State Transitions 45

3.4.2. Selection or Generation of Transitions 46

3.4.3. Smooth Transition of Gait Patterns for Different Speeds 47

3.5. FGGRL when There is an External Stability Problem of Leg Deficiency **47**

3.5.1. Quasi-Static-Stability for Five-Legged Walking 48

3.5.2. Learning Algorithm for Five-Legged Walking 49

3.6. Simulation Results **51**

3.7. Results with the Actual Robot-EA308 54

3.8. Conclusion 57

4. PROTRACTION OF A THREE-JOINT ROBOT LEG: TRAJECTORY OPTIMIZATION AND CONTROLLER DESIGN 59

4.1. Introduction: Protraction of Legs 59

4.2. Three Joint Robot Leg: Kinematic Model and Dynamic Equations 60

4.3. Trajectory Optimization 63

4.3.1. State Space Representation and Hamiltonian Formulation of a Three-Joint Leg System 64

4.3.2. Optimization with Objective Function Modification 68

4.4. Comparative Results of "Optimization With Single Objective Function" and "Optimization With Objective Function Modification" 70

4.4.1. Manipulation of Optimal Trajectories: 76

4.4.2. RBFNN for Interpolation: 76

4.5. Controller Design 79

4.6. Results 83

4.7. Conclusion 87

5. "TORQUE DISTRIBUTION" IN A SIX-LEGGED ROBOT SYSTEM: KINEMATIC-DYNAMIC FORMULATION AND ENERGY OPTIMAL DISTRIBUTION 89

5.1. Introduction 89

5.2. Kinematic and Dynamic Modeling of a Six-Legged Robot With Three-Joint Legs 92

5.2.1. Kinematics of a Single Three-Joint Leg 94

5.2.2. Kinematic Equations of the Links (Vectorial Relations) 95

5.2.3. Force and Moment Equations Based on Free Body Diagrams (Vectorial Relations) 97

5.2.4. Torque Calculations 100

5.2.5. Relating the Body Equations to Tip Point Forces and Joint Torques 101

5.2.6. Friction Constraints 104

5.2.7. The Minimization Problem 106

5.2.8. Computational Cost of the Formulation 106

5.3. Torque Distribution: Solution of the Quadratic Programming Problem 107

5.4. Comparison of "Torque Distribution" with Tip-point "Force Distribution" 110

5.4.1. Observations on Stationary Standing 111

5.4.2. Observing the Effect of Friction Coefficient 112

5.4.3. Observations on Tripod Gait Walking 113

5.5. Conclusion 116

6. ANALYSIS OF WAVE GAITS FOR ENERGY EFFICIENCY 119

6.1. Introduction 119

6.2. Energy Efficiency Analysis of Gaits Based on Simulation Data 122

6.2.1. Relations Between the Gait Parameters 123
6.2.2. Observations for the Constant Stroke – Wave Gaits 124
6.2.3. Observations for the Constant Stroke - Phase Modified Wave Gaits 127
6.2.4. Observations for the Constant Protraction Time – Wave Gaits 129
6.2.5. Observations for the Constant Protraction Time – Phase Modified Wave Gaits 130
6.3. Heuristic Strategies to Find an Energy Efficient Standard Gait 132
6.4. Testing the Strategies on the Experimental Setup 133
6.5. Conclusion 136
7. SUMMARY AND CONCLUSION: CONTRIBUTIONS 138

REFERENCES 143
APPENDIX 148
A. Inertia Matrices for the Body Frame and the Three-Joint Legs of the Six-Legged Robot-EA308 148
B. Entries of the Matrices in the Equation of the Dynamic Motion of a Three-Joint Leg 151
C. Joint Related Position Vectors 153
D. MATLAB Fuzzy System File for the Joint Acceleration Controller 153
CURRICULUM VITAE 155

LIST OF TABLES

TABLES

Table 3.1: Successive epochs and commands sent to the simulation robot within one episode of FGGRL. 51

Table 4.1: Denavit-Hartenberg link parameters for the three joint robot leg. 61

Table 4.2: Sequential values of the weights of the objective function for "objective function modified optimization" 69

Table 4.3: Set of initial and final tip-point positions used for optimization. 70

Table 5.1: The masses of the components of the Robot EA308. 91

Table 5.2: Computational cost of formulation for a three-joint *n*-legged robot with *m*-legs supporting. 107

Table 5.3: Two examples of standing on flat surface for comparison of "torque distribution" with "force distribution". 112

Table 6.1: *Ecpl.* values for wave gaits with constant stroke. 125

Table 6.2: *Ecpl.* values for phase modified wave gaits with constant stroke 128

Table 6.3: Comparison of the best *Ecpl.* values applicable to the Robot-EA308 for the constant stroke – wave gaits and constant stroke – phase modified wave gaits. 128

Table 6.4: Comparison of stability margins for the constant stroke – wave gaits and constant stroke – phase modified wave gaits. 128

Table 6.5: *Ecpl.* values for wave gaits with constant protraction time 129

Table 6.6: *Ecpl.* values for phase modified wave gaits with constant protraction time 131

Table 6.7: Comparison of the best *Ecpl.* values applicable to the Robot-EA308 for the constant stroke length wave gaits and constant protraction time phase modified wave gaits 131

Table 6.8: Comparison of the stability margins for the best applicable *Ecpl.* cases for the constant stroke length wave gaits and constant protraction time phase modified wave gaits 131

Table 6.9: The *acEpl.* values for different gait patterns for the Robot-EA308. *(a)* Constant stroke – wave gaits; *(b)* constant stroke – phase modified wave gaits; *(c)* constant protraction time – wave gaits; *(d)* constant protraction time – phase modified wave gaits. 134

Table A.1: Parameters for inertia matrix calculation of the body frame. (These parameters belong to the Robot-EA308) 149

Table A.2: Parameters for the inertia matrix calculation of leg-links. (These parameters belong to the legs of the Robot-EA308). 150

LIST OF FIGURES

FIGURES

Figure 1.1: The robot EA308 developed in our laboratory. 4

Figure 1.2: Tetrapod and tripod gaits for six legged insects (statically stable gaits)........................... 5

Figure 1.3: Three examples of four-legged animal gaits (dynamically stable gaits): (a) Ox in walk gait, (b) camel in rack gait, (c) horse in transverse-gallop gait............................... 5

Figure 1.4: Schematic diagram showing the oscillatory structure of the Pearson model. 8

Figure 1.5: Schematic representation of the six mechanisms in the Cruse model.............................. 9

Figure 2.1: The numbering of the six legs and the parameters related to legs-body structure. 17

Figure 2.2: Support polygon, front and rear longitudinal stability margins for the six-leg case. 19

Figure 2.3: Demonstration of the enclosure relations between the subsets of orderly gaits.............. 21

Figure 3.1: Discrete positions of stepping, and the possible transition for a velocity of two units per iteration. 39

Figure 3.2: Examples of stable and unstable support polygons. 40

Figure 3.3: Figure showing the pitch (P) and stroke (R) lengths. 42

Figure 3.4: Results for a successful FGGRL in computer simulation.*(a)* Reward for each step, *(b)* average reward of the last ten steps for each step, *(c)* stability margin for each step, *(d)* average stability margin of the last ten steps for each step, *(e)* number of memorized states for each step, *(f)* average number of utilization of the memorized transitions or states, *(g)* speed of the robot at each step. 53

Figure 3.5: Experimental setup for the Robot-EA308 for learning of five-legged walking............. 54

Figure 3.6: Results for a successful FGGRL on the actual Robot-EA308. *(a)* Reward for each step, *(b)* average reward of the last ten steps for each step, *(c)* stability margin for each step.. 55

Figure 3.7: Slights showing the Robot-EA308 while walking with the five-legged gait pattern composed of the sequential states [0 5 0 3 5 ~~3~~]→[4 0 2 4 0 ~~4~~]→[5 0 3 0 4 ~~0~~]→[0 5 0 3 5 ~~3~~]. 55

Figure 3.8: Results of a five-legged learning episode, in which the “on-the-border” problem is observed. *(a)* Reward for each step, *(b)* average reward of the last ten steps for each step, *(c)* the first and second patterns of states with their stability margins. 56

Figure 4.1: Three-joint robot leg: Reference frames and joint variables.. 61

Figure 4.2: Results of "optimization with single objective function". First row: joint angles; second row: joint angle velocities; third row: joint angle accelerations; fourth row: tip point position errors. 73

Figure 4.3: Results of “optimization with objective function modification". First row: joint angles; second row: joint angle velocities; third row: joint angle accelerations; fourth row: tip point position errors.. 74

Figure 4.4: Results of "optimization with single objective function" (the first row and the dashed lines in the last row) and "optimization with objective function modification" (the second row and solid lines in the last row). The result of "optimization with single objective function" is infeasible. 75

Figure 4.5: Results of "optimization with single objective function" (the first row and the dashed lines in the last row) and "optimization with objective function modification" (the second row and solid lines in the last row). The result of "optimization with single objective function" is feasible. 75

Figure 4.6: Effect of the modification on optimized trajectories. 76

Figure 4.7: Trajectory interpolating RBFNN. 77

Figure 4.8: The trajectories produced as a result of the interpolation with the RBFNN for two trained input of initial-final tip point positions........ 78

Figure 4.9: The trajectories produced as a result of the interpolation with the RBFNN for two untrained input of initial-final tip point positions........ 78

Figure 4.10: Representation of pseudo joint angle error for joint *j* of a multi-link revolute manipulator. 80

Figure 4.11: Block diagram of the overall control system of protraction. 82

Figure 4.12: SIMULINK block diagram illustrating the realization of overall protraction control. . 83

Figure 4.13: First example for comparison of the trajectory results generated by the protraction controller with the optimized results. The initial and final tip point positions are [0.11, -0.04, -0.09] and [0.09, 0.06, -0.09], respectively........ 85

Figure 4.14: Second example for comparison of the trajectory results generated by the protraction controller with the optimized results. The initial and final tip point positions are [0.09, -0.06, -0.09] and [0.11, 0.04, -0.09], respectively........ 86

Figure 4.15: Protraction of the right middle leg of robot EA308, following the trajectory generated by the controller. The initial and final tip point positions are [0.11, -0.06, -0.09] and [0.11, 0.06, -0.09], respectively........ 86

Figure 5.1: The free body diagrams for the robot body, and three links of one leg. 99

Figure 5.2: Effect of the friction coefficient on dissipated energy for "force distribution" and "torque distribution" schemes. 113

Figure 5.3: SIMULINK block diagram for the calculations of the six-legged robot model. (The robot body is moving with constant speed.) 114

Figure 5.4: The leg configuration and the tripod gait plan for four steps........ 117

Figure 5.5: The graphs of joint-torques for tripod gait walking on level surface with "torque distribution"........ 117

Figure 5.6: The graphs of joint-torques for tripod gait walking on level surface with "force distribution"........ 118

Figure 5.7: The ratios of the horizontal over vertical components of the tip point forces for “torque distribution” (left) and “force distribution” (right) cases. 118

Figure 6.1: Stroke trajectory for the left side legs for duty factor *0.875.* Low level: power stroke (retraction); high level: return stroke (protraction)........ 126

Figure 6.2: Experimental setup for recording energy dissipation of the Robot-EA308 while walking........ 134

Figure A.1: Robot body frame and reference frame configuration for the calculation of inertia matrix. 149

Figure A.2: Sample leg-link and the reference frame configuration for the calculation of the related inertia matrix. 150

CHAPTER 1

INTRODUCTION

1.1. A Comparison of Wheeled, Tracked, and Legged Locomotion

Although many animals in nature have legs for locomotion, the very first vehicles developed by human have been with wheels. Following the invention of the steam engine and widespread use of the railways, and then the development of the combustion engines, wheeled locomotion has become the most widespread technology of transportation. Despite its success on predetermined and plane surfaces, wheeled locomotion is not proper for unknown and rough terrains. The tracked (palette) locomotion is developed in order to cope with this problem. However, tracked locomotion is also not without problems, since it destroys the terrain on which the vehicle is moving. As an alternative to both wheeled and tracked forms, legged locomotion is developed by imitating the legged animals in nature (Mahajan et al., 1997).

The terrain properties are of most importance in determining the kind of robot to be used in applications. The “geometric properties” such as roughness and inclination, the “material properties” such as friction and hardness, and “temporal properties” such as changing of surface in time, like on the river surface under the water, are three items cited in Hardarson (1998) in this regard. Besides these terrain properties, the mission to be performed and the criteria efficiency are important items to be considered in choosing the robot. Following Hardarson (1998), it is worth here to mention the advantages and disadvantages of wheeled, tracked and legged locomotion respectively.

The wheeled locomotion is most advantageous in hard and even terrains. It provides smooth and fast locomotion with a high payload-weight-to-mechanism-weight ratio, and efficient energy consumption. However, wheeled locomotion is disadvantageous in uneven and smooth terrains. In the existence of obstacles higher than the radius of the wheels or big holes on the surface, the wheeled locomotion becomes impossible. Since the wheeled vehicle follows exactly the ground surface the locomotion is subject to rocking and vibrations due to the steps and holes on the ground. If the surface is not hard enough, for example sandy, muddy or snowy surfaces, the wheels are unable to drag the vehicle. From technology and application point of view, the manufacturing and control of wheeled systems is simple, very much studied and widespread in all field of transportation. This simplicity of application makes wheeled systems attractive also for robotics researchers.

Track locomotion is advantageous especially on loose surfaces, in unknown environments with steep hills, big holes and obstacles. Since the palettes provide a wide surface to hold on the ground, traction on sandy and snowy surfaces become possible. The technology of tracked locomotion is also simple, well developed, and the control is again easy. It has a good payload capacity. Despite the good payload capacity, high-energy consumption is an important disadvantage of tracked vehicles. The machine has to be heavy because it carries the whole palette mechanism. Due to the friction in the tracks there is a considerable amount of energy loss. Another important disadvantage is that tracked locomotion damages a wide part of the surface. Especially the turning of palette vehicles damages the surface since it has to be with slip friction. The vehicle contacts the ground with a plate, inflexible surface; therefore it suffers from the sharp orientation changes on the ground, mostly due to large-hard obstacles (rocks) and sharp inclinations.

Legged locomotion is a proper solution for movements on loose-rough-uneven terrains. This advantage of legged locomotion is mostly due to the fact that legged systems use isolated footholds. Wheeled and tracked systems follow the surface in a continuous manner; therefore their performance is limited by the worst parts on the terrain. A legged system, on the other hand, can choose the best places for foot placement. These footholds are isolated from the remaining parts, hence the performance of the legged system is limited by the best footholds. Besides using isolated footholds, the legged system can provide active suspension, which does not exist in wheeled or tracked systems. This means that the system can have control on the force distribution through the foothold points. In this way an efficient utilization of the footholds provides further improvement of the vehicle-ground interaction. A legged system is well adaptive to uneven terrains, namely the legs can be arranged (lengthened and shortened) according to the level changes, and they can jump over obstacles or holes. Therefore, the body can be moved in a desired orientation, for example horizontally, whatever the ground structure is. Considering the vehicle-environment interaction, the legged locomotion is the most advantageous, since it gives the less damage to the surface compared to wheeled and tracked systems.

The legged locomotion is disadvantageous considering the system control and energy consumption. Legged mechanisms have complicated kinematics and dynamics, and a lot of actuators have to be controlled in continuous coordination; therefore, control of legged systems is more difficult in comparison to wheeled and tracked systems. Since they are comparatively novel development, there are no well-established technologies for legged systems. Another drawback of legged systems is the high-energy consumption due to the active suspension of the body by the supporting legs; therefore, the payload-weight-to-mechanism-weight ratio is considerably low compared to wheeled and tracked systems. In multi legged systems generally each joint is driven by a separate motor, which is mostly the heaviest component in the system (there are designs outside of this general scheme (Ota et al., 1998)). Therefore, in general, there are as many motors as the joints in the multi legged systems. Since these motors are generally placed on the links of the legs (again different designs outside this general scheme exist (Gonzales de Santos et al., 2004; Lin and Song, 2001; Clark et al., 2001)), the weights of the legs are quite comparable to that of the main robot body. In the general scheme, the actuators on a leg are serially connected. Therefore the weight a leg can carry is limited by the power of the weakest actuator on the leg. While walking, some of the legs are in the swing phase for protraction (three of the legs are in swing in tripod gait), which means that they do not contribute to lifting although they exists as extra weight. These factors due to the "legs" also contribute to the low ratio of payload-weight-to-mechanism-weight of legged-systems. Another disadvantage mentioned in Hardarson (1998), is that the vehicle suffers a force impact in a pulse form in every step. These pulses may result in rocking on the body. (There are studies to overcome such impulses with proper force distribution (Martins-Filho et al., 2000). The results of such studies are promising to take out this last item out of the disadvantages list of legged systems.)

Due to its advantages over wheeled and tracked systems, legged locomotion has found applications in various fields with multi legged robots. Particle gathering from unknown surfaces (of the earth or moon), recoveries from the regions of fire or earthquake disasters, locomotion in forest for harvesting trees (Hardarson, 1998), de-mining tasks (Gonzalez de Santos et al., 2004), and safe transportation in crowded places are some of the areas that legged locomotion can be utilized. The studies to incorporate intelligent systems with robotic applications result in promising developments for learning in control of robot manipulators (Erden et al., 2004). When legged robots are equipped with intelligent self-learning and vision systems, autonomous systems that can adapt to different circumstances emerge.

In this work, a kinematically controlled three-joint six-legged robot, named the Robot-EA308 (Figure 1.1), is developed in our laboratory, to be used in the experimental parts of the study. This robot is only for experimental purposes; therefore, it is limited to indoor usages being connected to an external power supply and controlling computer. The control of the robot is achieved by a MATLAB program, which continuously sends the information for the joint angles of the legs via the serial port of the computer to the servo-controllers on the robot body. The controllers on the body control the angular positions of the servo-motors on the joints. All the mechanical construction and software development is performed during this study. (The motors and servo-controllers are purchased.) This robot is used for calculating the actual energy dissipation while walking with different gait patterns (Chapter 6), for real-time learning of five-legged gait patterns with an external reinforcement signal signaling the falls (Chapter 3), for visual testing and demonstrative purposes for the developed control structure of protraction (Chapter 4).

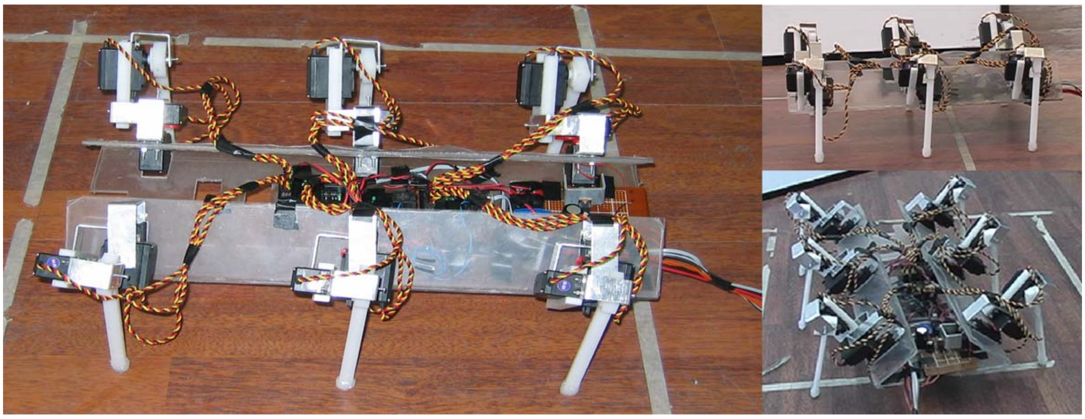


Figure 1.1: The robot EA308 developed in our laboratory.

1.2. Biological Inspirations for Legged Locomotion Systems

Many of the animals in nature have adopted legs for various environmental conditions. Centipedes, spiders, cockroaches, cats, camels, kangaroos, and human are among those, either with different number of legs or with different kind of walking. It is understandable that people turned their attention to those walking animals, after it was recognized that the human invented wheeled and tracked systems did not satisfy all the needs. In this sense, legged systems have a peculiarity of imitating the nature. This imitation is obvious in structural similarity between legged robots and imitated animals; however, for today the imitation is not limited to structural design. Today researchers are trying to understand the underlying biological principles of walking in animals, namely the operational and control structures (Hughes, 1965; Wilson, 1966; Pearson, 1976; Cruse, 1979; Cruse et. al, 1983; Cruse, 1990). The results of such biological researches have been utilized in robotics via inspirations. Among many of them are Cruse et al.(1998), Espenschied et al. (1996), Pfeiffer et al. (1995), and Clark et al. (2001).

In research in biological sciences and robotics applications the most important item is the plan and coordination of leg movements. The plan of walking, namely the “gait pattern”, determines the sequence of stepping of legs with their stance and swing durations in each step. Mahajan et al.(1997) gives the following definition for gait: “The gait of an articulated living creature, or a walking machine, is the corporate motion of the legs, which can be defined as the time and location of the placing and lifting of each foot, coordinated with the motion of the body, in order to move the body from one place to another.” Animal gaits can be divided into two main groups as *statically stable* and *dynamically stable*. In statically stable gaits, the ground projection of the center of gravity of the system always lies in the polygon determined by the supporting legs. In every step, a new polygon is formed and the center of gravity always stays inside these polygons. The six and eight-legged animals adopt, mostly, statically stable gaits (Figure 1.2). The disadvantage of such gaits is that the locomotion is considerably slow in order to sustain the static stability. Dynamically stable gaits, on the other hand, are the fast gaits, in which the center of the body is always in a dynamical motion. In dynamically stable gaits the center of gravity is not always inside a polygon determined by the supporting legs. This means that, in dynamically stable gaits there occur orientations in which the animal cannot stand statically without falling down. Therefore, the animal has to be in motion in all phases of dynamically stable gaits. Two legged animals have to adopt dynamically stable gaits, and four legged animals mostly adopt dynamically stable gaits, although they can also adopt statically stable gaits (Figure 1.3). The *walk* gait that is adopted by almost all four legged animals, the *trot* gait of horses, the *rack* gait of camels, the *canter* and *gallop* gaits of horses, and the *bound* gait of dogs are some of the dynamically stable gaits of four legged animals in nature. (Lasa, 2000)

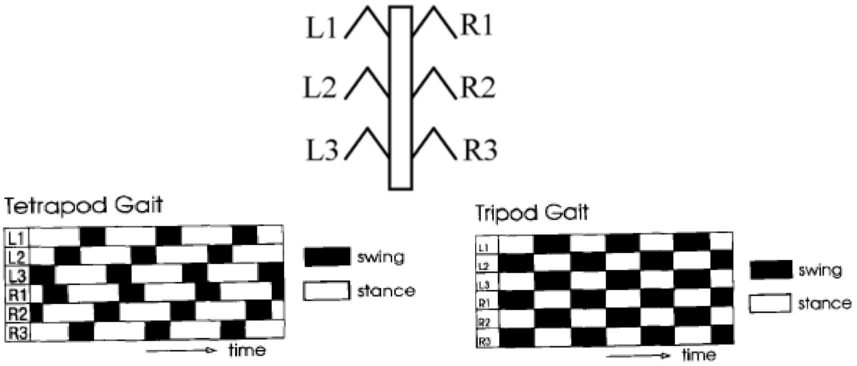
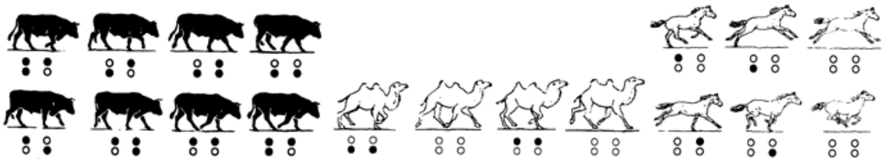


Figure 1.2: Tetrapod and tripod gaits for six legged insects (statically stable gaits) [Adapted from Pfeiffer et al. (1995)].



(a) (b) (c)

Figure 1.3: Three examples of four-legged animal gaits (dynamically stable gaits): (a) Ox in walk gait, (b) camel in rack gait, (c) horse in transverse-gallop gait. [Adapted from Lasa (2000).]

The aim of the studies in biological sciences is to understand the mechanisms underlying the gaits of living creatures and to develop models based on those. In robotic applications modified versions of such models are applied to artificial robots. In Ferrel (1995), six important terms that are frequently used in describing gaits are given. In Pfeiffer (1995), three variables that fully define a regular gait pattern, in which all the legs perform the same movements with phase differences, are given. For convenience of reading the introduction chapter here will be given slightly modified definitions of these nine terms adopted from Ferrel (1995) and Pfeiffer (1995). (For some of these terms a more mathematical definition is given in Chapter 2.)

Protraction: The movement of the leg in which the leg moves towards the front of the body.

Retraction: The movement of the leg in which the leg moves towards the rear of the body.

Power Stroke: The phase of stepping of one leg, in which the foot is contacting the ground, the leg is supporting the body (contributing to lifting) and propelling the body. In forward walking, the leg retracts during this phase. It is also called *stance phase* or *support phase*.

Return Stroke: The phase of stepping of one leg, in which the foot is in the air, being carried to the starting point of the next power stroke. In forward walking the leg protracts during this phase. It is also called *swing phase* or *recovery phase*.

Anterior Extreme Position (AEP): The extreme forward position that the foot can be placed. In forward walking, this is mostly the target of the foot in return stroke, and starting point of the foot for power stroke.

Posterior Extreme Position (PEP): The extreme backward position that the foot can be placed. In forward walking, this is mostly the position of the foot where the power stroke ends, and the return stroke starts.

Duty Factor: The ratio of the part of the walking cycle during which the leg makes contact with the ground.

Ipsilateral Phase: The phase lag between two neighboring legs on the same part of the body.

Contralateral Phase: The phase lag between opposing legs in the two parts.

The six-legged walking research reveal that insects mostly adopt regular walking (*regular-periodic gaits*) in which all the legs perform periodic motions; the motions of different legs are similar in structure with the same period, but differ in their phase of oscillation. The speed of walking is determined by the speed of oscillation of the legs. Research results reveal that six-legged insects adopt different gait patterns according to the speed of walking (Figurw 1.2; Pfeiffer et al., 1995). In slow walking the insects use gaits in which most of the legs have contact with the ground (for example the tetrapod wave gait in which four legs have contact to the ground at any time); when the speed is increased the gait changes towards the tripod gait where three legs have contact to the ground at any time. This passage between gaits is a smooth one and achieved by continuous change of the parameters mentioned by Pfeiffer (1995) (the last three parameters above).

Ferrel (1995) gives a brief review of the three biologically inspired models for six-legged insect gaits, proposed by Wilson (1966), Cruse (1990), and Pearson (Pearson, 1976; Wilson, 1966). Wilson's model is a descriptive one based on rules imposed on the variables and parameters. In fact, the work of Wilson seems to have established the basic principles that are used also in the other two models. The Pearson model is based on neurological models and oscillators working together to control the leg movements, speed, and coordination. In the Pearson model the motor neurons, the oscillators and the command neuron maintain the rhythmic walking. The servomechanism and reflex modules modify this rhythmic walking according to the feedbacks. Ferrell (1995) argues that this modification may even lead to walking with five legs in case of leg amputation. The Cruse model is based on mechanisms that excite or inhibit the starting and ending of return and power strokes. The modifications on the PEP make up the basics of the Cruse model. The other mechanisms can be considered as modifications on the basic pattern. In the following will be given a summary of these three models based on the explanations in the mentioned references.

The rules in the *Wilson model* are as follows: (adapted from Ferrell, 1995; Donner, 1987)

1. A wave of protraction runs from rear to front, and no leg recovers from the power stroke until the one behind is placed in a supporting position.
2. Contra-lateral legs of the same segment alternate in phase.
3. Protraction time is constant.
4. Frequency of the oscillation of legs varies (retraction time decreases as frequency increases).
5. The intervals between the steps of the hind leg and the middle leg and between the middle leg and the foreleg are constant, while the interval between the steps of foreleg and hind leg varies inversely with frequency.

Basic elements in the Pearson model (refer to Figure 1.4): (adapted from Ferrell, 1995)

1. *Flexor motor neurons***:** The neurons that protract the leg when activated. A flexor motor neuron exists for each leg.
2. *Extensor motor neurons***:** The neurons that retract the leg when activated. An extensor motor neuron exists for each leg.
3. *Oscillator*: Generates the stepping pattern by controlling the activation of the flexor and extensor neurons. Protraction time is constant; retraction time varies with the frequency of oscillation. Speed of walking is determined by the frequency of oscillation. The oscillators that control contra-lateral legs of the same segment oscillate with 180^0 phase difference. The three oscillators along either side of the body maintain the same temporal lag. When the frequency of oscillation changes, the phase difference between the ipsilateral legs changes due to the constant temporal difference. Therefore, changing the oscillation frequency generates different gaits. It should be noted that there exist a unique gait for each frequency of oscillation, namely for each value of speed. An oscillator exists for each leg.

4. *Command neuron(s)*: The neuron that determines the speed of walking by changing the frequency of oscillators controlling the legs.
5. *Servomechanism circuit*: Uses feedback from the joint receptors to adjust the strength of the supporting and pushing according to the load.
6. *Reflex circuit*: Delays or prevents the command that protracts the leg forward. This reflex considers whether another leg has taken up some of the load. This reflex is important in cases of amputated leg or gait transition.

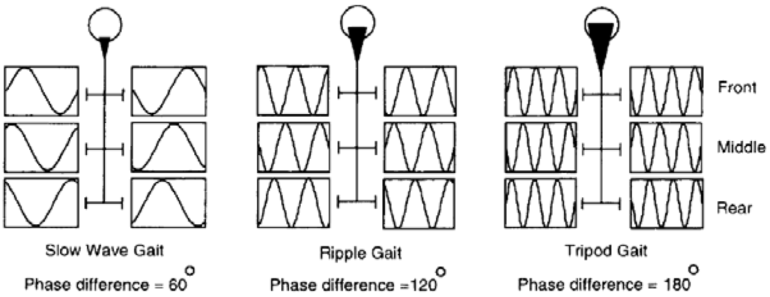


Figure 1.4: Schematic diagram showing the oscillatory structure of the Pearson model. The oscillators that control contra-lateral legs of the same segment oscillate with 180 degrees phase difference. The three oscillators along either side of the body maintain the same temporal lag. When the frequency of oscillation changes, the phase difference between the ipsilateral legs changes due to the constant temporal difference. Therefore, changing the oscillation frequency generates different gaits. [Adapted from Ferrell (1995).]

The six mechanisms of the Cruse model can be mentioned as follows (refer to Figure 1.5): (adapted from Cruse et al., 1998; Ferrell, 1995)

1. A rostrally directed influence inhibits the start of the return stroke in the anterior leg by shifting the PEP to a more posterior position. This is active during the return stroke of the posterior leg.
2. A rostrally directed influence excites the start of the return stroke in the anterior leg by shifting the PEP to a more anterior position. This is active during the beginnings of the power stroke of the posterior leg. It also applies to the contra-lateral leg.
3. A caudally directed influence, depending upon the position of the anterior leg, excites the start of a return stroke in the posterior leg by shifting the PEP to a more anterior position. The start of the return stroke is more strongly excited (occurs earlier) as the anterior leg is moved more rearward during the power stroke. This causes the posterior leg to perform the return stroke before the anterior leg begins its

return stroke. This is active during the power stroke of the anterior leg. It also applies to the contra-lateral leg.

4. The AEP is modulated by a single, caudally directed influence depending on the position of the anterior leg. This mechanism is responsible for the targeting behavior; namely the placement of the foot at the end of protraction is made close to the foot of the anterior leg. (This behavior is described as follow-the-leader strategy in Donner (1987), and its facilities for the walking system are discussed in detail. Briefly, the advantage of follow-the-leader strategy can be explained as follows: With this strategy the rear legs follow the placement of the one in the front. The animal searches foot places only for the front legs. Afterwards it is assured that the middle and hind legs will automatically be placed in safe positions.)
5. This mechanism has to do with distributing propulsive force among the legs.
6. This mechanism serves to correct errors in foot placement.

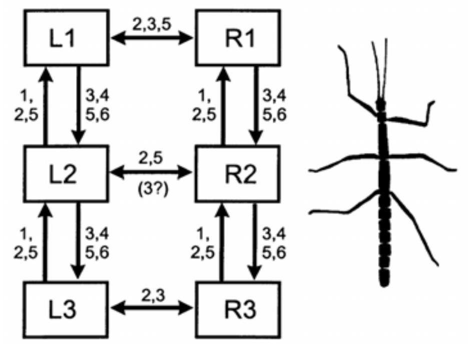


Figure 1.5: Schematic representation of the six mechanisms in the Cruse model. [Adapted from Cruse et al. (1998).]

The Wilson and Pearson models are more based on regular walk of six legged insects, namely they give descriptions and parameters of *regular-periodic gaits*, in which the legs make the same movements with phase differences. These two models do not seem to be compatible with *free gaits,* in which at least one of the legs on one side has a differently structured stepping (the duty factor of some of the legs may be different than the others, or there can be no constant ipsilateral or contralateral phase differences, or the protraction time may be different for different legs). Different structured legs, different robot configurations, and results of some learning algorithms might necessitate usage of free gaits. Especially deficiency or non-symmetries in the legs might necessitate adopting free gait patterns. Among the three models, the Cruse model seems to be more compatible with those. The Cruse model does not impose structured movement but it defines mechanisms whose results depend on the existing condition of the legs. For example, compared to the other two models, the Cruse model seems to be more applicable to five legged walking with some modifications on the excitation and inhibition mechanisms. Therefore, the Cruse model seems to be most significant of the three for robotic researches dealing with free gaits.

The above three models point to a common structure for six-legged systems. In this structure there is a central pattern generator mechanism and a reflex mechanism. (Klaassen, 2002) The central pattern generator mechanism produces rhythmic motions without the need of sensory feedback. It corresponds to the open loop control in the overall system. The central pattern generator mechanism is able to manage walking in normal conditions where no exceptions, such as obstacles, deficiencies, or slippage occur. Considering the lists for the three models above, all the rules of the Wilson model, the first four elements for the Pearson model, and the first three mechanisms of the Cruse model correspond to this central pattern generator mechanism in their own models.

The reflex mechanism, on the other hand, compensates for the deficiencies from the central pattern due to unexpected effects. It takes feedback from the actuators and tip point sensors, and modifies the central pattern accordingly. The reflex mechanism corresponds to the closed loop control in the overall system. Such reflexes may prolong the retraction phase, may increase the height of the foot in protraction, or may change the force on actuators. The fifth and sixth elements in the Pearson model, and the last three mechanisms in the Cruse model correspond to the reflex mechanism in their models. (In Klassen (2002), the Cruse model is interpreted as a model using only reflexes. This interpretation is correct, considering the fact that the Cruse model works only based on inhibitory and excitatory mechanisms, all of which need some feedback. Therefore, all the mechanisms in the list are reflexive. However, since the feedbacks necessary for the first three are only contact information, it makes sense to consider these as the central pattern generation mechanism. Despite of this consideration, it should be again noted that the Cruse model seems to be the most flexible one to handle free gait patterns.)

1.3. Multi Legged Walking

In light of the above reviewed biological inspirations, here will be given five items as the tasks to be accomplished in order to maintain a multi legged locomotion control in robotic applications. In literature there can be found various studies on each of these items. It should be noted that each of these items is a field of research in robotics, and there are various approaches and very different solutions for every of them. In these works, mostly the interested item of the research is studied in detail. The remaining items are mostly left out of the study either with the assumption that they are already maintained at the background, or with utilizing very simplified-straightforward solutions. Here, the listing below is intended to be a guide in classifying these researches, and to give an idea about the total detailed work on multi legged robots.

A generalized list of items of tasks for the work of a multi-legged robot system is as follows:

1. A *gait pattern generation module*.
2. A *coordination and control module* that controls the legs according to the gait pattern generator and compensates unexpected disturbances.
3. Control of *protraction* of one leg.
4. Control of *retraction* of one leg.
5. Regulation of the *force distribution* to the legs.

Gait pattern generation corresponds to the central pattern generation; namely, the generation of rhythmic plan that determines which leg will be in power or return stroke in any time. This plan also determines the duration of these strokes for each leg. Based on the biological inspirations above, this module is an open loop command generator that takes no feedback from the legs. It outputs the power or return stroke commands for the legs. In Pfeiffer (1995) an oscillatory gait pattern generator, very close to the Pearson model, is adopted, where the parameters of oscillation determine the gait and speed of walking. In Cymbalyuk (1998) the Cruse model is adapted, and the parameters of the mechanisms are optimized to obtain the most stable gait pattern generator. In Mahajan (1997) and Huang and Nomani (2003) constant gait patterns are adopted, where the gait pattern generator continuously generates these constant gaits.

The coordination and control module controls the legs according to the commands of gait pattern generator and unexpected disturbances from external or internal effects. The stroke commands of the gait pattern generator and the feedback from the legs (actuators and tip points) are input to the coordination and control module. In normal conditions, namely in existence of no disturbances, the coordination and control module directly applies the command of the gait pattern generator. However, in case of disturbances (a leg may be deficient, there may be obstacles or holes, the ground may be slippery), it delays or modifies the commands of the gait pattern generator. Furthermore, it adopts some reflexive behaviors for the legs in special cases. For example in case of an obstacle or a hole, it commands the leg to adopt search reflexes that would result to place the foot above, right or left of the obstacle or on the other side of the hole (Espenschied, 1996). Turning is also a task of the coordination and control module. It achieves turning by changing the AEP and PEP of the legs. The coordination and control module outputs commands for the control of retraction and protraction of the legs. Based on these, the coordination and control module can be considered as an intermediary reflex and control level between the central pattern generator and the individual legs. In Celaya and Porta (1998), the advancement of the robot is determined by an optimization on the coordinated retraction movement of the legs. In every step the PEPs of the supporting legs are determined by this optimization. This might be considered as an example of the coordination and control module that maintains the advancement of the robot in each step.

Retraction and protraction are the two basic movements for stepping of legs. Every leg has a retraction and protraction module. These modules are excited by the coordination and control module, with the input of AEP and PEP in every step. The retraction module generates the retraction movement from the given AEP to PEP; the general protraction module generates the protraction movement from the given PEP to AEP (Erden and Leblebicioğlu, 2004). The retraction module performs the retraction of the supporting legs whose trajectory are definite once the movement of the robot body is given. The protraction module performs the protraction of the returning legs, whose trajectory are not definite but might be the subject of an optimization. Therefore, while retraction is trivial once the body motion is given, protraction is non-trivial with existence of many possibilities. The protraction and retraction modules send the angle commands to the actuators.

The force distribution module manages the regulation of torques on the joints of both retracting and protracting legs. Force distribution to supporting legs is an important problem in multi legged systems. In statically stable robots at least three legs should support the body at any time. The equations relating the body dynamics to feet-ground contact forces make up an indeterminate system of six equations for nine (when three legs are on the ground) or more (more than three legs are on the ground) unknown foot forces. Therefore, the foot-ground interaction forces can be distributed in infinitely many ways. The problem in force distribution is to find the most efficient distribution among the infinitely many possibilities. The legs in protraction should also be dynamically controlled according to the trajectory of protraction. Given the body acceleration and leg kinematics the required torques on protracting leg-joints can be found using conventional robotic formulations. Therefore, while determination of forces on the protracting legs is a matter of calculation, the distribution of forces to the supporting legs is a matter of optimization among the infinite possibilities. The solution of the force distribution in all legs, maintains the required motion of the body platform with the given kinematic positioning of the legs determined by the other modules. The force distribution problem is more related to the dynamics of the robot, while the former items maintain the kinematics of walking.

1.4. Content of the Thesis

In this thesis five pieces of study are performed for the walking of the six-legged Robot-EA308, respectively in the following five chapters. These five chapters do not correspond one by one to the five items of tasks given above, but the five items are still useful for theexplanation of the content.

Chapter 2, and Chapter 6 are directly related to the first item, since they deal with analysis of gaits, and provide insight for constructing the gait pattern generation modules. In Chapter 2 an analytical study of "standard gaits" are performed. Analytical study of gaits is a rather rare work in the literature. In this chapter the definitions and mathematical tools for analysis are provided, a classification of orderly gaits (including the wave gaits) is introduced, various theorems for stability analysis and calculations are stated and proved. It is argued that, the chapter provides an easier and more insightful approach to stability analysis, although its results are redundant from theoretical point of view.

Chapter 3 is related with both the first and second items of tasks. In this chapter a free gait generation algorithm is presented, which can be considered mainly as a task of the first item, but also as a task of the second since the AEP and PEP are inherently determined by the free gait generation. This free generation is equipped with reinforcement learning in order to achieve improvement of stability and adaptation to unexpected conditions. This reinforcement learning can be considered as a reflexive mechanism that adapts the robot to changing environmental conditions, therefore strengthens the chapter's relation with the second item. The adaptation facility of the scheme is presented with learning of five-legged walking in case of deficiency on one of the rear legs. By using the free gait generation with reinforcement learning, the central pattern generation and reflex models are conciliated. The developed free gait generation with reinforcement learning is applied to the Robot-EA308 in real time. The robot improves its stability during regular walking, and learns to walk with five-legs without any falling in case of a deficiency on one of the rear legs. Another approach for conciliating the central pattern generation and reflexive model with reinforcement learning was presented in Erden and Leblebicioğlu (2005).

Chapter 4 is directly related with the third item of tasks, since it aims to find the optimum trajectory and perform the control for a leg in protraction. In this chapter the energy optimal trajectories of protraction are generated by using the optimal control technique, but with some modifications to avoid sticking in local optimums. Using the optimum trajectories, a controller is designed for the three-joint leg system, by making use of the approach of "multi agent system based fuzzy controller design for robot manipulators" (Erden et al., 2004). The performeance of the controller is demonstrated with the Robot-EA308.

Chapter 5 is related with the fifth item of tasks. In this chapter the required forces and moments for the kinematic motion of the robot are distributed to the supporting legs by making use of the approach of "torque distribution". Rather than the tip point forces, the square-sum of the joint torques are minimized. Since the square-sum of joint torques is a close indicator of dissipated power, the distribution proposed in the chapter is regarded as "energy optimal torque distribution". For the distribution, a quadratic programming problem corresponding to the desired objective and constraints, is solved. The protraction movement used in the example of tripod-gait walking in this chapter is generated by the controller developed in Chapter 4.

In Chapter 6, an energy analysis of wave gaits is performed. This chapter, though related with the first item of tasks, is put in the last; since, it extensively makes use of the torque distribution developed in Chapter 5 and the mathematical tools of stability analysis developed in Chapter 2. In Chapter 6, an energy efficiency analysis of wave gaits is performed by tabulating the calculated energy criteria using a simulation model. Based on this analysis, strategies are derived in order to determine the parameters of wave gaits for the most efficient walk. The strategies are justified with using the actual Robot-EA308.

There is no chapter specifically dealing with retraction of legs mentioned as the fourth item of tasks. This is because, as explained before, once the body motion is given, the trajectories of the retracting legs are definite. The only matter with this item is to perform the retraction of all supporting legs in coordination with each other. This is inherently achieved both in simulations and in applications with the Robot-EA308 throughout the work.

Chapter 7 concludes the work by summarizing the contributions of each of the cited chapters.

In the following, each chapter is presented as a work in itself. Although the results of the preceding chapters are used in some of them, these are not significant for comprehension of the work in a single chapter. Therefore, interested reader can read any of the chapters without having read the others. In each chapter there is an introduction section, main body sections, and a concluding section. In the introduction the problem handled in the chapter is introduced with extensive review of the related literature, and the distinguishing features of the chapter are mentioned in relation to those. In the main body sections the approaches to address the highlighted problems are presented. The concluding section summarizes the results of the chapter, and highlights the contributions.

CHAPTER 2

STABILITY ANALYSIS OF “STANDARD GAITS”

2.1. Introduction

Studies on the multi-legged animals in nature and the inspirations derived from those (Ferrell, 1995; Kar et al., 2003; Erden and Leblebicioglu, 2005) reveal that wave gaits are frequently observed in nature because of their high stability. Doner (1987) makes an intuitive illustration of why rear-to-front waves of stepping, which is the case in wave gaits, is more stable than front-to-rear waves. Besides, it mentions the advantage of the follow-the-leader strategy, which is realized with wave gaits, and with which the insect needs to make search for foot placement of only the front legs. As a result of their good stability, wave gaits are widely used in multi-legged walking machines; moreover, they have become the conventional frame of comparison for research results (Inagaki, 1997; Preumont et al., 1991; Pal et al., 1994; Inagaki and Kobayashi, 1994; Ye, 2003).

Despite their wide range of application, there are very few researches on the analytical analysis of the stability of multi-legged gaits. Song and Waldron (1987) provided the basic definitions for the analytical study of wave gaits; then, stated and proved the basic theorems to calculate their stability margin. This paper is the basic source of inspiration for the definitions and theorems presented in this chapter. Later, Song and Choi (1990) published another paper where the range of duty factors, in which the wave gaits are optimally stable, were determined for four, six, and larger number of legs. In this paper the phase conditions violating the general optimality of wave gaits are tabulated one by one, and the ranges of duty factor in which wave gaits are optimally stable for these conditions are given. Considering the six legged walking, this paper proves that the wave gait is optimally stable among all periodic and regular gaits (the definitions are to be given) for all possible values of duty factor (β), namely in the range of $1/2 \leq \beta < 1$. The nine-pages of proof for the single basic theorem in this paper clearly demonstrates how tedious it is to prove the optimal stability of wave gaits in the universe of periodic and regular gaits: the authors had to analyze 19 cases for derivation of the violating conditions, and then 31 cases to check these conditions for the six-legged case, and 5 cases to check them for the eight legged case; totally 55 cases are analyzed one by one. Though the proof is very important to show the optimum stability of wave gaits in the determined ranges, it is overwhelmed by analysis of various particular cases, rather than narrowing down from the general to the optimum. The approach of the paper makes it difficult to get a comprehensive understanding of the nature of stable walk in multi-legged systems. In fact, it is possible to come up with a simpler and more insightful proof of the optimal stability of wave gaits for six (and four)-legged walking if the universe of search is limited to periodic, regular, and constant phase increment gaits, which are defined to be “standard gaits” in this chapter (the definitions are to be given).

In the chapter here the stability margin for the gaits of this limited universe is derived based on some theorems, and the optimality of the wave gaits in this universe is proved for six (and four)-legged walking. To mention again, the ranges of the optimality of the wave gaits are already determined by Song and Choi (1990) considering all the periodic and regular gaits; hence, the optimality of wave gaits among all the standard gaits, which constitute a subset of regular and periodic gaits, is already proved. Therefore, what this chapter does is in fact redundant from a theoretical point of view. However, it is believed that the proof here is significant from analytical point of view for the following reasons: 1) the proof is not based on analyzing single cases one by one, but follows a path of narrowing down from the general (standard gaits) to the particular (wave gaits), 2) therefore, the proof is more comprehensible, 3) it provides more insight to the nature of the stability of wave gaits, and 4) it provides a more systematic approach which can be very useful to analyze the effect of modifications on wave gaits (this fourth item is especially important in relation to the “phase modified wave gaits” introduced in Chapter 6).

Furthermore, this chapter systematizes the basic definitions of multi-legged walking, most of which were provided by Song and Waldron (1987). This systematization is performed by adding new definitions and establishing the enclosure and intersection relations between the defined sets of gaits. As a result, the chapter provides the general set picture of the gaits under the “orderly gaits”. Such systematization is considered to be significant in order to guide the theoretical work on stability of multi-legged walking. Lastly, this chapter provides the analytical proofs of the very basic propositions (e.g., the range of duty factor, and superior stability of rear-to-front waves). To our knowledge, this chapter is the first to attempt to make such systematization and to prove of such basic propositions.

2.2. Definitions and Theorems for Stability Analysis of Periodic Gaits

In this section, some definitions and theorems are given for stability analysis of gaits for multi-legged walking. These definitions and theorems are for the 2n-legged system in Figure 2.1. The definitions in this section are adopted with some modifications and additions from Song and Waldron (1987). At the end of the section, the set relations among the defined gaits are demonstrated in Figure 2.3. The reader is invited to refer to this figure while reading the definitions.

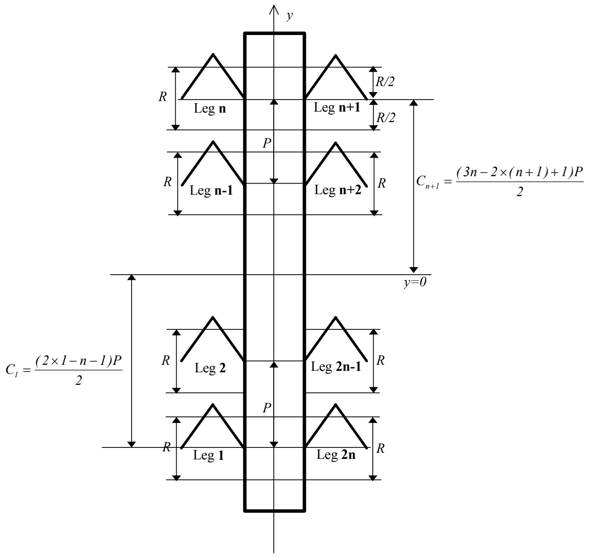

y
R
R/2
Leg n
Leg n+1
R/2
P
R
Leg n-1
Leg n+2
R
$C_{n+1} = \frac{(3n-2\times(n+1)+1)P}{2}$
y=0
$C_1 = \frac{(2\times 1-n-1)P}{2}$
R
Leg 2
Leg 2n-1
R
P
R
Leg 1
Leg 2n
R

Figure 2.1: The numbering of the six legs and the parameters related to legs-body structure.

2.2.1. Definitions

Fractional Function: A fractional function $F[\Psi]$ of a real number Ψ is defined as follows:

$$F[\Psi]=\begin{cases} \text{the fractional part of } \Psi, & \Psi \geqslant 0 \\ 1-(\text{the fractional part of } \{-\Psi\}), & \Psi<0 \end{cases}$$

Orderly Gait: A gait is called orderly if each leg steps with a period of its own, not necessarily being equal to the period of any other leg.

Periodic Gait: An orderly gait is said to be periodic if all the legs have the same period of one step cycle. This period is called the period of the gait. All the legs pass through the same phase of stepping in every period of the gait.

Phase difference: In a periodic gait, the ratio of the duration between two instants of walking to the duration of a full period is called the phase difference between those two instants.

Relative Phase and Local Phase: In a periodic gait, the phase difference with which a leg is leading the left-front leg (*leg-n*) is called the relative phase of the leg; the relative phase of *leg-i* is denoted by ϕ_i. By definition, ϕ_i is restricted to $0 \leq \phi_i < 1$. The self-phase difference of the leg with respect to the phase of putting the foot on the ground is called the local phase of the leg; the local phase of *leg-i* is denoted by ψ_i. For example, if the local phase of *leg-n* is given by t, and *leg-i* is leading *leg-n* with a phase difference of ϕ_i, the following equations hold:

$$\begin{array}{ll} \psi_n=t & \psi_{n+1}=F[t+\phi_{n+1}] \\ \vdots & \vdots \\ \psi_2=F[t+\phi_2] & \psi_{2n-1}=F[t+\phi_{2n-1}] \\ \psi_1=F[t+\phi_1] & \psi_{2n}=F[t+\phi_{2n}] \end{array}$$

Ipsilateral Phase Difference, $\boldsymbol{\phi}$**:** If a gait has the same phase difference between the successive legs on a side, this phase difference is called the ipsilateral phase difference of that side. By definition, the ipsilateral phase difference is limited to the interval [*0, 1*). Namely, $0 \leq \phi < 1$ holds.

Contralateral Phase Difference, $\boldsymbol{\varphi}$**:** If a gait has the same phase difference between the legs of all right-left couples, this phase difference is called the contralateral phase difference of that gait. By definition, the contralateral phase difference is limited to the interval [*0, 1*). Namely, $0 \leq \varphi < 1$ holds.

Constant Phase Increment Gait: A periodic gait is said to be constant phase increment if it has the same ipsilateral phase difference on each side. The ipsilateral phase difference is the constant phase increment of that gait. For a gait with the constant increment of ϕ, the local phases of the legs take the following form, where φ is the contralateral phase difference:

$$
\begin{array}{ll}
\psi_n = t & \psi_{n+1} = F[t+\varphi] \\
\vdots & \vdots \\
\psi_2 = F[t+(n-2)\phi] & \psi_{2n-1} = F[t+\varphi+(n-2)\phi] \\
\psi_1 = F[t+(n-1)\phi] & \psi_{2n} = F[t+\varphi+(n-1)\phi]
\end{array}
\tag{2.1}
$$

Symmetric Gait: A gait is said to be symmetric if the phase difference between the right and left legs (contralateral) is ½. If $\varphi = ½$, in (2.1), then a symmetric constant phase increment gait is obtained.

Support and Return Phases, Retraction and Protraction: Considering the stepping of one leg, the motion has two phases. In the support phase the leg is placed on the ground, supports the body and propels the body towards the direction of the motion. The movement of the leg in the support phase is called retraction. In the return phase of stepping the leg is lifted, and the foot is moved in the air towards the starting point of the support. The movement of the leg in the return phase is called protraction.

Duty Factor: In an orderly gait, each leg has a duty factor of its own, given by the ratio of the duration of the support phase to the duration of full stepping. For the possibility of advancement, the duty factor has to be less than *1*. The duty factor of the *leg-i* is denoted by β_i.

Regular Gait: An orderly gait is said to be regular if all the legs have the same duty factor, β_i.

Standard Gait: A gait is standard if it is (periodic,) constant phase increment, and regular.

Wave Gait: A periodic, regular, symmetric and constant phase increment gait is a wave gait if the constant phase increment is $(1-\beta)$. Namely, the gait defined by (2.1) is a wave gait if $\phi = 1-\beta$ and $\varphi = 1/2$. Wave gaits are rear-to-front gaits, in which a leg starts protraction immediately after the rear leg is put down. Therefore, the gait creates a wave of stepping from rear to front. A wave gait phase distribution can be given as follows:

$$\begin{array}{ll}
\psi_n = t & \psi_{n+1} = F[t+1/2] \\
\vdots & \vdots \\
\psi_2 = F[t+(n-2)(1-\beta)] & \psi_{2n-1} = F[t+(n-2)(1-\beta)+1/2] \\
\psi_1 = F[t+(n-1)(1-\beta)] & \psi_{2n} = F[t+(n-1)(1-\beta)+1/2]
\end{array}$$

For the case of six-legs, the phase distribution of wave gaits take the following form:

$$\begin{array}{ll}
\psi_3 = t & \psi_4 = F[t+1/2] \\
\psi_2 = F[t-\beta] & \psi_5 = F[t-\beta+1/2] \\
\psi_1 = F[t-2\beta] & \psi_6 = F[t-2\beta+1/2]
\end{array} \tag{2.2}$$

Support polygon: It is defined as the polygon whose vertexes are constructed on the horizontal plane by vertical projections of the foot-ground interaction points of the supporting legs. A support polygon is possible only if at least two legs on one side and one leg on the other are supporting. (Figure 2.2).

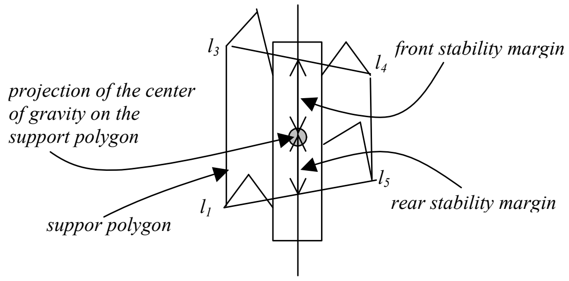


Figure 2.2: Support polygon, front and rear longitudinal stability margins for the six-leg case.

Front and Rear Stability Margins: For any support polygon, the distance between “the projection of the center of gravity on the support polygon” and “the intersection of the front edge (in the walking direction) of the polygon with the longitudinal centerline of the body” is called the *front stability margin*. The distance between “the projection of the center of gravity on the support polygon” and “the intersection of the rear edge (in the opposite direction of walking) of the polygon with the longitudinal centerline of the body” is called the *rear stability margin*.

Longitudinal Stability Margin: The minimum of the front and rear stability margins is called the *longitudinal stability margin* of the support polygon.

(Longitudinal Gait) Stability Margin: The minimum of the longitudinal stability margins of all possible support polygons during walking according to a gait is called the *stability margin* of that gait. If the stability margin of a gait is larger than that of another gait, the former gate is called to be *more stable* than the latter.

Statically Stable Gait: A gait is said to be *statically stable* if its stability margin is greater than zero. Otherwise, the gait is said to be *statically unstable*. If a gait is stable this means that the projection of the center of gravity on the horizontal plane always remains inside the support polygon; otherwise, the center of gravity remains outside the support polygon in some instant of the walk and the body falls down.

Stroke length (R): The distance that the body moves throughout the support phase of a leg. (Figure 2.1)

Pitch length (P): The distance between the centers of stroke of the ipsilateral legs. (Figure 2.1)

Throughout the chapter it will be assumed that the stroke length is less than the pitch length, namely $R<P$, which is the case in Figure 2.1. In Figure 2.3, the enclosure relations of the subsets of “orderly gaits” are demonstrated based on the given definitions. In this figure it is clearly observed that wave gaits are the regular, symmetric, and constant phase increment gaits with the ipsilateral phase difference of $\phi=1-\beta$. Song and Choi (1990) considered the gray colored region of periodic and regular gaits for the general case of 2n-legged walking. In this region, they determined the ranges where the wave gaits are optimally stable. Namely, wave gaits are not optimally stable among all periodic and regular gaits for 2n-legged walking, but only in the determined ranges. However, for the case of six and four-legged walking their paper proves that wave-gaits are optimally stable among all periodic and regular gaits for all possible ranges of duty factors. In this chapter the striped region of standard gaits is considered for six(and four)-legged walking. The stability margin of the standard gaits is derived and the wave gaits are proved to be optimally stable among all the standard gaits for the six(and four)-legged case. This figure clearly demonstrates that the proof of Song and Choi (1990) is more general than the one here. However, a comparison of the analysis of this chapter and that of Song and Choi (1990) reveals that the proof here is much more comprehensible and systematic.

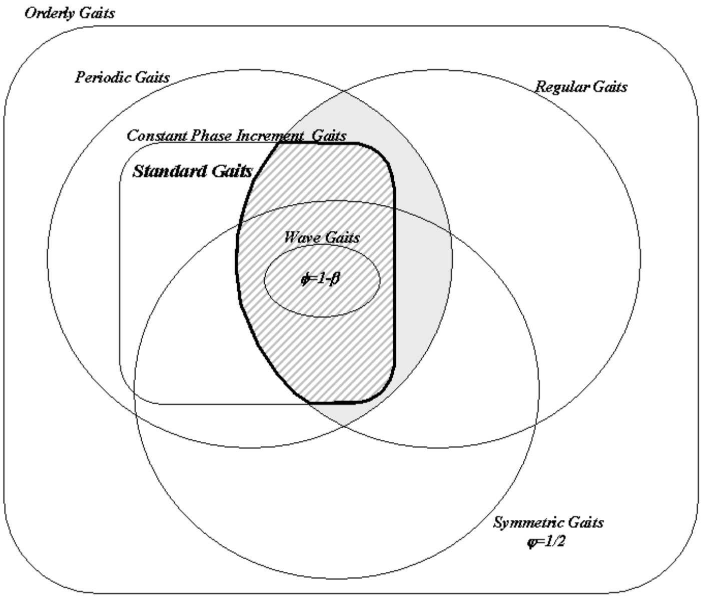


Figure 2.3: Demonstration of the enclosure relations between the subsets of orderly gaits.

2.2.2. Theorems

Song and Waldron (1987) stated and proved four theorems that simplify the stability analysis of periodic, regular, symmetric, and constant phase increment gaits. Afterwards, based on these four theorems and an additional fifth theorem, they derived the formula for the stability margin of wave gaits. The aim here is to generalize the analysis of Song and Waldron (1987) to standard gaits (periodic, regular, constant phase increment gaits), namely to perform the same analysis without the limitation of symmetry and fixation of the constant phase increment to $\phi=1-\beta$. With this generalization the stability formula of standard gaits is derived and the wave gaits, which compose a subset of standard gaits, are proved to be most stable in the universe of standard gaits for six(and four)-legged walking. Before going in those analysis, first the very basic *theorem of duty factor* is proved; then the three theorems of Song and Waldron (1987), which are applicable to standard gaits are stated without repeating the proofs; and lastly the theorems related to standard gaits are stated and proved.

Theorem 2.1: *(Basic Theorem of Duty Factor)* For a 2n-legged periodic and regular gait, the minimum possible duty factor with which static stability can be achieved is $3/(2n)$; namely $\beta\geq 3/(2n)$.

Proof: A gait can be stable only if at least three legs, two being on one side and one being on the other, are supporting at any instant of the walk. If this is violated at any instant the gait is statically unstable. Here will be shown that, this condition is violated for some instant if $\beta<3/(2n)$. Let's assume that $\beta<3/(2n)$ and consider one side of the robot. If none of the any two legs support at the same time, namely if only one leg supports at any time, the duration of total support resulting from all the legs on one side would be βn. This is the longest duration that can be achieved by single leg support on one side. Since $\beta<3/(2n)$, $\beta n<3/2$ holds. It is clear that $1<\beta n$ should hold, otherwise there would occur durations within which no leg is supporting on that side. As a result, the maximum duration of a single leg support would be more than one period, but less than one and a half. Let's denote the excessive part as γ, then $\gamma=\beta n-1$ holds. This excessive part can be transformed to the duration of one period, so that in one period a duration of this excessive amount happens to be supported by two legs. Therefore, the longest duration that can be supported by two legs on one side in a period is γ. If this duration of two-leg support is increased or any part of this duration is supported by more than two legs, in the remaining part of the period, there would occur instants without any support. Therefore, the shortest duration that has to be supported by single leg on one side is $\alpha=1-\gamma=2-\beta n$. Since $\beta n<3/2$ is valid, $\alpha>1/2$ holds. If this shortest duration of single leg support were less than half of the period, this duration would be compensated with the double leg support of the other side. However, this is not the case, and the shortest single leg support on the other side is also greater than half of the period; therefore, the longest double leg support on the other side is less than half the period. This results that there has to occur single leg support on both sides at least for the duration of $\delta=2\alpha-1=3-2\beta n$. Single leg support on both sides results in statical instability. As a result, this duration has to be made less than zero, which necessitates that $3-2\beta n\leq 0$ holds, or equivalently $\beta\geq 3/(2n)$. ■

Lemma 2.1: (Due to Theorem 2.1) The minimum duty factor for a statically stable periodic and regular gait of six-legged walking is $1/2$. Therefore for a six-legged statically stable periodic and regular gait, $1/2\leq\beta<1$. The minimum duty factor for a statically stable periodic and regular gait of four legged walking is $3/4$. Therefore for a four-legged statically stable periodic and regular gait, $3/4\leq\beta<1$.

Adaptations from (Song and Waldron, 1987):

Theorem 2.2: If the body is moved unidirectionally along the longitudinal body axis, then the minimum stability margin in the period of a support pattern is the lesser of the rear stability margin at the beginning and the front stability margin at the end of that support pattern. [Adopted from Song and Waldron (1987), the proof can be found on ibid.p.64.]

Theorem 2.3: If the body is moved unidirectionally along the longitudinal body axis, then the minimum stability margin during a full cycle of walking time is the minimum of the following – the rear margins immediately after a leg is lifted from the rear boundary and the front margins immediately before a leg is placed ahead of the front boundary. [Adopted from Song and Waldron (1987), the proof can be found on ibid.p.64.]

Theorem 2.4: For a 2n-legged periodic, regular, and constant phase increment gait, any support pattern has a second support pattern that is mirror symmetrical with it about the lateral body axis, which goes through the center of the body and is perpendicular to the longitudinal body axis in a locomotion cycle. [Adopted from Song and Waldron (1987), the proof can be found on ibid.pp.65-66.]

Lemma 2.2: (Due to Theorems 2.2-2.4) The stability margin of periodic, regular and constant phase increment gaits (standard gaits) can be found by taking the minimum of all the rear longitudinal stability margins after a leg is lifted from the rear boundary during a full locomotion cycle.

Lemma 2.3: For a 2n-legged walking machine with the configuration given in Figure 2.1, the y component of the support position of a supporting leg with phase ψ_i is given as follows:

$$P_i = C_i + R/2 - \psi_i(R/\beta)$$

where the y component of the position of the stroke center, C_i is given by,

$$C_i = \left\{ \begin{matrix} \frac{1}{2}(2i-n-1)P\,, i \leqslant n \\ \frac{1}{2}(3n-2i+1)P\,, i > n \end{matrix} \right\}$$

Theorems for Standard Gaits:

The Rule of Neighborhood: Considering all the legs in clockwise direction, namely in the circle of *1_2_3_..._(2n-1)_(2n)_1*, if a leg is lifted, then both of the legs in the neighborhood of the lifted leg should be supporting.

Theorem 2.5: The rule of neighborhood for a standard gait for 2n-legs with duty factor β is satisfied if and only if

$$\beta \geq \phi \geq 1-\beta \quad \text{and} \quad \beta \geq \varphi \geq 1-\beta \,. \tag{2.3}$$

Proof: According to the rule of neighborhood, the neighboring legs must be supporting throughout the period that a leg is lifted. It is obvious that the ipsilateral and contralateral phase differences have to be greater than zero; otherwise, the rule of neighborhood will not be satisfied between the successive legs on one side and between the front or rear legs of both sides, respectively. Therefore, $0<\phi<1$ and $0<\varphi<1$ hold from the beginning.

It should be noted that the duration of protraction is less than a period and there is only one continuous duration of protraction throughout a period. Therefore, if the neighboring legs are supporting at the instants that a leg is just lifted and just placed, then the neighboring legs are supporting throughout the protraction of the leg. The proof will be performed for all the legs, and it will be shown that the neighboring legs are supporting at the lifting and placing instants of all the legs provided that the given conditions hold.

Leg 1: At the instant rear-left leg is lifted the local phases of the *legs 2, 1*, and *2n* will be as follows:

$$\psi_2 = F[\beta - \phi] \quad ; \quad \psi_1 = \beta \quad ; \quad \psi_{2n} = F[\beta + \varphi] \, . \qquad (2.4)$$

The condition that *leg-2* is supporting is $0 \leq \psi_2 = F[\beta - \phi] \leq \beta$. If $\beta - \phi < 0$, then $\psi_2 = 1 + \beta - \phi$, and the condition that *leg-2* is supporting leads to $1 + \beta \geq \phi \geq 1$, which cannot happen since $0 < \phi < 1$. Therefore, $\beta - \phi \geq 0$, which leads to $\psi_2 = \beta - \phi$, and hence the condition that *leg-2* is supporting reduces to $\beta \geq \phi \geq 0$.

The condition that *leg-2n* is supporting is $0 \leq \psi_{2n} = F[\beta + \varphi] \leq \beta$. If $0 \leq \beta + \varphi < 1$, then $\psi_{2n} = \beta + \varphi$, and the condition that *leg-2n* is supporting leads to $0 \leq \beta + \varphi \leq \beta$, which cannot happen since $0 < \varphi < 1$. Therefore, $\beta + \varphi \geq 1$, which leads to $\psi_{2n} = \beta + \varphi - 1$, and hence the condition that *leg-2n* is supporting reduces to $1 \geq \varphi \geq 1 - \beta$.

At the instant rear left leg is placed the local phases of the *legs 2, 1*, and *2n* will be as follows:

$$\psi_2 = F[-\phi] \quad ; \quad \psi_1 = 1 \quad ; \quad \psi_{2n} = F[\varphi] \, . \qquad (2.5)$$

The condition that *leg-2* is supporting is $0 \leq \psi_2 = F[-\phi] \leq \beta$; this leads to the condition that $1 \geq \phi \geq 1 - \beta$.

The condition that *leg-2n* is supporting is $0 \leq \psi_{2n} = F[\varphi] = \varphi \leq \beta$; this leads to the condition that $\beta \geq \varphi \geq 0$.

The conditions that *leg-2* is supporting when *leg-1* is lifted and *leg-1* is placed should hold together. Then it is assured that *leg-2* is supporting throughout the protraction of *leg-1*. Therefore, $\beta \geq \phi \geq 0$ and $1 \geq \phi \geq 1 - \beta$ should hold together, which means $\beta \geq \phi \geq 1 - \beta$. The same argument is valid for *leg-2n*, and the conditions lead to $\beta \geq \varphi \geq 1 - \beta$. As a result, the rule of neighborhood around *leg-1* is satisfied only if both of the conditions of (2.3) are satisfied.

Leg i, 1<i<n: At the instant a non-extreme *leg-i* on the left side is lifted the local phases of the *legs (i+1)*, *i*, and *(i-1)* will be as follows:

$$\psi_{i+1} = F[\beta - \phi] \quad ; \quad \psi_i = \beta \quad ; \quad \psi_{i-1} = F[\beta + \phi] \, . \qquad (2.6)$$

At the instant *leg-i* is placed the local phases of the *legs (i+1)*, *i*, and *(i-1)* will be as follows:

$$\psi_{i+1} = F[-\phi] \quad ; \quad \psi_i = 1 \quad ; \quad \psi_{i-1} = F[\phi] \, . \qquad (2.7)$$

The phase difference between *leg-i* and *leg-(i+1)* in (2.6) and (2.7) is the same as the phase difference between *leg-1* and *leg-2* in (2.4) and (2.5). Therefore the condition that *leg-(i+1)* is supporting during the protraction of *leg-i* is the same as the condition that *leg-2* is supporting during the protraction of *leg-1*, namely $\beta \geq \phi \geq 1 - \beta$. The phase difference between *leg-i* and *leg-(i-1)* in (2.6) and (2.7) is the same as the phase difference between *leg-1* and *leg-2n* in (2.4) and(2.5), provided that φ is replaced with ϕ. Since the mathematical analysis will follow the same lines, the condition that "*leg-(i-1)* is supporting during the protraction of *leg-i*" is the same as the condition that "*leg-2n* is supporting during the protraction of *leg-1*", provided that φ is replaced with ϕ. The result is again *β≥ϕ≥1-β*. As a result, the rule of neighborhood around *leg-i* is satisfied only if the first condition of (2.3) is satisfied.

Leg n: At the instant front left leg is lifted and placed, the local phases of the *legs (n-1)*, *n*, and *(n+1)* will be as in the following equations, respectively:

$$\psi_{n-1}=F[\beta+\phi] \quad ; \quad \psi_n=\beta \quad ; \quad \psi_{n+1}=F[\beta+\varphi]$$

$$\psi_{n-1}=F[\phi] \quad ; \quad \psi_n=1 \quad ; \quad \psi_{n+1}=F[\varphi].$$

Leg (n+1): At the instant front right leg is lifted and placed, the local phases of the *legs n*, *(n+1)*, and *(n+2)* will be as in the following equations, respectively:

$$\psi_n=F[\beta-\varphi] \quad ; \quad \psi_{n+1}=\beta \quad ; \quad \psi_{n+2}=F[\beta+\phi]$$

$$\psi_n=F[-\varphi] \quad ; \quad \psi_{n+1}=1 \quad ; \quad \psi_{n+2}=F[\phi].$$

Leg i, n<i<2n: At the instant a non-extreme *leg-i* on the right side is lifted and placed, the local phases of the *legs (i-1)*, *i*, and *(i+1)* will be as in the following equations, respectively:

$$\psi_{i-1}=F[\beta-\phi] \quad ; \quad \psi_i=\beta \quad ; \quad \psi_{i+1}=F[\beta+\varphi]$$

$$\psi_{i-1}=F[-\phi] \quad ; \quad \psi_i=1 \quad ; \quad \psi_{i+1}=F[\varphi].$$

Leg 2n: At the instant rear right leg is lifted and placed, the local phases of the *legs (2n-1)*, *2n*, and *1* will be as in the following equations, respectively:

$$\psi_{2n-1}=F[\beta-\phi] \quad ; \quad \psi_{2n}=\beta \quad ; \quad \psi_1=F[\beta-\varphi]$$

$$\psi_{2n-1}=F[-\phi] \quad ; \quad \psi_{2n}=1 \quad ; \quad \psi_1=F[-\varphi].$$

In the equations of *leg n, (n+1), i (n<i<2n),* and *2n*, in all cases, the phase differences between the supporting legs and their neighbors are one of *ϕ, -ϕ, φ, -φ*. As it is demonstrated for the case of *Leg i,1<i<n,* the mathematical analysis for all these cases will be the same as that of *Leg1* with proper substitutions; therefore all these phase differences will lead to the same conditions derived for the case of *Leg 1*, which can be restated as *β≥ϕ≥1-β* and *β≥φ≥1-β*.

Since the rule of neighborhood is satisfied for all the legs with the conditions of (2.3), these are also the conditions for the rule of neighborhood for the gait.■

Theorem 2.6: *(Stability for the Rule of Neighborhood)* For the "standard gaits satisfying the rule of neighborhood" (SGSRN), the stability margin is given by $S_r=\min\{S_{r1}, S_{r2n}\}$, where

$$S_{r1}=\frac{1}{2}[(2n-4)P/2-R+(R/\beta)(2\beta-\phi+\varphi-1)]$$

$$S_{r2n}=\frac{1}{2}[(2n-4)P/2-R+(R/\beta)(2\beta-\phi-\varphi)] \tag{2.8}$$

Proof: Based on Lemma 2.2, the stability analysis, namely the search for the minimum stability margin throughout one period of walk, can be performed in the rear borders of all the supporting polygons, by analyzing all the instants that a leg is lifted from a rear boundary during a full locomotion cycle. If the rule of neighborhood is satisfied, the lift from the rear boundary occurs when either *leg-1* or *leg-2n* is lifted; because, when *leg-2* or *leg-(2n-1)* is lifted, respectively, *leg-1* or *leg-2n* will already be supporting due to the rule of neighborhood. Therefore, the minimum stability margin when the rule of neighborhood is satisfied is the minimum of margins at the instants *leg-1* and *leg-2n* are lifted.

At the instant *leg-1* is lifted, *leg-2* and *leg-2n* are supporting. The local phases of *legs 2*, *1*, and *2n* are

$$\psi_2 = F[\beta - \phi] \quad ; \quad \psi_1 = \beta \quad ; \quad \psi_{2n} = F[\beta + \varphi] \,.$$

Due to the condition of $-1 < -\phi \leq 0$, for the phase of *leg-2* the condition $\beta - 1 < \beta - \phi \leq \beta$ holds. Since *leg-2* is supporting, this interval has to be shrunk to $0 \leq \beta - \phi \leq \beta$; therefore $0 \leq \psi_2 = \beta - \phi \leq \beta$. Due to the condition of $0 \leq \varphi < 1$, for the phase of *leg-2n* the condition $\beta \leq \beta + \varphi < \beta + 1$ holds. Since *leg-2n* is supporting, this interval has to be shrunk to $1 \leq \beta + \varphi < \beta + 1$; therefore $0 \leq \psi_{2n} = \beta + \varphi - 1 < \beta$. For these local phases, the position of the supporting points of *legs 2* and *2n*, and the resulting rear margin are as follows:

$$\begin{aligned}
&P_2 = (3-n)P/2 + R/2 - (R/\beta)(\beta - \phi) \\
&P_{2n} = (1-n)P/2 + R/2 - (R/\beta)(\beta + \varphi - 1) \\
&\Rightarrow S_{r1} = -\frac{1}{2}(P_2 + P_{2n}) = \frac{1}{2}[(2n-4)P/2 - R + (R/\beta)(2\beta - \phi + \varphi - 1)]
\end{aligned}$$

At the instant *leg-2n* is lifted, *leg-1* and *leg-(2n-1)* are supporting. The local phases of *legs 1*, *2n*, *(2n-1)* are

$$\psi_1 = F[\beta - \varphi] \quad ; \quad \psi_{2n} = \beta \quad ; \quad \psi_{2n-1} = F[\beta - \phi]$$

Due to the condition of $-1 < -\varphi \leq 0$, for the phase of *leg-1* the condition $\beta - 1 < \beta - \varphi \leq \beta$ holds. Since *leg-1* is supporting, this interval has to be shrunk to $0 \leq \beta - \varphi \leq \beta$; therefore $0 \leq \psi_1 = \beta - \varphi \leq \beta$. Due to the condition of $-1 < -\phi \leq 0$, for the phase of *leg-(2n-1)* the condition $\beta - 1 < \beta - \phi \leq \beta$ holds. Since *leg-(2n-1)* is supporting, this interval has to be shrunk to $0 \leq \beta - \phi \leq \beta$ therefore $0 \leq \psi_{2n-1} = \beta - \phi \leq \beta$. For these local phases, the position of the supporting points of *legs 1* and *(2n-1)*, and the resulting rear margin are as follows:

$$\begin{aligned}
&P_1 = (1-n)P/2 + R/2 - (R/\beta)(\beta - \varphi) \\
&P_{2n-1} = (3-n)P/2 + R/2 - (R/\beta)(\beta - \phi) \\
&\Rightarrow S_{r2n} = -\frac{1}{2}(P_1 + P_5) = \frac{1}{2}[(2n-4)P/2 - R + (R/\beta)(2\beta - \phi - \varphi)]
\end{aligned}$$

The stability margin when the rule of neighborhood is satisfied can be stated as, $S_r = min\{S_{r1}, S_{r2n}\}$. ■

Theorem 2.7: For $n \geq 3$, any SGSRN is stable.

Proof: The stability margin of the SGSRN is given as the minimum of the two equations in (2.8). In both of these equations, the term $(2n-4)P/2-R$ is always greater than zero, since $P>R$. Then the phase terms should be dealt with. Due to the satisfaction of the rule of neighborhood the inequalities of $\beta \geq \phi \geq 1-\beta$ and $\beta \geq \varphi \geq 1-\beta$ hold. These inequalities lead to the inequalities of $0 \leq 2\beta-\phi+\varphi-1 \leq 4\beta-2$ and $0 \leq 2\beta-\phi-\varphi \leq 4\beta-2$, which mean that the phase terms in the equations are both greater than or equal to zero. Therefore, both of the margins are greater than zero; hence the stability margin of any SGSRN is greater than zero, namely the gait is stable. ■

Theorem 2.8: Among all the SGSRN with duty factor β, the most stable one is the wave gait corresponding to the duty factor β. This wave gait is given by $\phi=1-\beta$ and $\varphi=1/2$. The stability margin of the wave gait is

$$S_w = \frac{1}{2}\left[(2n-4)P/2 - R + (R/\beta)\left(3\beta - \frac{3}{2}\right)\right] \tag{2.9}$$

Proof: The most stable SGSRN with duty factor β, is the one that maximizes the stability margin, which corresponds to maximizing the minimum of the margins given in (2.8). The variables ϕ and φ, which determine the standard gait, are related only to the terms of $(2\beta-\phi+\varphi-1)$ and $(2\beta-\phi-\varphi)$; therefore the minimum of these two terms should be maximized with respect to these two parameters. Both of the terms are monotone decreasing with respect to ϕ; therefore ϕ should take the least possible value to maximize the minimum of them. The minimum of ϕ, given in (2.3) is $\phi=1-\beta$. The first term is monotone increasing, and the second term is monotone decreasing with respect to φ; therefore, in order to maximize the minimum of the two terms, φ should take the value that equates them. This value is $\varphi=1/2$, which is in the range allowed by (2.3). With these substitutions, the phase terms take the value of $(3\beta-3/2)$. With this phase term, both of the stability margins given in (2.8) take the value given in (2.9), hence the stability margin of the wave gait. ■

Theorem 2.9: Considering six(and four)-legged walking, the wave gait for a given duty factor β is more stable than any standard gait that does not satisfy the rule of neighborhood for the given duty factor.

Proof: First of all, it is obvious that, for four-legged walking, any gait that does not satisfy the rule of neighborhood is unstable (at most two legs are supporting if the rule is not satisfied). Therefore, the proof will deal with the six-legged case, where $n=3$. The proof will be based on examining the four cases, which cover all the range of standard gaits that do not satisfy the rule of neighborhood. For each case it will be assumed that the generic gait representing the case is more stable than the wave gait, and a contradiction will be found related to the minimum duty factor for six-legged walking (Lemma 2.1). As a result, it will be shown that no gait for any of the four cases can be more stable than the wave gait.

Based on the condition given in (2.3), the four cases that together cover the range of standard gaits that do not satisfy the rule of neighborhood are *(i)* $\phi<1-\beta$, *(ii)* $\phi>1-\beta$, *(iii)* $\varphi<1-\beta$, and *(iv)* $\varphi>1-\beta$.

(i) $\phi<1-\beta$:

Consider the case that leg-2 is lifted.

$$\begin{array}{ll} \psi_3=F[\beta-\phi] & \psi_{2n-2}=F[\beta-\phi+\varphi] \\ \psi_2=\beta & \psi_{2n-1}=F[\beta+\varphi] \\ \psi_1=F[\beta+\phi] & \psi_{2n}\ \ =F[\beta+\phi+\varphi] \end{array}.$$

Since $\beta<\beta+\phi<1$, *leg-1* is lifted. If *leg-3* or *leg-2n* is lifted, the stability margin will automatically be less than that of the wave gait, since $R<P$. Therefore, both *leg-3* and *leg-2n* should be supporting. Related to the phase of *leg-2n*, $\beta+\varphi<\beta+\phi+\varphi<1+\varphi$, which means $1<\beta+\phi+\varphi<1+\beta$ should hold for supporting; therefore, $\psi_{2n}=\beta+\phi+\varphi-1$. Related to the phase of *leg-3*, $2\beta-1<\beta-\phi<\beta$, which means $0<\beta-\phi<\beta$ should hold for supporting; therefore, $\psi_3=\beta-\phi$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{array}{l} P_3=(5-n)P/2+R/2-(R/\beta)(\beta-\phi) \\ P_{2n}=(1-n)P/2+R/2-(R/\beta)(\beta+\phi+\varphi-1) \\ \Rightarrow S_{i2}=-\frac{1}{2}(P_3+P_{2n})=\frac{1}{2}[(2n-6)P/2-R+(R/\beta)(2\beta+\varphi-1)] \end{array}$$

In order this stability margin to be larger than the wave gait margin, $S_{i2}>S_w$ should be satisfied. If this inequality is combined with the inequality of $P>R$, the inequality requirement reduces to $\beta<(2\beta+\varphi-1)-(3\beta-3/2)$, in which the right hand side is a comparison of the phase related terms in S_{i2} and S_w. The same reduction of inequality will occur in the other cases as well. Therefore, only this reduced form of inequality with the phase related terms will be mentioned in *(ii), (iii), (iv)* cases. The inequality, after some arithmetic, reduces to $2\beta-1/2<\varphi$.

Now, consider the case that *leg-(2n-1)* is lifted:

$$\begin{array}{ll} \psi_3=F[\beta-\phi-\varphi] & \psi_{2n-2}=F[\beta-\phi] \\ \psi_2=F[\beta-\varphi] & \psi_{2n-1}=\beta \\ \psi_1=F[\beta+\phi-\varphi] & \psi_{2n}\ \ =F[\beta+\phi] \end{array}$$

Since $\beta<\beta+\phi<1$, *leg-2n* is lifted. If *leg-1* or *leg-(2n-2)* is lifted, the stability margin will automatically be less than that of the wave gait, since $R<P$. Therefore, both *leg-1* and *leg-(2n-2)* should be supporting. Related to the phase of *leg-1*, $\beta-\varphi<\beta+\phi-\varphi<1-\varphi$, which means $0<\beta+\phi-\varphi<\beta$ should hold for supporting; therefore, $\psi_1=\beta+\phi-\varphi$. Related to the phase of *leg-(2n-2)*, $2\beta-1<\beta-\phi<\beta$, which means $0<\beta-\phi<\beta$, should hold for supporting; therefore, $\psi_{2n-1}=\beta-\phi$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{aligned}
&P_{2n-2}=(5-n)P/2+R/2-(R/\beta)(\beta-\phi)\\
&P_1=(1-n)P/2+R/2-(R/\beta)(\beta+\phi-\phi)\\
&\Rightarrow S_{i(2n-1)}=\frac{1}{2}[(2n-6)P/2-R+(R/\beta)(2\beta-\varphi)]
\end{aligned}$$

In order this stability margin to be larger than that of the wave gait, $S_{i(2n-1)}>S_w$ should be satisfied, which reduces to $\beta<(2\beta-\varphi)-(3\beta-3/2)$, and then to $\varphi<3/2-2\beta$.

The conditions $2\beta-1/2<\varphi$ and $\varphi<3/2-2\beta$ together require $\beta<1/2$, which is a contradiction related to the minimum duty factor for six-legged walking (Lemma 2.1). Therefore for six-legged walking, no standard gait satisfying $\phi<1-\beta$, can be more stable than the wave gait with the same duty factor.

(ii) $\phi>\beta$:

Consider the case that *leg-1* is lifted.

$$\begin{array}{ll}
\psi_3=F[\beta-2\phi] & \psi_{2n-2}=F[\beta-2\phi+\varphi]\\
\psi_2=F[\beta-\phi] & \psi_{2n-1}=F[\beta-\phi+\varphi]\\
\psi_1=\beta & \psi_{2n}\ =F[\beta+\varphi]
\end{array}$$

Since $\beta-1<\beta-\phi<0$, $\beta<1+\beta-\phi<1$, hence *leg-2* is lifted. If *leg-3* or *leg-2n* is lifted, the stability margin will automatically be less than that of the wave gait, since $R<P$. Therefore, both *leg-3* and *leg-2n* should be supporting. Related to the phase of *leg-2n*, $\beta<\beta+\varphi<1+\varphi$, which means $1<\beta+\varphi<1+\beta$ should hold for supporting; therefore, $\psi_{2n}=\beta+\varphi-1$. Related to the phase of *leg-3*, $-2+\beta<\beta-2\phi<-\beta$, which means $-1+\beta<\beta-2\phi+1<1-\beta$, therefore, $0<\beta-2\phi+1<\beta$ should hold for supporting; hence, $\psi_3=\beta-\phi+1$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{aligned}
&P_3=(5-n)P/2+R/2-(R/\beta)(\beta-2\phi+1)\\
&P_{2n}=(1-n)P/2+R/2-(R/\beta)(\beta+\varphi-1)\\
&\Rightarrow S_{ii1}=\frac{1}{2}[(2n-6)P/2-R+(R/\beta)(2\beta+\varphi-2\phi)]
\end{aligned}$$

In order this stability margin to be larger than the wave gait margin, $S_{ii1}>S_w$ should be satisfied, which reduces to $\beta<(2\beta+\varphi-2\phi)-(3\beta-3/2)$. When this inequality is combined with the inequality of $\phi>\beta$, the condition reduces to $\varphi>-3/2+4\beta$.

Now, consider the case that *leg-(2n)* is lifted:

$$\begin{array}{ll}
\psi_3=F[\beta-2\phi-\varphi] & \psi_{2n-2}=F[\beta-2\phi]\\
\psi_2=F[\beta-\phi-\varphi] & \psi_{2n-1}=F[\beta-\phi]\\
\psi_1=F[\beta-\varphi] & \psi_{2n}\ =\beta
\end{array}$$

Since $\beta-1<\beta-\phi<0$, *leg-(2n-1)* is lifted. If *leg-1* or *leg-(2n-2)* is lifted, the stability margin will automatically be less than that of the wave gait, since $R<P$. Therefore, both *leg-1* and *leg-(2n-2)* should be supporting. Related to the phase of *leg-1*, $\beta-1<\beta-\varphi<\beta$, which means $0<\beta-\varphi<\beta$, should hold for supporting; therefore, $\psi_1=\beta-\varphi$. Related to the phase of *leg-(2n-2)*, $-2+\beta<\beta-2\phi<-\beta$, which means $-1+\beta<\beta-2\phi+1<1-\beta$; therefore, $0<\beta-2\phi+1<\beta$ should hold for supporting; hence, $\psi_{2n-2}=\beta-2\phi+1$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{aligned}&P_{2n-2}=(5-n)P/2+R/2-(R/\beta)(\beta-2\phi+1)\\&P_1=(1-n)P/2+R/2-(R/\beta)(\beta-\varphi)\\&\Rightarrow S_{ii2n}=\frac{1}{2}[(2n-6)P/2-R+(R/\beta)(2\beta-\varphi-2\phi+1)]\end{aligned}$$

In order this stability margin to be larger than the wave gait margin, $S_{i2n}>S_w$ should be satisfied, which reduces to $\beta<(2\beta-\varphi-2\phi+1)-(3\beta-3/2)$. When this inequality is combined with the inequality of $\phi<\beta$, the condition reduces to $-4\beta+5/2>\varphi$.

The conditions $\varphi>-3/2+4\beta$ and $-4\beta+5/2>\varphi$ together require $\beta<1/2$, which is a contradiction related to the minimum duty factor for six-legged walking (Lemma 2.1). Therefore for six-legged walking, no standard gait satisfying $\phi>\beta$ can be more stable than the wave gait with the same duty factor.

(iii) $\varphi<1-\beta$:

Consider the case that leg-1 is lifted:

$$\begin{aligned}&\psi_3=F[\beta-2\phi] &&\psi_{2n-2}=F[\beta-2\phi+\varphi]\\&\psi_2=F[\beta-\phi] &&\psi_{2n-1}=F[\beta-\phi+\varphi]\\&\psi_1=\beta &&\psi_{2n}\ \ =F[\beta+\varphi]\end{aligned}$$

Since $\beta<\beta+\varphi<1$, *leg-2n* is lifted. If *leg-2* or *leg-(2n-1)* is lifted, the stability margin will automatically be less than that of the wave gait, since $R<P$. Therefore, both *leg-2* and *leg-(2n-1)* should be supporting. Related to the phase of *leg-2*, $-1+\beta<\beta-\phi<\beta$, which means $0<\beta-\phi<\beta$ should hold for supporting; therefore, $\psi_2=\beta-\phi$. Related to the phase of *leg-(2n-1)*, $-1+\beta<\beta-\phi+\varphi<1-\beta$; therefore, $0<\beta-\phi+\varphi<\beta$ should hold for supporting; hence, $\psi_{2n-1}=\beta-\phi+\varphi$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{aligned}&P_2=(3-n)P/2+R/2-(R/\beta)(\beta-\phi)\\&P_{2n-1}=(3-n)P/2+R/2-(R/\beta)(\beta-\phi+\varphi)\\&\Rightarrow S_{iii1}=\frac{1}{2}[(2n-6)P/2-R+(R/\beta)(2\beta+\varphi-2\phi)]\end{aligned}$$

In order this stability margin to be larger than the wave gait margin, $S_{iii1}>S_w$ should be satisfied, which reduces to $\beta<(2\beta+\varphi-2\phi)-(3\beta-3/2)$. When this inequality is combined with the inequality of $\varphi<1-\beta$, the condition reduces to $5/4-(3/2)\beta>\phi$.

Now, consider the case that *leg-2* is lifted:

$$\begin{array}{ll} \psi_3 = F[\beta - \phi] & \psi_{2n-2} = F[\beta - \phi + \varphi] \\ \psi_2 = \beta & \psi_{2n-1} = F[\beta + \varphi] \\ \psi_1 = F[\beta + \phi] & \psi_{2n} \ = F[\beta + \phi + \varphi] \end{array}$$

If *leg-1* is supporting, $1<\beta+\phi<1+\beta$ should hold, which means $1-\beta<\phi$. The conditions $1-\beta<\phi$ and $5/4-(3/2)\beta>\phi$ necessitate $\beta<1/2$, which is a contradiction related to the minimum duty factor; therefore, *leg-1* cannot be supporting; hence, $\beta<\beta+\phi<1$. Then, *leg-3* and *leg-2n* must be supporting. Related to the phase of *leg-3*, $0<\beta-\phi<\beta$ should hold for supporting; hence, $\psi_3=\beta-\phi$. Related to the phase of *leg-2n*, since $\beta<\beta+\phi<1$, $\beta<\beta+\phi+\varphi<2-\beta$ holds; therefore, $1<\beta+\phi+\varphi<1+\beta$ should hold for supporting; hence, $\psi_{2n}=\beta+\phi+\varphi-1$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{aligned} &P_3 = (5-n)P/2 + R/2 - (R/\beta)(\beta - \phi) \\ &P_{2n} = (1-n)P/2 + R/2 - (R/\beta)(\beta + \phi + \varphi - 1) \\ &\Rightarrow S_{iii2} = \frac{1}{2}[(2n-6)P/2 - R + (R/\beta)(2\beta + \varphi - 1)] \end{aligned}$$

In order this stability margin to be larger than the wave gait margin, $S_{iii2}>S_w$ should be satisfied, which reduces to $\beta<(2\beta+\varphi-1)-(3\beta-3/2)$. The conditions $\beta<(2\beta+\varphi-1)-(3\beta-3/2)$ and $\varphi<1-\beta$ together require $\beta<1/2$, which is a contradiction related to the minimum duty factor for six-legged walking (Lemma 2.2). Therefore for six-legged walking, no standard gait satisfying $\varphi<1-\beta$ can be more stable than the wave gait with the same duty factor.

(iv) $\varphi>\beta$:

Consider the case that *leg-2n* is lifted:

$$\begin{array}{ll} \psi_3 = F[\beta - 2\phi - \varphi] & \psi_{2n-2} = F[\beta - 2\phi] \\ \psi_2 = F[\beta - \phi - \varphi] & \psi_{2n-1} = F[\beta - \phi] \\ \psi_1 = F[\beta - \varphi] & \psi_{2n} \ = \beta \end{array}$$

Since $-1+\beta<\beta-\varphi<0$, $\beta<\beta-\varphi+1<1$ holds; therefore, *leg-1* is lifted. If *leg-2* or *leg-(2n-1)* is lifted, the stability margin will automatically be less than that of the wave gait, since $R<P$. Therefore, both *leg-2* and *leg-(2n-1)* should be supporting. Related to the phase of *leg-(2n-1)*, $-1+\beta<\beta-\phi<\beta$, which means $0<\beta-\phi<\beta$ should hold for supporting; therefore, $\psi_{2n-1}=\beta-\phi$. Related to the phase of *leg-2*, $-\beta<\beta-\phi-\varphi<-1+\beta$; therefore, $1-\beta<\beta-\phi-\varphi+1<\beta$, and $0<\beta-\phi-\varphi+1<\beta$ should hold for supporting; hence, $\psi_{2n}=\beta-\phi-\varphi+1$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{aligned} &P_{2n-1} = (3-n)P/2 + R/2 - (R/\beta)(\beta - \phi) \\ &P_2 = (3-n)P/2 + R/2 - (R/\beta)(\beta - \phi - \varphi + 1) \\ &\Rightarrow S_{iv2n} = \frac{1}{2}[(2n-6)P/2 - R + (R/\beta)(2\beta - 2\phi - \varphi + 1)] \end{aligned}$$

In order this stability margin to be larger than the wave gait margin, $S_{iv2n}>S_w$ should be satisfied, which reduces to $\beta<(2\beta-2\phi-\varphi+1)-(3\beta-3/2)$. When this inequality is combined with the inequality of $\varphi>\beta$, the condition reduces to $5/4-(3/2)\beta>\phi$.

Now, consider the case that *leg-(2n-1)* is lifted:

$$\begin{array}{ll} \psi_3=F[\beta-\phi-\varphi] & \psi_{2n-2}=F[\beta-\phi] \\ \psi_2=F[\beta-\varphi] & \psi_{2n-1}=\beta \\ \psi_1=F[\beta+\phi-\varphi] & \psi_{2n}\ =F[\beta+\phi] \end{array}$$

If *leg-2n* is supporting, $1<\beta+\phi<1+\beta$ should hold, which means $1-\beta<\phi$. The conditions $1-\beta<\phi$ and $5/4-(3/2)\beta>\phi$ necessitate $\beta<1/2$, which is a contradiction related to the minimum duty factor; therefore, *leg-2n* cannot be supporting; hence, $\beta<\beta+\phi<1$. Then, *leg-1* and *leg-(2n-2)* should be supporting. Related to the phase of *leg-(2n-2)*, $0<\beta-\phi<\beta$ should hold for supporting; hence, $\psi_{2n-2}=\beta-\phi$. Related to the phase of *leg-1*, since $\beta<\beta+\phi<1$, $\beta-1<\beta+\phi-\varphi<1-\beta$ hold; therefore, $0<\beta+\phi-\varphi<\beta$ should hold for supporting; hence, $\psi_1=\beta+\phi-\varphi$. With these local phases the resulting supporting point positions and the stability margin will be as follows:

$$\begin{array}{l} P_{2n-2}=(5-n)P/2+R/2-(R/\beta)(\beta-\phi) \\ P_1=(1-n)P/2+R/2-(R/\beta)(\beta+\phi-\varphi) \\ \Rightarrow S_{iv(2n-1)}=\frac{1}{2}[(2n-6)P/2-R+(R/\beta)(2\beta-\varphi)] \end{array}$$

In order this stability margin to be larger than the wave gait margin, $S_{iv(sn-1)}>S_w$ should be satisfied, which reduces to $\beta<(2\beta-\varphi)-(3\beta-3/2)$. The conditions $\beta<(2\beta-\varphi)-(3\beta-3/2)$ and $\varphi>\beta$ together require $\beta<1/2$, which is a contradiction related to the minimum duty factor for six-legged walking (Lemma 2.2). Therefore for six-legged walking, no standard gait satisfying $\varphi>\beta$ can be more stable than the wave gait with the same duty factor.

For all the four cases that together cover all the range of standard gaits that do not satisfy the rule of neighborhood, it is shown that the wave gait is more stable. Hence, for six(and four)-legged walking, for the same duty factor, the wave gait is more stable than any standard gait which does not satisfy the rule of neighborhood. ■

Theorem 2.10: For six(and four)-legged walking, among all the standard gaits with duty factor β, the most stable gait is the wave gait corresponding to that duty factor; namely, for six(and four)-legged walking, the most stable gait among the standard gaits with duty factor β occurs when $\phi=1-\beta$ and $\varphi=1/2$. The stability margin of the wave gait is given in (2.9).

Proof: The proof trivially follows from Theorems 2.8 and 2.9.■

Theorem 2.11: *(Theorem of ϕ limitation)* Any wave gait for *2n-legs*, is more stable than any standard gait with a constant-phase-increment of $\phi\leq(1-\beta)/2$.

Proof: Any wave gait has the phase variables of $\phi=1-\beta$ and $\varphi=1/2$; therefore, it satisfies the condition for the rule of neighborhood given in (2.3). As a result, the stability margin of any wave gait is larger than the minimum rear stability margin that can be obtained by the support of *leg-2* and *leg-2n*, or *leg-1* and *leg-(2n-1)*.

Consider the case that *leg-3* is lifted:

$$\begin{array}{ll}
\psi_3=\beta & \psi_{2n-2}=F[\beta+\varphi] \\
\psi_2=F[\beta+\phi] & \psi_{2n-1}=F[\beta+\phi+\varphi] \\
\psi_1=F[\beta+2\phi] & \psi_{2n}\ \ =F[\beta+2\phi+\varphi]
\end{array}$$

If $\phi\leq(1-\beta)/2$ holds, $\beta\leq\beta+\phi\leq1$ and $\beta\leq\beta+2\phi\leq1$ will both hold, and *leg-1* and *leg-2* will both be lifted. Then the maximum rear stability margin will be the one that can be achieved by the support of *leg-3* and *leg-2n*, which will be less than the minimum stability margin of the wave gait, since $R<P$. It should be noticed that, a stable standard gait with the condition $\phi\leq(1-\beta)/2$, is not possible for a six or four-legged walking for any duty factor. ■

Theorem 12: *(Theorem of rear-to-front waves)* For SGSRN, for the same duty factor β and contralateral phase difference φ, any rear-to-front gait is more stable than any front-to-rear gait.

Proof: A standard gait is said to be rear-to-front if the ipsilateral phase difference is in the region of $0\leq\phi\leq1/2$, and front-to-rear if $1/2\leq\phi\leq1$. The stability margin of the standard gaits satisfying the rule of neighborhood is given as the minimum of the two equations in (2.8). In both of these equations the term for the ipsilateral phase difference has a negative sign; namely, the smaller the ipsilateral phase difference the larger both of the margins are. Since the ipsilateral phase difference of any rear-to-front gait is less than that of any front-to-rear gait, it follows that the stability margin of the former is larger than the stability margin of the latter. Hence, the proof of the theorem. ■

2.3. Conclusion

This chapter dwells upon the stability of six(and four)-legged walking from an analytical point of view. The definitions and theorems presented in the chapter are developed on the basis of Song and Waldron (1987). The chapter proves the optimal stability of wave gaits in the range of standard gaits for six(and four)-legged walking. For six(and four)-legged walking, Song and Choi (1990) had in fact proved the optimal stability of wave gaits in the range of periodic and regular gaits, which is more general than standard gaits. However, their proof is tedious, and does not provide much insight regarding to the nature of stable walk. Although the proof of this chapter is redundant from a theoretical point of view, its systematics of narrowing down provides more insight compared to the proof of Song and Choi (1990).

For the proof of the optimal stability of wave gaits, the chapter makes use of the concepts of standard gaits and the rule of neighborhood, and demonstrates the functionality of these for gait analysis. The rule of neighborhood, for example, is used in some other multi-legged walking researches (Porta and Celaya, 2004; Song and Waldron, 1987). The chapter here defines these concepts in an analytically systematic way, rather than with intuitive definitions. It is believed that, this chapter provides a systematic approach to the tedious work of analytical study of the stability of multi-legged walking, and provides analytical proof for the intuitively accepted propositions.

The analyzes of both the chapter here and Song and Choi (1990) reveal that, though wave gaits are optimally stable in particular regions, they are not so in all the range of "orderly gaits". The systematic analytical representation of non-periodic, irregular, or non-standard gaits is not trivial; and the analysis in these regions is considerably difficult. For multi-legged walking, another challenging issue, besides stability, is energy consumption. The analysis of energy efficiency of wave gaits (Chapter 3) reveals that it is possible to obtain more energy efficient gaits by deviating from the optimally stable wave gaits. Therefore, analyzing the unexplored regions of orderly gaits from stability point of view still remains as a topic for future work.

CHAPTER 3

FREE GAIT GENERATION WITH REINFORCEMENT LEARNING

3.1. Introduction

In the literature there exists two opposing views about how gait control is achieved in nervous system of animals, and what is best to be applied in multi legged robots. These views lead to two models for multi-legged locomotion. The "reflex model" composes of local controllers in the legs based on the sensory-motor-feedback between the local agents. The well-known example of reflex model controlled walking is the one developed in Cruse et al. (1998), where the legs interact with each other via some mechanisms (Chapter 1). The "Central Pattern Generator (CPG) model", on the other hand, is based on a feed forward central controller which generates rhythmic motions of the legs without the need of sensory feedback. The various applications of CPGs utilizing oscillatory neural networks (Inagaki et al., 2003), are based on the Pearson model of insect locomotion (Ferrell, 1995; Chapter 1).

Donner (1987) and Klaassen et al. (2002) provide brief comparisons of CPG and reflex models. The core of the discussion about pattern generation is how to manage adaptability and smooth passage from one to another gait pattern during continuous walking. The CPG models generate continuous patterns, mostly by oscillators controlling the leg movements, without using any feedback. It is easy to manage smooth passages between gait patterns by smoothly changing the oscillation constants. However, the CPG models are not as adaptive as the reflex models considering the changing environment and system conditions. For example, if one of the legs is out of order the CPG model collapses. The reflex model, on the other hand, can manage such situations; because, it gets feedback from the legs and determines the commands accordingly. The problem with the reflex model is the lack of a durable walking pattern, and the necessity of processing the feedback from all legs for every step. The authors of both Donner (1987) and Klaassen et al. (2002) conclude that the CPG and reflex models should be conciliated in order to achieve the best performance. The approach presented in this chapter is an attempt to conciliate the CPG and reflex models with central generation of free gaits and reinforcement learning for state transitions. Free gait generation and reinforcement learning are performed on the developed discrete model of stepping. The resulting Free Gait Generation with Reinforcement Learning (FGGRL) is applied in order to generate gaits with large stability margin while walking with different speeds, and a stable gait in case of a rear leg deficiency.

In the literature reinforcement learning is widely applied to walking machines in order to obtain improvement in various robot behaviors. Among those, Ilg and Berns (1995) and Ilg et al., (1997) make use of reinforcement learning in order to develop suitable protraction and retraction movements with off-line learning. In Ilg et al. (1997), of-line learning is performed with a real robot leg. In Ilg and Berns (1995), it is stated that the developed learning structure is applied also for leg coordination during walking, but the results are not reported. In Kirchner (1998), real time reinforcement learning is applied to a six-legged robot in order to develop, first, the four elementary leg movements of moving up, moving down, swinging forward, and retracting, then, the tripod gait. However, the paper does not provide detailed results related to learning of the tripod gait. In Huber and Grupen (1997), real time reinforcement learning is applied to a four-legged robot in order to develop a turning gait, using a hybrid discrete event dynamic system structure. In Porta and Celaya, (2001), Kimura et al. (2001), Svinin et al. (2001), and Karalarli et al. (2004) reinforcement learning is specifically used in order to obtained efficient gait generations.

There are attempts of free gait generation, in which off-line learning or programming in advance is applied. In Pratihar et al. (2002), a genetic-fuzzy approach is followed in order to achieve an optimal path and gait generation simultaneously. In their structure the fuzzy logic controllers change the stroke lengths to handle the turning of the robot and to avoid ditches while walking. The rules of the fuzzy controllers are learned off-line by using genetic algorithms. In Pal et al. (1994) free-gait generation is performed by using a graph search among the predefined states delineating the supporting and returning legs. Their search maximizes the number of leg transfers in each step, therefore whenever it is possible it catches the state transitions that lead to wave gaits, which are noted to be optimal in that sense. The authors claim that their algorithm combines the efficiency of the wave gait with the terrain adaptivity of the free gait. In Porta and Celaya (2004), a reactive free-gait generation is proposed. In this scheme, also, the gait is generated and modified as a result of the interaction of the robot with the environment. For example, if one of the legs does not find a proper position to place the tip on the front side of its stroke, it places it somewhere in the middle or rear of the stroke. The gait of the machine is modified immediately depending on the current stroke positions of the legs. In their free-gait generation they make use of the stability rule, which is denoted as “the rule of neighborhood” in this thesis; therefore, their algorithm guarantees stability in every instant throughout the walk. This feature exists also for the free-gait generation proposed in this chapter. All of the three mentioned works are simulation based. All the three algorithms are intended for real-time modification of gaits according to the interaction of the robot with the environment. However, this modification is limited with the capabilities of the off-line developed structures. Neither the rule structure of the first, nor the recorded possible state transitions of the second, nor the rules used for free gait generation in the third are intended for continuous improvement. They cannot handle new situations which are not estimated during off-line training or programming; because, they do not have an adaptation mechanism to unexpected situations. For example, in case of deficiency in any of the legs, their algorithms are not expected to function properly. The superiority of reinforcement learning is that it provides a learning structure for continuous development and adaptation to unexpected situations. Therefore, researchers of multi-legged walking are attracted by application of reinforcement learning to their systems.

Frequently, the applications of reinforcement learning to gait generation aim to maximize the speed. In Kimura et al. (2001), for example, sequential angular positions of the actuators of a four-legged robot is aimed to be learned in order to maximize the speed. The reinforcement learning algorithm is realized on an actual robot system, in which two wheels attached to the rear of the robot measure the advancement and provide the reinforcement signal to the robot. In Karalarli et al. (2004), reinforcement learning is used to maximize the speed and stability margin using a gait generation frame with fuzzy controllers. The fuzzy controllers command the linguistic thresholds that determine the transition of legs from stance to swing, or vice versa. Reinforcement learning is used to tune the centers and spreads of the membership functions. The authors have applied their gait generation to the case of leg deficiency. Although the algorithm could not find a pattern, it managed to make the robot walk with some instances of instability. In Porta and Celaya (2001), reinforcement learning is used with their free-gait generation structure explained extensively in Porta and Celaya (2004). In their application they aimed to maximize the speed within the framework of their free-gait generation structure. The results show that the scheme with reinforcement learning is more efficient than their hand-programmed reactive free gait generation. The superiority of their learning scheme is that it is possible to be applied on real robot; because, as mentioned in relation to Porta and Celaya (2004), their free gait generation always guarantees stability and avoids the robot from falling. This facility exists also in the reinforcement learning scheme developed in this chapter. In the application here the free gait generation already adopts the commanded speed; therefore, there is no need of speed optimization. Rather, the stability of the gait is optimized. Besides that, the application here accepts an external reward signaling if the robot falls or not due to some unexpected reasons. For example, a rear-leg might be deficient and the robot does not know that. In that case, the reinforcement learning makes the robot learn the transitions that do not result in falling.

In Svinin et al. (2001), reinforcement learning is used to learn the sequential action patterns that result in a stable straight walk. In this application, as the one here, falling of the robot results in a negative reinforcement, and the algorithm learns to avoid those. However, the application is pure simulative. It is strongly emphasized that learning procedure may take too much time if it is applied to an actual robot in real-time. In fact, this difficulty is pointed out also in Ilg and Berns (1995), Ilg et al. (1997), and Kirchner (1998), noting that real-time application of reinforcement learning to actual robotic sytems is extremely difficult because of the necessity of large number of training cycles. It is recommended to incorporate a priori knowledge or to make use of a biased learning strategy in order to simplify and speed up the learning process. Therefore, application of a real-time learning to an actual walking robot in this chapter is noteworthy with the ones in Kimura et al.

(2001), Kirchner (1998), and Huber and Grupen (1997). In order to cope with large number of training cycles, this chapter applies the reinforcement learning, not starting from a tabula rasa, but with the peculiar free-gait generation algorithm that guarantees stability for all states. Such an approach, does not only guarantee stable walk in normal conditions, but also makes learning faster even when stability is not guaranteed when one of the legs is disfunctioning.

In the following, first the discrete model of stepping is introduced. Then the free gait generation on this model and the reinforcement learning scheme applied for learning the state transitions are explained. These together constitute the FGGRL. Following that the FGGRL is applied to learn walking when one of the rear legs is out of order. Lastly the results obtained on simulation and real experimental setup are given.

3.2. Discrete Model of Stepping and Explanation of States for the Learning Task

The model is developed on a discrete model of the possible positions of the tip point of a single leg. In Figure 3.1, the discrete positions of stepping are shown. There are five support positions (leg-states) on the ground labeled from 1 to 5. Considering the walking direction as positive, the *position 1* corresponds to the maximum possible anterior extreme position, while the *position 5* corresponds to the minimum possible posterior extreme position. The *position 3* is in the middle of the support line; it is the closest support position to the attachment of the leg to the body. In *position 3* the leg can support without any extension. The *position 0* is the single position that the tip point stays when the leg is not supporting. This will be mentioned as the return position.

In a regular stepping, which is not the case here, there is a single tip point trajectory. Therefore the anterior and posterior extreme positions are fixed. According to such a regular scheme the transitions between the six states in Figure 3.1 would be fixed as from *0* to *1*, *1* to *2*, *2* to *4*, *3* to *4*, *4* to *5*, and *5* to *0*; or if the intermediate support states are not mentioned, it becomes from *0* to *1*, *1* to *5*, and *5* to *0*. However, in order to generate a free gait, the stepping must also be free; namely, the leg must be able to start supporting on any position on the support line and to pass to the return position from any support position. The discretization in Figure 3.1 prepares the frame for such a free stepping.

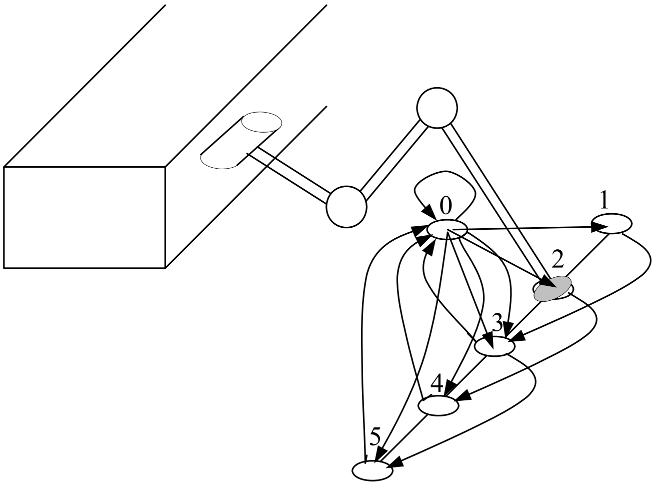
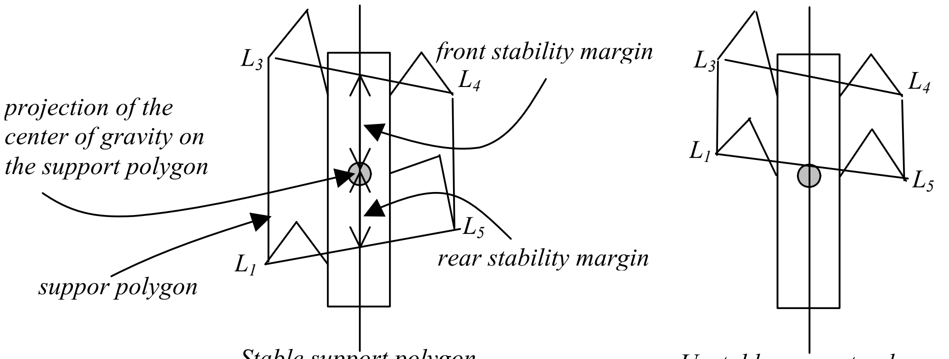


Figure 3.1: Discrete positions of stepping, and the possible transition for a velocity of two units per iteration.

In free stepping the steps are performed with discrete transitions between the leg-states. These transitions are conditioned by the speed of walk that determines how much unit the tip point will be retracted in the support phase. In the model here the tip point has to be in one of the leg-state positions at the end of every iteration. For the simulation and actual robot used here, one unit of distance corresponds to *2cm*. In Figure 3.1 the transitions for a walk with two units per iteration are given, which corresponds to *4cm per iteration* or *4cm per step*. With this speed the leg has to retract two units if it is supporting. Accordingly, the transitions from *1* to *3*, *2* to *4*, and *3* to *5* are allowed. The transition from *1* to *2*, for example, is not allowed; otherwise, the body would not be advanced by two units. The leg cannot pass to any supporting state from the *state 0*. The first supporting state it visits has to satisfy the condition that it allows the required amount of advancement in the following iteration. Accordingly, for the iteration of two units the first visits from *0* to *4*, or from *0* to *5* are not allowed. The first visit from *0* has to be either to *1*, *2*, or *3*.

The transition from *state 0* to the first supporting state has nothing to do with the advancement of the body; therefore, such a transition will be mentioned as a *sub-transition*. A sub-transition can take place before the following transition within the supporting states, during a single iteration. Then the stepping from *state 0* in one iteration can be as follows: first the sub-transition from *state 0* to the anterior supporting state takes place, then the transition from that anterior supporting state to the next supporting state takes place. As a result, in one iteration, considering two units of advancement, transitions from *0* to *3* ,*4* , or *5* are possible. Given the unit of advancement and any transition from state *0* to a supporting state, the sub-transition that takes place in advance of the transition on the supporting line can easily be determined. A leg in state *0* can stay in return position, so the transition from *0* to *0* is also possible.

The leg-states of the six legs, together, construct the state of the robot. Accordingly, there are 6^6=46656 possible states for the robot. The supporting legs in any state of the robot correspond to a supporting polygon. In order the state to be statically stable, the projection of the center of gravity of the robot on the horizontal plane should remain within this supporting polygon. In Figure 3.2 examples of stable and unstable support polygons are given. The stability margin of a support polygon, and that of the corresponding state of the robot, is defined as the minimum of the front and rear stability margins shown on the figure. Theoretically, a state is stable if this stability margin is greater than zero. However, with the actual Robot-EA308, the minimum margin for a stable walk is *2cm*. Therefore, throughout this chapter the minimum stability margin will be taken as *2cm*, rather than zero. In Figure 3.2 and 3.3 the convention of numbering the legs is revealed.



Stable support polygon *Unstable support polygon*

Figure 3.2: Examples of stable and unstable support polygons.

According to the *2cm* minimum stability margin assumption, 41510 of the 46656 states are statically stable. However, not all of these stable states can be used during an actual walk. This is because some of those are either dead-end states, or states that are impossible to be reached. For example the state *5-5-5-5-5-5* is a dead end state, since the next state after one iteration has to be *0-0-0-0-0-0*, which is an unstable state. The state *2-2-2-2-2-2* is impossible to be reached, because the robot has to be in the unstable state of *0-0-0-0-0-0* in advance of the iteration, considering the minimum speed of one unit per iteration. The state *1-1-1-1-1-1* is impossible to be reached even with a sub-transition due to the same reason. While the states that are impossible to be reached do not create a problem while learning state transitions, the dead-end states do.

The learning of walking in general corresponds to learning the possible transitions from any visited stable state to another stable state that is not dead-end. Although the total number of unstable states (5146) is much less than the number of stable states (41510), many of the unstable states are always reachable while only some of the stable states conditioned by the walking speed are reachable from any given stable state. For example the state *0-0-0-0-0-0* is reachable from any given stable state. Here is given an example based on computer calculations: For the state of *3-0-3-0-3-3*, with the walking speed of *2 units per iteration*, there are $2^6=64$ possible transitions (the legs in leg-state *3* can go to the leg-state of either *5* or *0*; the legs in leg-state *0* can go to the leg-state of either *0* or *2*). Among these only 21 of them are statically stable, considering a minimum of *2cm* stability margin. Among the stable ones only 8 of them are not *-1* or *-2 states* (to be explained in the following). Therefore, the machine has to learn to choose 8 of the 64 possible transitions from this particular state. This ratio is as low as 2/64 for the state *3-3-3-3-3-3* with the same speed. These are low ratios for real time learning applications. Therefore, learning of walking in the very general sense is a tedious task (Svinin et al., 2001; Ilg and Berns, 1995; Ilg et al., 1997; Kirchner, 1998). In order to cope with the problem of high dimension of possible transitions, the paper here adopts the idea of stable free state generation, which guarantees generation of statically stable next states that are not dead-end.

3.3. Free Gait Generation (FGG)

The FGG developed here is based on successive stable free state generations (FSG). This FSG is performed by a random choice of a state from the subset of stable non-dead-end states that satisfy the rule of neighborhood and fulfill the advancement of the robot with the commanded velocity. In this generation satisfaction of the rule of neighborhood is critical in order to guarantee the stability.

3.3.1. The Rule of Neighborhood

The idea of "satisfying the rule of neighborhood" is used also in Song and Waldron (1987), and to our knowledge, first applied in generation of free next states in Porta and Celaya (2004). In Section 2.2.2 of this thesis, the definition of the rule of neighborhood is given and it is analytically proved that for robots with more than six-legs, standard-gaits (a set including the wave gaits) satisfying the rule of neighborhood are statically stable. In fact it can be easily shown that any support polygon satisfying the rule of neighborhood is statically stable, provided that the pitch length (the distance between the middle points of stroke lines) is greater than the stroke length.

Theorem 3.1: Any support polygon, and the corresponding state of the six-legged robot, satisfying the rule of neighborhood is statically stable provided that the pitch length is greater than the stroke length ($P>R$). The minimum possible stability margin of a support polygon satisfying the rule of neighborhood is $(P-R)/2$.

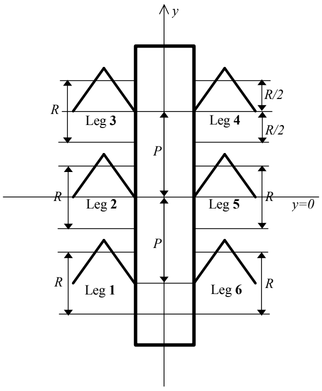


Figure 3.3: Figure showing the pitch (*P*) and stroke (*R*) lengths.

Proof: Due to the rule of neighborhood the front stability margin is determined by the support of one of the following three couples of legs: *2-4*, *3-4*, *3-5*. Due to the geometry in Figure 3.3 the minimum front stability margin occurs when the couples *2-4* or *3-5* are supporting. For both cases the minimum margin occurs when the front leg (*3* or *4*) and rear leg (*2* or *5*) of the couples support at their posterior extreme positions. In this case the line connecting the support positions of the couples crosses the centerline at a point where $y=(P-R)/2$. Then the minimum front stability margin is $(P-R)/2$. The same arguments go for the rear stability margin, whose possible minimum can again be calculated as $(P-R)/2$. Then the possible minimum of the stability margin of the polygon, given as the minimum of these two margins, is $(P-R)/2$. Due to the assumption of ($P>R$), the minimum possible stability margin of a polygon satisfying the rule of neighborhood is always greater than zero, hence the polygon is stable.■

In the case of Robot-EA308, the pitch length is *17cm*, and the stroke length is *8cm*. The minimum possible stability margin of a polygon satisfying the rule of neighborhood for the actual robot is then $(17-8)/2=4.5cm$, which is greater than the required minimum of *2cm*. Therefore, the theorem applies to the Robot-EA308, and in its discrete stepping model any state satisfying the rule of neighborhood (SSRN) is statically stable.

As stated before dead-end states are a problem for learning of walking. Therefore, the free gait generation should avoid generation of any dead-end states. The following definitions and theorems deal with avoiding dead-end state generation.

-1 State: Given the speed of walk in *units per iteration*, a SSRN is designated as a *-1 state* if it does not lead to any SSRN; in other words, a dead-end state is a *-1 state*.

-2 State: Given the speed of walk in *units per iteration*, a SSRN is designated as a *-2 state* if it does not lead to any SSRN which is not *-1*.

-3 State: Given the speed of walk in *units per iteration*, a SSRN is designated as a *-3 state* if it does not lead to any SSRN which is not *-2*.

Theorem 3.2: Given the speed of walk in *units per iteration* as *v_unit*, where *v_unit* takes one of the values from *{1,2,3,4}*, a *-1 state* has to have two successive supporting leg states greater than or equal to (*6-v_unit*).

Proof: A *-1* state, P_{-1}, has to lead only to states that violate the rule of neighborhood. This means, the state of P_{-1} is such that it cannot avoid two of the legs to have the leg state of *0* in any of the succeeding states. Without violating the generality, let's assume the pair of legs L_i-L_{i+1} have to take the values *0-0* in the succeeding state of P_0. Since P_{-1} satisfies the rule of neighborhood it is known that at least one of these two legs is supporting in state P_{-1}. If the remaining is not supporting, namely if it is in state *0* in P_{-1}, then that leg can take any leg state in accordance with the speed in the succeeding state; namely, that leg does not have to be in state *0* in P_0. However, this contradicts with the condition that L_i-L_{i+1} have to take the values *0-0* in P_0. Therefore, both of the legs are supporting in P_{-1}. In order to have to generate *0-0* in the succeeding state, these two legs have to have the state of at least (*6-v_unit*) in P_{-1}, otherwise they could take another supporting state. For example if one of the legs were (*5-v_unit*), it could take the leg state *5* in the succeeding state. As a result, any *-1 state*, P_{-1}, has to have two successive leg states greater than or equal to (*6-v_unit*).■

Theorem 3.3: Given the speed of walk in *units per iteration* as *v_unit*, where *v_unit* takes one of the values from *{1,2,3,4}*, a *-2 state* has to have the pattern of s_{i-1}-s_i-s_{i+1}-s_{i+2} in any of its four successive legs, where (*6-2v_unit*)≤$\{s_i,s_{i+1}\}$<(*6-v_unit*) and $\{s_{i-1},s_{i+2}\}$≥ (*6-v_unit*) holds.

Proof: A *-2* state, P_{-2}, has to lead to only *-1* states. This means, according to *theorem 3.2*, the state P_{-2} is such that it cannot avoid two of the legs to have the successive leg states greater than or equal to (*6-v_unit*) in any of the succeeding states. Without violating the generality, let's assume the pair of legs L_i-L_{i+1} have to be in this condition in the succeeding state of P_{-1}. Using the same reasoning in the proof of *theorem 3.2*, these two legs have to be supporting with the states s_1, s_2 in state P_{-2}, where (*6-v_unit*) <$\{s_1,s_2\}$≤ (*6-2v_unit*) holds. However, this condition is necessary but not sufficient to have the states greater than or equal to (*6-v_unit*) in the succeeding state. This is because any supporting leg can take a supporting leg state or the leg-state *0* in the succeeding state. There must be a condition that forces these two legs to stay in supporting in the succeeding state. This condition is due to the satisfaction of the rule of neighborhood by the succeeding state of P_{-1}. The couple of legs L_i-L_{i+1} have to be supporting in state P_{-1} only if the two legs in the vicinity of the couple are in leg-state *0*. Again following the reasoning in the proof of *theorem 3.2*, if these two legs have to be in leg-state *0* in state P_{-1}, they have to have a state greater than or equal to (*6-v_unit*) in state P_{-2}.■

Theorem 3.4: There does not exist any *-3 state*.

Proof: A *-3* state is defined as a state which has to lead to a *-2* state. By *theorem 3.3*, a *-2 state*, P_{-2}, has to have the pattern of s_{i-1}-s_i-s_{i+1}-s_{i+2} in any of its four successive legs, where $(6-2v\_unit) \leq \{s_i, s_{i+1}\} < (6-v\_unit)$ and $\{s_{i-1}, s_{i+2}\} \geq (6-v\_unit)$ holds. Following the reasoning used in the previous theorems, in order to have to be supporting with these specific states in P_{-2}, the legs must be supporting in the previous state. Any four successive supporting legs, whatever their supporting states are, can not be forced to support in the successive state with the rule of neighborhood; since, any of the two legs in between can pass to the leg-state *0* without violating the rule of neighborhood. Then a SSRN, which is not *-2* is achieved. Therefore, it is impossible to have a *-3* state which has to be succeeded by a -2 state.■

Theorem 3.5: If a free gait generation algorithm avoids any *-1* or *-2* states in successive state generations, the gait will visit no dead-end states.

Proof: If *-1* states are avoided, there will be no visits to any direct dead-end states. Then, *-2 states* will become the dead-ends of the new situation. If those are also avoided, there will be no visits to the states that have to lead to a *-1 state*. Once, *-2* and *-1 states* are avoided there will be no dead-end states of the new situation; since, there are no *-3 states*. As a result the free gait generation will be free of any dead-ends.■

3.3.2. Free State Generation Algorithm

Based on the above theorems, the following state generation algorithm is developed. This algorithm guarantees the satisfaction of the rule of neighborhood and avoids any *-1* or *-2 states*. It takes the current state, which is neither *-1* nor *-2*, and the speed of walk in *units per iteration* as the input, and generates the next state as the output.

Given: Current state $P=[s_1, s_2, s_3, s_4, s_5, s_6]$ and the speed of walk $v\_unit=v$.

Step 1: Find a succeeding state in which all the legs that can support are supporting:

$P'=[s_1', s_2', s_3', s_4', s_5', s_6']$, where $$s_i' = \begin{Bmatrix} mod(6, s_i + v_{unit}) & , \ if\ s_i \neq 0 \\ random\{1, 2, \ldots, 5 - v_{unit}\}, & if\ s_i = 0 \end{Bmatrix}$$

Step 2: Construct the state P'' by transferring the supporting legs in state P' to the leg-state *0* with a probability of 0.2, 0.3, 0.4, 0.5, and 0.6 for the leg states *1*, *2*, *3*, *4*, and *5*, respectively. (These probability values are determined so that the leg is more likely to stay in support if its tip point is in a front position, and it is more likely to transfer to the return state if its tip point is in a rear position on the support line.)

Step 3: Check if P'' satisfies the rule of neighborhood. If it does not satisfy, change the state of the legs with leg-state zero to their values in P' with a probability of 0.5. Repeat this process till a state satisfying the rule of neighborhood is obtained. This state will be P'''.

Step 4: Check if P''' is a *-1* or *-2 state*. If it is either *-1* or *-2*, go to *step 1*. Otherwise P''' is a succeeding state of P, which is neither *-1* nor *-2* and satisfies the rule of neighborhood.

Output: P'''.

This random generation of the next state is guaranteed to stop in finite cycles, because the initial state P is already a non-dead-end state generated by the algorithm, hence there exists at least one SSRN to be reached by the random generation. The successive usage of the free state generation algorithm results in the FGG of this paper. With this FGG the machine can walk with a given speed without any fall. However, the states generated by the FGG do not have any specificity rather than being a SSRN. Among the possible gaits with a given speed there are some with more static stability, while there are some that hardly satisfy the stability condition. Although the FGG guarantees stable states, it is still possible to improve the stability of walk by making use of the most stable states. This improvement can be achieved by a learning with which the best state transitions are memorized and utilized, besides new state generations. The next section will introduce the application of reinforcement learning for this purpose.

3.4. Free Gait Generation with Reinforcement Learning (FGGRL)

The reinforcement learning adopted here is based on updating the utilities of state transitions according to a reinforcement signal. A similar application of reinforcement learning of walking to an eight-legged robot can be found in Svinin et al. (2001).

3.4.1. Updating the utilities of State Transitions

After every *(n)th* iteration, the transition from a state $P^{(n)}$ to $P^{(n+1)}$ is assigned a reinforcement signal depending on the stability margin of the next state $P^{(n+1)}$. For this purpose a nominal stability margin value of *7cm* is used, and the reinforcement takes a value in comparison to this nominal margin as in (3.1). The subscript *sm* in the reinforcement term indicates that it is used to maximize the stability margin.

$$r_{sm}^{(n)} = \left\{ \begin{array}{l} \dfrac{2}{1+\mathrm{e}^{(7-s_m{}^{n+1})}} - 1 \quad , \quad \mathit{if}\ s_m{}^{n+1} \geq 7 \\ \dfrac{2}{1+\mathrm{e}^{(7-s_m^{n+1})}} - 1.5 \quad , \quad \mathit{if}\ s_m{}^{n+1} < 7 \\ -1.5 \quad , \quad \mathit{if}\ v_{unit} \neq v_{commanded} \end{array} \right\} \tag{3.1}$$

In (3.1), $s_m^{(n+1)}$ stands for the stability margin of the next state $P^{(n+1)}$. The third line of the reinforcement signal might be encountered during speed transitions, which will be explained in the end of this section. The visited states, and the experienced transitions within these visited states are recorded with their corresponding utilities in a lookup table. After every iteration the utilities of memorized transitions, which can take a value in the range of [0,1], are updated. In the following the items of updating are explained.

Utility initialization: If the transition is experienced the first time it is assigned the utility of 0.5.

$$u_{P^{(n)}\rightarrow P^{(n+1)}}=0.5 \tag{3.2}$$

Evaporation: After every iteration, all the utilities in the lookup table are discounted with a factor of *0.9*. In this way, non-used utilities and states are eliminated in time.

Utility elimination: The transitions with utilities less than *0.02* are cleaned from the lookup table.

Memorized state elimination: If the sum of the utilities of all experience transitions from a state are less than *0.1*, that state is totally eliminated from the lookup table.

Reward distribution: The reward $r_{sm}^{(n)}$ achieved by the transition from $P^{(n)}$ to $P^{(n+1)}$ at iteration *(n)*, is used to update the utilities of the last five transitions that lead to $P^{(n+1)}$. The updating of the utilities is as in (3.3). The learning rate denoted by the term γ is taken to be *0.8*.

$$u_{P^{(i)}\rightarrow P^{(i+1)}}:=u_{P^{(i)}\rightarrow P^{(i+1)}}+\gamma^{(i-n)}r_{sm}^{(n)}(1-u_{P^{(i)}\rightarrow P^{(i+1)}}) \quad , \quad i=n-5,n-4,\ldots n. \tag{3.3}$$

3.4.2. Selection or Generation of Transitions

The next state is determined either by selection among the memorized transitions, or by generation of a new transition, depending on the utilities of the current state $P^{(n)}$. If there are *k* memorized transitions from the state $P^{(n)}$ each with utility $u_{P^{(n)},j}$,one of these transitions are realized with probability p_u, and a new transition is created with probability $(1-p_u)$, where p_u is given as in (3.4).

$$p_u=\sum_{j=1}^{k}\frac{u_{P^{(n)},j}}{k} \tag{3.4}$$

If one of the memorized transitions is to be realized, a random selection is performed with a distribution proportional to the utilities of the transitions. Otherwise a new transition is created.

A new transition can be created in two ways. First, the FSG algorithm can be used to generate the next state. The generated next state might still coincide with one of the memorized states; then the table is updated with the existing states, otherwise the generated state is considered as a new state. Second, one of the memorized states can directly be used as the next state, although it does not exist within the transitions experienced from the current state $P^{(n)}$. Such a direct utilization makes it easier for the algorithm to catch cycles of transitions, which correspond to a patterned periodic walk. In case of existence of such a transition to a memorized state, the two possibilities of new transition generation are performed with equal probabilities of *0.5*. If there are more than one possible transitions to a memorized state, and it is decided to make transition to a memorized state, then one of them is selected randomly with a distribution proportional to the sum of the transition utilities of the candidates. If there does not exist any memorized state proper for transition, then the FSG algorithm is directly used.

If the command of speed is changed at an instant, the memorized transitions cannot be used. Therefore, the memorized state transition intended to be utilized has to be checked if it is fitting to the commanded speed. It is easy to achieve this check by comparing the state of supporting legs in the next state candidate and the current state. If the speed requirement is not satisfied the memorized transition selected for realization cannot be used; then, a new state is generated using the FSG algorithm according to the commanded speed.

3.4.3. Smooth Transition of Gait Patterns for Different Speeds

The FSG described above guarantees SSRN which are not *-1* or *-2* for the given speed. However, when the speed command is changed at an instant, the machine might be in a state which is *-1* or *-2* according to the new commanded speed. (Such a situation might occur when the speed command is increased. In the case of decreasing the speed command, it does not occur, because any state which is not *-1* or *-2* for a given speed, is also not *-1* or *-2* for any speed lower than that. Consider that the supporting legs will retract less in lower speeds. Less retraction is always less likely to create a dead end state compared to more retraction.) In order to detect such situations, the current state input to the FSG is tested if it is a *-1* or *-2 state* regarding to the given speed. If it is so, the speed is temporarily reduced to *1 unit per iteration* till a state which is not *-1* or *-2* is reached with respect to the newly commanded speed. It is clear that, any state visited by the robot, according to whichever speed it is generated, is neither *-1* nor *-2* with respect to the speed of *1 unit per iteration*. Moreover, it is possible to reach every SSRN which is not *-1* or *-2* for any given speed, with generation according to the speed of *1 unit per iteration*. Therefore, generations with *1 unit per iteration* are guaranteed to reach a *non -1* and *non -2* SSRN for the commanded speed, and from that state on the robot can walk with the commanded speed. In this way smooth transitions between different speeds is achieved. The robot changes the speed only with slowing down the walk, not with stopping. The state transitions get the negative reward of *-1.5* when the speed is lowered to *1 unit per iteration* for transition purposes. In this way memorization of the transition states are avoided. That is why there is the third line in the formula of the reward in (3.1).

3.5. FGGRL when There is an External Stability Problem of Leg Deficiency

The necessity of handling the case of deficiency in one of the legs is addressed in Inagaki, (1997) and Karalarli et al. (2004). In Karalarli et al. (2004), the authors content with observation of the behavior of their gait generation algorithm, making use of reinforcement learning, when one of the legs is deficient. Although the algorithm results in an improvement, it does not end up with a stable pattern. In Inagaki (1997), on the other hand, the author dwells upon generating a stable periodic gait for five legs. The analysis reveals that, with the settings of the paper, there exists no stable periodic gait with five legs for the general rectangular leg arrangement of hexapod robots. The paper shows that stable periodic gaits with five legs can be achieved with an assymmetric pentagonal leg arrangement. The paper proposes to shift to the developed assymmetric pentagonal leg arrangement in case of deficiency in any of the legs. However, in order to achieve this, the workspace of the legs should be quite large. The analysis and results of these two papers reveal that static-stable five legged walking is a hard task. Considering this, the chapter here does not claim to have solved the five-legged walking problem. Rather, five-legged walking is used as a case in which adoptability of the robot to an unexpected failure is tested. Moreover, the five-legged walking in this chapter is limited to a *pseudo-static-stable* walk, which will be introduced in the following, with deficiency in one of the *rear-legs*. The peculiarity of the chapter, however, is that the learning of five legged walking is realized on the actual robot.

In the above introduced FGGRL, the generated transitions are guaranteed to be stable, and only learning of state transitions with larger stability margins is performed. When there is an external stability problem, as a defect in one of the legs, a SSRN cannot guarantee static stability any more. Moreover, the detections for *-1* and *-2 states* developed till here, do not work for the five-legged case. When a rear leg is out of order the FSG algorithm will not know it, and will continue to generate states as if all the six legs were in order. Then, the task of learning stable transitions with states neither *-1* nor *-2* for the five-legged case will be loaded on the reinforcement learning part of FGGRL. The robot will fall down with many of the transitions, will stand up again and will try new transitions. During all these, learning will be performed. For the case of five-legged walking only the speed of *1 unit per iteration* is used, since any other speed is impossible with five legs for the Robot-EA308.

The stability margin with six-legged walking is assured to be larger than *4.5cm* in any instant of the walk, due to the satisfaction of the rule of neighborhood. For the five-legged case such a stability margin cannot be guaranteed. Moreover, even satisfying static stability in all instants is impossible for the Robot-EA308. Therefore, the concept of quasi-static-stability is developed for five-legged walking.

3.5.1. Quasi-Static-Stability for Five-Legged Walking

In explaining the discrete model of stepping the transitions from leg-state *0* to the first supporting states were mentioned as sub-transitions, which do not have anything to do with advancement of the robot. It was again stated that these sub-transitions could take place before a transition within the supporting states during a single iteration. In walking of the Robot-EA308 in control of the FGGRL, this is exactly realized. In this way, the legs that will transform to a supporting position from the return position are put in support position, before the legs that will transform to the return state of *0* are raised. Then, in the very beginning of retraction, more legs than the actual ones in the next state are supporting the robot. Since retraction happens in a short time, a statically unstable situation in the beginning of retraction is easily compensated. Then it is enough for a state to sustain the minimum static stability margin of *2cm* at the end of retraction. For that reason, any state that sustains the minimum static stability margin of *2cm* after retraction is designated as *quasi-static-stable* for the Robot-EA308. In the simulations for learning of five-legged walking, and also in calculation of the reinforcement signal for the six-legged walking in (3.1), this quasi-static-stability margin is used. (Since the quasi-static-stability is calculated after the legs are retracted, there is a dissymmetry considering the front and rear stability margins. This is because when the legs are retracted, the front stability margin is more likely to be less than the rear stability margin. As a result the quasi-static-stability margin is limited mostly by the positions of the front legs. Therefore, it becomes very hard for the Robot-EA308 to walk even with a quasi-static-stable gait when one of the front or middle legs is out of order. Because of this reason, the five legged walking adopted here is limited to one of the rear legs being out of order, rather than any of the six.)

In the learning scheme for the five-legged case the reinforcement signal and the probability of using an existing state for transition generation are modified. For the learning of five-legged walking, it is not meaningful to try to improve the static stability margin, because to achieve a five-legged quasi-static-stable state is already a hard task, and five-legged quasi-static-stable states can hardly satisfy the minimum stability criterion. What is learned for five-legged walking is just the transitions to quasi-static-stable states. The reinforcement signal is modified as in (3.5). The subscript *sw* in the reinforcement term indicates that it is used to achieve a stable walk.

$$r_{sw}^{(n)} = \begin{Bmatrix} 1 \;\;, & if\ P^{(n+1)} is\, quasi-static-stable \\ -1.5 \;\;, & if\ P^{(n+1)} is\, statically\, unstable \end{Bmatrix} \tag{3.5}$$

In this way only transitions to quasi-static-stable states get positive rewards, and transitions to unstable states are cleaned from the memory just after they are experienced. As a result, the memorized states are all quasi-static-stable, and it is reasonable to make use of those whenever it is possible. Therefore, for the five-legged case, the probability of using an existing state for transition generation is made *1*, rather than *0.5*. The robot creates transition to a memorized state whenever it is possible. In this way it is made easier to catch a cycle of state transitions that will result in five legged walking.

3.5.2. Learning Algorithm for Five-Legged Walking

The following algorithm is adopted for learning walking when one of the rear legs is out of order:

Initialization state: P_i=[$\underline{1}$ 1 1 1 1 $\underline{1}$] This state is used to initialize the walk if the robot cannot find a stable next state succeeding its current state. It should be noted that, because of the rear-leg deficiency, either the first or the last leg-state is realized as *0*, although it is commanded as *1*.

Current state at iteration n: $P^{(n)}$ This is equalized to the initialization state at the beginning of learning.

Number of learning steps: N

for n=1 to N,

Step 1: Determine the next state candidate P^{*}.

If there is any state possible to make a transition, it will be realized. Such memorized candidates might or might not have been memorized as a possible transition from $P^{(n)}$

Step 1.i: Check if there is any memorized transition from $P^{(n)}$. If not go to *Step 1.ii*. If there is, the next state will be selected among them with probability p_u (3.4). The selection will be performed randomly according to a distribution proportional to the transition utilities of $P^{(n)}$. If this is performed, go to *Step 2*. With probability $1-p_u$, go to *Step 1.ii*.

Step 1.ii: Check if there is any memorized state that is possible to make a transition. If not, go to *Step 1.iii*. If there is, the selection will be performed randomly according to a distribution proportional to the sum of the transition utilities of the memorized next state candidates. Then, go to *Step 2*.

Step 1.iii: Use FSG algorithm to generate a new next state from $P^{(n)}$.

Step 2: Make the transition from $P^{(n)}$ to P^{*}.

If the transition is stable, memorize the transition, update the lookup table according to the reinforcement signal with value *1*, assign $P^{(n+1)}:=P^{*}$, $n:=n+1$, and go to *Step 1*.

Step 3: Transition is unstable. The robot falls down. Make the robot stand at the state $P^{(n)}$.

Step 4,5: Repeat *Step 1*, *Step 2*, and *Step 3*.

In this way try to find a stable transition from $P^{(n)}$ for two more times.

Step 6: Three attempts have been performed to make a stable transition from $P^{(n)}$, but non of them has created a stable next state. Give up trying to find a stable transition from $P^{(n)}$. Assign $P^{(n+1)}$ to the initialization state, update the lookup table according to the reinforcement signal with value *-1.5*, assign $P^{(n+1)}:=P_i$, $n:=n+1$, and go to Step 1.

3.6. Simulation Results

The success of the FGGRL is evaluated first with computer simulation. The simulation is performed for 500 steps of the robot. Table 3.1 shows the successive commands that are sent to the simulation robot during these 500 steps.

Table 3.1: Successive epochs and commands sent to the simulation robot within one episode of FGGRL.

Epochs	***A***	***B***	***C***	***D***	***E***
Commands ***speed: (cm/step)***	***v_unit=1***	***v_unit=4***	***v_unit=2***	***v_unit=3***	***Deficiency on rear left leg*** ***(v_unit=1)***
Step numbers	1-100	101-200	201-300	301-400	401-500

In the first four epochs the reinforcement in (3.1), for the last epoch the reinforcement in (3.5) are used. In the first four epochs the FGGRL is used as it is described in Section 3.4, and for the last epoch it is used with its form given in Section 3.5. In the epochs of A and C the FGGRL is expected to generate gaits that have a stability margin close to or larger than the nominal value of *7cm*. There is a single gait with which the robot can walk with the speed of *4 units per iteration*: the tripod gait with the successive states of 0-5-0-5-0-5 and 5-0-5-0-5-0. Therefore there cannot be any improvement in the stability margin in epoch B. The reinforcement in epoch B is always a large negative value, so that no state transition is memorized. In other words, the reinforcement learning part of FGGRL is not functioning. The epoch B demonstrates only the smooth passage from the speed of *1* to *4 units per iteration*, and the successful generation of the single choice of states by free state generation algorithm. The passage from epoch C to D might also demonstrate the smooth passage feature of the FGGRL. Within epoch D, although there are a few different gaits that can be followed, very few of them have a stability margin close to *7cm*. Although a slight improvement can be expected in stability margin, it is impossible to come close to the nominal value in the average.

In the end of epoch D and start of epoch E, the rear left leg is disfunctioned. The FGGRL algorithm is not aware of this situation. In the simulation the robot falls while walking with the speed of *3 units*, and it is assumed that it gets an external signal. With this signal the robot only understands that there is a stability problem without knowing its reason. Then it passes to the speed of *1 unit per iteration* and adopts the form of FGGRL for external stability problem situations. In this epoch the robot tries successive states making state generations as if all the legs were in order.

Many of these states result in falling of the robot, while some result in stable transitions. What is expected from the robot is to memorize the stable transitions it experiences and come up with a five-legged gait pattern that results in a continuously stable walk.

The success of the FGGRL is evaluated based on its performance in epochs A, C, and E. If the robot achieves a walk with continuous stability margin larger than *6cm* in the last ten steps of the epochs A and C it is considered to be successful, respectively in these epochs. If the robot can achieve a continuous stable walk with five legs in the last ten steps it is considered to be successful in epoch E.

In computer simulation the described 500 step walk was performed for 1000 episodes. The FGGRL was successful 891 times in epoch A, 996 times in epoch C, and 685 times in epoch E. Among those, in 604 of the trials the algorithm was successful in three of the epochs together. The algorithm is more successful with the speed of *2 units per iteration* compared to *1 units*. This inclines one to comment that the ratio of the number of states with stability margin larger than *7cm* to the all possible stable states is larger for the speed of *2 units per iteration* compared to *1*. Although the computer calculations for the state 3-0-3-0-3-3 given in Section 3.2 support this comment, it needs a mathematical proof, which will not be dealt here.

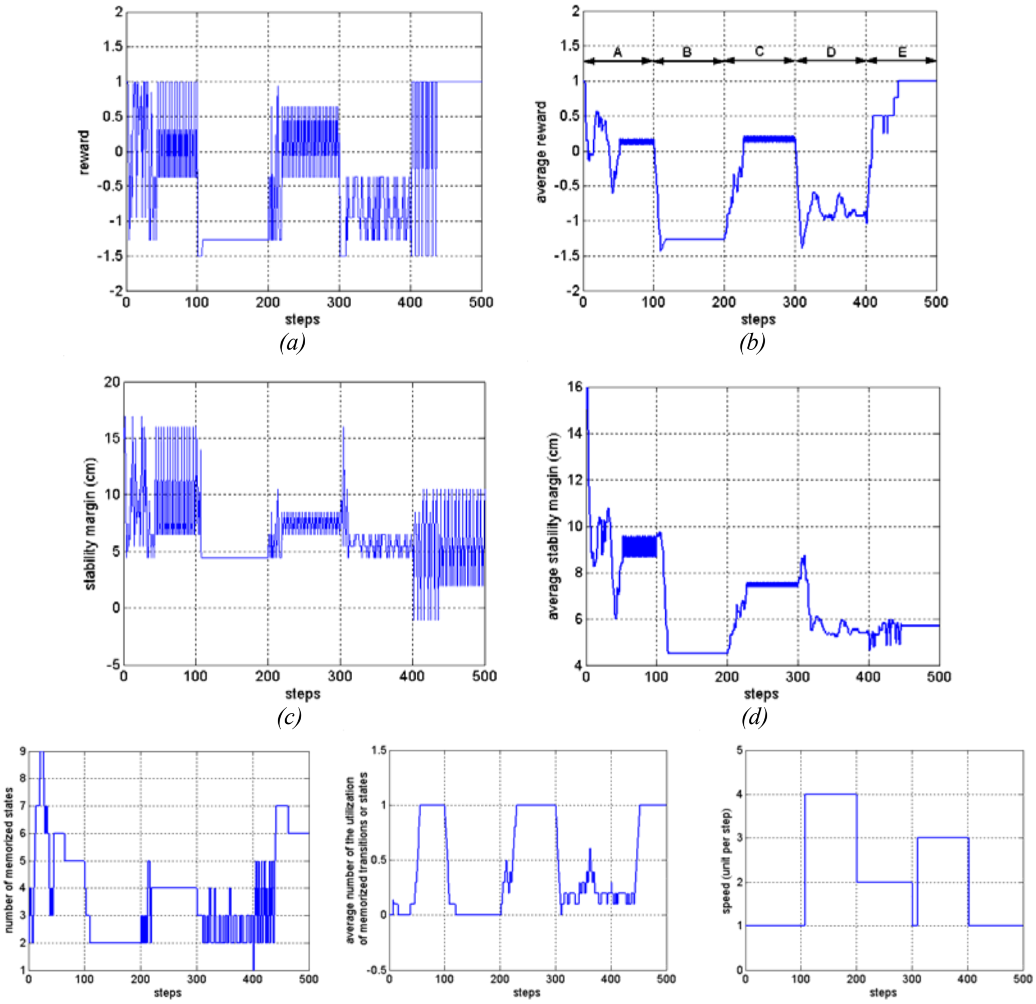
In Figure 3.4, the results of a successful trial is shown. In *(a)* the actual reward returned after each iteration is plotted. In *(b)* the average reward in the vicinity of each step is given. The averages in the figures are calculated using the results of preceding ten steps. In the epochs A, C, and E one can observe the increase and stabilization of the value of rewards. In epochs A and C the average reward is managed to be raised above *0* in the last 40 and 60 steps, respectively. In epoch E, the average reward is *1* in the last 50 steps. This shows that the robot learned to walk with five legs in about 50 steps. In this figure, the number of attempts performed after falling, which are explained in 4th and 5th steps of the five-legged learning algorithm, are not included. The number of these attempts for this result is 23. The pattern of successive states learned for five legged walking is as follows,

[4 3 0 4 0 ~~4~~] → [5 4 0 5 2 ~~0~~] → [0 5 3 0 3 ~~3~~] → [2 0 4 2 4 ~~4~~] → [3 2 5 3 5 ~~0~~] → [4 3 0 4 0 ~~4~~]

with the respective stability margins of 2, 5.5, 2, 8.5, 10.5, and again 2 cm. The state of the sixth leg is signed with double strike, because it is not used by the robot, although commanded by the FGGRL.

In *(c)* and *(d)*, in Figure 3.4, the average stability margin for each iteration, and the actual stability margin after each iteration are given, respectively. One can observe the affinity between the stability margin and reward traces for the epochs A, B, C, and D. Such an affinity is already not expected for the epoch E, because the reward does not depend directly on the stability margin as in the others. In *(e)* and *(f)*, the average number of utilization of memorized states or transitions and the number of states memorized are given respectively. As it is observed, in the beginning of the epochs

A, C, and E the number of memorized states increases, and in the middle or towards the end it decreases and remains at a constant value. Accordingly, the utilization of the memorized states is low at the beginning of the three epochs, and full utilization is performed after the middle. For the epoch B the utilization is always in the zero level, while in C it remains in very low values. The two states memorized in epoch B are the initialization state which always exists in the memory, and the current state; namely, no new state memorization is performed in epoch B. In figure *(f)* the actual speed of the robot is given for each step. One should observe that immediately after the 100th and 300th steps, the actual speed is not the commanded speed but *1 unit per step*. This is because the robot advances with the speed of *1 unit per step* in order to catch a suitable state from which it can change to the commanded speed.



(a) *(b)*

(c) *(d)*

(e) *(f)* *(g)*

Figure 3.4: Results for a successful FGGRL in computer simulation.*(a)* Reward for each step, *(b)* average reward of the last ten steps for each step, *(c)* stability margin for each step, *(d)* average stability margin of the last ten steps for each step, *(e)* number of memorized states for each step, *(f)* average number of utilization of the memorized transitions or states, *(g)* speed of the robot at each step.

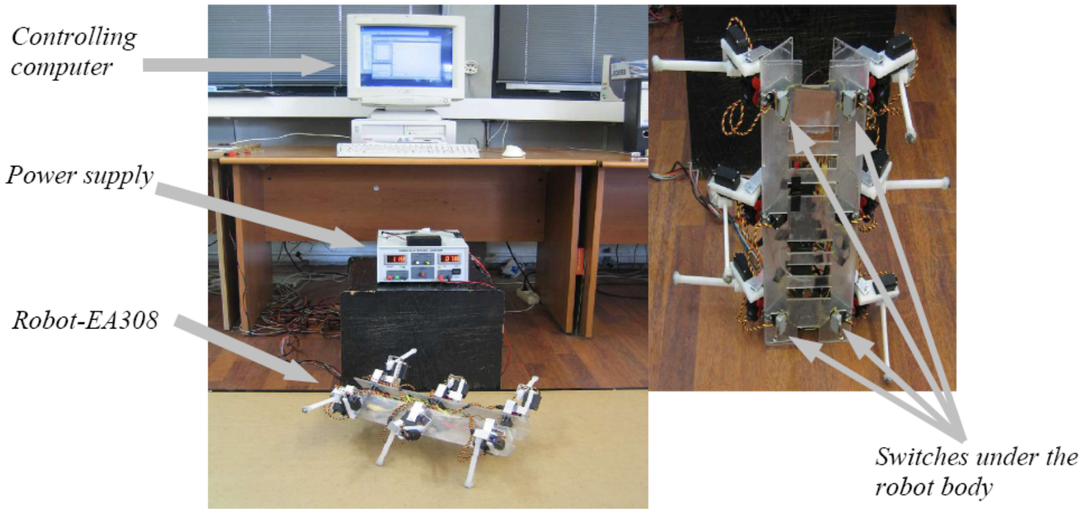


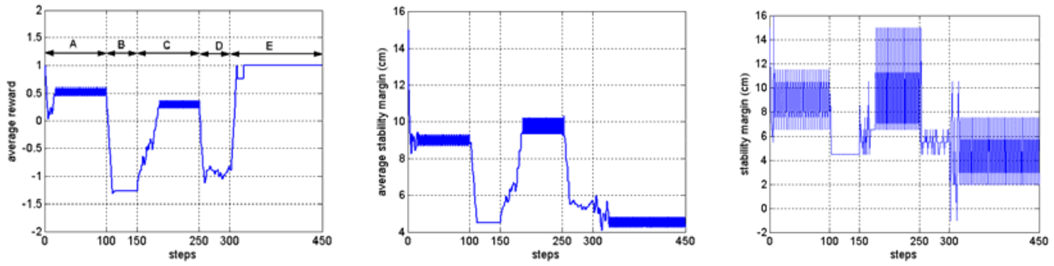
Figure 3.5: Experimental setup for the Robot-EA308 for learning of five-legged walking.

3.7. Results with the Actual Robot-EA308

The FGGRL is applied to the Robot-EA308 with the experimental setup shown in Figure 3.5. For the learning of five-legged walking the robot is equipped with four switches beneath its body frame, and the necessary hardware to send the signal of falling to the computer. When one of these switches is closed, the robot is considered to fall down, and the controlling computer can detect this through the connection between the hardware on the robot and its parallel port. The successive joint angle commands, which result in motions of the legs, are calculated by the controlling computer according to the output of the FGGRL algorithm. These comments are sent to the servo-controllers on the robot via the serial port of the computer.

It should be remembered that in the first 400 steps, the reinforcement is based on stability margin; therefore, it is calculated internally by the computer simulation. It is only in the last 100 steps, namely after one of the rear legs is out of order, that a real feedback from the actual robot is used as the reinforcement signal.

In Figure 3.6, the results of a successful FGGRL application on the actual Robot-EA308 are shown. For this result, again the right rear leg (sixth leg) is out of order in epoch E. Since the results obtained on the real robot are very similar to the ones obtained from the computer simulations, only the figures for average reward, average stability margin, and stability margin are given. In the application on the real robot, the length of epochs B and D are limited to 50 steps, because the gait patterns generated in these epochs do not differ much in time. The epoch E, on the other hand, is enlarged to 150 steps, in order to see the robot walking with five legs for some time.



(a) *(b)* *(c)*

Figure 3.6: Results for a successful FGGRL on the actual Robot-EA308. *(a)* Reward for each step, *(b)* average reward of the last ten steps for each step, *(c)* stability margin for each step.

For the epochs A and C, the robot manages a walk with large stability margins towards the middle of the epochs. Especially for the epoch C, it was possible to observe that the robot was walking with a poor stability in the beginning, and could manage a safe walk only afterwards. It was also possible to observe the smooth passage from *2 units* to *3 units per step* in the very beginning of the epoch D: the robot first made steps with *1 unit* advancement, and passed to *3 units* after three steps.

One can observe in Figure 3.6, that the robot learned to walk with five legs after about 23 real steps. The number of attempts due to the falling throughout these 23 steps was 18. The states that constitute the five-legged walk after the 23 steps are

[0 5 0 3 5 ~~3~~]→[4 0 2 4 0 ~~4~~]→[5 0 3 0 4 ~~0~~]→[0 5 0 3 5 ~~3~~]

with stability margins of 4, 2, 7.5, and again back to 4cm respectively. The slights that show the robot in these states are given in Figure 3.7, with the same order.

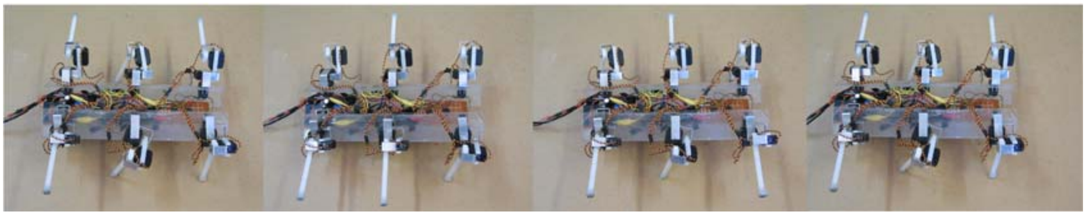
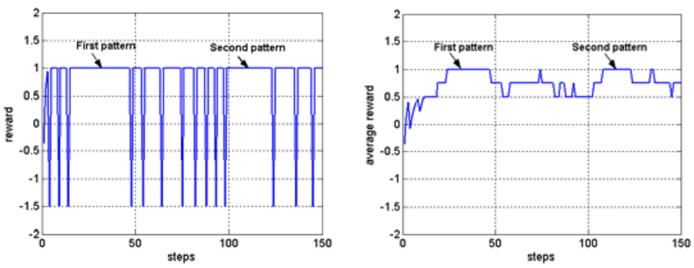


Figure 3.7: Slights showing the Robot-EA308 while walking with the five-legged gait pattern composed of the sequential states [0 5 0 3 5 ~~3~~]→[4 0 2 4 0 ~~4~~]→[5 0 3 0 4 ~~0~~]→[0 5 0 3 5 ~~3~~], from left to right, respectively.



First Pattern	*Stab.marg. (cm)*
[4 0 2 0 5 ~~0~~]	7.5
[5 3 0 5 0 ~~2~~]	4
[0 4 5 0 5 ~~3~~]	3
[3 5 0 5 0 ~~4~~]	2
[4 0 2 0 5 ~~0~~]	7.5

Second Pattern	*Stab.marg. (cm)*
[3 5 2 0 4 ~~4~~]	8.5
[4 0 3 5 0 ~~5~~]	3
[5 3 4 0 2 ~~0~~]	8.5
[0 4 0 5 3 ~~3~~]	1
[3 5 2 0 4 ~~4~~]	8.5

Figure 3.8: Results of a five-legged learning episode, in which the “on-the-border” problem is observed. *(a)* Reward for each step, *(b)* average reward of the last ten steps for each step, *(c)* the first and second patterns of states with their stability margins.

For the application in Figure 3.6, it is observed that although the robot found a pattern of five legged walking, it occasionally lost its stability on the state with the minimum stability margin of *2cm*. However, in its first attempt it managed to have a stable step with the same state. (Since the attempts are not considered as real steps, the fallings are not observed on the figures.) Therefore, these fallings did not alter the learned pattern in this example. However, in some other episodes of learning of five-legged walking with the actual robot, the irregular stability with a state created problems for learning. For example, a state with stability margin of *2cm* resulted in falling in some cases; or, a state with stability margin of *1cm*, which was considered to be unstable in the simulation, resulted in a stable standing in some cases, but falling in others. Such an “on-the-border” state results the following: First the robot considers the state as stable, and starts to establish a pattern with it. But after some time the state either becomes totally unstable, or sometimes stable but sometimes unstable. In such a case, if the robot manages to make the state stable in its attempts, the learned pattern is preserved, with occasional falls and recoveries. Otherwise, the learned pattern is lost with cleaning of the problematic state from the memory. These two phenomena are easily observable; because, the on-the-border state damages an already established pattern. However, an on-the-border state might also result in delaying of learning of a pattern, which is harder to be observed.

In Figure 3.8, the results of a five-legged learning episode is given as an example in which the problem of on-the-border state is observed. During this episode a pattern (First Pattern) is established and used between the 24th and 47th steps. In step 48, the state with stability margin of *2cm*, behaved unstable and the pattern was lost. The robot managed to establish another pattern (Second Pattern) with a state of *1cm* stability margin, and used it between the 118th and 123rd steps. However, at the 124th step this state behaved unstable and the pattern was again lost.

With the actual robot 30 episodes of 150 steps were performed separately for learning of five-legged walking. In half of those the right, and in the other half the left rear leg was deficient. No difference was observed between the right or left one being deficient, concerning the success of learning. The result of each episode is categorized with one of the four success categories of “perfect”, “acceptable”, “late”, and “not-learned”. “Perfect” means that the robot learned a five-legged walking pattern in the first 100 steps and managed to finish the total 150 steps with the same pattern. “Acceptable” means that the robot managed to learn a five-legged walking pattern in the first 100 steps, but because of on-the-border states, lost the learned pattern. “Late” means that the robot could learn a five-legged walking pattern only in the last 50 steps. And lastly, “not-learned” means the robot could not learn a pattern in all 150 steps. Among the 30 episodes, the results of 9 were “perfect”, 10 were “acceptable”, 5 were “late”, and 6 were “not-learned”. With the perfect results the robot learned five-legged walking in about 5 to 10 minutes and finished the 150 steps in less than 15 minutes. With the acceptable results, although a pattern is generated again in 5 to 10 minutes, the fulfillment of 150 steps lasted up to 20 minutes. The late and not-learned episodes were finished in between 20 to 25 minutes.

If having established a five-legged walking pattern in the first 100 steps is considered as “successful”, as done for the simulation results, 19 of the episodes (“perfect” or “acceptable”) were successful, and the remaining 11 (“late” and “not-learned”) were unsuccessful, with a success ratio of 63%. This is slightly larger than that of the five-legged learning with the simulation. Why is there such a slight increase despite the negative effects of the “on-the-border” phenomena? Because, with the actual robot some states with a stability margin of *1* or even *0cm*, result in stable standings and are used in the patterns. This is not possible with the simulation, where any state with a stability margin less than *2cm* is consistently unstable. Existence of such states in the real robot application, enlarges the space of stable states in the search space, and makes learning easier. It should be noted, however, that the negative effect of on-the-border states, might be larger than the mentioned positive effect in other samples of episodes. Therefore, “on-the-border” effect should be stressed here as a severe problem to be considered in application of reinforcement learning to learning of walking.

3.8. Conclusion

In this chapter the problems of free gait generation, continuous improvement of gaits, and adaptation to unexpected conditions are addressed for a six-legged robot. While the free gait generation corresponds to the central part of gait generation, the reinforcement learning incorporated with this corresponds to the reactive part. The possible walking patterns are generated by the free gait generation, and among these the ones most successful for adaptation are memorized and utilized by the reinforcement learning. In this way the approaches of central pattern generation and reflex model are somewhat conciliated and their efficiency are utilized in a single gait generation scheme.

The reinforcement learning makes the robot adopt more stable gaits in normal conditions of no external effect of instability. As a result, the stability of walking with a commanded speed is continuously increased till a pattern with each state having good stability margin is achieved. When the speed command is changed, the robot performs a smooth passage with probably slowing down to the speed of *1 units per step*, and then adopting the new speed in a few steps. This smooth passage between commanded speeds should be noted as a peculiar feature of the algorithm here. Both increasing the stability margin and smooth passage between different speeds are successfully observed on the real Robot-EA308.

The adaptation capability of the learning scheme is tested on an unexpected case for the robot, namely when there occurs a deficiency on one of the rear-legs. In this application the robot is not aware of the cause, but experiences falls although it generates statically stable gaits. The fallings result in negative reinforcement and the robot learns to avoid them by memorizing and making use of the stable states. The learning of five-legged walking is realized with the actual Robot-EA308. The switches beneath the robot body enabled the computer to realize the falls.

The main difference between applying the reinforcement learning with simulation and with the actual robot is designated as the effect of “on-the-border” situations. This problem rises especially because the reinforcement signal taken from the actual robot has a binary character. The robot either falls or not. The scheme here cannot handle the situations when a state behaves indefinitely. If the reinforcement signal could designate a continuous stability value it would be better for the algorithm to avoid on-the-border states. However, for the case of avoidance of falling, such a reinforcement signal is not easy to be generated with simple switches. One way to overcome this problem might be to incorporate the internal reinforcement, designating the stability margin depending on the simulation model, with the reinforcement taken from the actual robot. Future work based on this chapter might focus on developing the reinforcement signal in order to handle the external indefiniteness in learning of durable five-legged stable gaits.

CHAPTER 4

PROTRACTION OF A THREE-JOINT ROBOT LEG: TRAJECTORY OPTIMIZATION AND CONTROLLER DESIGN

4.1. Introduction: Protraction of Legs

The focus of this chapter is the protraction movement of a single three-joint robot leg. Protraction of a leg corresponds to carrying the tip point from a given PEP to a given AEP, by following a proper trajectory. In the literature there are quite scarce studies that explicitly deal with the protraction movement of robot legs. Ilg and Berns (1995) and Ilg et al. (1997) are two papers that deal with application of reinforcement learning for the power stroke (retraction) and return stroke (protraction) of a robot leg. In these papers the focus is the reinforcement learning scheme which is used to learn the parameters of some oscillatory networks controlling the leg movements. The efficiency of the resulting movements are out of concern. In Cruse et al. (1998) a general control structure for protraction is described under the subtitle of “control of the swing movement”. Similarly, in Dürr et al. (2004) protraction is explicitly mentioned as one of the four mechanically uncoupled swing movements of robot legs. The other three movements mentioned are searching, grasping, and grooming. These papers content with description of very general, and in fact similar, kinematic control models, which aim to imitate the biologically inspired movements observed in insects. However, the stress on the similarities between the mentioned four movements in Dürr et al. (2004) reveal the importance of the control for such group of swing movements. This points to the fact that a protraction movement generator should be applicable to be used for similar swing movements other than protraction. Such movements are especially the reflex movements to be performed when the robot comes across unexpected obstacles, holes, or hills (Espenschied et al., 1996). The optimization and control structure developed in this chapter is generalizable to any similar movement once the initial-final tip point positions are provided. In Erden and Leblebicioğlu (2004) we had performed a preliminary work for the approaches developed in this paper.

The biological observations reveal that the protraction time in six-legged insects is constant regardless of the speed of walking. In two of the models of legged locomotion (Section 1.2; Ferrell, 1995), namely in the Wilson and Pearson models, it is explicitly stated that the protraction time is constant. This is sense-full since there is no power load on the leg during the swing motion, and it is possible to protract in a quite short and constant time. Therefore, it is a reasonable approach to follow this biological observation in robotic applications. The aim in this work is to generate an efficient trajectory to carry the tip point of the leg from any given PEP to any given AEP in a given time which is taken to be *1.5 seconds*, as a value proper for the Robot-EA308. Efficiency means dissipation of minimum energy in the three actuators during protraction.

The chapter is organized as follows: Section 4.2 gives the kinematic and dynamic modeling of a three-joint leg. Section 4.3 gives the problem formulation and introduces the "object function modified optimization" approach for trajectory optimization. In this section, also the usage of radial basis function neural network (RBFNN) for interpolation of optimal trajectories is mentioned. In Section 4.4 the controller design in order to make the tip point to follow any guiding trajectory is presented. Section 4.5 gives the results that compare the performance of the controller with optimized trajectories. In this section, also some slights of the Robot EA308 are given while protracting through the trajectory generated by the designed controller. Section 4.6 gives concluding remarks with mentioning of probable future work.

4.2. Three-Joint Robot Leg: Kinematic Model and Dynamic Equations

A three-joint robot leg can be considered as a three-link revolute joint (RRR) manipulator, which is attached to a stationary base (robot body). Therefore, the kinematic modeling and derivation of dynamic equations can be performed following the conventional robotics approaches (Fu et al., 1987). The kinematic model here is derived by defining the reference frames according to the Denavit-Hartenberg convention. In Figure 4.1 a graphical representation of a three joint robot leg is given, with the attached reference frames and corresponding joint variables. In this figure the body *(b)* and the zeroth *(0)* reference frames are attached to the stationary robot body. Therefore they can be both considered as inertial frames. (It is assumed throughout the chapter that the motion of the body is much more slower than the motion of the leg during protraction. Therefore, the body is assumed to be stationary.) The axes of the body frame are arranged to be in accord with the actual robot-body orientation. The Denavit-Hartenberg link parameters based on Figure 4.1 are given in Table 4.1. The resulting homogeneous transformation matrices between the body and the zeroth frame, and between the sequential link frames are given in (4.1). In the formulas the variables represented by a_i stand for the length of the *ith* link (namely, the length of the portion of the link between the origins of *i-1st* and *ith* reference frames). The variables represented by θ_{ij} mean the sum of the *ith* and *jth* joint angles $(\theta_{ij}=\theta_i+\theta_j)$. *C* and *S* correspond to cosine and sinus functions, respectively. The values used for these parameters belong to the Robot-EA308: $\Psi=45^0$; $a_1=1.5\ (cm)$; $a_2=5.25\ (cm)$; $a_3=13.25\ (cm)$. Homogeneous matrices are used in derivation of positional relations between the successive frames. In (4.3) the tip point position with respect to the body frame is given. The rotation matrices between the frames are given in (4.2). These rotation matrices are used in vector equations, especially while deriving the dynamic equations.

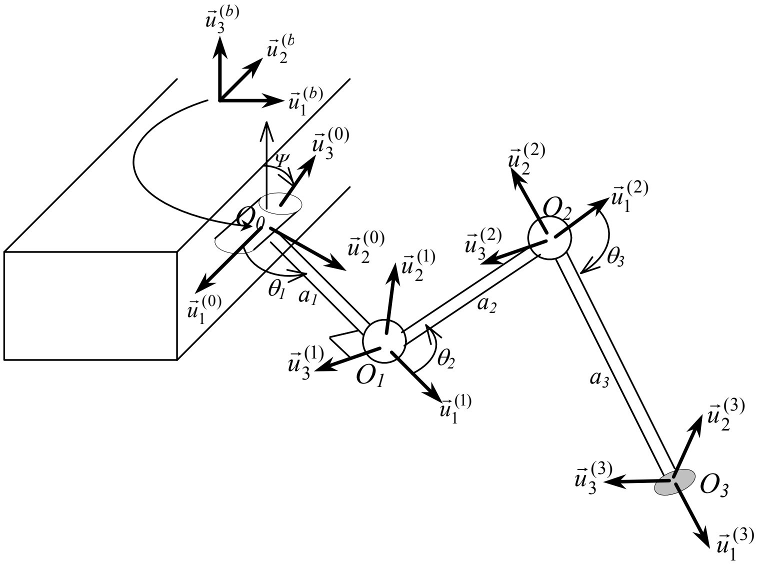


Figure 4.1: Three-joint robot leg: Reference frames and joint variables.

Table 4.1: Denavit-Hartenberg link parameters for the three joint robot leg.

Joint	$\boldsymbol{\theta_i}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{a_i}$	$\boldsymbol{d_i}$
1	θ_1	$\pi/2$	a_1=1.5cm	*0*
2	θ_2	*0*	a_2=5.25cm	*0*
3	θ_3	*0*	a_3=13.25cm	*0*

$$\hat{H}^{(b,0)}=\begin{bmatrix}0 & \cos(\Psi) & \sin(\Psi) & 0\\ -1 & 0 & 0 & 0\\ 0 & -\sin(\Psi) & \cos(\Psi) & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad \hat{H}^{(k-1,k)}=\begin{bmatrix}\cos(\theta_k) & -\sin(\theta_k)\cos(\alpha_k) & \sin(\theta_k)\sin(\alpha_k) & a_k\cos(\theta_k)\\ \sin(\theta_k) & \cos(\theta_k)\cos(\alpha_k) & -\cos(\theta_k)\sin(\alpha_k) & a_k\sin(\theta_k)\\ 0 & \sin(\alpha_k) & \cos(\alpha_k) & d_k\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{4.1}$$

$$\hat{H}^{(0,3)}=\hat{H}^{(0,1)}\,\hat{H}^{(1,2)}\,\hat{H}^{(2,3)}\quad \hat{H}^{(b,3)}=\hat{H}^{(b,0)}\,\hat{H}^{(0,3)}$$

$$\hat{C}^{(b,0)}=\begin{bmatrix}0 & \cos(\Psi) & \sin(\Psi)\\ -1 & 0 & 0\\ 0 & -\sin(\Psi) & \cos(\Psi)\end{bmatrix}\quad \hat{C}^{(k-1,k)}=\begin{bmatrix}\cos(\theta_k) & -\sin(\theta_k)\cos(\alpha_k) & \sin(\theta_k)\sin(\alpha_k)\\ \sin(\theta_k) & \cos(\theta_k)\cos(\alpha_k) & -\cos(\theta_k)\sin(\alpha_k)\\ 0 & \sin(\alpha_k) & \cos(\alpha_k)\end{bmatrix} \tag{4.2}$$

$$\bar{P}_e^{(b)}(\bar{\theta})=\begin{bmatrix}C\Psi(a_1S\theta_1+a_2S\theta_1C\theta_2+a_3S\theta_1C\theta_{23})+S\Psi(a_2S\theta_2+a_3S\theta_{23})\\ -(a_1C\theta_1+a_2C\theta_1C\theta_2+a_3C\theta_1C\theta_{23})\\ -S\Psi(a_1S\theta_1+a_2S\theta_1C\theta_2+a_3S\theta_1C\theta_{23})+C\Psi(a_2S\theta_2+a_3S\theta_{23})\end{bmatrix} \tag{4.3}$$

In order to derive the dynamic equations, first the inertia matrices of the links should be determined. Since the *kth* reference frame is stationary with respect to the *kth* link, the inertia tensor of the *kth* link around its center of mass appears to be a constant matrix with respect to the *kth* reference frame, as in (4.4). In Appendix A one can find the formulas for calculation of link inertia matrices. The values used in these formulations belong to the Robot- EA308. The resulting matrices for each link are in the form of (4.5).

$$\{\check{J}_k\}^{(k)} = \hat{J}_k^{(k)} = \hat{J}_k \tag{4.4}$$

$$\hat{J}_k = \begin{bmatrix} J_{k1} & 0 & 0 \\ 0 & J_{k2} & 0 \\ 0 & 0 & J_{k3} \end{bmatrix} \tag{4.5}$$

The dynamic equations are derived using the Lagrangian formulation. The Lagrange function *(L)* is defined as the difference between the kinetic *(K)* and potential *(U)* energies. In the general sense, the generalized forces of a conservative system are determined as in (4.6).

$$L = K - U \qquad \tau_k = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} \tag{4.6}$$

Application of this principle to the three-joint leg system leads to (4.7) where, q_k =joint variables (joint angles θ_k); $\dot{q}_k$ =derivative of joint variables (joint angle velocities $\dot{\theta}_k$); Q_k =joint actuator torques; D_k =effect of the external forces on the joints. Since there is no external effect on the leg throughout the protraction phase, the term D_k is equal to $\bar{0}$; therefore it will be omitted from the other equations.

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_k}\right) - \frac{\partial K}{\partial q_k} + \frac{\partial U}{\partial q_k} = Q_k + D_k \tag{4.7}$$

The kinetic energy and potential energy of the system are given as in (4.8) and (4.9), respectively, where $\vec{v}_k$ stands for the mass-center velocity, $\vec{\omega}_k$ for the angular velocity, and $\vec{m}_k$ for the mass of the k^{th} link; $\vec{p}_k$ stands for the mass-center position of the k^{th} link from the origin of the zeroth reference frame, and $\vec{g}$ stands for the gravitation.

$$K = \sum_{k=1}^{3}\left[\frac{1}{2} m_k \vec{v}_k \cdot \vec{v}_k + \frac{1}{2} \vec{\omega}_k \cdot \check{J}_k \cdot \vec{\omega}_k\right] = \sum_{k=1}^{3}\left[\frac{1}{2} m_k \bar{v}_k^{(0)t} \bar{v}_k^{(0)} + \frac{1}{2} \bar{\omega}_k^{(k)t} \hat{J}_k^{(k)} \bar{\omega}_k^{(k)}\right] \tag{4.8}$$

$$U = -\sum_{k=1}^{3} m_k \vec{g} \cdot \vec{p}_k = -\sum_{k=1}^{3} m_k \bar{g}^{(0)t} \bar{p}_k^{(0)} \tag{4.9}$$

Once the kinetic energy is represented in the form of (4.10), and the gravitational term of G_k is defined as in (4.11), the general dynamic equation of (4.7) can be written more explicitly as in (4.12) and (4.13).

$$K=\frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} m_{ij}\dot{q}_i\dot{q}_j=\frac{1}{2}\dot{\bar{q}}^t\hat{M}\dot{\bar{q}} \qquad \frac{\partial K}{\partial \dot{q}_k}=\sum_{i=1}^{3} m_{ik}\dot{q}_i=\hat{M}\dot{\bar{q}} \tag{4.10}$$

$$G_k=\frac{\partial U}{\partial q_k} \tag{4.11}$$

$$\frac{d}{dt}\left[\sum_{i=1}^{3} m_{ki}\dot{q}_i\right]-\frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j+G_k=Q_k \tag{4.12}$$

$$\sum_{i=1}^{3} m_{ki}\ddot{q}_i+\sum_{i=1}^{3}\sum_{j=1}^{3}\left[\frac{\partial m_{ki}}{\partial q_j}-\frac{1}{2}\frac{\partial m_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j+G_k=Q_k \tag{4.13}$$

Equation (4.13) can be put in a matrix form, which will gather all three joint variables, as in (4.14). In equation (4.14), $\hat{M}$ is the mass (inertia) matrix; $\bar{C}$ is the vector of coriolis, centrifugal, and gyroscopic terms; $\bar{G}$ is the vector of gravitational terms; and $\bar{Q}$ is the vector of the three joint torques. The values of these variables can be found in Appendix B.

$$\hat{M}(\bar{q})\ddot{q}+\bar{C}(q,\dot{q})+\bar{G}(q)=\bar{Q} \tag{4.14}$$

4.3. Trajectory Optimization

The trajectory optimization is performed by using the gradient based “optimal control theory” (Kirk, 1970, pp.184-209, 236-240, 330-343). It is well known that gradient based optimization algorithms suffer from sticking to local optimums. Especially when the dimension of vectors to be optimized is large, this problem is more severe. In trajectory optimization problems the dimension of vectors that represent the whole trajectory happen to be quite large. In order to cope with this problem it is a common approach to represent trajectories with parametrized polynomials which satisfy the initial and final position requirements. Cubic splines are widely used for this purpose (Bobrow et al., 2001; Chettibi et al, 2004; Garg and Kumar, 2002; Saramago and Stefen, 1998). Once the trajectories are represented by a small number of spline parameters, the trajectory optimization problem is reduced to optimize those parameters with respect to a given objective function. Application of non-gradient based optimization algorithms is also common to be used with such reduced representation of trajectories. Genetic algorithms is one of the most popular of those applied in trajectory optimizations. In Garg and Kumar (2002), trajectory optimization for a two-link manipulator is reduced to optimization of two parameters with genetic algorithms.

With polynomial representations trajectories are prisoned to a limited search space defined by the structure of the polynomial. Many of the feasible trajectories are excluded from the solution space. Therefore the most optimistic solution of those approaches is the best solution in the space represented by the structure of the polynomial. The search is then not performed everywhere in the solution space. If it is not desired to sacrifice the freedom of the trajectory, it is not proper to adopt reduced representations of trajectories based on polynomial parameters. The attempt in this chapter is to utilize full trajectory representation in optimization, in order to have access to the whole solution space. However, the usage of full trajectory representation necessitates a large number of unknowns to be optimized. This makes it difficult to be handled by non-gradient based optimization algorithms. For example, the coding of a large input vector leads to too large chromosomes for genetic algorithms. Moreover, in evolutionary optimizations, the parameters are optimized independently, which may lead to non-smooth (non-continuous) solution sets. Gradient based optimization is more proper from the stand point of handling high dimensionality, and producing analytical solutions, however, with the problem of sticking to local optimums. The approach here attempts to utilize gradient-based optimization using the optimal control theory. The problem of local optimality is intended to be overcome by objective function modification in different epochs of gradient-based optimization. In the following the formulation for optimal control, the objective function modification, and optimization results are presented. The optimal trajectories are processed and an interpolating RBFNN is constructed to produce near-optimal trajectories for any given initial and final point positions.

4.3.1. State Space Representation and Hamiltonian Formulation of a Three-Joint Leg System

In a robotic manipulator the actual inputs to the system are the voltages supplying the energy to the actuators. Once the actuators are omitted from the formulation the inputs to the mechanical system remain to be the actuator forces and torques. Namely, for a mechanical formulation of three joint leg system the actual inputs are the three joint torques, $(\bar{Q})$. However, if the aim is trajectory optimization, some auxiliary control variables can be chosen as the input, and then the required torques can be calculated using (4.14) (this approach is followed also in Frangos et al. (2001)). Accordingly, the inputs to the system here are chosen to be the angular accelerations of the joints, $(\bar{\theta})$.

There are three kinds of requirements in the formulation: system dynamics, initial and final state conditions, and joint angle constraints. In handling these requirements the approaches in Kirk (1970) are followed. Initial state conditions are imposed in the initialization of the optimization; the final state conditions are imposed in the objective function as a penalty term; joint angle constraints are imposed using a supplementary state variable, named as the constraint-state; and the system dynamics are imposed in the dynamic equations of the state variables. There is no restriction on the control input. The states are of three groups: the joint angles $(\bar{\theta})$; the joint velocities $(\dot{\bar{\theta}})$; and a constraint-state. The state and input vectors of the overall system are given in (4.15).

$$\bar{x}=\begin{bmatrix}\bar{x}_q\\ \bar{x}_{\dot{q}}\\ x_c\end{bmatrix}=\begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\\ x_c\end{bmatrix}=\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\end{bmatrix} \qquad \bar{u}(t)=\begin{bmatrix}\ddot{\theta}_1(t)\\ \ddot{\theta}_2(t)\\ \ddot{\theta}_3(t)\end{bmatrix} \tag{4.15}$$

Following the notation in Kirk (1970) the system dynamics can be given as

$$\dot{\bar{x}}=\bar{a}(\bar{x}(t),\bar{u}(t),t) \tag{4.16}$$

where the state derivatives are defined as follows:

$$\begin{aligned}
\dot{x}_1(t)&=a_1(\bar{x}(t),\bar{u}(t),t)=\dot{\theta}_1=x_4\\
\dot{x}_2(t)&=a_2(\bar{x}(t),\bar{u}(t),t)=\dot{\theta}_2=x_5\\
\dot{x}_3(t)&=a_3(\bar{x}(t),\bar{u}(t),t)=\dot{\theta}_3=x_6\\
\dot{x}_4(t)&=a_4(\bar{x}(t),\bar{u}(t),t)=\ddot{\theta}_1=u_1\\
\dot{x}_5(t)&=a_5(\bar{x}(t),\bar{u}(t),t)=\ddot{\theta}_2=u_2\\
\dot{x}_6(t)&=a_6(\bar{x}(t),\bar{u}(t),t)=\ddot{\theta}_3=u_3\\
\dot{x}_7(t)&=a_7(\bar{x}(t),\bar{u}(t),t)=f_c(\bar{x}(t),\bar{u}(t),t)
\end{aligned}
\qquad
\dot{\bar{x}}=\begin{bmatrix}\dot{\bar{x}}_q\\ \dot{\bar{x}}_{\dot{q}}\\ \dot{x}_c\end{bmatrix}=\begin{bmatrix}\bar{x}_{\dot{q}}\\ u\\ f_c(\bar{x}(t))\end{bmatrix} \tag{4.17}$$

The seventh state is the constraint-state whose dynamics is defined by the function $f_c(\bar{x}(t),\bar{u}(t),t)$. There are two kinds of constraints on the joint angles: the ones imposed by the mechanical construction of the joints, and the one that limits the tip point from going under the ground level. These constraints are given in (4.18), where $t_{1a}=30^0$; $t_{1b}=150^0$; $t_{2a}=0^0$; $t_{2b}=135^0$; $t_{3a}=-150^0$; $t_{3b}=-35^0$;and the ground level is taken to be $z_g=-9\ cm$.

$$\begin{aligned}
&t_{1a}\leqslant\theta_1\leqslant t_{1b}\\
&t_{2a}\leqslant\theta_2\leqslant t_{2b}\\
&t_{3a}\leqslant\theta_3\leqslant t_{3b}\\
&p_z(\bar{\theta})=-S\Psi(a_1S\theta_1+a_2S\theta_1C\theta_2+a_3S\theta_1C\theta_{23})+C\Psi(a_2S\theta_2+a_3S\theta_{23})\geqslant z_g
\end{aligned} \tag{4.18}$$

Related to those constraints the following seven f functions are determined, all of which should be greater than 0 in order the constraints to be satisfied.

$$\begin{aligned}
f_1(\bar{x}(t),t)&=\theta_1-t_{1a}\geqslant 0\\
f_2(\bar{x}(t),t)&=t_{1b}-\theta_1\geqslant 0\\
f_3(\bar{x}(t),t)&=\theta_2-t_{2a}\geqslant 0\\
f_4(\bar{x}(t),t)&=t_{2b}-\theta_2\geqslant 0\\
f_5(\bar{x}(t),t)&=\theta_3-t_{3a}\geqslant 0\\
f_6(\bar{x}(t),t)&=t_{3b}-\theta_3\geqslant 0\\
f_7(\bar{x}(t),t)&=1000(p_z(\bar{\theta})-z_g)\geqslant 0
\end{aligned} \tag{4.19}$$

The dynamics of the constraint-state is defined by

$$\dot{x}_c(t)=f_c(\bar{x}(t))=\sum_{i=1}^{7}[f_i(\bar{x}(t))]^2\varphi(-f_i) \tag{4.20}$$

where

$$\varphi(-f_i)=\begin{Bmatrix} 0, & for\ \ f_i(\bar{x}(t),t)\geqslant 0\\ 1, & for\ \ f_i(\bar{x}(t),t)<0 \end{Bmatrix} \qquad for\ \ i=1,2,...7 \tag{4.21}$$

The initial and final conditions for the constraint-state are set to be *0*. Since the φ function is positive if the constraint is not satisfied, and zero if the constraint is satisfied, the derivative of the constraint-state happens to be either positive (if any of the constraints is not satisfied) or zero. The conditions that the initial and final values are zero, and the derivative is non-negative force the constraint-state to remain zero throughout the protraction period. In this way all the seven constraints in (4.19) are satisfied.

It is a common approach to use the integral of torque squares as an index of energy dissipation in the actuators of robotic manipulators (Bobrow et al., 2001; Garg and Kumar 2002; Liu and Abdel-Malek 2000). The torque vector of the three-joint leg system is given in (4.22). Combining the torque square integration with the final state conditions leads to the objective function in (4.23), in which again the notation of Kirk (1970) is followed.

$$\bar{Q}(t)=\bar{Q}(\bar{x}(t),\bar{u}(t))=\bar{Q}(\bar{\theta}(t),\dot{\bar{\theta}}(t),\ddot{\bar{\theta}}(t))=\hat{M}(\bar{q})\ddot{\bar{q}}+\bar{C}(\bar{q},\dot{\bar{q}})+\bar{G}(\bar{q}) \tag{4.22}$$

$$\begin{aligned}
J(\bar{u})&=h(\bar{x}(t_f),t_f)+J_s(\bar{x}_s,\bar{u})\\
&=h(\bar{x}(t_f),t_f)+\int_{t_0}^{t_f} g(\bar{x}(t),\bar{u}(t),t)\,dt\\
&=(\bar{x}(t_f)-\bar{x_{tf}})^T\hat{D}_h(\bar{x}(t_f)-\bar{x_{tf}})+\int_{t_0}^{t_f}\bar{Q}(\bar{x}(t),\bar{u}(t))^T\hat{D}_g\bar{Q}(\bar{x}(t),\bar{u}(t))\,dt
\end{aligned} \tag{4.23}$$

The optimization problem can be stated as

$$\begin{array}{ll} Minimize & J(\bar{u}) \\ subject\ to & \\ & \dot{\bar{x}}(t)=\bar{a}(\bar{x}(t),\bar{u}(t),t) \\ & \bar{x}(t_o)=\bar{x}_o \end{array} \tag{4.24}$$

This leads to a two-point boundary value problem in which the state and control variables are not constrained by any boundaries, the final time t_f is fixed, and $\bar{x}(t_f)$ is free (in fact it is not free, but it can be considered as free in the formulation since the requirement is imposed in the objective function by $h(\bar{x}(t_f),t_f)$). Following the optimal control approach with the notation in Kirk (1970), the Hamiltonian function for this optimization can be written as:

$$H(\bar{x}(t),\bar{u}(t),\bar{p}(t),t)=g(\bar{x}(t),\bar{u}(t),t)+\bar{p}(t)^T\bar{a}(\bar{x}(t),\bar{u}(t),t) \tag{4.25}$$

where $\bar{p}(t)$ corresponds to the costate vector.

The necessary and boundary conditions for an optimal solution can be written as follows:

Necessary conditions for optimality:

$$\dot{\bar{x}}^*(t)=\bar{a}(\bar{x}^*(t),\bar{u}^*(t),t) \tag{4.26}$$

$$\dot{\bar{p}}^*(t)=\frac{-\partial H}{\partial \bar{x}}(\bar{x}^*(t),\bar{u}^*(t),\bar{p}^*(t),t) \tag{4.27}$$

$$\bar{0}=\frac{\partial H}{\partial \bar{u}}(\bar{x}^*(t),\bar{u}^*(t),\bar{p}^*(t),t) \tag{4.28}$$

Boundary conditions:

$$\bar{x}(t_0)=\bar{x}_0 \tag{4.29}$$

$$\bar{p}(t_f)=\frac{\partial h}{\partial \bar{x}}(\bar{x}(t_f)) \tag{4.30}$$

The first two equations of the necessary conditions make up two differential equations whose initial and final conditions are given by the boundary conditions equations. The trajectory of input vector is approximated with a dense discretization of *51* instants, which results in quite a high dimensional optimization input of *3×51* length. Starting with an initial input trajectory, namely *51* values of the vector $\bar{u}(t)$ for the discretized instances, the state and costate equations can be solved numerically, by forward and reverse integrations, respectively. The third equation of necessary conditions is in fact nothing but the gradient of the objective function with respect to the input vector. Therefore, $\bar{u}(t)$ can be updated in the negative direction of this gradient in order to minimize the objective function, as in (4.31), where *i* stands for the iteration number. After some iteration the optimal $\bar{u}(t)$ trajectory, which makes the third necessary condition as close as possible to zero, can be achieved. This technique is called "the method of steepest descent for two-point boundary-value problems" (Kirk, 1970). The initial $\bar{u}(t)$ trajectory in this work is taken to be a zero matrix of 3×51 dimension. The value of α in (4.31) is determined by a one-dimensional search in every step. The optimization is terminated when the difference between two successive objective functions is less than *0.01*.

$$\bar{u}^{(i+1)}(t_k) = \bar{u}^{(i)}(t_k) - \alpha \frac{\partial H}{\partial \bar{u}}(t_k) \quad , \quad t_k = (k-1)\frac{t_f}{50} \quad , \quad k = 1,2,\ldots,51 \tag{4.31}$$

4.3.2. Optimization with Objective Function Modification

In the objective function of (4.23), there are two terms, respectively related to final state conditions and energy integration. These terms are weighted by the constant matrices of $\hat{D}_h$ and $\hat{D}_g$, respectively. These two are taken to be diagonal matrices as follows:

$$\hat{D}_h = d_h \begin{bmatrix} 500 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^7 \end{bmatrix} \qquad \hat{D}_g = d_g \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{bmatrix} \tag{4.32}$$

The elements of $\hat{D}_h$ and $\hat{D}_g$ are arranged to have comparable values for the corresponding two terms in the objective function. The entry corresponding to the constraint-state in the $\hat{D}_h$ matrix is chosen to be very large in order to guarantee the satisfaction of the constraints. The d_h and d_g values determine the relative strength of the two terms with respect to each other. Choosing a large d_h and a small d_g will result in a better achievement of final state requirements with little decrease in total energy dissipation; while a small d_h and a large d_g will result in a big decrease of total energy dissipation with non-achievement of final state requirements. Therefore, there is a trade-off between the two considerations represented by these two terms. Another problem is that, whatever values these two parameters are assigned, the optimization will stop in a local optimum, which does not give a satisfactory trajectory. For example, even though d_h is chosen very large compared to *dg* the optimization will stop at a local minimum in which the tip point is not brought to the desired tip point position. Therefore, the problem is of two fold: sticking to local optimums, and trade-off between the two terms.

The approach adopted to overcome these two problems is to make optimizations with changing the d_h and d_g parameters sequentially. In this way the local optimum is jumped over by the change of objective function, and the two terms are enforced sequentially one after the other. The optimization with objective function modification can be described as follows:

```
Step 1:
    i) Initialize $\bar{u}(t_k) = \bar{0}$ for k=0,1,...,51.
    ii) $d_h=d_{h,1}$; $d_g=d_{g,1}$.
    iii) Perform optimization.
    iv) Output $\bar{u}(t_k) = \bar{u}(t_k)^{[2]}$ for k=0,1,...,51.
Step i:
    v) Initialize $\bar{u}(t_k) = \bar{u}(t_k)^{[i-1]}$ for k=0,1,...,51.
    vi) $d_h=d_{h,i}$; $d_g=d_{g,i}$.
    vii) Perform optimization.
    viii) Output $\bar{u}(t_k) = \bar{u}(t_k)^{[i]}$ for k=0,1,...,51.
for i=2,...,6.
```

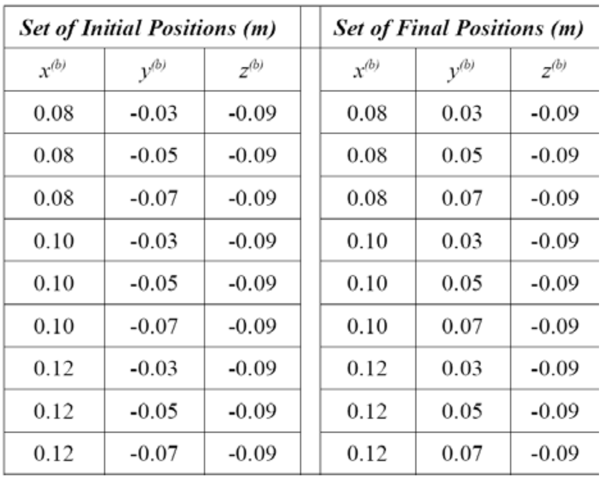
The changing values of d_h and d_g are given in Table 4.2. In *Steps 1-4*, a regular change is followed. In the step before the last, the jump of d_g is increased to give a last impulse in the direction of increased d_g. It should be noted that the last step corresponds to the actual objective function intended to be minimized by the overall optimization.

The movement of protraction is carrying the tip-point of the leg from a given position located on the ground in the backward of the leg origin, to another given position located on the ground in the forward of the leg origin. Therefore it is possible to determine a region of departure in the backward of the leg, and a region of arrival in the forward. These regions are determined to be two squares whose corners are given as [*(8,-7,-9);(12,-7,-9);(12,-3,-9);(8,-3,-9)*] and [*(8,3,-9);(12,3,-9);(12,7,-9);(8,7,-9)*] (*cm*), respectively, with respect to the body frame. The tip point is assumed to depart from a point in the square of departure to a point in the square of arrival. The ground level is taken to be *-9cm* in the *z*-direction. Nine points from each square are chosen as the positions for optimization. Eight of these are distributed homogeneously throughout the periphery of the square, and the remaining is taken to be the center point. As a result, $9\times 9=81$ initial-final position pairs are used in the optimization; therefore, a total of *81* optimal trajectories are obtained to represent the protraction from the square of departure to the square of arrival. These trajectories will be used to interpolate a trajectory for any given initial-final point positions in those squares. The set of the initial and final point positions used for optimization are given in Table 4.3.

Table 4.2: Sequential values of the weights of the objective function for "objective function modified optimization"

	Step 1	*Step 2*	*Step 3*	*Step 4*	*Step 5*	*Step 6*
d_h	1	10	1	10	1	10
d_g	10	1	10	1	100	1

Table 4.3: Set of initial and final tip-point positions used for optimization.



Set of Initial Positions (m)			*Set of Final Positions (m)*		
$x^{(b)}$	$y^{(b)}$	$z^{(b)}$	$x^{(b)}$	$y^{(b)}$	$z^{(b)}$
0.08	-0.03	-0.09	0.08	0.03	-0.09
0.08	-0.05	-0.09	0.08	0.05	-0.09
0.08	-0.07	-0.09	0.08	0.07	-0.09
0.10	-0.03	-0.09	0.10	0.03	-0.09
0.10	-0.05	-0.09	0.10	0.05	-0.09
0.10	-0.07	-0.09	0.10	0.07	-0.09
0.12	-0.03	-0.09	0.12	0.03	-0.09
0.12	-0.05	-0.09	0.12	0.05	-0.09
0.12	-0.07	-0.09	0.12	0.07	-0.09

4.4. Comparative Results of "Optimization With Single Objective Function" and "Optimization With Objective Function Modification":

Before giving the optimization results with the "objective function modification", here will be given the optimization results with the conventional approach, which corresponds to using the last objective function determined by the values in *Step 6* of Table 4.2. These results will be a basis to observe the improvement by the objective function modified optimization.

In Figure 4.2 the joint angles, velocities, and accelerations resulted from the *81* optimizations with "single objective function" are given in the first three rows, respectively. The last row shows the tip point position errors, namely the differences between the actual tip point and desired final tip point positions in the three coordinate axes. Among these results, a few of the optimizations resulted in feasible trajectories, which came close to the desired final tip point position with a movement of lifting up the tip point in the start. Some of these feasible ones are shown by arrows on some of the figures. Most of the remaining optimizations created unfeasible trajectories, which cannot be considered as a proper protraction at all. The important point here is that, not only the unfeasible ones but also the few feasible trajectories are the local minimums of the single objective function. A much more better improvement of those trajectories will be possible with the objective function modified optimization.

In Figure 4.3 the joint angles, velocities, and accelerations resulted from the *81* optimizations with "objective function modification" are given in the first three rows, respectively. Again the last row shows the tip point position errors. It is immediately realized how structured the figures are compared to the ones in Figure 4.2. In 79 of the results given in Figure 4.3, the optimization was successful to create a reasonable behavior of protraction: In the starting phase the leg is first retracted towards the body with raising the tip point above the ground level; in the middle phase the protraction towards the front is maintained; and in the end phase the leg is extracted carrying the tip point towards the desired final position. Among the *81* optimizations, only two of them resulted in a different behavior. These two are shown with arrows on the figures. In these two, the optimization did not manage to bring the tip point to the desired final position.

A comparison of the two optimizations regarding to cost minimization are given in Figures 4.4 and 4.5. Figure 4.4 shows the improvement for an infeasible case with “single objective function optimization”. The fist row figures show the leg movement from different sights for the optimization result with “single objective function”. The second row figures show the leg movement for the optimization result with “objective function modification”. It is clear how feasible and reasonable the movement on the second row figures is. The figures in the last row of Figure 4.4 show the main cost function values (nominal cost), whose parameters are given by the *Step 6* of Table 4.2. The dashed lines show the values for “optimization with single objective function”, and the solid lines show the values for “optimization with modified objective function”. The most left one of these depict the change of the main objective function (nominal cost). The ultimate aim of both optimizations is to minimize this function. As expected the cost decreases continuously in the case of “optimization with single objective function”. In the case of “optimization with objective function modification” the nominal cost increases and decreases with the modifications in the cost function. However, ultimately when the nominal cost function of *Step 6* is used, the resulting nominal cost is considerably lower than the one with “optimization with single objective function”. The middle figure in the row depicts the value of $h(\bar{x}(t_f),t_f)$, which is related with the achievement of final state requirements with the d_h value in *Step 6*. A lower value of this function means that the final state requirements are attained better. It is seen that this value is lower in the “optimization with objective function modification” compared to the case of “optimization with single objective function”. The right most figure in the bottom row of Figure 4.4 depicts the value of $\int_{t_0}^{t_f} g(\bar{x}(t),\bar{u}(t),t)dt$, which signify the sum of torque squares with the d_g value in *Step 6*. The difference for this value is striking for the two cases of “optimization with objective function modification” and “optimization with single objective function”. In the case of “optimization with objective function modification” a step-by-step decrease to much more lower values than those of the “optimization with single objective function” is observed.

Figure 4.5 shows the improvement in the case of a feasible trajectory found with “optimization with single objective function”. Similar arguments made above go for these figures, too. The only difference is that the result of “optimization with single objective function” is feasible, in the sense that the trajectory both reaches to the desired final position and follows a reasonable path. However, the result of “optimization with objective function modification” is both feasible and much more efficient compared to the former.

The “optimization with single objective function” suffers from the local optimums, which mostly lead to unfeasible trajectories. With the “optimization with objective function modification” it is possible to jump over these local optimums and generate more feasible and efficient trajectories. With the six steps and the associated cost parameters mentioned in Table 4.2, the “optimization with objective function modification” was successful to generate feasible and more efficient trajectories for the *79* cases of the *81* optimizations. These results, however, are sure not the global optimums, since the optimization is still based on the gradient information. A problem related to “optimization with objective function modification” is the long time required for the termination. As can be seen in Figures 4.4 and 4.5 the “optimization with objective function modification” necessitates around 200 iterations, which is quite much for a gradient-based optimization. However, the improvement is considerable and leads to satisfactory results to be used in controller design. The long time of termination did not create a problem for this work, since the optimization is performed only to create a training set to be used in the design of protraction controller.

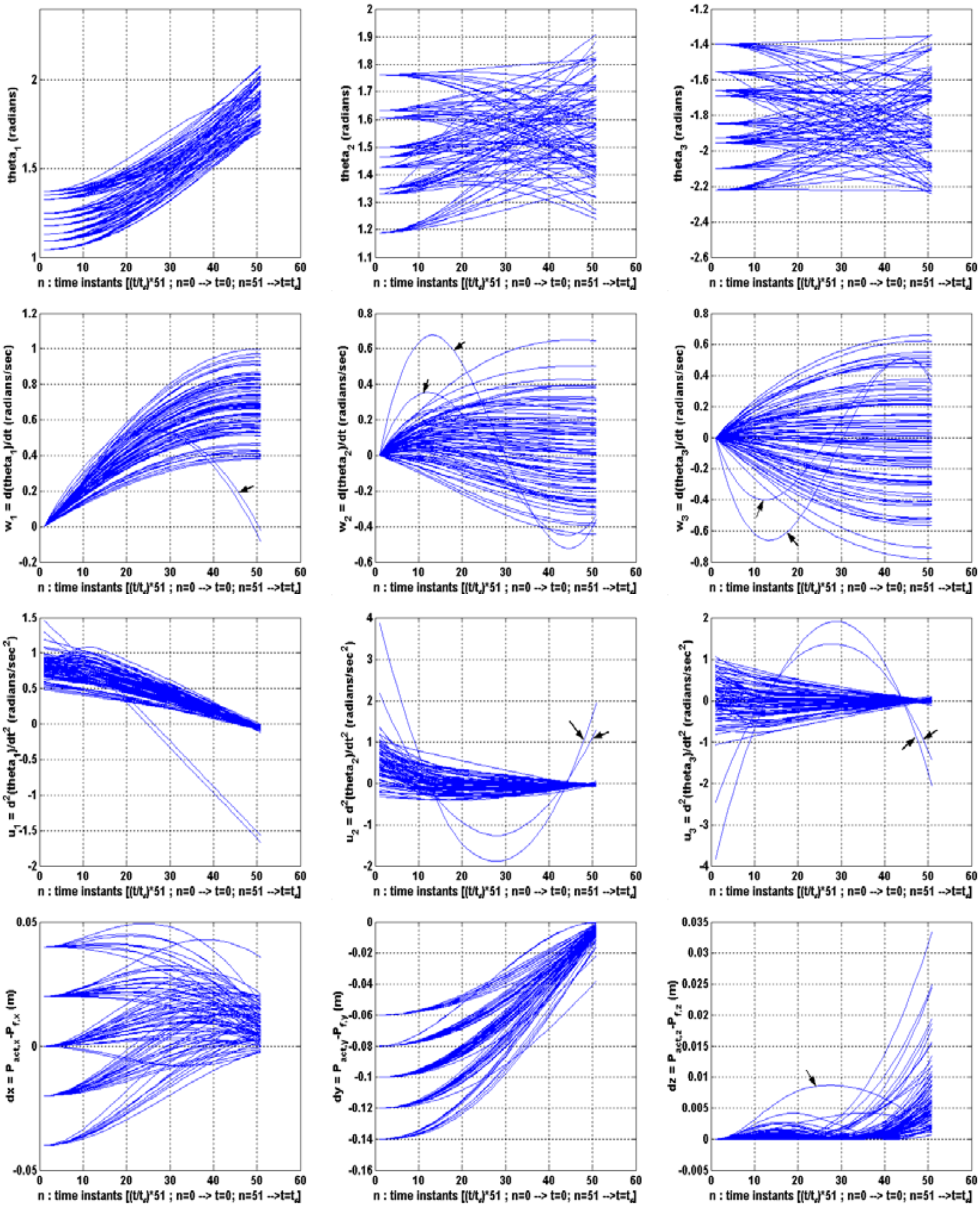


Figure 4.2: Results of "optimization with single objective function". First row: joint angles; second row: joint angle velocities; third row: joint angle accelerations; fourth row: tip point position errors.

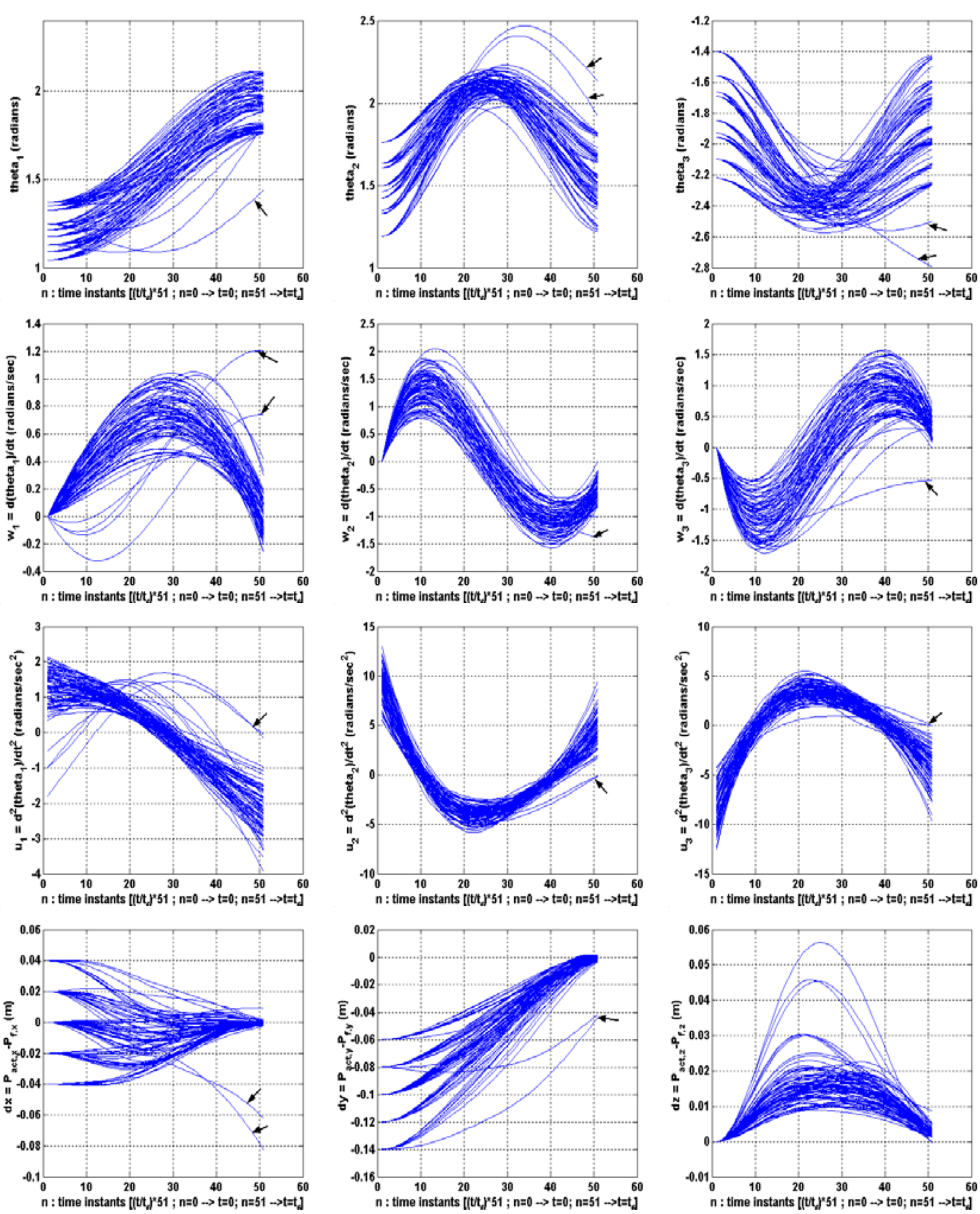


Figure 4.3: Results of "optimization with objective function modification". First row: joint angles; second row: joint angle velocities; third row: joint angle accelerations; fourth row: tip point position errors.

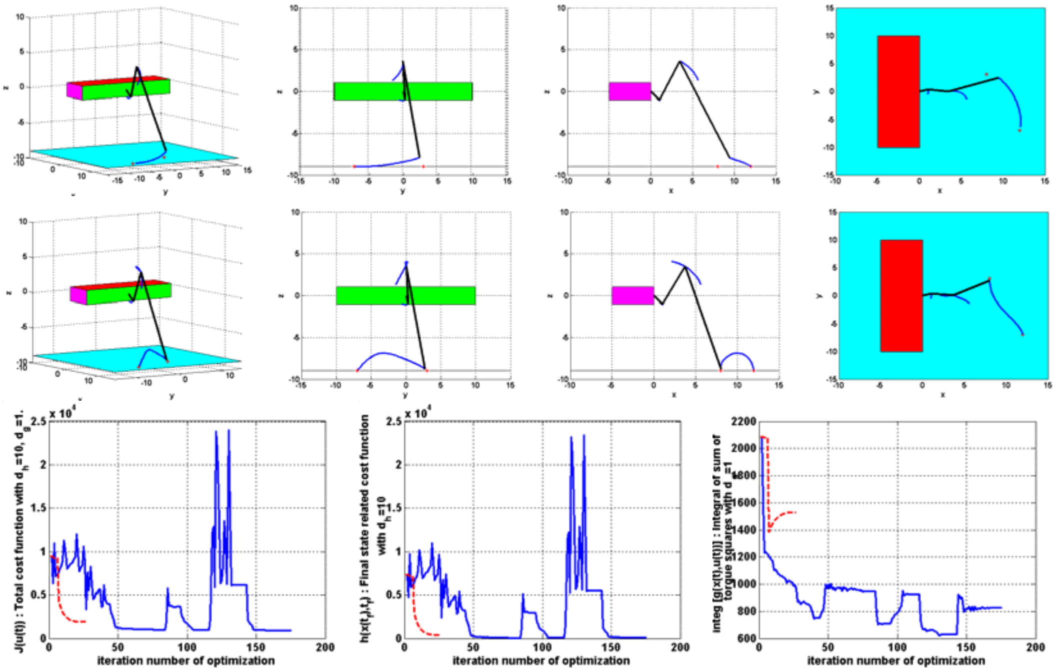


Figure 4.4: Results of "optimization with single objective function" (the first row and the dashed lines in the last row) and "optimization with objective function modification" (the second row and solid lines in the last row). The result of "optimization with single objective function" is infeasible. (Unit of first two row figures: cm; initial tip point position: [12, -7, -9] (cm); final tip point position: [8, 3, -9] (cm).)

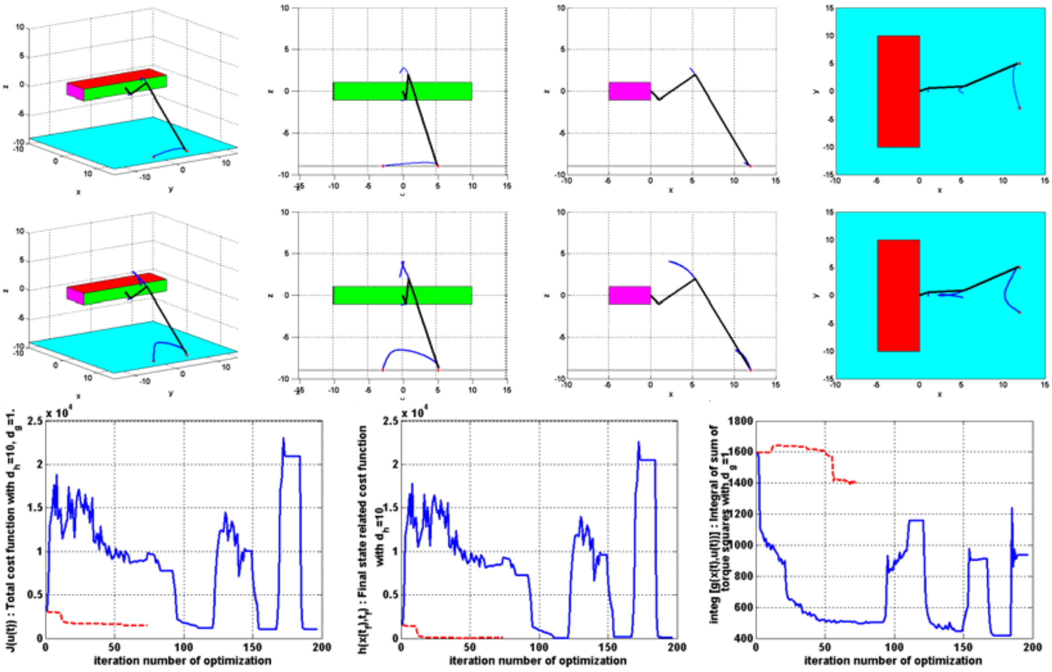


Figure 4.5: Results of "optimization with single objective function" (the first row and the dashed lines in the last row) and "optimization with objective function modification" (the second row and solid lines in the last row). The result of "optimization with single objective function" is feasible. (Unit of first two row figures: cm; initial tip point position: [12, -3, -9] (cm); final tip point position: [12, 5, -9] (cm).)

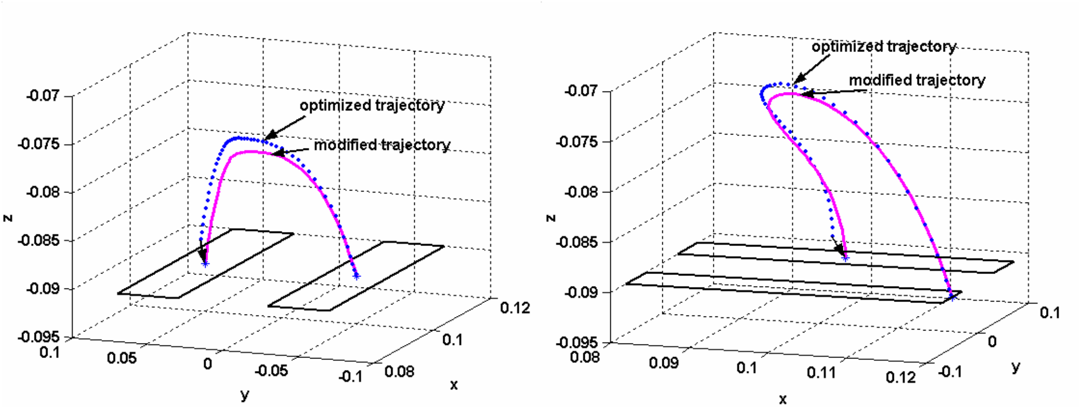


Figure 4.6: Effect of the modification on optimized trajectories. (unit: m)

4.4.1. Manipulation of Optimal Trajectories:

The optimization is capable of decreasing the torque-square-sum with approaching the tip point to the final position; however, this approaching does not result in exact placement of the tip point to the desired final position. Most of the time slight, but sometimes considerable, position differences occur between the actual and desired final tip point positions. For the interpolation purposes, this is an improper situation regarding to the data set. In order to overcome this final position deviation, the optimized trajectories are modified to bring the final tip-points to the desired positions. This is performed by adding the difference vector between the actual and desired final tip-point positions to all position vectors, multiplied with a weight decreasing towards the start of the trajectory, as in (4.33). This manipulation effects the most successful optimizations in a negligible amount. However, the less successful optimizations, in which the final point position is not close to the desired position, are manipulated in a considerable way. The effect of optimal trajectory manipulation is shown in Figure 4.6 for two sample initial-final position pairs.

$$\bar{P}_{mod}(t_k)=\bar{P}_{op}(t_k)+(\bar{P}_f-\bar{P}_{mod}(t_{51}))\frac{k}{51} \qquad k=0,1,2,\ldots,51. \qquad (4.33)$$

4.4.2. RBFNN for Interpolation:

In order to utilize the optimal trajectories for control of protraction, it is necessary to construct an interpolation algorithm, which will generate the trajectory for any given initial and final tip-point positions. For this purpose a RBFNN with pure-linear output neurons is a proper choice (Figure 4.7). The inputs of this interpolating network are the coordinates of the initial and final tip point positions; that makes up *6* input variables. The outputs are the tip point positions corresponding to the *51* discretized instances throughout the protraction period, which makes up a total of *153* output variables. The inner neurons in an RBFNN correspond to the representatives of clusters of the input space. The weights between the input and inner neurons determine the location of the representatives in the input space.

Given any input, its distance to the representatives are calculated, and the inner neurons are activated with an inverse proportion to this distance. Namely, the representative, which is closest to the input is activated most. The weights between the inner and output neurons correspond to the output vectors associated with the representative neurons. Since these weights are multiplied with the activation of inner neurons, the output is a weighted sum of the effect of all representatives. Training of the RBFNN according to the optimized trajectories is a matter of generating the inner neurons, namely the cluster representatives, and the weights corresponding to those. An approach to this problem is to introduce an inner neuron, and then to optimize all the weights according to the data set, in each iteration. The center of the introduced neuron is chosen to be the data to which the RBFNN fits worst at that instant. Then the weights between all the inner neurons and output layer are optimized to minimize the least square error of all the data set. The number of neurons to be generated is a matter of design preference, in which data storage capabilities should be considered. In this work the function, *newrb()*, in the neural network toolbox of MATLAB is utilized for performing the described procedure with *30* inner neurons. The resultant RBFNN is used as the guiding trajectory generator for the controller introduced in the next section. The output values of the RBFNN are given in (4.34). Two trajectories produced by the interpolation of the RBFNN for two sample initial-final position points from the training set are shown in Figure 4.8.

$$[output]_j = \sum_{n=1}^{30} w_{nj} a_n = \sum_{n=1}^{30} w_{nj}\, gaussian\left(\sum_{i=1}^{6} \left|[input]_i - c_{in}\right|\right) \tag{4.34}$$

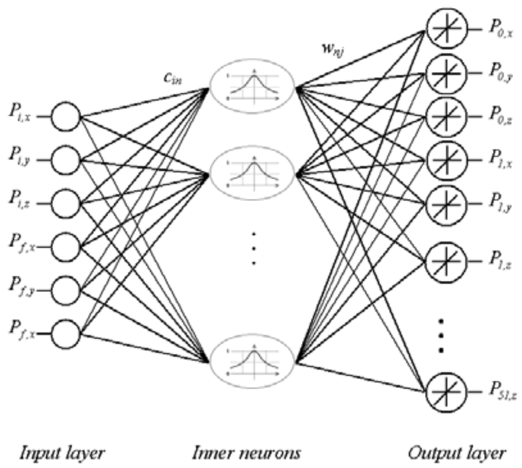


Figure 4.7: Trajectory interpolating RBFNN. i=1,2..,6 ; n=1,2,..,30 ; j=1,2,...,153. The vector c_n=[c_{1n} c_{2n} ... c_{6n}] determine the center of the nth neuron; a_n=gaussian(|[input]-c_n|) determines the activation of the nth neuron; the vector w_n =[w_{n1} w_{n2} ... w_{n3}] determine the effect of the nth neuron on the output.

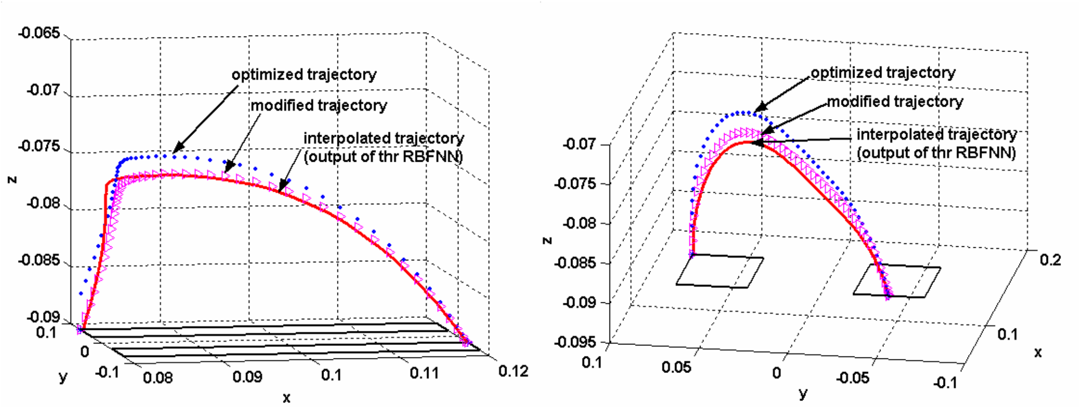


Figure 4.8: The trajectories produced as a result of the interpolation with the RBFNN for two trained input of initial-final tip point positions. (unit: m)

For convenience, here two more results are given in order to present the performance of the interpolating RBFNN for two untrained data. The initial-final tip point positions, the optimized trajectories, and the output of the interpolating RBFNN are depicted in Figure 4.9 for two untrained cases.

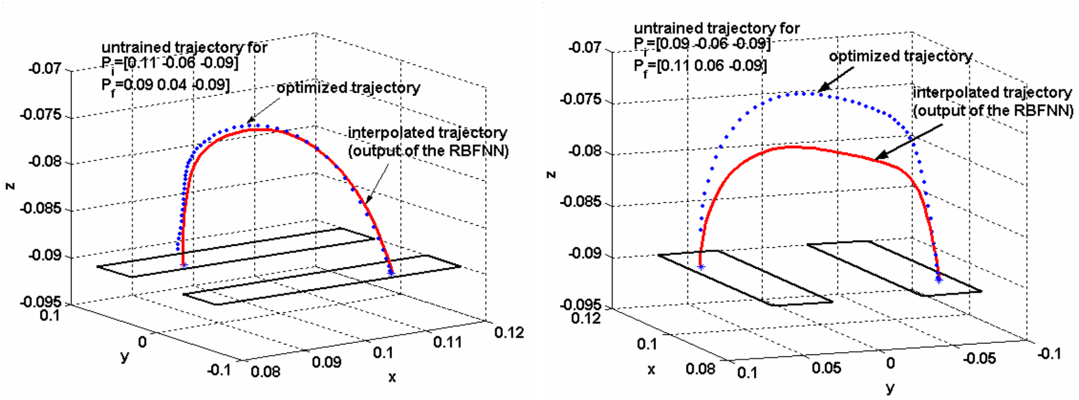


Figure 4.9: The trajectories produced as a result of the interpolation with the RBFNN for two untrained input of initial-final tip point positions. (unit: m)

4.5. Controller Design

The controller design problem for the protraction of a three-joint robot leg is formulated as follows: “to generate the joint angle accelerations in order to bring the tip point of the leg from any given initial position to any given final position following a near-optimal path, with handling of disturbances and deviations, and with minimum oscillations”. The near-optimal path will be generated using the RBFNN designed in the previous section. This path will be the guiding path for the controller. Then the aim is to make the leg follow a guiding trajectory in the required time interval, with stability at the final point. In order to achieve this, first it is necessary to design a controller that will bring the tip point from any given initial position to any given final position. Afterwards, this controller can be incorporated with the guiding trajectory.

Fuzzy logic based controller design technique is used to control the joint accelerations. The major problem in the controller design is to achieve fast response and stability at the same time. The protraction movement has to be accomplished in a given time (taken *1.5* seconds here) and the tip point must remain stationary without any oscillation at the final tip point position. The trade-off between fast response and oscillation is easier to be handled with fuzzy controllers due to the flexibility of rules and their weights.

In Erden et al. (2004) the idea of “multi agent based fuzzy controller design for robot manipulators” was already developed. The idea there was to design a fuzzy controller for each joint, in a way that each joint would act independently, and their total sum of actions would result in the desired system response. The controllers in that work were designed considering the infinitesimal tip point position change due to an infinitesimal change in the joint angles. The inputs to the controllers were directly the tip point position errors. In this work similar ideas are followed with a more sophisticated model. A single controller is designed to be used by all the joints independent of the others. Rather than taking the tip point position error as a direct input, this error is transformed to an angle error for each joint. Namely, a “pseudo joint angle error”, which signifies the share of the joint in the tip point position error, is defined for every single joint. The pseudo joint angle errors and angular velocities of the joints are input to the fuzzy controllers. The output of the controllers are the angular accelerations of the joints. These ideas will be described for a general manipulator in the following.

In Figure 4.10 a graphical representation of the *jth* joint of any multi-link revolute joint manipulator is given. In this figure the actual and desired tip point position vectors are depicted by $\vec{P}_{tip}$ and $\vec{P}_d$, respectively. The aim is to generate the *Pseudo_theta_error*, which corresponds to the share of the *jth* joint in the position error of the tip point. The contribution of the *jth* joint to carry the tip point to the desired position can only be with a rotation of the *jth* joint, around its direction of rotation. This direction is represented by $\vec{u}_3^{(j-1)}$ in the figure. The amount of rotation of the *jth* joint can be represented as an angle in the plane perpendicular to the direction of rotation, $\vec{u}_3^{(j-1)}$. This plane is shown as the big rectangle in Figure 4.10. In order to find the necessary rotation of the *jth* joint, it is necessary to obtain the projections of the actual and desired tip point positions on the plane perpendicular to the direction of rotation. Then the vectors between the origin of the *(j-1)st* frame and these projection points can be determined as $\vec{v}_{tip}$ and $\vec{v}_d$, respectively, as in the figure. Once $\vec{v}_{tip}$ and $\vec{v}_d$ are determined, the angle between these two vectors can easily be found using the inverse cosine formula. The actual pseudo joint angle error (*Pseudo_error*) related to the *jth* joint is generated by multiplying the *Pseudo_theta_error* with the length of the projection vector of the actual tip point position. The change in the tip point position error due to the rotation of the *jth* joint will be proportional to the magnitude of $\vec{v}_{tip}$. The vector equations and formulas related to those calculations are given in (4.35). The values of the vectors of $\vec{P}_{tip}$ and $\vec{r}_{0,j-1}$ with respect to the zeroth reference frame can be found in Appendix C.

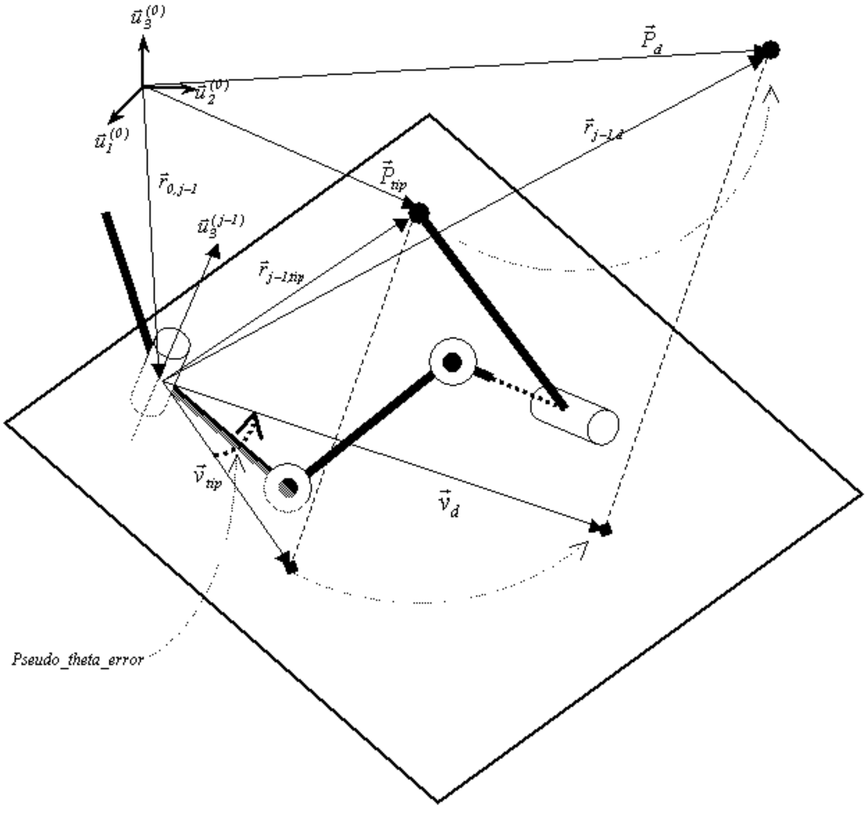


Figure 4.10: Representation of pseudo joint angle error for joint *j* of a multi-link revolute manipulator.

$$
\begin{aligned}
\vec{r}_{j-1,tip} &= \vec{P}_{tip} - \vec{r}_{0,j-1} \\
\vec{r}_{j-1,d} &= \vec{P}_{d} - \vec{r}_{0,j-1} \\
\vec{v}_{tip} &= \vec{r}_{j-1,tip} - \frac{(\vec{u}_3^{j-1} \cdot \vec{r}_{j-1,tip}) \cdot \vec{u}_3^{j-1}}{\|\vec{u}_3^{j-1}\|^2} \\
\vec{v}_{d} &= \vec{r}_{j-1,d} - \frac{(\vec{u}_3^{j-1} \cdot \vec{r}_{j-1,d}) \cdot \vec{u}_3^{j-1}}{\|\vec{u}_3^{j-1}\|^2} \\
Pseudo\ theta\ error &= acos\left(\frac{(\vec{v}_{tip} \cdot \vec{v}_d)}{|\vec{v}_{tip}| \cdot |\vec{v}_d|}\right) \\
Pseudo\ error &= (Pseudo\ theta\ error)|\vec{v}_{tip}|
\end{aligned}
\tag{4.35}
$$

A fuzzy controller is designed to be used in the three joints. The inputs of this controller are the *Pseudo_error* and joint angle velocity; the output is the joint angle acceleration. The flexibility of fuzzy rules gives way to a very effective design to overcome the trade off between fast response and oscillations. Two groups of rules are written to handle the decreasing of error, and repression of oscillations, respectively. The first group of rules depend on the magnitude of *Pseudo_error*: If *Pseudo_error* is large, the output acceleration is big in the direction to decrease the error, otherwise it is small. The method for rule aggregation is chosen to be "sum", rather than the more commonly used "max" option. This is because, using the "sum" aggregation, it is possible to easily arrange the output to any value between the tip points of output membership functions. For this purpose, two similar rules can be written with the same premises but different consequences, which results in a rule with a weighted average consequence. This technique is followed in the first group of rules. The second group of rules depend on the *Pseudo_error* and joint angle velocity: In the fuzzy regions that *Pseudo_error* is zero, these stabilization rules contribute to the joint acceleration in the opposite direction of joint velocity. The details of the fuzzy rules can be followed in the MATLAB fuzzy system file in Appendix D.

The guiding trajectory gives the coordinates of *51* instances throughout the protraction. The actual tip point position in any instant is closest to one of these *51* positions. Once this closest position is found the reference tip point position can be generated by following the guiding trajectory, and taking the position which is some time ahead of the closest position. When the actual position comes near to the desired final position (nearer than the second entry from the last instant of the guiding trajectory), the desired final tip point can be generated as the reference tip point position. In this way, the reference tip point positions are iterated through the guiding trajectory till the desired final position. The controller follows this guiding. The overall block diagram of the control system for the protraction is given in Figure 4.11. In this diagram the generation of reference position from the guiding trajectory, and generation of the pseudo error signal are performed in the "Pseudo Error Generator" box.

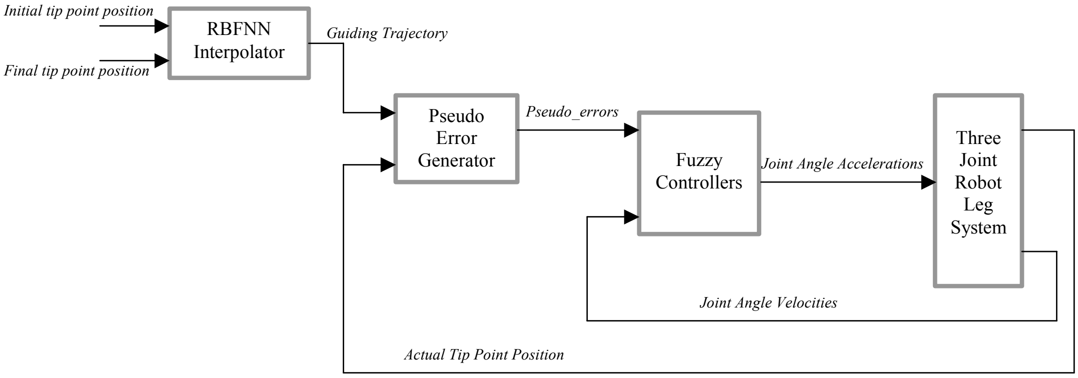


Figure 4.11: Block diagram of the overall control system of protraction.

In Figure 4.12 the SIMULINK block diagram is given to illustrate the realization of the overall protraction control. The association with the block in Figure 4.11 can easily be observed. The system in Figure 4.11 gives the very basic structure of the control system. Adopting this structure to the actual control case means arranging some parameters in order to maintain desired behavior for the specific cases. This adaptation procedure corresponds to arranging the gain parameters in Figure 4.12. The arrangement of these parameters is performed by trial and error; however, it is still possible to give the basic logic followed in tuning. Since the controllers take normalized inputs and provide normalized outputs, first the velocity and acceleration gains are arranged. The acceleration gains (*Gains 7, 8, 9*) are arranged to a high value, and the velocity gains (*Gains 1, 3, 5*) are divided by the same high value. This value is one that produces a reasonably fast output on which work can be performed. Then the pseudo_error gains (*Gains 2, 4, 6*) are arranged in order to have a close trace of the guiding trajectory. Since the responses of the second and third joints are faster than the first joint (Figure 4.3, acceleration figures) it necessitates to have higher gains for the pseudo_errors of the second and third joints with respect to the first one. A ratio of 1:10, with values 0.5:5, was proper to have a close tracing of the guiding trajectory. Lastly, the velocity and acceleration gains are tuned to arrange the velocity properly for changing initial-final position pairs. The resultant values for the velocity and acceleration gains are as in (4.36). It might be realized that increasing μ in these equations corresponds to decreasing the velocity gains that are input to the controllers, hence increasing the actual accelerations and velocities of the system and speeding up the system response.

$$\begin{gathered} dP = |\bar{P}_i - \bar{P}_f| \\ \mu = 400 + 160\frac{(dP - 0.08)}{0.04} \\ acceleration\ gains: [Gains\ 7,8,9] = \mu \\ velocity\ gains: [Gains\ 1,3,5] = \frac{1}{\mu} \end{gathered} \tag{4.36}$$

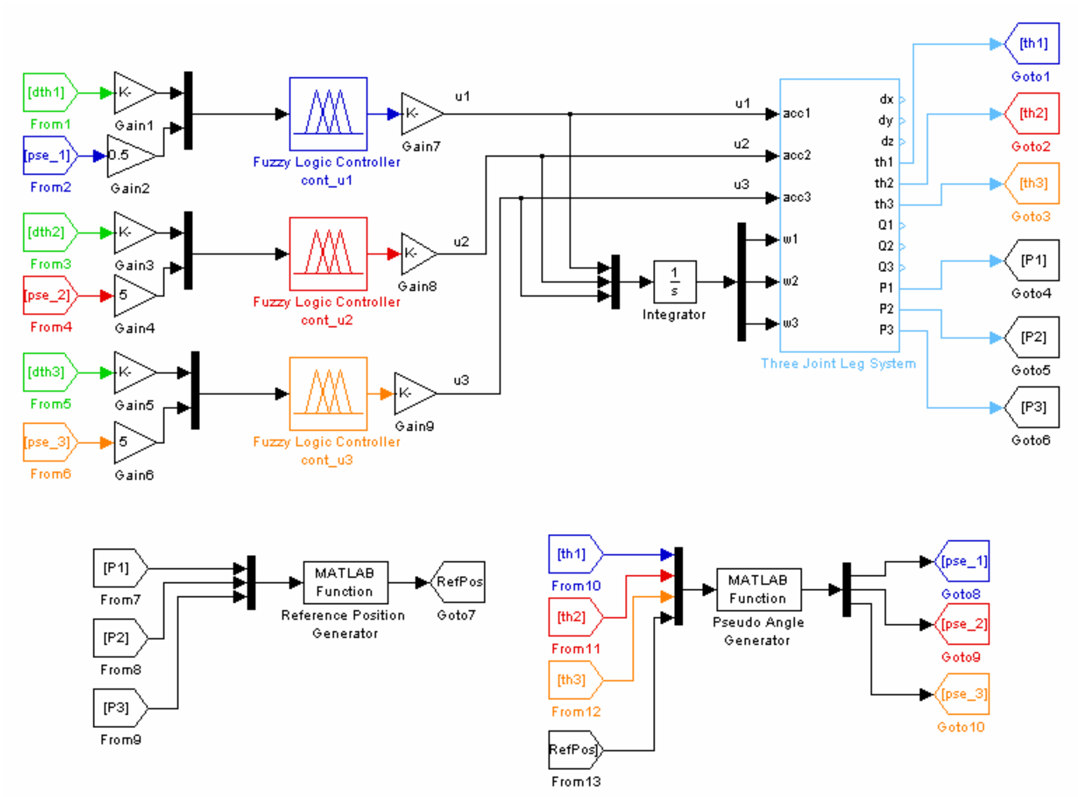


Figure 4.12: SIMULINK block diagram illustrating the realization of overall protraction control.

4.6. Results

In order to test the performance of the overall controller, initial-final tip point position pairs, which do not exist in the training set are used. The optimization algorithm is run with the input of these pairs, and the output trajectories of the actual controller are compared with the optimized ones. Here will be given the results for two such cases.

In Figure 4.13, the optimal and controller output results for the initial and final tip point positions of [*0.11, -0.04, -0.09*] and [*0.09, 0.04, -0.09*] are given. The interpolated trajectory in this case is quite close to the optimal one. The tip point trajectory created by the controller is very close to the interpolated trajectory, which is used as the guiding trajectory throughout the control. In the joint angle figures it is apparent that the resulting trajectories are very close to the optimal ones in the position level. In the velocity and acceleration levels, the optimal behavior is closely imitated by the controller.

In Figure 4.14, the results for the initial and final tip point positions of [*0.09, -0.06, -0.09*] and [*0.11, 0.06, -0.09*] are given. In this case the interpolated trajectory, namely the guiding trajectory is not very close to the optimal one but the structure is similar. This can be observed in the tip point trajectory and joint velocity graphs. The controller is again successful to follow the reference trajectory quite closely.

In some cases the interpolation might result trajectories comparatively far from the optimal ones (even from the manipulated forms of the optimized trajectories), as in the case of Figure 4.14. This is a matter related to the interpolation structure, which is not the main focus here. However, it can be stated that interpolations closer to the optimal trajectories is always possible by increasing the dimension of the interpolating network, with the cost of more memory storage.

In Figure 4.15, three slights are given to show the Robot-EA308 while the right-middle leg is in protraction, following the path generated by the controller. The left most figure shows how the leg is pulled towards the body while the tip point is raised up. In the middle and right-most figures the whole protraction can be seen from different perspectives. These results, and also the joint angle trajectories in Figures 4.13 and 4.14, reveal that the energy optimal protraction has the following character: The leg is pulled towards the body while the tip point is raised up; then, it is stretched out while the tip point is going down to approach to the destination. This behavior is quite different from the conventionally adapted protraction movements in which the leg is first stretched out while raising the tip point, and then pulled towards the body while the tip point is approached down to the destination.

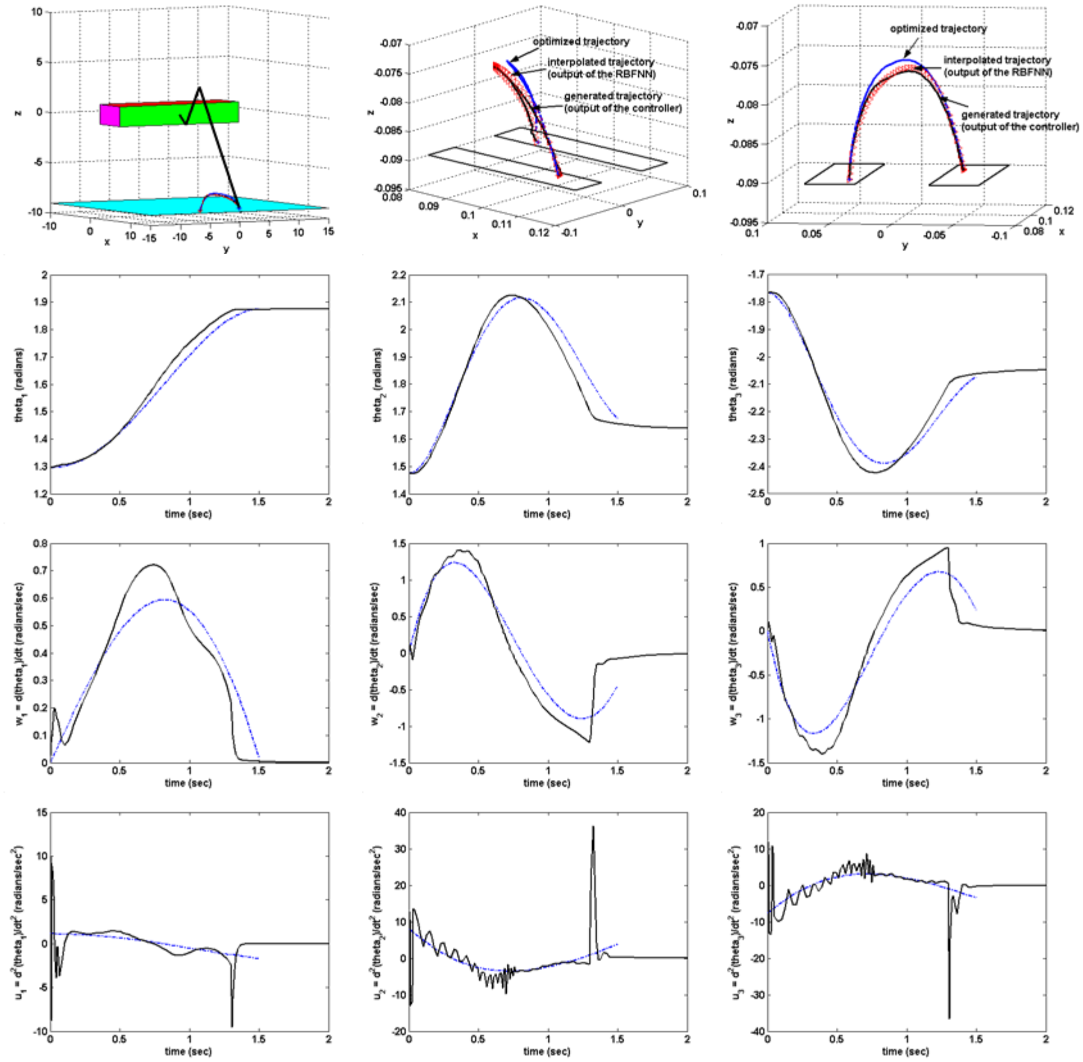


Figure 4.13: First example for comparison of the trajectory results generated by the protraction controller with the optimized results. The initial and final tip point positions are [0.11, -0.04, -0.09] and [0.09, 0.04, -0.09], respectively. Dotted lines are for the optimized results, solid lines are for the generated results by the controller.

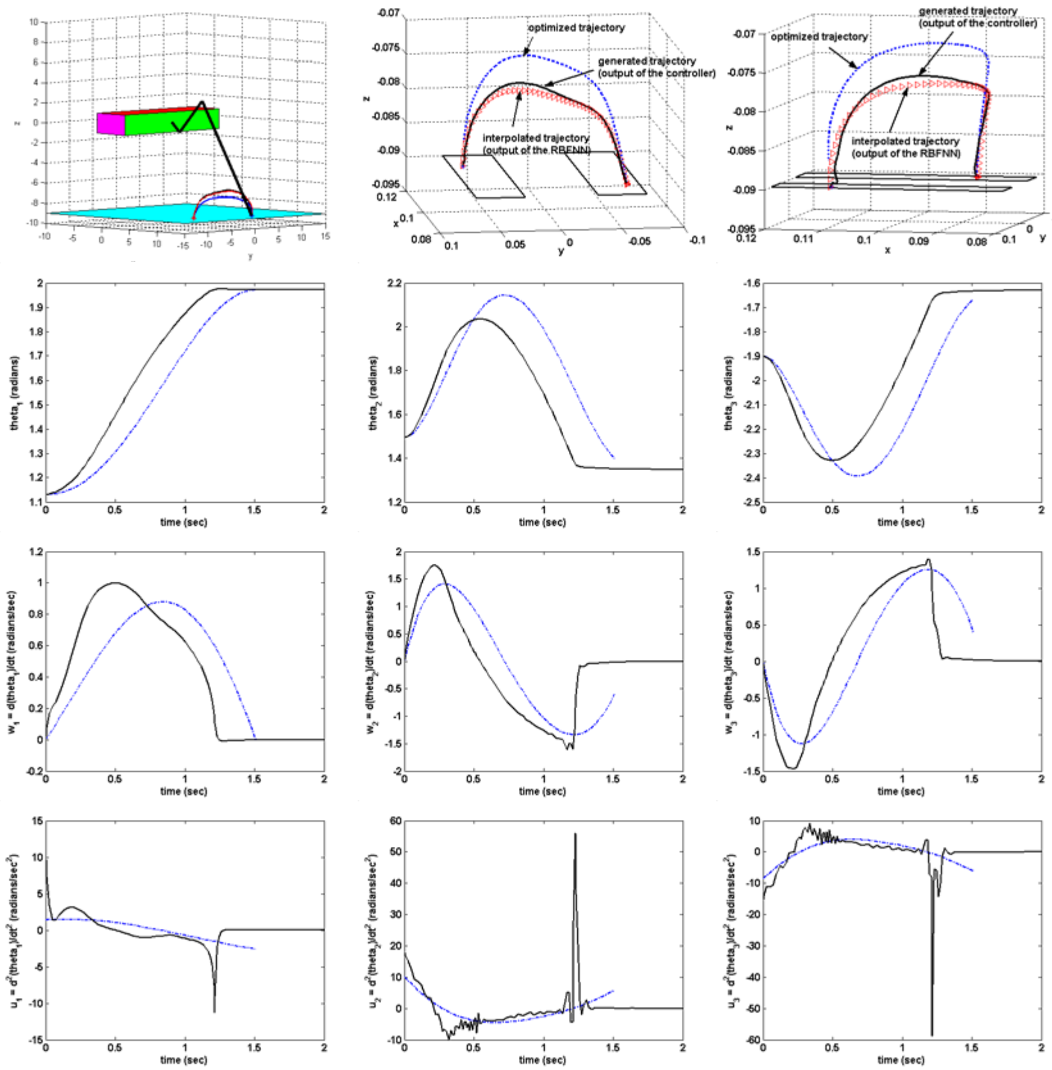


Figure 4.14: Second example for comparison of the trajectory results generated by the protraction controller with the optimized results. The initial and final tip point positions are [0.09, -0.06, -0.09] and [0.11, 0.06, -0.09], respectively. Dotted lines are for the optimized results, solid lines are for the generated results by the controller.

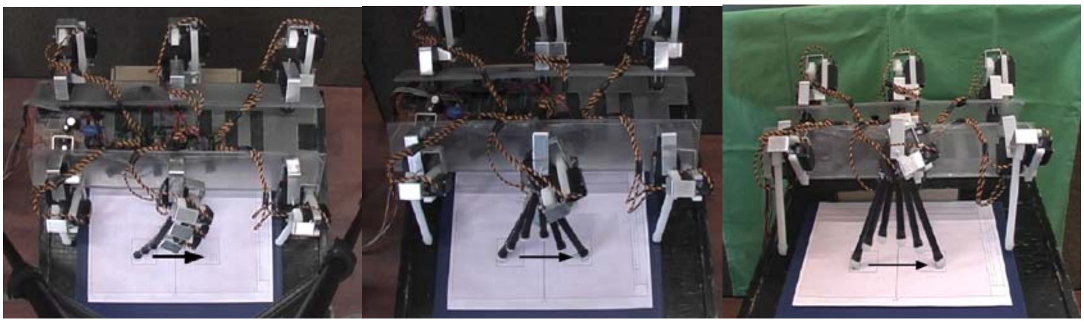


Figure 4.15: Protraction of the right middle leg of robot EA308, following the trajectory generated by the controller. The initial and final tip point positions are [0.11, -0.06, -0.09] and [0.11, 0.06, -0.09], respectively.

4.7. Conclusion

In this work the protraction movement of a three-joint robot leg is handled. Considering the literature on multi legged walking, protraction of legs is a very little worked subject. In multi-legged system applications, mostly, some limited protraction behaviors are adopted based on intuitive feeling of physical behavior. Those applications limit the freedom of leg placements since the system can use only the preplanned structures. Moreover, mostly, the adapted behaviors are not optimized considering energy consumption. The work here presents two crucial applications to overcome these limitations: The optimization of protraction creates energy optimal trajectories for any given initial-final tip-point position pairs. The controller for protraction generates near-optimal trajectories based on the generalization of *79* optimal trajectories. With this generalization the system does not need to be limited to some specific protraction behaviors, but it can handle any protraction once the initial-final tip-point positions are given in the range of the design. The results of this work show that an energy optimal protraction is characterized by pulling the leg towards the body, rather than stretching out, in the rising phase of the movement.

The optimization is based on the gradient descent algorithm, with the Hamiltonian formulation for optimal control problems. The conventional application suffers from sticking to unfeasible and inefficient local optimums. To overcome this, the approach of "optimization with objective function modification" is introduced. In this application the objective function is changed in steps once the optimization stops with the previous step. The last step makes optimization with the nominal objective function. The results reveal that this method is successful to overcome both feasible and unfeasible local optimum solutions of conventional "optimization with single objective function". The obtained trajectories are efficient and feasible. The interpolation of optimal trajectories are used as guiding trajectories for the protraction controller.

The controller design is based on the idea of controlling each joint independently of the others. A similar idea was introduced in Erden et al. (2004) as "multi agent system based control of robot manipulators". For the independent control of a joint a *pseudo_error* is produced, which corresponds to the share of the joint in the tip point position error. The joint velocity and the *pseudo_error* are input to the fuzzy controller of each joint. The controller commands the joint angle accelerations to the three-joint leg system. The overall system response is generated together by the all three resultant actions. The significance of this novel approach in the controller design is its simplicity of application and its requirement of no inverse kinematic solutions throughout the movement. The protraction controller designed with this approach is successful to follow any given guiding trajectory with closeness sufficient for protraction movements.

For the optimization algorithm, a considerable drawback is the long duration for termination. This is because a full optimization is performed for all the objective functions described separately by the steps. This was not a problem for the work presented here, because the optimization is performed once and the recorded results are used in generalizations. However, it may be a significant problem in some applications in which fast responses of optimization are required. The improvement of the idea of “optimization with modified objective function”, in the direction of speeding up the algorithm, remains as a future work. A crucial attempt may be to modify the objective function not in steps of sequential optimizations, but continuously throughout a single optimization. This might be performed by some intelligent and continuous modification of the objective function based on the values of a nominal objective function and its gradient. Considering the control structure developed here, in some cases the guiding trajectories generated by the RBFNN happen to be somewhat different from the optimized trajectories. As mentioned before, this is a matter of the interpolation algorithm. It is possible to improve the interpolation in a way to make all the trajectories very close to the optimized ones used for training. However, whatever is done, the trade off between better performance and high memory storage will persist. Despite this trade off, the performance of interpolation can always be improved by adopting more successful structures.

CHAPTER 5

"TORQUE DISTRIBUTION" IN A SIX-LEGGED ROBOT SYSTEM: KINEMATIC-DYNAMIC FORMULATION AND ENERGY OPTIMAL DISTRIBUTION

5.1. Introduction

An important drawback of legged systems is the high-energy consumption due to the active suspension of the body by the supporting legs. Most of the energy is consumed in the retraction (repelling back while supporting in power stroke) phases of the steps. The trajectories that the supporting legs follow throughout retraction are definite once the motion of the robot body is given. However, the torques to be applied on their joints in order to achieve this determined motion is not definite. This chapter deals with the energy efficient "torque distribution" problem in multi legged systems. In contrast to the general literature, the distribution of required forces and moments is not handled as a tip point "force distribution". It is demonstrated that the proposed "torque distribution" scheme results in a much more energy efficient distribution than the conventional "force distribution"; because, it makes use of the internal forces and friction to minimize the required joint-torques. In this comparison, as in many robotic applications (Bobrow et al., 2001; Garg and Kumar, 2002; Liu and Abdel-Malek, 2000; Marhefka and Orin, 1998), the integral of the squares of the joint-torques is used as an index of energy dissipation. In the chapter a detailed mathematical formulation of the dynamics of the six-legged robot with the primary variables being the joint-torques is given. A compact matrix representation for handling the "torque distribution" problem is formulated.

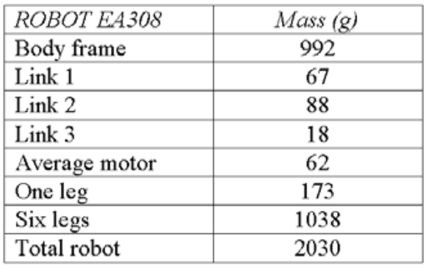
Since the topology of a multi-legged robot is continuously altered, the dynamic analysis tools developed for serial manipulators (Fu et al., 1987) and parallel manipulators (Ji and Wu, 2003; Beji and Pascal, 1999; Carvalho and Ceccarelli, 2001; Liu et al., 2003) are not directly applicable. Therefore, the dynamic equations here are derived using the approach of "free body diagrams" (Barreto et al., 1998; Lin and Song, 2001), which is based on relating the force-moment equations of rigid pieces to each other. Systematic applications of this approach results in the recursive formulations of the dynamics of multi-linked vehicles (Cheng and Orin, 1991; Hung et al., 2000). In Cheng and Orin (1991), a computationally efficient formulation of the closed-chain mechanisms is presented for the force distribution problem. Hung et al. (2000) presents a recursive and computationally efficient algorithm for general tree-structured mechanisms. The effects of the inertial-gravitational forces and contact forces on the links are decoupled, and later they are combined by using the superposition principle. Both Cheng and Orin (1991) and Hung et al. (2000) use the contact-friction forces, rather than the joint-torques, as the primary variables. If the robot is joint-torque controlled, the results of the distribution problem formulated by any of the two has to be converted into the torque space, so that a computation burden will be introduced. In the case that an energy-related optimization is performed, the objective function has to have a quadratic term of joint-torques (Marhefka and Orin, 1998). Then, if the primary variables of formulation are not the joint-torques but tip-point forces, some linear terms will appear and this will also introduce a computational burden. In this chapter, the formulation of dynamics of the six-legged robot uses the joint-torques as the primary variables; therefore, the results of the distribution do not need any conversion for a torque-controlled system, and the objective function is pure quadratic.

The general optimization problem for force-moment distribution requires minimization of an objective function with respect to linear equality and inequality constraints. This is the general form the problem is reduced whether the matter is force distribution as in Klein and Kittivatcharapong (1990), Cheng and Orin (1990), and Nahon and Angeles (1992) or torque distribution as formulated here. Cheng and Orin (1990) formulates the force distribution problem as a linear-programming problem. Klein and Kittivatcharapong (1990) formulates the objective function as the pure sum of the squares of the tip point force components and concludes that the linear programming approach leads to “discontinuities” in the solution. Nahon and Angeles (1992) formulates the force distribution problem as a quadratic programming problem. The paper concludes that the quadratic programming solution leads to continuous and faster solutions compared to linear programming formulations. Following the discussions in these three papers, this chapter formulates the torque distribution as a quadratic programming problem.

Minimizing the sum of the squares of the components of the tip point forces in the multi-legged robot studies seems to be an extension of the application of force distribution in robot hand studies (Cheng and Orin, 1990; Nahon and Angeles, 1992; Kumar and Waldron, 1998). Kumar and Waldron (1998) introduces the decomposition of contact forces into “equilibrating forces” and “interaction forces”. Accordingly, in Galvez et al. (2003) the tip-point forces are classified as the ones that “contribute to the robot's support and drive without causing the feet to fight each other”, and “the ones that cause the feet to fight each other”, respectively. Following this idea, the distribution problem is formulated in order to get rid of the interaction forces. The problem then turns out to be minimizing the sum of the squares of the tip point force components. This approach is applied in many other studies on multi-legged systems (Lin and Song, 2001; Klein and Kittivatcharapong, 1990; Preumont et al., 1997; Martins-Filho and Prajoux, 2000; Zhao et al., 2000). In these works either the direct pseudo-inverse of the matrix related to the equality constraints (Lin and Song, 2001; Preumont et al., 1997) or the pseudo-inverse of the enlarged form of that matrix with linearized active constraints (Martins-Filho and Prajoux, 2000) are used, or the direct pseudo-inverse solution of that equality matrix is used to initialize the optimization of the forces in the space determined by the inequality constraints (Klein and Kittivatcharapong, 1990; Galvez et al., 2003). In fact, the pseudo-inverse solution of the force equality constraints is indeed a practical and effective solution for those cases, whether it is used directly or as a starting guess. The reason is that, the resulting forces are minimal; therefore, the effect of inequality constraints related to friction are either eliminated or reduced to a minimum. This is the underlying reason that the pseudo-inverse solution for force distribution works so effectively in practice, hence receives such attention.

However, as it is demonstrated in Kar et al. (2001), Jiang et al. (2004), and Marhefka and Orin (1998), behind this computational efficiency, underlies energy inefficiency due to formulating the problem as minimization of the tip point force components. In Kar et al. (2000), the objective function is formulated as the sum of the multiplication of joint-torques with joint velocities. In Jiang et al. (2004), one of the three objective functions, formalized in an approximate and simplified way to achieve fast solutions, aims to minimize the energy consumption during locomotion. The results show that this simplified formulation performs better than the foot-force formulation for the objective it represents. In both of these papers, the formulation for minimization of energy dissipation does not consider the fact that there is still power dissipation on the motors (due to gravitation) even though the joints are not moving. In Marhefka and Orin (1998) the power consumption is calculated based on a DC motor model in terms of the joint-torques. It is concluded that “minimization of a weighted norm of joint-torques” produces good results considering real power dissipation; however “minimization of internal forces” may often produce poor power performance. The results of this paper are important for the justification of formulating the objective as “minimizing the sum of the squares of the joint-torques” for minimum energy consumption. In all of these three papers – in fact in most of the other literature that deal with “force distribution” (Klein and Kittivatcharapong, 1990; Preumont et al., 1997; Martins-Filho and Prajoux, 2000; Galvez et al., 2001; Pfeiffer et al., 2000) – either the inertial effects or totally the masses of the legs are ignored; hence, the tip point forces are related to the joint-torques via a simple Jacobian matrix multiplication. In Table 5.1, the masses of the components of the Robot-EA308 are given. The total mass of the legs is more than the mass of the body frame. Therefore, the assumption of negligible leg masses and inertia is not valid in practical applications. Interestingly, and to our knowledge, the work here seems to be unique to make use of the full dynamics of the legs in solving the “distribution” problem.

Table 5.1: The masses of the components of the Robot EA308.



ROBOT EA308	*Mass (g)*
Body frame	992
Link 1	67
Link 2	88
Link 3	18
Average motor	62
One leg	173
Six legs	1038
Total robot	2030

In Galvez et al. (2001) and Pfeiffer et al. (2000), the force-moment distribution problem is formulated as torque distribution for the two-joint eight-legged tube crawling robot. The optimization problem is formulated to minimize the maximum of the joint-torques. Although this is a torque distribution approach, it differs from that of this chapter: while the objective function for the tube crawling robot is "oriented to avoid overload on the motors", the objective function here is "oriented to dissipate minimum total energy". The formulation for the pipe crawling robot results in a min-max problem that can be transformed into a linear programming problem (Gill et al., 1981) and solved by the simplex method. In Galvez et al. (2001), in order to overcome the problem of discontinuity, a quadratic term of the joint-torques is added to the objective function. With this addition, it becomes a quadratic programming problem, which is then transformed into a linear complementarity problem and solved by using Lemke's algorithm. The aim of introducing the quadratic term is to penalize the fast changes of the objective function, rather than formulating the objective in a genuine quadratic form. In their objective function, there is always a trade-off between the optimum values and the discontinuous character of the solution; it has to be handled by arranging the constants in the quadratic term. In the study here the objective function is genuinely quadratic; therefore, there appears no problem of discontinuity. In the solution of the quadratic programming problem, the approach of eliminating the equality constraints is followed. Afterwards, the problem is converted into a linear complementary problem and solved by using Lemke's pivoting algorithm. The details of operations and handling the cycling problem in this solution are presented.

Considering all the reviewed papers, the distinguishing features of the chapter here are as follows: 1) the dynamic formulation and solution of the torque distribution are presented *together*; 2) the formulation is performed with using the joint-torques as the primary variables *throughout the derivation*, not only in the formulation of the optimization problem; 3) no simplification related to *the mass and inertia of the legs* is assumed in the solution of the distribution problem; 4) the simulation results are performed considering the whole robot system, namely the *protracting legs* are not ignored in the torque distribution solution.

5.2. Kinematic and Dynamic Modeling of a Six-Legged Robot With Three-Joint Legs

In the literature dealing with dynamics of walking machines it is usual that the very general equation of multi-link system dynamics is assumed to be given and that the explanations are performed using this general model (Zhao et al., 2000; Garcia et al., 2003; Pfeiffer et al., 1991). However, with such a general equation, it is hard to get insight into the particular case of a parallel system with changing topology as in walking-machines. In reality, while the system configuration changes, the matrices and variables in the very general formulas change both in value and dimension. The approach of "free body diagrams" (Barretto et al., 1998; Lin and Song, 2001), which is straightforward but applicationally very insightful, can be considered as an attempt to cope with these difficulties. As mentioned before, the method of free body diagrams is based on writing the force moment equations (Newton-Euler) for every rigid piece of the system, and to relate them to each other considering the physical interaction principles. The crucial thing is to perform the calculations and derive the equations in a systematic way, and to come up with a compact-simple-useful-insightful representation of the whole system dynamics. The chapter presented here makes use of this approach for deriving the dynamic equations with the joint-torques being the primary variables.

In mathematical modeling of kinematics and dynamics of the six-legged robot, it is assumed that the kinematics (translational and angular positions, velocities, and accelerations) of the main body of the robot is given. Then the aim is, first, to derive the kinematic equations of all the links on the legs and to determine the required joint velocities in order to achieve the given body motion; and second, to derive the dynamic equations related to all links and determine the force-moment relations that sustain the kinematic motions. The four steps below are followed to serve these calculations:

1) Kinematics of a single three-joint leg is handled (Chapter 4). The Jacobian matrix, which relates the "tip point velocity with respect to the robot body" to the "three-joint angular velocities" is derived.

2) The joint angle velocities of all the legs are determined. The legs of the robot are grouped into two: the ones in protraction (swing), and the ones in retraction (stance) phases. The legs in protraction are assumed to have predetermined tip point trajectories with respect to the robot body (this predetermined trajectory is assumed to be a proper one that will carry the tip point from the anterior to the posterior position, Chapter 4). The legs in retraction are assumed to be supporting the body without any slippage on their tip points. Namely, the tip points of the legs in retraction are stationary with respect to the ground frame. This means that the tip point velocity of a leg in retraction with respect to the robot body is equal to the negative of the velocity of the robot body with respect to the ground frame. As a result, given any desired body trajectory, the tip point velocities of the legs both in protraction and retraction are definite. Then using the inverse of the Jacobian matrix, it is possible to determine the joint angle velocities (accelerations) of all legs.

3) The kinematic equations of the links relating the link velocities and accelerations, both translational and rotational, to the joint angle velocities and accelerations are derived.

4) The dynamic equations of the links are derived using the method of free body diagrams. There are two basic equations, namely the Newton and Euler equations, that are related to translational and rotational motions, respectively. These equations are based on the principles that relate the external effects (forces and moments) acting on a rigid body to its inertia (mass and moment of inertia) and kinematics (translational acceleration,

angular acceleration, and angular velocity). Since the kinematics and inertia of the body and links are known, the total external forces and moments can be determined.

5.2.1. Kinematics of a Single Three-Joint Leg

A three joint robot leg (Figure 4.1) can be considered as a three link revolute joint (RRR) manipulator which is attached to a base (robot body). Therefore, the kinematic modeling can be performed following the conventional robotics approaches (Fu et al., 1987). The kinematic model of a three-joint leg was introduced in the previous chapter, Section 4.2. Here will only be mentioned some extra information needed for incorporation of the leg kinematics to the whole robot body.

The reference frame attachment in Figure 4.1 is valid for the three right side legs, but not for the ones on the left. One should notice that the frame attachments for the left side legs is not achieved by merely taking the symmetric form of the ones on the right with respect to the robot body. (In such a case the principality of the axes would not be sustained.) In the left side case, for the same leg configuration with the same joint angle values, the directions of the third components of the left leg frames are in the opposite direction with respect to the symmetric form of the ones on the right leg. The directions of the first components are determined accordingly to fit all the system to the Denavit-Hartenberg convention. For the left side legs, only the transformation matrices between the body and zeroth frames, and between the zeroth and first frames change. The change in the transformation matrix between the zeroth and first frames is simply a change on one of the Denavit-Hartenberg link parameters in Table 4.1. The change is $\alpha_1=-\pi/2$, for the left legs. With these changes for the legs on the left side, the formulations in Section 4.2 fit all the six legs.

The equations of kinematic relations are given in (5.1-5.7). Among these, some of them were given before in Section 4.2, but they are repeated here for convenience of gathering all the equations in relation to this section. The resulting transformation matrices between the ground and body frame, between the body and the zeroth frame, and between the sequential link frames are given in (5.1) and (5.2). The transformation matrices between the body and first frames are given separately for the right and left sides (5.2). In the formulas, (as in Section 4.2) the variables represented by a_i stand for the length of the *ith* link. The variables represented by θ_{ij} mean the sum of the *ith* and *jth* joint angles $(\theta_{ij}=\theta_i+\theta_j)$. *C* and *S* correspond to cosine and sinus functions, respectively. For the transformation between the body and ground frames (5.1), the Euler sequence of 312 (yaw: Φ_y, pitch: Φ_p, roll: Φ_r) is used. In (5.3) the tip point position (end point) with respect to the body frame, $P_e^{(b)}$, is given. In (5.4), the position of the tip point with respect to the zeroth reference frame, $P_e^{(0)}$, is given. Using the partial derivatives of this position with respect to the joint angles, as in (5.5), the Jacobian matrix, $\hat{J}_{p_v}$, related to the tip point translational velocity can be obtained, as given in (5.6). This matrix can be used to relate the tip point velocity of the leg, $\bar{v}_p^{(0)}$, to the joint angle velocities as in (5.7).

$$\hat{C}^{(g,b)} = \begin{bmatrix} C\Phi_y C\Phi_r - S\Phi_y S\Phi_p S\Phi_r & -S\Phi_y C\Phi_p & C\Phi_y S\Phi_r + S\Phi_y S\Phi_p C\Phi_r \\ S\Phi_y C\Phi_r + C\Phi_y S\Phi_p S\Phi_r & C\Phi_y C\Phi_p & S\Phi_y S\Phi_r - C\Phi_y S\Phi_p C\Phi_r \\ -C\Phi_p S\Phi_r & S\Phi_p & C\Phi_p C\Phi_r \end{bmatrix} \tag{5.1}$$

$$\hat{C}^{(b,0)}_{right} = \begin{bmatrix} 0 & C\Psi & S\Psi \\ -1 & 0 & 0 \\ 0 & -S\Psi & C\Psi \end{bmatrix} \qquad \hat{C}^{(b,0)}_{left} = \begin{bmatrix} 0 & -C\Psi & S\Psi \\ -1 & 0 & 0 \\ 0 & -S\Psi & -C\Psi \end{bmatrix}$$
$$\hat{C}^{(k-1,k)} = \begin{bmatrix} C\theta_k & -S\theta_k C\alpha_k & S\theta_k S\alpha_k \\ S\theta_k & C\theta_k C\alpha_k & -C\theta_k S\alpha_k \\ 0 & S\alpha_k & C\alpha_k \end{bmatrix} \tag{5.2}$$

$$\bar{P}_e^{(b)}(\bar{\theta}) = \begin{bmatrix} C\Psi(a_1 S\theta_1 + a_2 S\theta_1 C\theta_2 + a_3 S\theta_1 C\theta_{23}) + S\Psi(a_2 S\theta_2 + a_3 S\theta_{23}) \\ -(a_1 C\theta_1 + a_2 C\theta_1 C\theta_2 + a_3 C\theta_1 C\theta_{23}) \\ -S\Psi(a_1 S\theta_1 + a_2 S\theta_1 C\theta_2 + a_3 S\theta_1 C\theta_{23}) + C\Psi(a_2 S\theta_2 + a_3 S\theta_{23}) \end{bmatrix} \tag{5.3}$$

$$\bar{P}_e^{(0)}(\bar{\theta}) = \begin{bmatrix} a_1 C\theta_1 + a_2 C\theta_1 C\theta_2 + a_3 C\theta_1 C\theta_{23} \\ a_1 S\theta_1 + a_2 S\theta_1 C\theta_2 + a_3 S\theta_1 C\theta_{23} \\ a_2 S\theta_2 + a_3 S\theta_{23} \end{bmatrix} \tag{5.4}$$

$$\bar{J}_{pj} = \frac{\partial P_e^{(0)}}{\theta_j} \qquad \hat{J}_{p_v} = \left[\bar{J}_{p1} \vdots \bar{J}_{p2} \vdots \bar{J}_{p2}\right] \tag{5.5}$$

$$\hat{J}_{p_v} = \begin{bmatrix} -a_1 S\theta_1 - a_2 S\theta_1 C\theta_2 - a_3 S\theta_1 C\theta_{23} & -a_2 C\theta_1 S\theta_2 - a_3 C\theta_1 S\theta_{23} & -a_3 C\theta_1 S\theta_{23} \\ a_1 C\theta_1 - a_2 C\theta_1 C\theta_2 + a_3 C\theta_1 C\theta_{23} & -a_2 S\theta_1 S\theta_2 - a_3 S\theta_1 S\theta_{23} & -a_3 S\theta_1 S\theta_{23} \\ 0 & a_2 C\theta_2 + a_3 S\theta_{23} & a_3 C\theta_{23} \end{bmatrix} \tag{5.6}$$

$$\bar{v}_p^{(0)} = \hat{J}_{p_v} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \qquad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \hat{J}_{p_v}^{-1} \bar{v}_p^{(0)} \tag{5.7}$$

5.2.2. Kinematic Equations of the Links (Vectorial Relations)

It is assumed that the translational and angular accelerations of the body are already given as parameters of the desired motion. Besides, it is assumed that the legs in protraction follow a predetermined path with respect to the body; the legs in retraction also follow a determined path according to the body motion, since their tip points remain stationary on the ground. Since the tip point motions of the legs with respect to the body frame are all known, the velocities of the joint angles can easily be determined using the Jacobian matrix. These joint angle velocities and accelerations (derivatives of the velocities) will be used in deriving the link kinematics.

The kinematics of the links, namely translational accelerations, rotational velocities and rotational accelerations of the links, are considered to be the corresponding accelerations and velocities of the frames fixed to those links. Since each *jth* frame is fixed to the *jth* link, the velocities and accelerations of the *jth* frame on the *ith* leg will give the kinematics of the *jth* link of the *ith* leg. All the kinematic relations in this section will be represented as vectorial relations. Once these vectorial relations are obtained, they can be computed in any desired reference frame and converted to any other frame using the transformation matrices.

Using the general kinematic relations of moving frames (Fu et al., 1987, pp. 107-109), the angular velocities, angular accelerations, and translational accelerations of the three links of the *ith* leg can be calculated sequentially starting from the body going to the third link. Since the zeroth frame of the *ith* leg is stationary with respect to the body frame, the rotational velocity and acceleration of the zeroth frame with respect to the ground frame is equal to the rotational velocity and acceleration of the body frame with respect to the ground frame, which is already given (5.8 and 5.9).

Robot Body:

$$\vec{\omega}_{i0/g}=\vec{\omega}_{b/g}=\vec{\omega}_b \quad (\vec{\omega}_{i0/b}=\vec{0}) \tag{5.8}$$

$$\vec{\alpha}_{i0/g}=\vec{\alpha}_{b/g}=\vec{\alpha}_b \quad (\vec{\alpha}_{i0/b}=\vec{0}) \tag{5.9}$$

From now on, all the rotational and translational vectors will be with respect to the ground frame; therefore, the term *g* will be dropped from the equations. In the following are the rotational velocity and acceleration equations for the links of the *ith* leg. These equations are sequential in the sense that each uses the variable calculated in the previous equation. These variables are encircled in the equations.

Link 1 of the ith leg:

$$\vec{\omega}_{i1}=\enclose{circle}{\vec{\omega}_b}+\dot{\theta}_{i1}\vec{u}_3^{(i0)} \tag{5.10}$$

$$\vec{\alpha}_{i1}=\enclose{circle}{\vec{\alpha}_b}+\ddot{\theta}_{i1}\vec{u}_3^{(i0)}+\dot{\theta}_{i1}\vec{\omega}_b\times\vec{u}_3^{(i0)} \tag{5.11}$$

Link 2 of the ith leg:

$$\vec{\omega}_{i2}=\enclose{circle}{\vec{\omega}_{i1}}+\dot{\theta}_{i2}\vec{u}_3^{(i1)} \tag{5.12}$$

$$\vec{\alpha}_{i2}=\enclose{circle}{\vec{\alpha}_{i1}}+\ddot{\theta}_{i2}\vec{u}_3^{(i1)}+\dot{\theta}_{i2}\vec{\omega}_{i1}\times\vec{u}_3^{(i1)} \tag{5.13}$$

Link 3 of the ith leg:

$$\vec{\omega}_{i3}=\enclose{circle}{\vec{\omega}_{i2}}+\dot{\theta}_{i3}\vec{u}_3^{(i2)} \tag{5.14}$$

$$\vec{\alpha}_{i3}=\enclose{circle}{\vec{\alpha}_{i2}}+\ddot{\theta}_{i3}\vec{u}_3^{(i2)}+\dot{\theta}_{i3}\vec{\omega}_{i2}\times\vec{u}_3^{(i2)} \tag{5.15}$$

To relate the translational acceleration of the zeroth frame of the *ith* leg ($\vec{a}_{0_{i0}}$) to the acceleration of the body frame ($\vec{a}_{0_b}$), the position vector from the origin of the body frame to the origin of the zeroth frame of the *ith* leg ($\vec{r}_{bi}$) is needed (5.16). These vectors can be observed in Figure 5.1.

Robot Body:

$$\vec{a}_{O_{i0}}=\vec{a}_{O_b}+\vec{\alpha}_{i0}\times\vec{r}_{bi}+\vec{\omega}_{i0}\times(\vec{\omega}_{i0}\times\vec{r}_{bi})=\vec{a}_{O_{i0}}+\vec{\alpha}_{b}\times\vec{r}_{bi}+\vec{\omega}_{b}\times(\vec{\omega}_{b}\times\vec{r}_{bi}) \quad (5.16)$$

It is possible to obtain the acceleration of link mass centers by using the accelerations of the link frame origins. For this purpose the “vector from the link frame origin to the link mass center” ($\vec{c}_{ij}$) is needed. This vector is the negative of the vector ($\vec{c}_{ij(j+1)}$) in Figure 5.1: ($\vec{c}_{ij}=-\vec{c}_{ij(j+1)}$). In the following are the translational acceleration equations for the origins of the link frames ($\vec{a}_{O_{ij}}$) and the link mass centers ($\vec{a}_{ij}$).

Link 1 of the ith leg:

$$\vec{a}_{O_{i1}}=\enclose{circle}{\vec{a}_{O_{i0}}}+a_{i1}\vec{\alpha}_{i1}\times\vec{u}_1^{(i1)}+a_{i1}\vec{\omega}_{i1}\times(\vec{\omega}_{i1}\times\vec{u}_1^{(i1)}) \quad (5.17)$$

$$\vec{a}_{i1}=\vec{a}_{O_{i1}}+\vec{\alpha}_{i1}\times\vec{c}_{i1}+\vec{\omega}_{i1}\times(\vec{\omega}_{i1}\times\vec{c}_{i1}) \quad (5.18)$$

Link 2 of the ith leg:

$$\vec{a}_{O_{i2}}=\enclose{circle}{\vec{a}_{O_{i1}}}+a_{i2}\vec{\alpha}_{i2}\times\vec{u}_1^{(i2)}+a_{i2}\vec{\omega}_{i2}\times(\vec{\omega}_{i2}\times\vec{u}_1^{(i2)}) \quad (5.19)$$

$$\vec{a}_{i2}=\vec{a}_{O_{i2}}+\vec{\alpha}_{i2}\times\vec{c}_{i2}+\vec{\omega}_{i2}\times(\vec{\omega}_{i2}\times\vec{c}_{i2}) \quad (5.20)$$

Link 3 of the ith leg:

$$\vec{a}_{O_{i3}}=\enclose{circle}{\vec{a}_{O_{i2}}}+a_{i3}\vec{\alpha}_{i3}\times\vec{u}_1^{(i3)}+a_{i3}\vec{\omega}_{i3}\times(\vec{\omega}_{i3}\times\vec{u}_1^{(i3)}) \quad (5.21)$$

$$\vec{a}_{i3}=\vec{a}_{O_{i3}}+\vec{\alpha}_{i3}\times\vec{c}_{i3}+\vec{\omega}_{i3}\times(\vec{\omega}_{i3}\times\vec{c}_{i3}) \quad (5.22)$$

5.2.3. Force and Moment Equations Based on Free Body Diagrams (Vectorial Relations)

In order to derive the dynamic equations, the Newton and Euler equations should be applied to every rigid piece of the total system. Considering a moving rigid body, Newton's Equation relates “the acceleration of the mass center of the body with respect to the inertial frame” to “the total external forces applied on the body”. Euler's Equation, on the other hand, relates “the change of the angular momentum of the body with respect to the inertial frame” to “the total external moments applied on the body”.

In the case of the six-legged robot, there are *19* rigid pieces: the robot body, and the three links of the six legs. In Figure 5.1, the free body diagrams of the robot body and three links of a leg are shown with the external forces and moments acting on each. It should be noted that all the motions, namely the translational and rotational accelerations and velocities of all the rigid pieces, are determined by the preceding formulations. The aim now is to find the forces and moments required to achieve these motions.

For the robot body, the left-up figure in Figure 5.1 can be considered. The two equations for all rigid pieces take the following form. In these equations the variables related to the other equations are again encircled.

Robot Body:

$$m_b\vec{a}_b=m_b\vec{g}+\sum_{i=1}^{6}\enclose{circle}{\vec{F}_{ib}} \tag{5.23}$$

$$\check{J}_b\cdot\vec{\alpha}_b+\vec{\omega}_b\times(\check{J}_b\cdot\vec{\omega}_b)=\sum_{i=1}^{6}\enclose{circle}{\vec{M}_{ib}}+\sum_{i=1}^{6}\vec{r}_{ib}\times\vec{F}_{ib} \tag{5.24}$$

Link 1 of the ith leg:

$$m_{i1}\vec{a}_{i1}=m_{i1}\vec{g}-\vec{F}_{ib}+\enclose{circle}{\vec{F}_{i21}} \tag{5.25}$$

$$\check{J}_{i1}\cdot\vec{\alpha}_{i1}+\vec{\omega}_{i1}\times(\check{J}_{i1}\cdot\vec{\omega}_{i1})=-\vec{M}_{ib}+\enclose{circle}{\vec{M}_{i21}}-\vec{c}_{i1b}\times\vec{F}_{ib}+\vec{c}_{i12}\times\vec{F}_{i21} \tag{5.26}$$

Link 2 of the ith leg:

$$m_{i2}\vec{a}_{i2}=m_{i2}\vec{g}-\vec{F}_{i21}+\enclose{circle}{\vec{F}_{i32}} \tag{5.27}$$

$$\check{J}_{i2}\cdot\vec{\alpha}_{i2}+\vec{\omega}_{i2}\times(\check{J}_{i2}\cdot\vec{\omega}_{i2})=-\vec{M}_{i21}+\enclose{circle}{\vec{M}_{i32}}-\vec{c}_{i21}\times\vec{F}_{i21}-\vec{c}_{i23}\times\vec{F}_{i32} \tag{5.28}$$

Link 3 of the ith leg:

$$m_{i3}\vec{a}_{i3}=m_{i3}\vec{g}-\vec{F}_{i32}+\enclose{circle}{\vec{F}_{ig3}} \tag{5.29}$$

$$\check{J}_{i3}\cdot\vec{\alpha}_{i3}+\vec{\omega}_{i3}\times(\check{J}_{i3}\cdot\vec{\omega}_{i3})=-\vec{M}_{i32}+\enclose{circle}{\vec{M}_{ig3}}-\vec{c}_{i32}\times\vec{F}_{i32}-\vec{c}_{i3g}\times\vec{F}_{ig3} \tag{5.30}$$

For the legs in protraction, the tip points are on the air, namely there is no external force acting on the last link of the leg other than the joint torque and gravity ($\vec{F}_{ig3}=\vec{M}_{ig3}=\vec{0}$). Since the "total external force and moment on the link" can be found using the kinematics and inertia relations, the "gravitational effect" can be subtracted from that and the "required interaction force and torque between the last and the second links" can be determined. For the second link, this time, "the interaction effect between the second and first links" can be found by subtracting "the interaction effect between the second and last links" and "the gravity effect" from "the required force and moment". The same logic can be applied to the first link. In this way, for a leg in protraction, its effect (force and moment effects) on the body can be determined. With this logic, rearranging the force and moment equations of the links leads to the equations of (5.31-38).

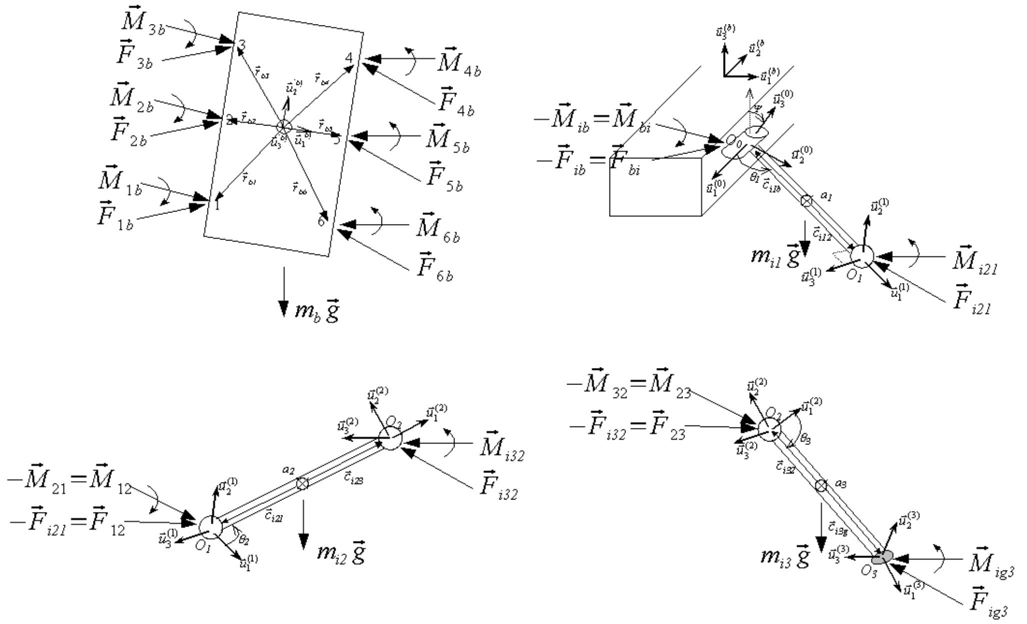


Figure 5.1: The free body diagrams for the robot body, and three links of one leg.

$$\vec{F}_{ib}=\vec{F}_{i21}+m_{i1}(\vec{g}-\vec{a}_{i1}) \tag{5.31}$$

$$\vec{F}_{i21}=\vec{F}_{i32}+m_{i2}(\vec{g}-\vec{a}_{i2}) \tag{5.32}$$

$$\vec{F}_{i32}=\vec{F}_{ig3}+m_{i3}(\vec{g}-\vec{a}_{i3}) \tag{5.33}$$

$$\vec{F}_{ib}=\vec{F}_{ig3}+m_{i3}(\vec{g}-\vec{a}_{i3})+m_{i2}(\vec{g}-\vec{a}_{i2})+m_{i1}(\vec{g}-\vec{a}_{i1}) \;=\; \vec{F}_{ig3}+\vec{f}_{imga} \tag{5.34}$$

$$\vec{f}_{imga} \;=\; m_{i3}(\vec{g}-\vec{a}_{i3})+m_{i2}(\vec{g}-\vec{a}_{i2})+m_{i1}(\vec{g}-\vec{a}_{i1}) \tag{5.35}$$

$$\vec{M}_{ib}=\vec{M}_{i21}-\vec{c}_{i1b}\times\vec{F}_{ib}+\vec{c}_{i12}\times\vec{F}_{i21}-[\check{J}_{i1}\cdot\vec{\alpha}_{i1}+\vec{\omega}_{i1}\times(\check{J}_{i1}\cdot\vec{\omega}_{i1})] \tag{5.36}$$

$$\vec{M}_{i21}=\vec{M}_{i32}-\vec{c}_{i21}\times\vec{F}_{i21}+\vec{c}_{i23}\times\vec{F}_{i32}-[\check{J}_{i2}\cdot\vec{\alpha}_{i2}+\vec{\omega}_{i2}\times(\check{J}_{i2}\cdot\vec{\omega}_{i2})] \tag{5.37}$$

$$\vec{M}_{i32}=\vec{M}_{ig3}-\vec{c}_{i32}\times\vec{F}_{i32}+\vec{c}_{i3g}\times\vec{F}_{ig3}-[\check{J}_{i3}\cdot\vec{\alpha}_{i3}+\vec{\omega}_{i3}\times(\check{J}_{i3}\cdot\vec{\omega}_{i3})] \tag{5.38}$$

Equating the ground tip point interaction terms to zero, $\vec{F}_{ig3}=\vec{M}_{ig3}=\vec{0}$, and performing the above calculations sequentially, all the force moment relations for the protracting legs can be found, and most importantly the effect of retracting legs on the robot body ($\vec{F}_{ib}$ and $\vec{M}_{ib}$) can be computed.

For the legs in retraction, the ground tip point interaction terms ($\vec{F}_{ig3}$ and $\vec{M}_{ig3}$) are not necessarily zero. However, it is a common and reasonable approach, followed in all reviewed literature of legged systems, to assume a hard point contact between the tip point and ground; which means there is only a translational force interaction, and no moment components. Consequently, the moment interaction is taken to be zero ($\vec{M}_{ig3} = \vec{0}$) and the force interaction $\vec{F}_{ig3}$ is left to be determined. (In fact, the formulation here is not limited to this assumption. The cases where moment interactions are considered, especially robot hand applications, can also be handled with the formulation presented here.) Since the tip point-ground force interaction is unknown, the method of computation from the tip towards the body cannot be applied to retracting legs. In fact, the tip point forces on these legs are dependent on the torques applied on the joints. It should be noticed that the only unknown for the group of force moment equations (5.31-38) for a retracting leg is the force interaction with the ground, $\vec{F}_{ig3}$. Once the force equations (5.31-34) are substituted into the moment equations (5.36-38), and a rearrangement is performed, the moment equations take the following form:

$$\vec{M}_{ig3}=\vec{0} \tag{5.39}$$

$$\begin{aligned}\vec{M}_{i32}=&[-\vec{c}_{i32}+\vec{c}_{i3g}]\times\vec{F}_{ig3}-\vec{c}_{i32}\times[m_{i3}(\vec{g}-\vec{a}_{i3})]\\&-[\check{J}_{i3}\cdot\vec{\alpha}_{i3}+\vec{\omega}_{i3}\times(\check{J}_{i3}\cdot\vec{\omega}_{i3})]\end{aligned} \tag{5.40}$$

$$\begin{aligned}\vec{M}_{i21}=&[-\vec{c}_{i32}+\vec{c}_{i3g}-\vec{c}_{i21}+\vec{c}_{i23}]\times\vec{F}_{ig3}\\&+[-\vec{c}_{i32}-\vec{c}_{i21}+\vec{c}_{i23}]\times[m_{i3}(\vec{g}-\vec{a}_{i3})]-[\vec{c}_{i21}]\times[m_{i2}(\vec{g}-\vec{a}_{i2})]\\&-[\check{J}_{i3}\cdot\vec{\alpha}_{i3}+\vec{\omega}_{i3}\times(\check{J}_{i3}\cdot\vec{\omega}_{i3})]-[\check{J}_{i2}\cdot\vec{\alpha}_{i2}+\vec{\omega}_{i2}\times(\check{J}_{i2}\cdot\vec{\omega}_{i2})]\end{aligned} \tag{5.41}$$

$$\begin{aligned}\vec{M}_{ib}=&[-\vec{c}_{i32}+\vec{c}_{i3g}-\vec{c}_{i21}+\vec{c}_{i23}-\vec{c}_{i1b}+\vec{c}_{i12}]\times\vec{F}_{ig3}\\&+[-\vec{c}_{i32}-\vec{c}_{i21}+\vec{c}_{i23}-\vec{c}_{i1b}+\vec{c}_{i12}]\times[m_{i3}(\vec{g}-\vec{a}_{i3})]\\&+[-\vec{c}_{i21}-\vec{c}_{i1b}+\vec{c}_{i12}]\times[m_{i2}(\vec{g}-\vec{a}_{i2})]\\&+[-\vec{c}_{i1b}]\times[m_{i1}(\vec{g}-\vec{a}_{i1})]\\&-[\check{J}_{i3}\cdot\vec{\alpha}_{i3}+\vec{\omega}_{i3}\times(\check{J}_{i3}\cdot\vec{\omega}_{i3})]-[\check{J}_{i2}\cdot\vec{\alpha}_{i2}+\vec{\omega}_{i2}\times(\check{J}_{i2}\cdot\vec{\omega}_{i2})]\\&-[\check{J}_{i1}\cdot\vec{\alpha}_{i1}+\vec{\omega}_{i1}\times(\check{J}_{i1}\cdot\vec{\omega}_{i1})]\end{aligned} \tag{5.42}$$

These equations for the moments between the links (5.40-42) can be written in a more compact form as in (5.43-45). The values of the vectors ($\vec{r}_{i03}, \vec{r}_{i13}, \vec{r}_{i23}, \vec{m}_{ib}, \vec{m}_{i21}, \vec{m}_{i32}$) in these equations are explicitly revealed by the comparison of (5.43-45) and (5.40-42).

$$\vec{M}_{ib}=\vec{r}_{i03}\times\vec{F}_{ig3}+\vec{m}_{ib} \tag{5.43}$$

$$\vec{M}_{i21}=\vec{r}_{i13}\times\vec{F}_{ig3}+\vec{m}_{i21} \tag{5.44}$$

$$\vec{M}_{i32}=\vec{r}_{i23}\times\vec{F}_{ig3}+\vec{m}_{i32} \tag{5.45}$$

5.2.4. Torque Calculations

Up to this point, all the moment relations between the links and between the first links and robot body are determined for both protracting and retracting legs. For the protracting legs, the determined equations can directly be calculated since there are no unknowns. For the retracting legs, on the other hand, the tip point ground interaction forces do still remain as unknowns. The aim in this section is to derive the equations for the joint-torques, and with these equations, to relate the joint-torques to the tip point-ground force interactions for the retracting legs. It is well known that the magnitude of a joint-torque is equal to the magnitude of the component of the moment interaction between the two links in the rotation direction. Therefore, the three torques on the joints of a leg can be written as follows:

$$\tau_{i1}=-\vec{M}_{ib}\cdot\vec{u}_3^{(i0)} \qquad \begin{aligned}\tau_{i1}&=-(\vec{r}_{i03}\times\vec{F}_{ig3}+\vec{m}_{ib})\cdot\vec{u}_3^{(i0)}\\ &=-(\vec{u}_3^{(i0)}\times\vec{r}_{i03})\cdot\vec{F}_{ig3}+\vec{m}_{ib}\cdot\vec{u}_3^{(i0)}\end{aligned} \tag{5.46}$$

$$\tau_{i2}=-\vec{M}_{i21}\cdot\vec{u}_3^{(i1)} \qquad \begin{aligned}\tau_{i2}&=-(\vec{r}_{i13}\times\vec{F}_{ig3}+\vec{m}_{i21})\cdot\vec{u}_3^{(i1)}\\ &=-(\vec{u}_3^{(i1)}\times\vec{r}_{i13})\cdot\vec{F}_{ig3}+\vec{m}_{i21}\cdot\vec{u}_3^{(i1)}\end{aligned} \tag{5.47}$$

$$\tau_{i3}=-\vec{M}_{i32}\cdot\vec{u}_3^{(i2)} \qquad \begin{aligned}\tau_{i3}&=-(\vec{r}_{i23}\times\vec{F}_{ig3}+\vec{m}_{i32})\cdot\vec{u}_3^{(i2)}\\ &=-(\vec{u}_3^{(i2)}\times\vec{r}_{i23})\cdot\vec{F}_{ig3}+\vec{m}_{i32}\cdot\vec{u}_3^{(i2)}\end{aligned} \tag{5.48}$$

If these equations are put in a matrix form the following relations can be obtained. In these relations the values of the tensor $\check{R}_i$ and the column $\overline{m}_i$ are explicit.

$$\begin{bmatrix}\tau_{i1}\\ \tau_{i2}\\ \tau_{i3}\end{bmatrix}=-\begin{bmatrix}\vec{u}_3^{(i0)}\times\vec{r}_{i03}\\ \vec{u}_3^{(i1)}\times\vec{r}_{i13}\\ \vec{u}_3^{(i2)}\times\vec{r}_{i23}\end{bmatrix}\vec{F}_{ig3}-\begin{bmatrix}\vec{m}_{ib}\cdot\vec{u}_3^{(i0)}\\ \vec{m}_{i21}\cdot\vec{u}_3^{(i1)}\\ \vec{m}_{i32}\cdot\vec{u}_3^{(i2)}\end{bmatrix} \tag{5.49}$$

$$\overline{\tau}_i=-\check{R}_i\cdot\vec{F}_{ig3}-\overline{m}_i \tag{5.50}$$

$$\vec{F}_{ig3}=\check{R}_i^{-1}(-\overline{\tau}_i-\overline{m}_i)=-\check{R}_i^{-1}\overline{\tau}_i-\check{R}_i^{-1}\overline{m}_i \tag{5.51}$$

5.2.5. Relating the Body Equations to Tip Point Forces and Joint Torques

The total force and moment affect of the six legs on the robot body can be found by the following equations:

$$m_b\vec{a}_b=m_b\vec{g}+\sum_{i=1}^{6}\vec{F}_{ib} \quad\Rightarrow\quad \vec{F}_b=\sum_{i=1}^{6}\vec{F}_{ib}=m_b(\vec{a}_b-\vec{g}) \tag{5.52}$$

$$\begin{aligned}&\check{J}_b\cdot\vec{\alpha}_b+\vec{\omega}_b\times(\check{J}_b\cdot\vec{\omega}_b)=\sum_{i=1}^{6}\vec{M}_{ib}+\sum_{i=1}^{6}\vec{r}_{bi}\times\vec{F}_{ib}\\ &\Rightarrow\ \vec{M}_b=\sum_{i=1}^{6}\vec{M}_{ib}+\sum_{i=1}^{6}\tilde{r}_{bi}\cdot\vec{F}_{ib}=\check{J}_b\cdot\vec{\alpha}_b+\vec{\omega}_b\times(\check{J}_b\cdot\vec{\omega}_b)\end{aligned} \tag{5.53}$$

The two equations above give the effect of all the legs on the robot body. Since the effect of the protracting legs on the body is explicitly known, it is possible to subtract these effects from the total, and remain with the equations of effects of the retracting (supporting) legs on the robot body. Let's assume that the set of legs in protraction is given by $L_P=\{i : \textit{if the leg i is in retraction}\}=\{r_1, r_2, ..., r_s\}$, and the set of legs in protraction is given by $L_P=\{i : \textit{if the leg i is in protraction}\}=\{p_1, p_2, ..., p_t\}$. It is clear that $s+t=6$. Then (5.52) and (5.53) can be rewritten in the following form:

$$\vec{F}_b{}'=\sum_{r_i=1}^{s}\vec{F}_{r_ib}=m_b(\vec{a}_b-\vec{g})-\sum_{p_i=1}^{t}\vec{F}_{p_ib} \tag{5.54}$$

$$\begin{aligned}\vec{M}_b{}' &= \sum_{r_i=1}^{s} \vec{M}_{r_i b} + \sum_{r_i=1}^{s} \tilde{r}_{b r_i} \cdot \vec{F}_{r_i b} \\ &= \check{J}_b \cdot \vec{\alpha}_b + \vec{\omega}_b \times (\check{J}_b \cdot \vec{\omega}_b) - \sum_{p_i=1}^{t} \vec{M}_{p_i b} - \sum_{p_i=1}^{t} \tilde{r}_{b p_i} \cdot \vec{F}_{p_i b}\end{aligned} \tag{5.55}$$

If these two equations are put in a matrix form, they reduce to (5.56). In this equation, the right hand side is known, and the left hand side is to be rearranged in order to lead the calculations to the joint torques.

$$\begin{bmatrix} \sum_{r_i=1}^{s} \vec{F}_{r_i b} \\ \sum_{r_i=1}^{s} \vec{M}_{r_i b} + \sum_{r_i=1}^{s} \vec{r}_{b r_i} \times \vec{F}_{r_i b} \end{bmatrix} = \begin{bmatrix} \vec{F}_b{}' \\ \vec{M}_b{}' \end{bmatrix} \tag{5.56}$$

Rearranging (5.56) and representing the vectors with respect to the ground frame, the matrix equality in terms of the force and moment effect of the retracting legs can be achieved as in (5.57). The notation *(g)* on the force and moment columns is used to mean “the column representation of the vector with respect to the ground frame”.

$$\underbrace{\left[\begin{array}{cccccccccccccccccccc} 1&0&0&1&0&0&&1&0&0&0&0&0&0&0&0&&0&0&0 \\ 0&1&0&0&1&0&\cdots&0&1&0&0&0&0&0&0&0&\cdots&0&0&0 \\ 0&0&1&0&0&1&&0&0&1&0&0&0&0&0&0&&0&0&0 \\ \hdashline &&&&&&&&&&1&0&0&1&0&0&&1&0&0 \\ &\tilde{r}^{(g)}_{br_1}&&&\tilde{r}^{(g)}_{br_2}&&\cdots&&\tilde{r}^{(g)}_{br_s}&&0&1&0&0&1&0&\cdots&0&1&0 \\ &&&&&&&&&&0&0&1&0&0&1&&0&0&1 \end{array}\right]}_{\hat{L}} \begin{bmatrix} \bar{F}^{(g)}_{r_1 b} \\ \bar{F}^{(g)}_{r_2 b} \\ \cdots \\ \bar{F}^{(g)}_{r_s b} \\ \cdots\cdots \\ \bar{M}^{(g)}_{r_1 b} \\ \bar{M}^{(g)}_{r_2 b} \\ \cdots \\ \bar{M}^{(g)}_{r_s b} \end{bmatrix} = \begin{bmatrix} \vec{F}_b'^{(g)} \\ \vec{M}_b'^{(g)} \end{bmatrix} \tag{5.57}$$

The large matrix on the left side can be denoted by ($\hat{L}$), and the vector representation can be replaced by the column representation of those vectors with respect to the ground frame. Then the equation reduces to the simple representation of

$$\Rightarrow \hat{L} \cdot \begin{bmatrix} \bar{F}^{(g)}_{r_1 b} \\ \bar{F}^{(g)}_{r_2 b} \\ \cdots \\ \bar{F}^{(g)}_{r_s b} \\ \cdots\cdots \\ \bar{M}^{(g)}_{r_1 b} \\ \bar{M}^{(g)}_{r_2 b} \\ \cdots \\ \bar{M}^{(g)}_{r_s b} \end{bmatrix} = \begin{bmatrix} \bar{F}_b'^{(g)} \\ \bar{M}_b'^{(g)} \end{bmatrix} = \begin{bmatrix} F'_{bx} \\ F'_{by} \\ F'_{bz} \\ M'_{bx} \\ M'_{by} \\ M'_{bz} \end{bmatrix} \tag{5.58}$$

Substitution of (5.51) in (5.34) and (5.43), and representation with respect to the ground frame leads to the following equations relating the actuator torques to the effect of the legs on the robot body:

$$\vec{F}_{ib}^{(g)} = \vec{F}_{ig3}^{(g)} + \vec{f}_{imga}^{(g)} \;=\; \hat{T}_{if}\bar{\tau}_i + \bar{t}_{if} + \bar{f}_{imga}^{(g)} \tag{5.59}$$

$$\vec{M}_{ib}^{(g)} = \tilde{r}_{i03}^{(g)} \cdot \vec{F}_{ig3}^{(g)} + \vec{m}_{ib}^{(g)} \;= \hat{T}_{im}\bar{\tau}_i + \bar{t}_{im} + \bar{m}_{ib}^{(g)} \tag{5.60}$$

$$\begin{array}{lcl} \hat{T}_{if} = (-\check{R}_i^{(g)})^{-1} & ; & \bar{t}_{if} = \hat{T}_{if}\bar{m}_i \\ \hat{T}_{im} = \tilde{r}_{i03}^{(g)}\hat{T}_{if} & ; & \bar{t}_{im} = \tilde{r}_{i03}^{(g)}\bar{t}_{if} \end{array} \tag{5.61}$$

For the legs in retraction the two equations of (5.59) and (5.60) can be put in a matrix form, in which the vector and tensor representations with respect to the ground frame are replaced by their column and matrix representations.

$$\begin{bmatrix} \bar{F}_{r_1 b}^{(g)} \\ \bar{F}_{r_2 b}^{(g)} \\ \vdots \\ \bar{F}_{r_s b}^{(g)} \\ \cdots\cdots \\ \bar{M}_{r_1 b}^{(g)} \\ \bar{M}_{r_2 b}^{(g)} \\ \vdots \\ \bar{M}_{r_s b}^{(g)} \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{T}_{r_1 f} & \hat{0} & \hat{0} & \cdots & \hat{0} \\ \hat{0} & \hat{T}_{r_2 f} & \hat{0} & \cdots & \hat{0} \\ & & \vdots & & \\ \hat{0} & \hat{0} & \hat{0} & \cdots & \hat{T}_{r_s f} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \hat{T}_{r_1 m} & \hat{0} & \hat{0} & \cdots & \hat{0} \\ \hat{0} & \hat{T}_{r_2 m} & \hat{0} & \cdots & \hat{0} \\ & & \vdots & & \\ \hat{0} & \hat{0} & \hat{0} & \cdots & \hat{T}_{r_s m} \end{bmatrix}}_{\hat{C}_{fm}} \begin{bmatrix} \bar{\tau}_{r_1} \\ \bar{\tau}_{r_2} \\ \vdots \\ \bar{\tau}_{r_s} \end{bmatrix} + \underbrace{\begin{bmatrix} \bar{t}_{r_1 f} + \bar{f}_{r_1 mga}^{(g)} \\ \bar{t}_{r_2 f} + \bar{f}_{r_2 mga}^{(g)} \\ \vdots \\ \bar{t}_{r_{sf}} + \bar{f}_{r_s mga}^{(g)} \\ \cdots\cdots \\ \bar{t}_{r_1 m} + \bar{m}_{r_1 b}^{(g)} \\ \bar{t}_{r_2 m} + \bar{m}_{r_2 b}^{(g)} \\ \vdots \\ \bar{t}_{r_s m} + \bar{m}_{r_s b}^{(g)} \end{bmatrix}}_{\bar{c}_{fm}} \tag{5.62}$$

Representing the big matrix and the column on the right side by ($\hat{C}_{fm}$) and ($\bar{c}_{fm}$), respectively, (5.62) takes the following simple form:

$$\begin{bmatrix} \bar{F}_{r_1 b}^{(g)} \\ \bar{F}_{r_2 b}^{(g)} \\ \vdots \\ \bar{F}_{r_s b}^{(g)} \\ \cdots\cdots \\ \bar{M}_{r_1 b}^{(g)} \\ \bar{M}_{r_2 b}^{(g)} \\ \vdots \\ \bar{M}_{r_s b}^{(g)} \end{bmatrix} = \hat{C}_{fm} \begin{bmatrix} \bar{\tau}_{r_1} \\ \bar{\tau}_{r_2} \\ \vdots \\ \bar{\tau}_{r_s} \end{bmatrix} + \bar{c}_{fm} \tag{5.63}$$

Substituting (5.63) into (5.58) leads to the equality of

$$\underbrace{\hat{L}\cdot\hat{C}_{fm}}_{\hat{A}}\begin{bmatrix}\bar{\tau}_{r_1}\\ \bar{\tau}_{r_2}\\ \vdots\\ \bar{\tau}_{r_s}\end{bmatrix}=\underbrace{\begin{bmatrix}F'_{bx}\\ F'_{by}\\ F'_{bz}\\ M'_{bx}\\ M'_{by}\\ M'_{bz}\end{bmatrix}-\hat{L}\cdot\bar{c}_{fm}}_{\bar{b}} \tag{5.64}$$

The very standard equality for the joint torques of retracting (supporting) legs is obtained as follows, where ($\hat{A}$) is a $6\times 3s$ matrix and ($\bar{b}$) is a 6×1 column:

$$\hat{A}\begin{bmatrix}\bar{\tau}_{r_1}\\ \bar{\tau}_{r_2}\\ \vdots\\ \bar{\tau}_{r_s}\end{bmatrix}=\bar{b} \tag{5.65}$$

It is clear that at least three legs have to be supporting the body in order to achieve the static stability. This means that the value of s is at least 3 ($s\geq 3$). Considering that there are three joints in each leg, the number of unknowns (joint torques) in (5.65) will always be greater than the number of equations, which is always 6. There are infinitely many solutions to (5.65). Consequently, the calculation of torques for the legs on the ground is a “determination” among the infinite possibilities.

The problem now is to solve (5.65) in a way to minimize energy consumption. This optimization should consider the constraints due to friction and joint-torque limits. However, since minimizing the energy consumption is considered to be equivalent to minimizing the joint-torque squares, it makes sense to omit the constraints on the joint-torque limits. (In cases where these limits are considerably small, they should also be included in the formulation.)

5.2.6. Friction Constraints

The relation between the tip point forces and joint-torques of a supporting leg are determined by (5.51) and (5.61) as

$$\vec{F}^{(g)}_{ig3}=\bar{F}_{ig3}=\begin{bmatrix}f_{ix}\\ f_{iy}\\ f_{iz}\end{bmatrix}=\hat{T}_{if}\bar{\tau}_i+\hat{t}_{if} \tag{5.66}$$

Here two kind of constraints for each supporting leg on a flat surface are considered:

1) The third component of the force applied to the tip from the ground has to be in the negative direction of the gravitation. Therefore,

$$f_{iz}\geqslant 0 \tag{5.67}$$

2) The horizontal component of the force applied to the tip from the ground has to be smaller than the limit determined by the friction coefficient and the vertical component. This constraint takes the following form where μ_f is the friction coefficient:

$$\mu_f^2 f_z^2 \geq (f_x^2 + f_y^2) \tag{5.68}$$

This non-linear inequality corresponds to a friction cone (Nahon and Angeles, 1992). Due to the difficulty of handling non-linear constraints, it is a common approach to transform them into linear constraints that are tighter than the original. In the case here, the friction cone is transformed into a friction pyramid (Klein and Kittivatcharapong, 1990; Nahon and Angeles, 1992) by the following four linear inequalities:

$$(\mu_{eff}=\frac{\mu_f}{\sqrt{2}}) \qquad \Rightarrow \qquad \begin{array}{c} \mu_{eff} f_{iz} - f_{ix} \geqslant 0 \\ \mu_{eff} f_{iz} + f_{ix} \geqslant 0 \\ \mu_{eff} f_{iz} - f_{iy} \geqslant 0 \\ \mu_{eff} f_{iz} + f_{iy} \geqslant 0 \end{array} \tag{5.69}$$

The inequalities of (5.69) already imply (5.67). Therefore, for each supporting leg, the four constraints in (5.69) are sufficient for the formulation. For representational purposes, (5.69) will be put in a matrix form, and then converted to a matrix inequality in terms of joint-torques. Considering (5.66), the following equations perform this transformation. The values of the matrix $\hat{M}$ and column $\bar{m}$ are explicit in (5.72).

$$\hat{\mu}_{eff} = \begin{bmatrix} -1 & 0 & \mu_{eff} \\ 1 & 0 & \mu_{eff} \\ 0 & -1 & \mu_{eff} \\ 0 & 1 & \mu_{eff} \end{bmatrix} \quad \textit{and} \ (5.69) \ \Rightarrow \ \hat{\mu}_{eff} \bar{F}_{ig3} \geqslant \bar{0} \tag{5.70}$$

$$\begin{bmatrix} \hat{\mu}_{eff}\hat{T}_{r_1 f} & \hat{0} & \cdots & \hat{0} \\ \hat{0} & \hat{\mu}_{eff}\hat{T}_{r_2 f} & \cdots & \hat{0} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{0} & \hat{0} & \cdots & \hat{\mu}_{eff}\hat{T}_{r_s f} \end{bmatrix} \begin{bmatrix} \hat{\tau}_{r_1} \\ \hat{\tau}_{r_2} \\ \vdots \\ \hat{\tau}_{r_s} \end{bmatrix} \geqslant \begin{bmatrix} -\hat{\mu}_{eff}\bar{t}_{r_1 f} \\ -\hat{\mu}_{eff}\bar{t}_{r_2 f} \\ \vdots \\ -\hat{\mu}_{eff}\bar{t}_{r_s f} \end{bmatrix} \tag{5.71}$$

$$\hat{M}\,\bar{\tau} \geqslant \bar{m} \tag{5.72}$$

5.2.7. The Minimization Problem

The objective function to be minimized was mentioned as “the sum of the squares of the joint-torques on the supporting legs”. This objective function is quadratic, and can be represented by the following term, where the positive definite weight matrix $\hat{H}$ takes the proper entries.

$$Objective\ Function \rightarrow \frac{1}{2}\bar{\tau}^T \hat{H}\,\bar{\tau} \tag{5.73}$$

It should be stated that the formulations and solution algorithms proposed from now on, require only the term $\hat{H}$ to be positive-definite and symmetric. Combining (5.65), (5.72), and (5.73) the overall optimization problem can be written as follows:

$$\begin{array}{ll} Minimize & \frac{1}{2}\bar{\tau}^T \hat{H}\,\bar{\tau} \\ subject\ to & \hat{A}\,\bar{\tau} = \bar{b} \\ & \hat{M}\,\bar{\tau} \geqslant \bar{m} \end{array} \tag{5.74}$$

5.2.8. Computational Cost of the Formulation

For convenience, the results of the computational cost analysis for the formulation of (5.74) are given in Table 5.2. The formulation presented here is compared with the efficient formulations in which the tip point force components are the primary variables (Cheng and Orin, 1991; Hung et al., 2000). For the analysis of Table 5.2, the operations are assumed to be performed in an efficient way (e.g., symmetry of matrices are considered, multiplications and additions with zero entries are avoided, efficient matrix inversion with Gauss elimination (Noble and Daniel, 1977) is assumed.) The number of multiplications and additions for a three-joint *n*-legged robot, with *m* legs supporting is calculated to be [384*n*+128*m*+100 (×); 270*n*+109*m*+67 (+)]. This means [3172 (×); 2341 (+)] operations for a six-legged robot with all the legs supporting.

The number of operations for the formulation in Cheng and Orin (1991) is given by [(180*N*-46)*m*+27 (×); (147*N*-46)*m*+27 (+)] for an *m* legged, each with *N* degrees of freedom, robot. This makes [2991 (×); 2415 (+)] operations for a three-joint six-legged robot with all the legs supporting. If the algorithm in Hung et al. (2000) is applied to a three-joint *n*-legged robot for the case of *m* supporting legs, the number of necessary multiplications and additions can be calculated to be [354*n*+134*m*+29 (×); 267*n*+108*m*+45 (+)]. For the three-joint six-legged robot with six legs supporting, this makes [2957 (×); 2295 (+)] operations. The amount of computation in this chapter is slightly more than the ones in Cheng and Orin (1991) and Hung et al. (2000). The operations of matrix inversion in (5.61), which are needed for the transformation of force equations into torque equations, do not exist in the formulations of these two papers. These inversions result in [27*m* (×); 12*m* (+)]=[142 (×); 72 (+)] operations, which constitute much of the difference. However, since the difference is not significant (less than 8%), it can be concluded that the computational cost of the formulation here is almost on the level of the mentioned two efficient formulations.

Table 5.2: Computational cost of formulation for a three-joint n-legged robot with m-legs supporting.

	Related equations	**Number of multiplications**	**Number of additions**
Transformation matrices	(1), (2)	$12n+16$	4
Transformation matrices from each reference frame to the ground frame	(matrix-matrix multiplications)	$72n+12$	$45n+12$
Transformation of the inertia matrices to the ground frame	(matrix-matrix-matrix multiplications)	$135n+45$	$90n+30$
Transformation of the link vectors to the ground frame	(matrix-vector multiplications)	$33n$	$3n$
Force moment equations and torque calculations for the protracting legs	(12)-(17), (21)	$126n-126m$	$120n-120m$
Vectors of force-moment relations for the supporting legs	(18)-(20)	$117m$	$126m$
Vectors and matrices for relating the torques to the effect of the supporting legs on the robot body.	(33)	$87m$	$45m$
The forces and moments required to be supplied by the supporting legs on the robot body	(27), (28)	$6n-6m+27$	$12n-12m+21$
The matrix A	(30), (34), (35)	$18m$	$39m$
The vector b	(30), (34), (35)	$6m$	$12m+6$
Constraint matrices	(40-42)	$32m$	$16m$
Total Computation		**$384n+128m+100$**	**$270n+109m+67$**

5.3. Torque Distribution: Solution of the Quadratic Programming Problem

The minimization problem stated in (5.74) is a typical quadratic programming problem with linear equality and inequality constraints. Although some general-purpose packages are available in commercial software for the solution of such quadratic programs, it is significant to mention the steps of solution explicitly for two purposes. First, the microprocessors used in robotics applications have limited memory capacities, so that they cannot make use of the packages of mathematical programs; hence, every step of the computational routines has to be implemented on microprocessors. The algorithm here is simple to be implemented on microprocessors. This is because it makes use of the tableau method, which is commonly used in linear programming problems, and which is based on simple row operations. Second, the general-purpose routines might not be very efficient for the problem here. In the paper here, the approach of eliminating the equality constraints in advance (Cheng and Orin, 1990; Nahon and Angeles, 1992) is followed. It is observed that by this elimination the required time for the solution is decreased by almost 50%. Furthermore, the approach here uses a "modified simplex type method" (Goldfarb and Idnani, 1983), but, by transforming the primal problem into a "linear complementary problem" (Bazaraa et al., 1993). In this way, the computationally costly phase-I solution of other modified simplex type methods for quadratic programming (Bazaraa et al., 1993; Dantzig, 1963) is avoided. The steps of the solution are as follows:

1) The equality constraints are eliminated in an optimal way. With this, the optimal particular solution for the equality constraints is obtained. The search for the optimal homogeneous solution is formulated as a quadratic programming with inequality constraints only.

2) The problem is put in a standard form by manipulating the joint-torque variables in the homogeneous part.

3) The linear complementary problem is generated using the Karush-Kuhn-Tucker Conditions of the system.

4) The linear complementary problem is solved using Lemke's Complementary Pivoting Algorithm (Bazaraa et al., 1993).

5) The homogeneous solution is obtained from the solution of the quadratic programming.

6) The general solution is obtained by combining the particular and homogeneous solutions.

For the case in Figure 5.4, the average CPU time required to solve the optimization problem is *15-16ms*, with an unoptimized MATLAB coding on a 2.4 GHz Pentium IV processor. This time is considered to be fast enough for real-time applications. (In Nahon and Angels (1992), *38ms* is considered to be fast enough to supply the set points for a force controller.)

The *elimination of the equality constraints* corresponds to solving the following problem:

$$\begin{aligned} &Minimize && \frac{1}{2}\bar{\tau}^T \hat{H}\,\bar{\tau} \\ &subject\ to && \hat{A}\,\bar{\tau}=\bar{b} \end{aligned} \tag{5.75}$$

This problem can easily be solved by using Lagrange multipliers (Sabes, 2001). Considering the fact that $\hat{H}$ is symmetric, the solution for (5.75), hence the particular solution for (5.74), can be given as follows:

$$\bar{\tau}_0=\hat{H}^{-1}\hat{A}^T[\hat{A}\,\hat{H}^{-1}\hat{A}^T]^{-1}\bar{b} \tag{5.76}$$

The general solution to the problem in (5.74) can now be stated as

$$\bar{\tau}=\bar{\tau}_0+\hat{N}_A\bar{\tau}_1 \tag{5.77}$$

where the columns of $\hat{N}_A$ are a basis for the null-space of $\hat{A}$ (Desoer, 1970). Substituting (5.77) in (5.74) the equivalent modified problem can be stated as follows:

$$\begin{aligned} &Minimize && \frac{1}{2}\bar{\tau}_1^T \hat{H}_1\bar{\tau}_1 \\ &subject\ to && -\hat{M}_1\bar{\tau}_1\leqslant-\bar{m}_1 \end{aligned} \tag{5.78}$$

where

$$\hat{H}_1=\hat{N}_A^T\hat{H}\,\hat{N}_A\ ;\ \hat{M}_1=\hat{M}\,\hat{N}_A\ ;\ \bar{m}_1=\bar{m}-\hat{M}\,\bar{\tau}_0. \tag{5.79}$$

It should be noted here that the matrix $\hat{H}_1$ is positive- definite (due to the positive definiteness of $\hat{H}$ and the columns of $\hat{N}_A$ being linearly independent); and there appears no linear term in the objective function after the substitution. However, the problem in (5.78) is not in a standard form since the torque variables can take both positive and negative values. In order to transform this into a *quadratic programming problem in the standard form* (Bazaraa et al., 1993), the torque variables can be decomposed into two positive components as

$$\bar{\tau}_1 = \bar{\tau}_{1p} - \bar{\tau}_{1n} \quad \textit{where} \quad \bar{\tau}_{1p} \geqslant \bar{0} \quad \textit{and} \quad \bar{\tau}_{1n} \geqslant \bar{0} \,. \tag{5.80}$$

If the new vector of variables is defined as

$$\bar{\tau}_{1\,pn} = \begin{bmatrix} \bar{\tau}_{1p} \\ \bar{\tau}_{1n} \end{bmatrix} \tag{5.81}$$

the problem in standard form can be stated as

$$\begin{array}{ll} \textit{Minimize} & \dfrac{1}{2} \bar{\tau}^T_{1\,pn} \hat{H}_{1\,pn} \bar{\tau}_{1\,pn} \\ \textit{subject to} & \hat{A}_{1\,pn} \bar{\tau}_{1\,pn} \leqslant \bar{b}_{1\,pn} \\ & \bar{\tau}_{1\,pn} \geqslant \bar{0} \end{array} \tag{5.82}$$

where

$$\hat{H}_{1\,pn} = \begin{bmatrix} \hat{H}_1 & -\hat{H}_1 \\ -\hat{H}_1 & \hat{H}_1 \end{bmatrix} \; ; \; \hat{A}_{1\,pn} = \begin{bmatrix} -\hat{M}_1 & \hat{M}_1 \end{bmatrix} \; ; \; \bar{b}_{1\,pn} = [-\bar{m}_1] \tag{5.83}$$

In (5.82), the symmetric weight matrix $\hat{H}_{1pn}$ is positive-semi-definite (guaranteed by its special form and the positive-definiteness of $\hat{H}_1$); hence, the objective function is “convex” (Bertsekas, 1999; Gill et al., 1981). For the problems of convex objective functions with linear constraints, the Karush-Kuhn-Tucker (KKT) necessary conditions for local optimality turn out to be the sufficient conditions for global optimality (Bazaraa et al., 1993). With some algebraic manipulations the KKT conditions for problem (5.82) can be written in a special form (Bazaraa et al., 1993) as in (5.83).

$$\begin{gathered} \hat{A}_{1\,pn} \bar{\tau}_{1\,pn} + \bar{y} = \bar{b}_{1\,pn} \\ -\hat{H}_{1\,pn} \bar{\tau}_{1\,pn} - \hat{A}^T_{1\,pn} \bar{u} + \bar{v} = \bar{0} \\ \bar{\tau}^T_{1\,pn} \bar{v} = \bar{0} \quad , \quad \bar{u}^T \bar{y} = \bar{0} \\ \bar{\tau}_{1\,pn} \; , \; \bar{y} \; , \; \bar{u} \; , \; \bar{v} \geq \bar{0} \end{gathered} \tag{5.83}$$

These KKT conditions can be put in a matrix form as

$$\begin{gathered} \bar{w} - \hat{\Phi} \bar{z} = \bar{q} \\ \bar{w}^T \bar{z} = \bar{0} \\ \bar{w} \; , \; \bar{z} \geqslant \bar{0} \end{gathered} \tag{5.84}$$

where

$$\hat{\Phi} = \begin{bmatrix} \hat{0} & -\hat{A}_{1\,pn} \\ \hat{A}^T_{1\,pn} & \hat{H}_{1\,pn} \end{bmatrix} \; ; \; \bar{q} = \begin{bmatrix} \bar{b}_{1\,pn} \\ \bar{0} \end{bmatrix} \; ; \; \bar{w} = \begin{bmatrix} \bar{y} \\ \bar{v} \end{bmatrix} \; ; \; \bar{z} = \begin{bmatrix} \bar{u} \\ \bar{\tau}_{1\,pn} \end{bmatrix} . \tag{5.85}$$

The system in (5.83) is a *linear complementary problem*, which can be solved by using “Lemke's Complementary Pivoting Algorithm” (Bazaraa et al., 1993). This algorithm can easily be implemented using the tableau method described in the mentioned reference. In case of non-degeneracy (which is important for pivoting), the positive-semi-definiteness of the matrix $\hat{H}_{1pn}$ and existence of no linear terms in the objective function of (5.82) are sufficient conditions for the algorithm to stop in a finite number of iterations with a KKT point (Bazaraa et al., 1993). (If a linear term exist in the objective function – which is the case for the “force distribution” applied in this paper for comparative purposes – in case of non-degeneracy, the positive-semi-definiteness of $\hat{H}_{1pn}$ is a sufficient condition for the algorithm to find a KKT point in finite steps only if it exists, namely only if the problem is consistent. In our case, the “force distribution” system is guaranteed to be consistent; because, a change of the primal variables to the tip-point forces in the “force distribution” problem will again result in a system like (5.82) without any linear terms.) However, the problem of degeneracy, which is noted to exist in most real-world problems (Bazaraa et al., 1990), exists also here and leads to infinite cycling (circling) of pivoting (Bazaraa et al., 1993; Dantzig, 1963). A practical solution to overcome degeneracy is perturbing the column of constants ($\overline{q}$) by adding very small random numbers (in the level of 10^{-9}) to its entries (Bazaraa et al., 1990). Any feasible basis derived from the solution of the non-degenerate problem obtained in this way is also a basis for the degenerate problem (Murty, 1983). The feasibility for the original degenerate problem is violated only in the range of 10^{-9}; therefore, this basis can be considered as a feasible basis for the original degenerate problem. It should be noted that the analytical rules for avoiding cycling are mostly computationally expensive (Bazaraa et al., 1990); hence, such a practical solution is desirable for robotic applications.

5.4. Comparison of “Torque Distribution” with Tip-point “Force Distribution”

In this section the results of the comparison of “torque distribution” and “force distribution” schemes are given. For convenience, a brief introduction for the handling of “force distribution” will be given first.

The “force distribution” scheme is obtained by replacing the objective function in (5.74) with the quadratic term of the tip-point forces. The new objective function corresponds to the sum of the squares of the tip-point force components. As a result, the problem in (5.74) takes the new form of

$$\begin{array}{ll} \textit{Minimize} & \frac{1}{2}\overline{F}^{T}_{g3}\hat{H}\,\overline{F}_{g3} \\ \textit{subject to} & \hat{A}\overline{\tau}=\overline{b} \\ & \hat{M}\,\overline{\tau}\geqslant\overline{m} \end{array} \tag{5.86}$$

where $\overline{F}_{g3}$ represents the column of the components of all the tip-point forces on the supporting legs. This column can easily be found in terms of joint-torques, by using the equalities of (5.66).

Then, the objective function in (5.86) has both a linear and quadratic term of the joint-torque variables. The linear term changes the form of the homogeneous torque solution in (5.76) as follows:

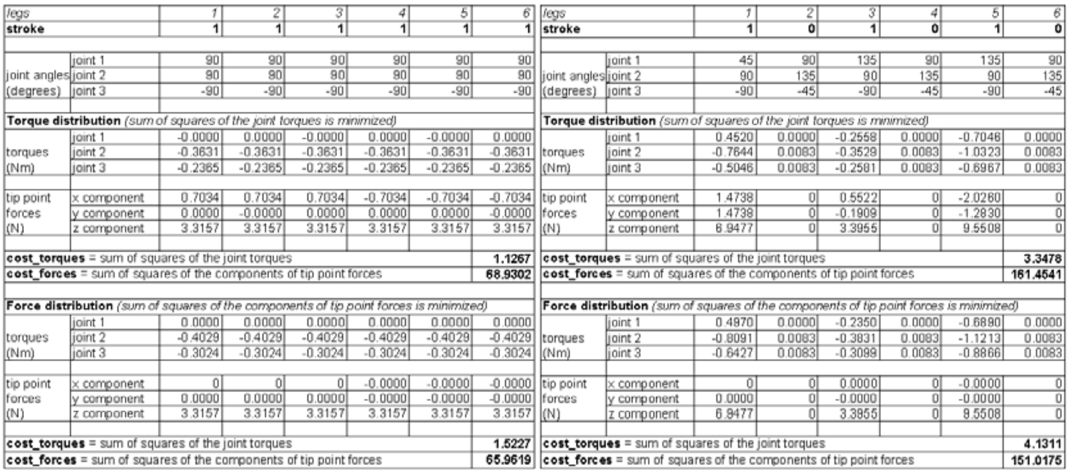
$$\bar{\tau}_0=\hat{H}_T^{-1}\hat{A}^T(\hat{A}\hat{H}_T^{-1}\hat{A}^T)^{-1}(\bar{b}+\hat{A}\hat{H}_T^{-1}\bar{c}_T)-\hat{H}_T^{-1}\bar{c}_T \tag{5.87}$$

Here, $\bar{c}_T$ and $\hat{H}_T$ are the resulting column and matrix for respectively the linear and quadratic terms of joint-torques in (5.86). The remaining formulation and solution follows the described procedure for the “torque distribution” scheme.

5.4.1. Observations on Stationary Standing

In Table 5.3, two stationary standing situations on a flat surface are considered with friction coefficient of $\mu=0.3$ between ground and tip-points. In the example of the left table, the robot is standing with six supporting legs (the value of “stroke” is 1 for all the six legs), all the legs being in a nominal supporting position with joint angles of $\pm 90^0$. For the case of the right table, the robot is supported by three legs, two extreme legs of the left side and the middle leg of the right side (the value of “stroke” is [1 0 1 0 1 0] for the legs). The left rear leg supports with an extension to the back, while the left front and right middle legs support with an extension to the front. In the tables the resulting joint torques and tip point forces are given for both “torque distribution” and “force distribution”. For both of the cases, the “force distribution” resulted in forces with zero horizontal components, which means that with those distributions the robot can stand on ice (frictionless environment). One can feel that the robot must be dissipating excessive energy since it does not need to behave as if it is standing on ice. This means that the robot does not make good use of the friction. In the torque distribution cases, however, there are significant horizontal components of the tip point forces. In both tables, the “cost_torques”, namely the sum of the squares of the joint-torques, is smaller in the torque distribution case, whereas the “cost_forces”, namely the sum of the squares of the components of the tip-point forces, is smaller in the force distribution case. It should be noted that both of these solutions satisfy the dynamic equations as well as the constraints imposed by the friction between the ground and the tip-point. What shall be highlighted about the results is that, the decrease in the “cost_torques” by using the torque distribution instead of the force distribution is not insignificant. The decrease is more than 25% for the case of the left table, and more than 18% for the case of the right table. Since the “sum of the squares of the torques”, rather than the “sum of the squares of the components of tip-point forces”, is a close indicator of dissipated energy, the decrease in the “cost_torques” corresponds to more or less the same amount of decrease in the dissipated power in the motors driving the joints. As a result it can be concluded that, the “torque distribution” is much more energy efficient than the “force distribution”, since it makes use of the friction and does not eliminate the useful contribution of internal forces. This conclusion is briefly affirmed, also in (Kar et al., 2001), which states “the horizontal foot force components reduce energy consumption significantly”.

Table 5.3: Two examples of standing on flat surface for comparison of "torque distribution" with "force distribution".



legs		*1*	*2*	*3*	*4*	*5*	*6*
stroke		**1**	**1**	**1**	**1**	**1**	**1**
joint angles (degrees)	joint 1	90	90	90	90	90	90
	joint 2	90	90	90	90	90	90
	joint 3	-90	-90	-90	-90	-90	-90
Torque distribution *(sum of squares of the joint torques is minimized)*							
torques (Nm)	joint 1	-0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
	joint 2	-0.3631	-0.3631	-0.3631	-0.3631	-0.3631	-0.3631
	joint 3	-0.2365	-0.2365	-0.2365	-0.2365	-0.2365	-0.2365
tip point forces (N)	x component	0.7034	0.7034	0.7034	-0.7034	-0.7034	-0.7034
	y component	0.0000	-0.0000	0.0000	0.0000	0.0000	-0.0000
	z component	3.3157	3.3157	3.3157	3.3157	3.3157	3.3157
cost_torques = sum of squares of the joint torques							**1.1267**
cost_forces = sum of squares of the components of tip point forces							**68.9302**
Force distribution *(sum of squares of the components of tip point forces is minimized)*							
torques (Nm)	joint 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	joint 2	-0.4029	-0.4029	-0.4029	-0.4029	-0.4029	-0.4029
	joint 3	-0.3024	-0.3024	-0.3024	-0.3024	-0.3024	-0.3024
tip point forces (N)	x component	0	0	0	-0.0000	-0.0000	-0.0000
	y component	0.0000	0.0000	0.0000	-0.0000	-0.0000	-0.0000
	z component	3.3157	3.3157	3.3157	3.3157	3.3157	3.3157
cost_torques = sum of squares of the joint torques							**1.5227**
cost_forces = sum of squares of the components of tip point forces							**65.9619**

legs		*1*	*2*	*3*	*4*	*5*	*6*
stroke		**1**	**0**	**1**	**0**	**1**	**0**
joint angles (degrees)	joint 1	45	90	135	90	135	90
	joint 2	90	135	90	135	90	135
	joint 3	-90	-45	-90	-45	-90	-45
Torque distribution *(sum of squares of the joint torques is minimized)*							
torques (Nm)	joint 1	0.4520	0.0000	-0.2558	0.0000	-0.7046	0.0000
	joint 2	-0.7644	0.0083	-0.3529	0.0083	-1.0323	0.0083
	joint 3	-0.5046	0.0083	-0.2581	0.0083	-0.6967	0.0083
tip point forces (N)	x component	1.4738	0	0.5522	0	-2.0260	0
	y component	1.4738	0	-0.1909	0	-1.2830	0
	z component	6.9477	0	3.3955	0	9.5508	0
cost_torques = sum of squares of the joint torques							**3.3478**
cost_forces = sum of squares of the components of tip point forces							**161.4541**
Force distribution *(sum of squares of the components of tip point forces is minimized)*							
torques (Nm)	joint 1	0.4970	0.0000	-0.2350	0.0000	-0.6890	0.0000
	joint 2	-0.8091	0.0083	-0.3831	0.0083	-1.1213	0.0083
	joint 3	-0.6427	0.0083	-0.3099	0.0083	-0.8866	0.0083
tip point forces (N)	x component	0	0	0.0000	0	-0.0000	0
	y component	0.0000	0	-0.0000	0	-0.0000	0
	z component	6.9477	0	3.3855	0	9.5508	0
cost_torques = sum of squares of the joint torques							**4.1311**
cost_forces = sum of squares of the components of tip point forces							**151.0175**

5.4.2. Observing the Effect of Friction Coefficient

To our knowledge the only work that mentions the effect of friction coefficient on the dissipated energy of a multi-legged robot is (Kar et al., 2001). Here, the conclusion that the "energy consumption...reduces gradually with an increase in the value of the coefficient of friction" will be approved, but it will be demonstrated that this is possible only if "torque distribution", rather than the "force distribution", is used.

The example will be given for the stationary standing with six-legs described on the left table of Table 5.3. In Figure 5.2, the effect of the friction coefficient on the sum of the squares of the joint-torques on the six supporting legs is depicted for both "force distribution" and "torque distribution" cases. As it is observed there is no change of dissipated energy if the "force distribution" is used. This is expected, since the "force distribution" eliminates the horizontal tip point forces and gives the same result of distribution whatever the friction coefficient is. The "torque distribution", on the other hand, makes quite good use of the margin allowed by the friction coefficient. The energy dissipation decreases with the increasing friction coefficient.

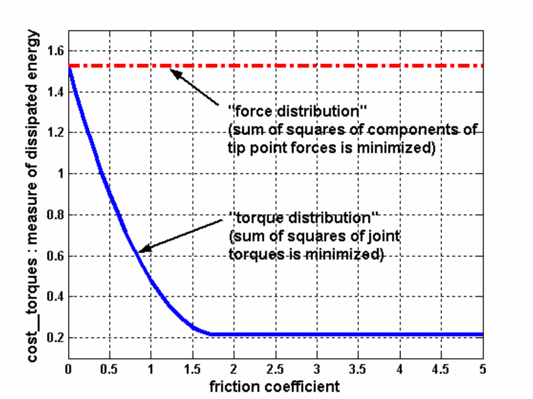


Figure 5.2: Effect of the friction coefficient on dissipated energy for "force distribution" and "torque distribution" schemes.

5.4.3. Observations on Tripod Gait Walking

In this section the results mentioned above will be demonstrated on a full walking of a simulative six-legged robot. The aim is to make the comparison of the results of the two distribution schemes, based on a full epoch of walk, rather than a stationary standing position. The data of this simulative robot is derived from the actual three-joint six-legged Robot-EA308. The approximated values of the physical variables can be found in Appendices A and B. The arrangement of calculations for a whole period of walking necessitates a detailed programming and tedious combining of the calculations described in previous sections. In the simulative model here, all the mathematical calculations of dynamics, both of retracting and protracting legs, are performed without any simplification mentioned in relation to the reviewed references. The protraction movement of the legs is constructed in an energy optimal way as introduced in Chapter 4. To demonstrate the arguments mentioned above, a four step constant speed walking is performed on a flat surface using the tripod gait. The friction coefficient between the ground and the leg tips is taken to be *0.3*. Since, in a regular tripod gait the duty factor (ratio of the protraction time to the elapsed time for a full step) is ½, the retraction and protraction times are equal for all legs. The protraction time here is taken to be *1.5 seconds*. The walk is repeated for both “torque distribution” and “force distribution” calculations.

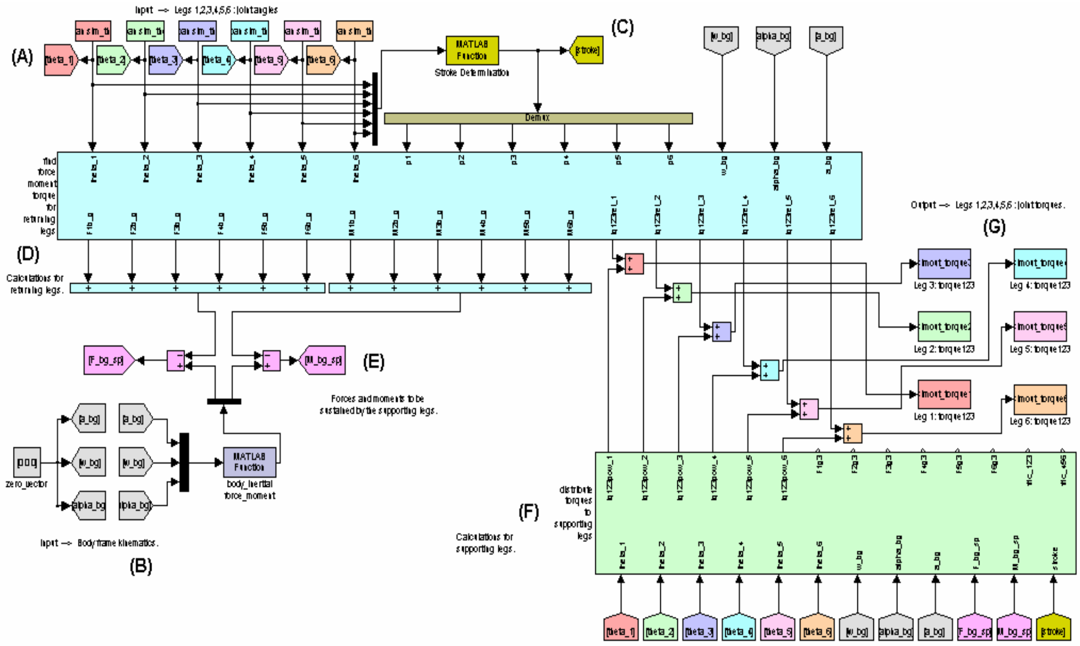

Input —> Legs 1,2,3,4,5,6 : joint angles
(A)
[theta_1]
[theta_2]
[theta_3]
[theta_4]
[theta_5]
[theta_6]
MATLAB Function
Stroke Determination
[stroke]
(C)
Demux
[w_bg]
[alpha_bg]
[a_bg]
find force moment torque for returning legs
(D)
Calculations for returning legs.
[F_bg_sp]
[M_bg_sp]
(E)
Forces and moments to be sustained by the supporting legs.
zero_vector
MATLAB Function
body_inertial force_moment
Input —> Body frame kinematics.
(B)
Output —> Legs 1,2,3,4,5,6 : joint torques.
(G)
Leg 1: torque123
Leg 2: torque123
Leg 3: torque123
Leg 4: torque123
Leg 5: torque123
Leg 6: torque123
distribute torques to supporting legs
(F)
Calculations for supporting legs.

Figure 5.3: SIMULINK block diagram for the calculations of the six-legged robot model. (The robot body is moving with constant speed.)

In Figure 5.3, the MATLAB SIMULINK model used for the calculations throughout the walking is given. This model, in a way, summarizes the basics of the calculations described in previous sections. According to the steps of calculations the diagram is labeled by sequential letters. In region (A) the predetermined traces of joint angles, in region (B) the kinematics of the body frame are input to the system. In region (B) the required force and moment for the body frame are calculated using body kinematics variables. Since the example is for a constant speed walk, the accelerations of the body frame are all zero. In region (C) the stroke configuration is determined based on the joint angle inputs of the legs. This stroke configuration is used in region (D) to determine the returning (protracting) legs. In this region the calculations related to the returning legs are performed; the effect of these on the body frame and the joint torques on these returning legs are calculated. In region (E) the effect of returning legs is combined by the required force and moment for the body frame. In this way, the amount of force and moment to be supplied by the supporting legs is determined. In region (F) these force and moment are distributed to the supporting legs. It is this point that either the “torque distribution” or “force distribution” scheme is used. Lastly, in region (G) all the joint-torques of the six legs are obtained, combining the results of the calculations for returning legs and torque/force distribution.

In Figure 5.4 the leg configurations and the tripod gait plan for four steps are given. In Figure 5.5 and the left side of Figure 5.7, the results for the “torque distribution” scheme, in Figure 5.6 and the right side of Figure 5.7, the results for the “force distribution” scheme are given.

In Figure 5.5, the joint-torque traces of the six legs are depicted with the figures being placed in accordance to the given leg configuration. One can easily observe that for all the legs, the joint-torques in retraction take significant values while the joint-torques in protraction are close to zero. Throughout the walk, the average of the sum of the torque squares is recorded to be *0.7178*. In Figure 5.6, the results for the "force distribution" scheme are given in the same order. One can observe that the joint-torques take considerably larger values than those of the "torque distribution" scheme. Therefore, the average of the sum of the torque squares, which is recorded to be *1.3295*, is almost twice that of the "torque distribution" scheme. Since these are considered to be proportional to the average dissipated power on the motors, it can be concluded that the "torque distribution" scheme results in an almost twice-efficient distribution compared to the "force distribution", for the tripod-walking example given here.

On the left side of Figure 5.7, one can observe that the torque distribution limited the ratios of horizontal over vertical components to the friction coefficient; namely, the maximum possible value. On the right side, on the other hand, these ratios remain very close to zero. This shows that the "torque distribution" achieves efficiency by making use of the friction between the ground and tip-points of the supporting legs. In other words, it uses the internal forces to decrease the dissipated energy. It can easily be guessed that if the friction coefficient were taken to be larger, the efficiency of the "torque distribution" would be much better compared to the "force distribution". (In fact, *0.3* seems to be a small value compared to many possible surface interactions. To give an idea, some kinetic friction coefficients for some surface interactions are as follows: rubber on concrete: 0.8; steel on steel: 0.57; aluminum on steel: 0.47; copper on steel:0.36; glass on glass: 0.4; wood on wood: 0.2; teflon on teflon: 0.04.)

5.5. Conclusion

In this chapter a full kinematic and dynamic modeling of a three-joint six-legged robot is derived, taking into account the detailed dynamical formulations of all the links on both retracting and protracting legs. The derivation is performed by using the joint-torques as the primary variables. The distribution of forces and moments is formulated as a “torque distribution” problem with an energy efficiency oriented objective function. The resultant quadratic programming problem is solved by eliminating the equality constraints, converting the problem into a linear complementary problem, and using Lemke's Complementary Pivoting Algorithm. The simulation results demonstrate that by formulating the objective function as “minimizing the sum of the squares of the joint-torques”, rather than the more common approach of “minimizing the sum of the squares of the tip point force components”, a much more energy efficient distribution is achieved. Moreover, it is demonstrated that, the “torque distribution” scheme makes good use of the friction between the tip points of the supporting legs and the ground. The “force distribution” scheme, on the other hand, does not make use of this friction since it eliminates all the internal forces in the system.

The results presented here emphasize the following three items: 1) Rather than the tip point forces, the joint-torques should be minimized in distributing the required forces and moments to the supporting legs; 2) the legs on walking machines usually have considerable weight and inertia with respect to the body frame, therefore it is not realistic to ignore the mass and inertia of either supporting or returning legs in dynamic formulations; 3) friction helps considerably to minimize the dissipated energy if the “torque distribution” scheme is used; therefore any system should have an adaptive structure to estimate the friction coefficient in different surfaces, and perform the torque distribution calculations accordingly.

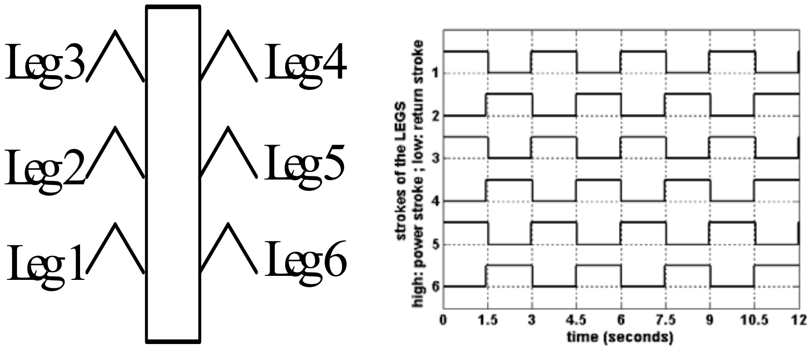


Figure 5.4: The leg configuration and the tripod gait plan for four steps.

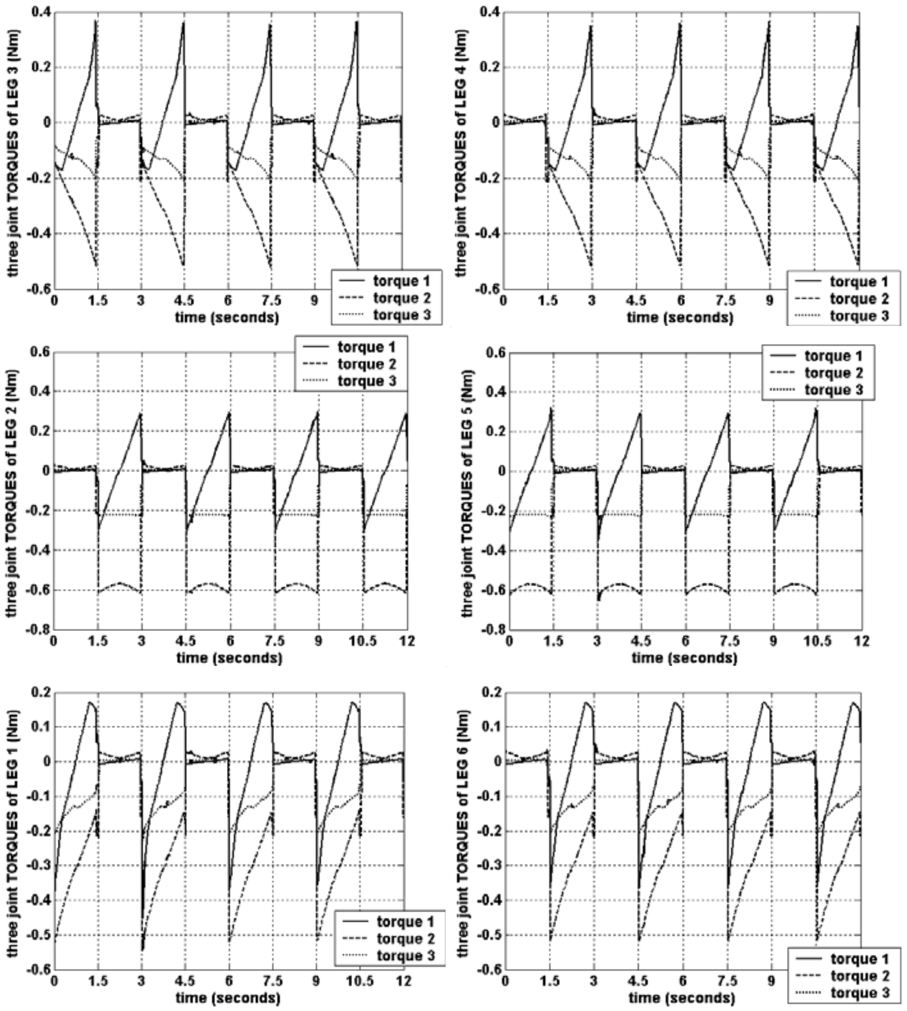


Figure 5.5: The graphs of joint-torques for tripod gait walking on level surface with "torque distribution". (Duty factor is ½: retraction and protraction times are 1.5 seconds. Friction coefficient: 0.3. Average of the sum of the torque squares: **0.7178**.)

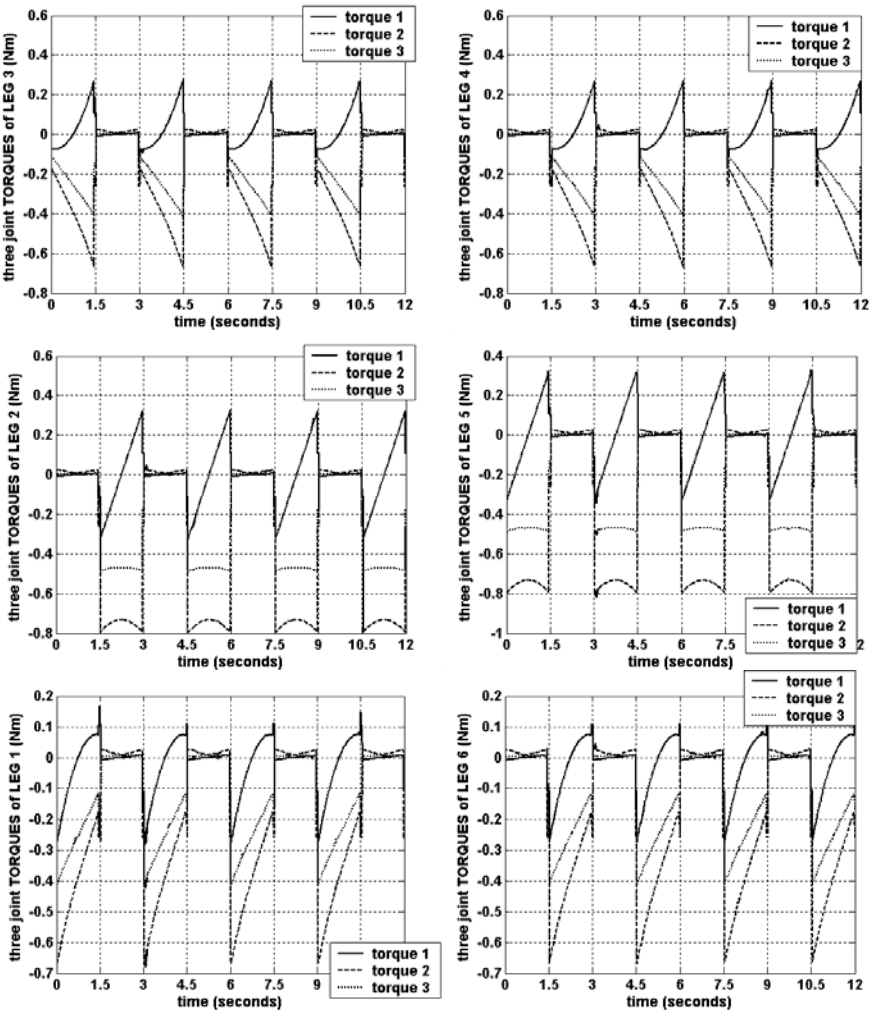


Figure 5.6: The graphs of joint-torques for tripod gait walking on level surface with "force distribution". (Duty factor is ½: retraction and protraction times are 1.5 seconds. Friction coefficient: 0.3. Average of the sum of the torque squares: **1.3295**.)

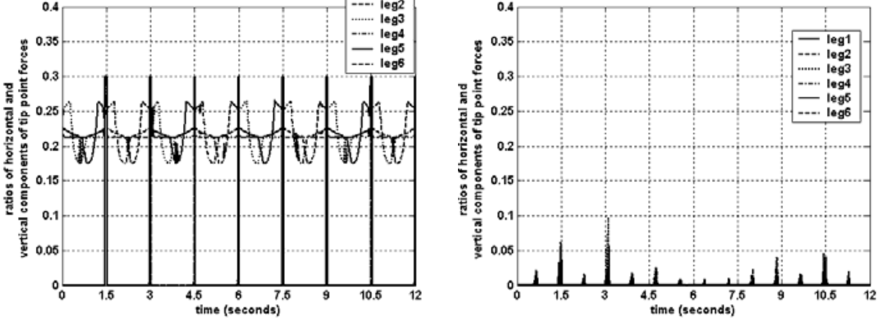


Figure 5.7: The ratios of the horizontal over vertical components of the tip point forces for “torque distribution” (left) and “force distribution” (right) cases.

CHAPTER 6

ANALYSIS OF WAVE GAITS FOR ENERGY EFFICIENCY

6.1. Introduction

As mentioned in Chapter 2, wave gaits are widely used in multi-legged walking machines due to their high stability. The choice of a particular wave gait for a walking machine corresponds to determining the parameters delineating the gait (duty factor, stroke length, protraction time), in accordance with the mechanical design limitations (maximum stroke length, minimum protraction time, etc.) and preferences of walking (speed). However, since there are more than one choices, this is not a trivial task. In this chapter, minimizing the power consumption throughout the walk is taken to be the criterion to determine such parameters. The chapter aims to develop some strategies to adopt the best walking, in the vicinity of wave gaits and in the set of standard gait (Chapter 2), in order to minimize the power consumption.

In Kar et al. (2001) an analysis of energy efficiency of wave gaits is performed with respect to the structural parameters, interaction forces, friction coefficient, and duty factor, based on a simulation model of a three-joint six-legged robot. The aim is mainly to introduce the developed method of force distribution rather than arriving to some guiding principles for finding the most efficient parameters of wave gaits. The structural parameters that are chosen are the lateral offset of the legs from the robot body and the ratio of the links on the legs. It is observed that considering "specific energy consumption" (energy consumption per weight per traveled length), there is an optimum lateral offset which makes the ground reaction force vectors pass close to the hip or knee points of the legs throughout the walk. The link ratio, as well, is observed to have an optimum value minimizing the specific energy consumption. In the paper a force distribution scheme that minimizes the power consumption is used, though different from that of Chapter 5 (Section 5.1). It should be noted that the analysis of the paper makes the assumption of massless legs. The following three conclusions of Kar et al. (2001) should be considered:

1) Power consumption characteristics improve when legs are at their midstroke.
2) Power consumption characteristics improve when more legs are on the ground.
3) Based on the depicted figure in the paper, the specific energy consumption has more than one local minimum with respect to the duty factor. It should be noted that the analysis is performed with a fixed value of stroke length.

Kar et al. (2001) does not make an analysis on the effect of the stroke length; therefore, it is not possible to observe the interfering effects of duty factor and stroke length in that paper.

Marhefka and Orin (1997) made a more focused analysis of the gait parameters considering energy efficiency. This analysis is also based on a simulative model of a six-legged robot, in which each leg has a three-degrees of freedom. In this paper also, a force distribution scheme that minimizes the power consumption, rather than the tip point force components, is used (Marhefka and Orin, 1998) (again somewhat different from that of Chapter 5, Section 5.1). The analysis is based on observations on the tabulated data obtained from the simulation of a cycle of walk. The effects of duty factor, stroke length, body height, midstroke foothold, and velocity are investigated. The paper concludes the analysis with some "rules" which point to the relation between the average power consumption and the analyzed parameters. The rules are aimed to give an intuitive idea, rather than providing solid strategies, to determine the best parameters. The tabulated data in the paper give the average power consumption during a cycle of walk; in these tables, the authors make no consideration of the traveled distance. To observe the effect of velocity on power dissipation the authors make use of the mechanical "specific resistance", which corresponds to the energy consumption per weight per traveled length. The following can be concluded based on the arguments of the paper:

1) Average power consumption increases with increasing velocity.
2) Specific resistance decreases with increasing velocity.
3) For constant stroke walk, for all the ranges of velocity they observed, average power consumption tends to decrease with increasing duty factor. This is because more legs are supporting when the duty factor is increased. However, for very high duty factors, the legs have to be transferred very fast in the protraction phase. The power required for fast leg transfer may dominate the decreasing effect of increased duty factor; therefore, for a constant stroke length walk, power consumption may increase if the duty factor is increased too much. In their tabulated data, this has occurred only for the case of *0.2m/s* speed walk with the duty factor of *0.8* and stroke length of *0.3m*. For this case the protraction time can be calculated to be *0.375sec.*, which is much less than many realistic situations.
4) For a fixed duty factor, for a constant speed walk, there is an optimum value of the stroke length that minimizes the power consumption. When the stroke length is increased from this optimum value, power consumption increases due to larger deviation from the midstroke position. When it is decreased from the optimum value the power consumption again increases; this time due to the decrease in transfer time in the protraction phase.

The analysis of the lateral offset of the legs and the ratio of the links in Kar et al. (2001), and the body height and midstroke foothold in Marhefka and Orin (1997) do not lead to generalizable conclusions. This is because either the optimum value of these variables are dependent on the robot structure, or other considerations – such as enlargement of workspace – are more dominant than the power consumption in their determination. Therefore, in the work here those structural parameters will not be examined. Rather, the parameters of gait, namely *duty factor, stroke length, phase difference between ipsilateral legs, protraction time, retraction time* and *velocity* will be considered. Similar to Marhefka and Orin (1997), the analysis and derivation of strategies will be based on the tabulated results.

Nishii (1998) performed an analysis of "energetic cost" (dissipated energy per weight per traveled length) with respect to the duty factor and velocity in walking of a two-joint six-legged robot. The analysis is mainly motivated to demonstrate that energy efficiency is the major cause shaping the walking of legged animals; hence, the paper does not provide solid strategies for robotic applications. In their analysis the authors used the optimal stroke length, rather than a fixed stroke length or fixed protraction time, in every case. Therefore, the analysis does not provide simultaneous observation of the interfering effects of the duty factor and stroke length. On the other hand, the results are in agreement with those of the aforementioned two papers: cost of transport decreases with increasing velocity; in low speeds cost of transport decreases with increasing duty factor; but in high speeds the inertia of the leg links start to be significant due to the decreased protraction time, hence the cost of transport increases with increasing duty factor. The paper also provides a graph depicting the optimal duty factor as a function of velocity, provided that the stroke length is again optimized for each value. In this graph, the optimal duty factor decreases with increasing velocity, which is in accordance with the fact that six-legged insects make use of the tripod gait (the minimum duty factor) for fast walking.

Lin and Song (2001) introduced a model for dynamic formulation and force distribution, and made an analysis of stability and energy efficiency with respect to the duty factor and velocity for a quadrupled walking machine. The paper uses a concept of "dynamic stability margin" for stability analysis of statically stable locomotion. The analysis shows that this dynamic stability margin decreases with increasing velocity and decreasing stroke length; it takes a maximum value for an optimal value of the duty factor. The machine used for this paper has massive legs; therefore, the results concerning energy dissipation are dominated by the effect of the high inertia of the legs. Because of this fact, the analysis presents conflicting results to those of the aforementioned three papers: for constant stroke length, "specific resistance" increases with increasing duty factor, and increases with increasing velocity. For a constant stroke length walk, both of increasing duty factor and increasing velocity are equal to decreasing protraction time. Since the inertia of the legs are high, the decrease in protraction time, which means faster transfer of legs, causes a considerable increase in dissipated power. Based on the data provided in their paper the ratio of the mass of one leg to the total mass of the machine is *0.1437*. For the Robot-EA308 used in this thesis, this ratio is as low as *0.086*. (Kar et al. (2001) makes the assumption of massless legs; although the data is not provided in Marhefka and Orin (1997), the ratio there is expected to be close to that of the Robot-

EA308). This difference is intelligible, since a leg in a four-legged machine comprises more of the total mass compared to a leg in a six-legged machine. Consequently, this comparison should reveal that the analysis of energy dissipation and gait parameters is very much dependent on the number of legs and range of velocity. Hence, the analysis here is significant for only the six-legged machines, whose ratio of leg-mass-to-total-body-mass and velocity of walk are around those of the Robot-EA308. The conclusions and strategies of this paper should not arbitrarily be generalized to any number of legs. Lastly, Muraro et al. (2003) made an analysis of energy efficiency for the three dynamically stable (statically unstable) trot, amble and curvet gaits of a quadruped robot, with a virtual leg model. The paper concludes that the curvet gait is the less efficient of the three for all velocities, the amble gait is more efficient in high velocities, and the trot gait is more efficient for low velocities.

In all these reviewed papers the work is based on simulations, and neither of the results are tested on a realistic system. Moreover, in neither of them the aim is to arrive to some solid strategies for finding the best parameters of the gait for the most energy efficient walk. This chapter aims to fill the gap in these two regards. In this work, a concentrated energy efficiency analysis of wave gaits on a three-joint six-legged robot is performed. The analysis intends to arrive to some strategies that will be guiding to determine the parameters of wave gaits for the most energy efficient walk. In developing these strategies, a phase modification in the wave gaits is proposed. This deviation from the wave gaits, which are optimally stable, results in a loss of the stability margin. Therefore, the analysis of stability of standard gaits, performed in Chapter 2, will be extensively used to observe the amount of loss with the proposed modification. The resultant strategies are justified on the realistic data obtained from an actual experimental setup. This setup is composed of the three-joint six-legged Robot-EA308, a controlling software, a computer that communicates with the robot, and the data management system that records the data of the energy consumption of the robot while walking.

In the following section, an analysis of energy efficiency of wave gaits is performed based on the simulation model of the Robot-EA308, by tabulating the energy consumption with respect to duty factor and velocity. Based on these tables strategies are derived in the third section. In the fourth section similar tables are constructed using the actual experimental setup. The strategies derived from the simulation data are justified on this realistic data. The last section conclude the work.

6.2. Energy Efficiency Analysis of Gaits Based on Simulation Data

For robot applications one of the most challenging problems is to achieve a task with minimum energy consumption. In fact, often, the total energy consumption for a given task is more important than the average power consumption. It may be possible that a task is accomplished in a shorter time with higher average power consumption, and the total dissipated energy happens to be less. Therefore, the energy consumption per length of travel is more relevant considering the multi-legged walking. In Kar et al. (2001), Nishii (1998), and Lin and Song (2001) the criterion of "the energy per weight per traveled length" is considered. In Marhefka and Orin (1997), average power consumption is depicted on the tables and "energy per weight per traveled length" is considered for the variations in velocity.

In Marhefka and Orin (1997) a model of a DC motor with a gear system is used in order to obtain the power consumption from the required current and voltage values. The power variables of this paper are therefore dependent on the motor-gear model. In Kar et al. (2001) and Lin and Song (2001), instantaneous power is taken to be "the product of instantaneous joint torques and joint velocities". Such modeling ignores the fact that a considerable amount of power is dissipated on the joints of the supporting legs even though the joints are not moving. In order to eliminate such limitations it is a common approach to consider "the integral of the sum of the squares of the joint-torques" as a criterion of dissipated energy in the actuators of robotic manipulators (Bobrow et al., 2001; Garg et al., 2002; Liu et al., 2000). Accordingly, Nishii (1998) used the integral of the weighted sum of "the product of instantaneous joint torques and joint velocities" and "the sum of the squares of the joint-torques" as the energetic cost; and Muraro et al. (2003) used the criteria of "the integral of the square of the joint-torques per traveled length" and "the energy loss due to Joule effect per traveled length" separately. For the simulations in this chapter, "the integral of the sum of the squares of the joint-torques", both of retracting and protracting legs, is used to correspond to the energy dissipation. Then, the criterion of "dissipated energy per length" is chosen to be minimized, and it is denoted by the term *Ecpl.* (energy criterion per length). (The criterion is not divided by the total weight of the robot *(2.03kg)* as in the other papers.)

The simulation is performed in MATLAB and its SIMULINK environments. The trajectories of all the joints of the legs, both supporting and returning, are generated for a cycle of walk for the given gait parameters. Then using these joint trajectories the overall model of the six-legged Robot-EA308 is simulated. In this simulation, the distribution of necessary forces and moments to the supporting legs is performed in an energy optimal way by the "torque distribution" scheme introduced in Chapter 5. The friction coefficient between the tip points of the supporting legs and ground is taken to be *0.3*.

6.2.1. Relations Between the Gait Parameters

Before going into the analysis, it is worth mentioning the mathematical relations between the investigated gait parameters (duty factor (β), stroke length (R), ipsilateral phase difference (ϕ), protraction time (t_p), retraction time (t_r), and velocity (v)). The relation between the protraction time and retraction time is determined by the duty factor: $\beta=t_r/(t_r+t_p)$. Increase in duty factor results in the increase of the retraction time, or decrease of the protraction time. With high duty factors the leg has to have faster transfer in the return phase. The ipsilateral phase difference, ϕ, is an independent variable. The velocity is given by the relation: $v=R/t_r$. Based on these equations the stroke length is given by $R=vt_p\beta/(1-\beta)$. For a constant stroke length and constant velocity, the protraction time decreases with increasing duty factor. If the velocity and protraction time are held constant, the stroke length increases with increasing duty factor. These relations will be helpful in the following analyzes.

The observations follow four steps; each step is performed on a particular table of data. The first step examines the most generally applied walk, namely constant stroke wave gaits. The second step introduces the phase modified-wave gaits in order to minimize the dissipated energy. In the third and fourth steps the wave gaits and phase modified wave gaits, respectively, are observed for the case of constant protraction time.

6.2.2. Observations for the Constant Stroke – Wave Gaits

The first observation is for the constant stroke walk, with the stroke length of *0.14m*, which is the maximum for the Robot-EA308. This kind of walk is very much used in multi-legged robot applications due to its simplicity regarding to trajectory generation for protraction and retraction phases. In Table 6.1, the *Ecpl.* values are given with respect to the duty factor and velocity. In constant stroke walk, for a given velocity the retraction time is constant. The protraction time, on the other hand, decreases with increasing duty factor. The actual mechanical and control systems have a limitation that the protraction time cannot be decreased without limit. For the Robot-EA308, the minimum proper protraction time is *1.5sec*. When this minimum is exceeded the protraction becomes improper; namely, the legs are commanded too fast, so that they are not raised enough above the ground, and the commanded trajectory is not properly followed. If it is exceeded more and more, the protraction is totally unacceptable, in the sense the legs are not lifted from the ground and the walk of the robot is impossible. For simulations, however there is no such limitation; therefore, the data in Table 6.1 is extended to the regions that cannot be properly handled by the actual Robot-EA308. In Table 6.1, the region with the white background corresponds to the proper protraction for the actual Robot-EA308. The region with the gray background corresponds to protraction times between *1.5* and *0.5sec*. In all the tables demonstrating the *Ecpl.* values, the minimums for the particular velocities (in one column) are written in bold.

The first striking observation is the decrease of *Ecpl.* with increasing velocity. This is in accordance with the observation in Marhefka and Orin (1997). It should be noted that if the *Ecpl.* values are multiplied by the velocity, the outcome is the energy dissipation per time, i.e. the power dissipation. If this operation is performed on Table 6.1, it will be observed that power dissipation increases with increasing velocity, which is again in accordance with the tabulated data in Marhefka and Orin (1997). In the following three tables, the relations between the *Ecpl.* values and velocity, and the power dissipation and velocity will remain the same. Therefore, this observation will not be repeated.

In order to observe the relation between the *Ecpl.* values and the duty factor, the table is separated into two regions with a thick line. Above the thick line is the region for duty factors less than two third (*β≤2/3*). In this region, for every velocity, the *Ecpl.* values decrease continuously with increasing duty factor, without any exception. Therefore, it can be stated that energy per traveled length decreases continuously with increasing duty factor for constant stroke wave gaits in the region *β≤2/3*.

Table 6.1: *Ecpl.* values for wave gaits with constant stroke of *0.14m*. (*Duty factor (df)* versus *velocity (vel, m/sec)*).

df / vel	**0,005**	**0,010**	**0,015**	**0,020**	**0,025**	**0,030**	**0,035**	**0,040**	**0,045**	**0,050**	**0,055**	**0,060**	**0,065**	**0,070**	**0,075**	**0,080**	**0,085**	**0,090**	**0,095**	**0,100**
0,500	90.9465	45.4914	30.7886	22.8744	18.2077	15.4452	12.9644	11.3256	10.2042	9.1894	8.3156	7.6610	7.0687	6.5435	6.0884	5.7247	**5.3381**	**5.0954**	4.8562	4.5873
0,525	76.3919	37.3877	26.0520	19.0676	15.6804	12.7877	10.9192	9.4132	8.3352	7.6872	6.9113	6.4103	5.7466	5.4603	**5.0259**	**4.8876**	4.4641	4.1409	3.9971	3.8669
0,550	63.0653	31.5824	20.6389	15.4492	12.3650	10.3750	8.9610	7.8019	6.9385	6.2786	5.7596	5.2755	4.9605	**4.4820**	4.2984	4.0288	3.6786	3.5363	3.3102	3.1961
0,575	53.5772	26.7167	17.8631	13.3326	10.6775	8.7303	7.6850	6.7051	5.9590	5.3875	4.7711	4.3727	**4.0392**	3.8591	3.5562	3.3894	3.2407	2.9121	2.8644	2.7366
0,600	46.2489	23.1154	15.5652	11.5741	9.3178	7.7263	6.6252	5.8510	5.2055	4.6376	4.2596	**3.8411**	3.5982	3.3287	3.0712	2.9396	2.7776	2.6166	2.4859	2.3463
0,625	41.7924	21.1999	13.9329	10.4746	8.4613	7.0433	6.0452	5.3063	4.6951	4.2637	**3.8629**	3.5491	3.2784	3.0740	2.8290	2.6815	2.5413	2.3821	2.2903	2.1977
0,650	39.6800	19.9080	13.3476	9.9231	7.9347	6.6009	5.7088	5.0340	4.4177	**3.9365**	3.6964	3.3959	3.0893	2.9186	2.7079	2.5598	2.4136	2.2552	2.1902	2.0426
0,675	38.9310	19.4910	12.9711	9.7759	7.8259	6.5492	5.6607	4.9090	**4.3612**	3.9019	3.5310	3.2747	3.0011	2.7810	2.6338	2.4661	**2.2817**	2.1671	2.1273	2.0734
0,700	38.9195	19.4703	13.0105	9.6729	7.7551	6.4910	5.5456	**4.8767**	4.3612	3.8574	3.5432	**3.2164**	**2.9837**	**2.7685**	**2.5832**	2.4482	2.2836	**2.1638**	**2.0325**	**1.9413**
0,725	38.5751	19.4020	12.9741	**9.6448**	**7.7512**	**6.4314**	**5.5441**	4.8918	4.3055	**3.8445**	3.5954	3.3050	3.0270	2.7877	2.6578	2.4826	2.3843	2.2224	2.1149	2.0287
0,750	38.7579	19.3288	13.1571	9.6895	7.7679	6.5146	5.6610	4.8466	4.3344	3.9073	3.7269	3.3015	3.0729	2.7873	2.6645	**2.4461**	2.3232			
0,775	39.0983	19.4413	13.1319	9.7735	7.9029	6.4825	5.5690	4.8717	4.3761	3.9596	3.5957	3.2587	3.1346	2.9019	2.7324					
0,800	38.5708	19.0102	12.8933	9.7213	7.5961	6.4577	5.5043	4.8974	4.3562	3.9064	3.5547	3.3145	3.1151	2.7998						
0,825	38.0073	19.1491	12.7162	9.5112	7.6688	6.4404	5.4567	4.8715	4.3201	3.8737	**3.4777**									
0,850	37.5388	18.7104	**12.4537**	9.2730	7.6695	6.2667	5.4036	4.8890	**4.2209**											
0,875	36.7854	18.6948	12.2428	9.1738	7.4402	6.1918	**5.4221**	**4.6965**												
0,900	35.2898	**17.6491**	11.9877	8.9451	**7.2691**	**6.0600**														
0,925	**34.7462**	17.0492	**11.3652**	**8.4927**																
0,95	**33.3278**	**16.1290**																		

Below the thick line is the region for duty factors larger than two third (*β>2/3*). In this region the *Ecpl.* values for a given velocity do not demonstrate a regular behavior; rather, some waves of decrease and increase are observed (for example, observe the increases and decreases for the case of *vel=0.04m/s*). Such an irregular behavior was observed also in Marhefka and Orin (1997), as mentioned as the third item in the relevant review. The comment of Marhefka and Orin (1997) for this behavior was that, the decrease in protraction time increases the required power to move the leg faster in the return stroke. Although this might be the reason for systems with comparatively massive legs, here will be proposed another explanation. It will be demonstrated that the proposed explanation, rather than the effect of decreased protraction time, will provide the dominant reason for the irregularity. Moreover, this explanation will lead to a modification on the wave gaits in order to achieve more energy efficient walk.

In Figure 6.1, the stroke trajectories of the left side legs are given for the duty factor of *0.875*. Because of the increased duty factor the protraction time takes a very small fraction (*0.125*) of the step cycle. Due to the characteristics of wave gaits the protractions take place one after the other as a wave from rear to front. As a result, the protraction of three legs are concentrated on less than a half cycle of the whole period. This configuration results in durations in which all the supporting legs are in either the first or the second half of their retraction phases, hence they are supporting with their tip-points being either all on the first or all on the second half of the stroke line. For example, in the encircled region in Figure 6.1, all the legs have passed their mid-stroke positions and are supporting in the second half of their stroke lines. Since a big and constant stroke length is used, the deviation from the mid-stroke means a lot of power dissipation for the support. For the situation in Figure 6.1, all the left legs are supporting, but in an inefficient way. This effect will be designated as *inefficient phase distribution*. The reason of the irregular behavior of the *Ecpl.* values for increased duty factor might be the interference of "the decreasing effect of the increased duty factor" with "the increasing effect of the inefficient phase distribution". If this proposal is correct, or at least if it captures the dominant effect, the wavy behavior should disappear once the inefficient phase distribution is overcome.

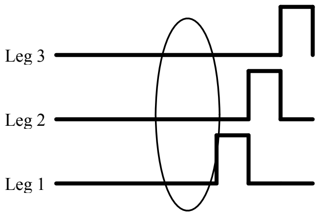


Figure 6.1: Stroke trajectory for the left side legs for duty factor *0.875*. Low level: power stroke (retraction); high level: return stroke (protraction).

To overcome the inefficient phase distribution it is necessary to make a slight modification on the wave gait ipsilateral phases given in (2.2). In wave gaits, the ipsilateral phase differences are determined according to the duty factor: $\phi=1-\beta$. In order to distribute the protraction phases of the same side legs equally, the ipsilateral phase differences should be chosen to be 1/3: $\phi=1/3$. However, this cannot be made for the duty factors less than 2/3; otherwise, the condition for the rule of neighborhood is not satisfied, and the gait is not guaranteed to be stable. In fact, there is already no necessity for modification for the range $\beta\leq 2/3$; because, the effect of inefficient phase distribution is not observed in this range. The resulting gait pattern will be designated by *phase modified wave gait* and the phase distribution is given as follows:

$$\begin{aligned}
&\psi_3=t && \psi_4=F[t+1/2] \\
&\psi_2=\begin{Bmatrix} F[t-\beta], & \beta\leqslant 2/3 \\ F[t+1/3], & \beta>2/3 \end{Bmatrix} && \psi_5=\begin{Bmatrix} F[t-\beta+1/2], & \beta\leqslant 2/3 \\ F[t+5/6], & \beta>2/3 \end{Bmatrix} \\
&\psi_1=\begin{Bmatrix} F[t-2\beta], & \beta\leqslant 2/3 \\ F[t+2/3], & \beta>2/3 \end{Bmatrix} && \psi_6=\begin{Bmatrix} F[t-2\beta+1/2], & \beta\leqslant 2/3 \\ F[t+1/6], & \beta>2/3 \end{Bmatrix}
\end{aligned} \tag{6.1}$$

6.2.3. Observations for the Constant Stroke - Phase Modified Wave Gaits

With the phase modification of (6.1), only the region of $\beta>2/3$ will be affected. Therefore, for the phase modified wave gait tables, only the region for $\beta>2/3$ will be given. In Table 6.2, again the region with white background corresponds to protraction time larger than *1.5sec.*, and the region with gray background corresponds to protraction time between *1.5* and *0.5sec.*. As it is observed, the irregular behavior in Table 6.1 disappeared in Table 6.2: the *Ecpl.* values continuously decrease with increasing duty factor for all the velocities. This observation proves that the irregularity in Table 6.1 was determined dominantly by the inefficient phase distribution, rather than the decrease of protraction time. Considering the combination of the upper part of Table 6.1 with Table 6.2, it can be stated that *energy per traveled length decreases continuously with increasing duty factor for constant stroke – phase modified wave gaits.*

Moreover, for all combinations of duty factor and velocity, the values in Table 6.2 are less than the values in Table 6.1. This points to the observation that *constant stroke – phase modified wave gaits are more energy-efficient than constant stroke – wave gaits*. In Table 6.3, the best *Ecpl.* values applicable to the Robot-EA308 for the wave gaits and phase modified wave gaits are depicted. As it is observed the energy dissipation is decreased by up to 30% in low velocities. For higher velocities the energy decrease by shifting to the phase modified wave gaits is less significant, although still on a positive level. The reason is that the best applicable *Ecpl.* values for higher velocities occur with higher duty factors (close to *0.675*), where the effect of the inefficient phase distribution is weak.

As a result of the continuous decrease with increasing duty factor, the minimum *Ecpl.* values of the *constant stroke – phase modified wave gaits* occur on the extreme regions determined by the minimum protraction time (observe the regularity of the placing of the minimums). Therefore, considering the regions allowed by the physical system, it can be stated that *the less the protraction time, the better the energy efficiency is* (this is proved by the given data at least for the range of protraction time larger than *0.5* seconds for the simulated mechanical configuration of the Robot-EA308). In the next two subsections the analysis of wave gaits and phase modified wave gaits will be performed with the minimum applicable protraction time. But before that, the phase modified wave gaits need to be compared with the wave gaits considering the variation in stability margin.

The mentioned phase modification results in a SGSRN (Chapter 2). Therefore, the phase modified wave gaits are guaranteed to be stable. However, as will be remembered, wave gaits are the most stable among the standard gaits. Therefore, any modification on a wave gait, provided that the new gait is standard, results in a less stable gait. This is the case for the mentioned modification, and it matters how much of the stability is lost. Based on (2.8), the stability margin of a phase modified wave gait for the region of $\beta>2/3$ is given as follows:

$$S_{r1}=\frac{1}{2}[(2n-4)P/2-R+(R/\beta)(2\beta-\frac{5}{6})] \tag{6.2}$$

Table 6.2: *Ecpl.* values for phase modified wave gaits with constant stroke of *0.*14m. *(Duty factor(df) versus velocity(vel, m/sec).)*

df / vel	0,005	0,010	0,015	0,020	0,025	0,030	0,035	0,040	0,045	0,050	0,055	0,060	0,065	0,070	0,075	0,080	0,085	0,090	0,095	0,100
0,675	38.7687	19.2797	12.8975	9.6444	7.7183	6.4476	5.5407	4.8145	**4.2894**	3.9148	3.5656	3.2121	2.9840	2.8539	2.5665	2.5327	2.2779	2.1900	2.0893	1.9247
0,700	36.5597	18.4139	12.2563	9.0092	7.3482	6.1193	5.2534	**4.5582**	4.0444	3.6464	3.3782	3.0049	2.8636	2.6227	2.4556	2.3003	2.1611	2.0914	1.9742	1.8672
0,725	34.8662	17.1777	11.6114	8.6684	6.9752	5.8103	**5.0032**	4.3040	3.8575	3.5055	3.1996	2.9231	2.7015	2.5247	2.3141	2.1875	2.0878	**1.9893**	**1.8704**	**1.9069**
0,750	33.2715	16.5039	10.9178	8.1874	6.5659	**5.5367**	4.6992	4.1014	3.7001	3.3105	3.0077	2.8640	2.5207	2.4385	2.2415	**2.1385**	**1.9990**			
0,775	30.8783	15.5607	10.4502	7.7800	**6.2560**	5.1922	4.4817	3.9331	3.4929	3.1653	2.9083	2.6215	2.5060	2.2871	**2.1839**					
0,800	29.6924	14.7936	9.9304	**7.3813**	5.9145	4.9325	4.2416	3.7046	3.3162	3.0062	2.7115	**2.5409**	**2.3768**	**2.2167**						
0,825	28.0506	14.2191	9.4938	6.9961	5.5687	4.7359	4.1219	3.6240	3.1888	**2.8529**	**2.6544**									
0,850	26.8125	13.5292	**8.8942**	6.6882	5.4009	4.5147	3.8827	3.4551	**3.0396**											
0,875	25.3060	12.7222	8.4559	6.3654	5.1385	4.3144	**3.7457**	**3.3531**												
0,900	24.0414	**12.1179**	8.0417	6.0414	**4.8426**	**4.0946**														
0,925	**23.0009**	11.5218	**7.8109**	**5.9222**																
0,95	**21.9988**	**10.9077**																		

Table 6.3: Comparison of the best *Ecpl.* values applicable to the Robot-EA308 for the constant stroke – wave gaits and constant stroke – phase modified wave gaits.

velocity (m/sec)	constant stroke wave gait best Ecpl.	constant stroke phase modified wave gait best Ecpl.	%decrease
0,005	34,7462	23,0009	33,8031
0,010	17,6491	12,1179	31,3398
0,015	12,4537	8,8942	28,5819
0,020	9,6448	7,3813	23,4686
0,025	7,7512	6,2560	19,2899
0,030	6,4314	5,5367	13,9114
0,035	5,5441	5,0032	9,7563
0,040	4,8767	4,5582	6,5311
0,045	4,3612	4,2894	1,6463

Table 6.4: Comparison of stability margins for the constant stroke – wave gaits and constant stroke – phase modified wave gaits.

df	constant stroke wave gait stab. marg.	constant stroke phase modified wave gait stab. marg.	%decrease
0,5000	0,0150	0,0150	0,0000
0,5250	0,0250	0,0250	0,0000
0,5500	0,0341	0,0341	0,0000
0,5750	0,0424	0,0424	0,0000
0,6000	0,0500	0,0500	0,0000
0,6250	0,0570	0,0570	0,0000
0,6500	0,0635	0,0635	0,0000
0,6750	0,0694	0,0686	1,2444
0,7000	0,0750	0,0717	4,4444
0,7250	0,0802	0,0745	7,0251
0,7500	0,0850	0,0772	9,1503
0,7750	0,0895	0,0797	10,9309
0,8000	0,0938	0,0821	12,4444
0,8250	0,0977	0,0843	13,7468
0,8500	0,1015	0,0864	14,8792
0,8750	0,1050	0,0883	15,8730
0,9000	0,1083	0,0902	16,7521
0,9250	0,1115	0,0919	17,5354
0,9500	0,1145	0,0936	18,2375

This stability margin is the same as that of the wave gait (2.9) if $\beta=2/3$, and smaller than that of the wave gait if $\beta>2/3$. In Table 6.4, the stability margins for the constant stoke-wave gaits and constant stroke-phase modified wave gaits are given with respect to the duty factor. For these constant stroke gaits, the stability margin does not change with changing velocity. As it is observed, the decrease in stability margin is less than 20% for all the duty factors. Therefore, it can be concluded that by passing to the constant-stroke phase modified wave gaits the increase in energy efficiency happens in the expense of no significant stability loss.

Table 6.5: *Ecpl.* values for wave gaits with constant protraction time of *1.5sec. (Duty factor(df) versus velocity(vel, m/*sec*).)*

df / vel	**0.005**	**0.010**	**0.015**	**0.020**	**0.025**	**0.030**	**0.035**	**0.040**	**0.045**	**0.050**	**0.055**	**0.060**	**0.065**	**0.070**	**0.075**	**0.080**	**0.085**	**0.090**	**0.095**	**0.100**
0.500	42.2655	21.2429	14.3716	11.0182	9.0570	7.7748	6.8677	6.2317	5.7428	5.4397	5.2345	5.0309	4.8844	4.8432	4.7539	4.7525	**4.8627**	**4.8913**	4.9004	5.0854
0.525	40.1280	20.0722	13.7949	10.4573	8.4034	7.2110	6.3892	5.8340	5.4126	5.0557	4.8572	4.7143	4.5269	4.4548	4.4573	**4.4724**	4.4742	4.5812	4.6943	
0.550	38.6626	19.1255	12.8605	9.8283	7.9179	6.7839	6.0598	5.4263	5.0763	4.7452	4.4759	4.2988	4.2053	**4.0720**	**4.0929**	4.0961	4.0923	4.2790		
0.575	36.6796	18.4996	12.0584	9.2711	7.5819	6.3904	5.6050	5.1257	4.6185	4.3303	4.1292	3.9781	**3.8936**	3.9257	3.8671	3.9407				
0.600	34.9793	17.3802	11.7762	8.8579	7.1779	6.0572	5.3180	4.9063	4.4483	4.1091	3.9756	**3.7643**	3.7061	3.6295	3.6315					
0.625	33.0379	16.8756	11.0536	8.3937	6.8264	5.7740	5.1682	4.6220	**4.2033**	**3.9801**	**3.7884**	3.7122	3.5778							
0.650	31.0336	15.6570	10.3979	7.9199	6.5988	**5.5449**	**5.0129**	**4.6045**	4.2689	4.0082	3.8122	3.7615								
0.675	29.3503	14.9678	10.1518	7.7596	**6.3591**	5.5719	5.0411	4.6311	4.3743	4.1910										
0.700	28.4611	14.3514	9.9711	**7.6511**	6.4556	5.6942	5.1742	4.8767	4.6558											
0.725	27.2374	14.1026	**9.8149**	7.7953	6.7274	5.9641	5.5098	5.2456												
0.750	26.2926	13.7914	10.0017	8.0888	7.0577	6.3713	5.9847													
0.775	25.9336	13.6971	10.0479	8.1994	7.3566	7.0172														
0.800	24.7297	**13.6817**	10.1809	8.7275	8.1696															
0.825	24.3944	13.8034	10.6946	9.6199																
0.850	23.5268	14.2456	11.6988	11.5210																
0.875	**22.9857**	14.9350	13.7126																	
0.900	23.3299	17.0970																		
0.925	25.2384																			
0.950	32.0959																			

6.2.4. Observations for the Constant Protraction Time – Wave Gaits

In this section the wave gaits are analyzed with respect to the duty factor and velocity with a constant protraction time, which is taken to be *1.5sec.*, the minimum applicable to the Robot-EA308. Since the protraction time is constant, the retraction time is determined by the duty factor, and then the stroke length is determined by the velocity. Namely, the stroke length changes with changing duty factor and velocity, and the maximum possible stroke length will be the limiting factor of the following constant protraction time analyzes. The physically applicable maximum stroke length to the Robot-EA308 is *0.14m*. In the constant protraction time tables the region with white background corresponds to stroke length less than this maximum ($st_l \leq 0.14m$). However, within the simulation environment the analysis is performed also for the range where the stroke length is larger than the applicable maximum but smaller than the pitch length ($0.14m < st_l \leq 0.17m$). The region with gray background corresponds to this range.

In Table 6.5, it is observed that, for all velocities, the change of the *Ecpl.* values with increasing duty factor has a regular characteristic: it first decreases, makes a minimum, and then increases. The minimum applicable *Ecpl.* values for constant protraction time wave gaits are observed to be less than those of the constant stroke wave gaits, except for the velocity of *0.05m/s*. (For this single case, the minimum applicable *Ecpl.* value for constant protraction time wave gait (*3.9801*) is only slightly larger than that of the constant stroke wave gait (*3.9365*).) It should be noted that the decrease in the *Ecpl.* values before the minimum and the increase after are both continuous. This regular behavior points that there is a dominating factor other than the inefficient phase distribution for the constant protraction time wave gaits: obviously it is the stroke length. In the regions where the stroke length is small, the increase of the duty factor is the dominant factor in decreasing behavior of the *Ecpl.* values; whereas, in the regions where the stroke length is large the increase of the stroke length becomes the dominant factor in the increasing behavior of the *Ecpl.*

values. Although the inefficient phase distribution is not dominant in the constant protraction time wave gaits, the experience of the former analysis reveals that there can be further improvement in the high duty factor regions by shifting to the phase modified wave gaits. The next section will analyze this case.

6.2.5. Observations for the Constant Protraction Time – Phase Modified Wave Gaits

In Table 6.6, the *Ecpl.* values are given for constant protraction time – phase modified wave gaits for the region of $\beta>2/3$. The region with white background again corresponds to stroke length less than the maximum applicable ($st_l \leq 0.14m$). The region with gray background corresponds to stroke length greater than the maximum applicable but smaller than the pitch length ($0.14m<st_l \leq 0.1mm$). The same regular behavior of the *Ecpl.* values in Table 6.5 is observed in Table 6.6; but, the minimums occur with higher duty factors. It is further observed that the minimum applicable *Ecpl.* values in Table 6.6 are the smallest of the ones in all the previous tables. This leads to the conclusion that *the minimum Ecpl. is achieved with the constant protraction time – phase modified wave gaits*.

In Table 6.7, a comparison of the best *Ecpl.* values applicable to the Robot-EA308 for the constant stroke length – wave gaits and constant protraction time – phase modified wave gaits is given. In this table the region with gray background corresponds to the range where $\beta \leq 2/3$; namely, there is no phase modification for these parts and the values for the constant protraction time – phase modified gaits are taken from the upper part of Table 6.5. The region with white background corresponds to the best values in Table 6.6, where $\beta>2/3$. Similarly, a comparison of the stability margins for the best applicable *Ecpl.* cases for the constant stroke length – wave gaits and constant protraction time – phase modified wave gaits is given in Table 6.8. It can be observed that there is a significant energy gain in the phase modified region, and a moderate energy gain in the other region (a slight loss occurs for the velocity of *0.05m/sec.*). The stability margin loss is less than 20% in all cases; there occur significant stability gains in many cases (especially in the part for $\beta \leq 2/3$). Therefore in general, it can be stated that the shift to constant protraction time – phase modified wave gaits from the constant stroke – wave gaits results in more energy efficient gaits, with negligible exceptions, with tolerable stability losses.

Table 6.6: *Ecpl.* values for phase modified wave gaits with constant protraction time of *1.5sec. (Duty factor(df) versus velocity(ve, m/secl).)*

df / vel	0.005	0.010	0.015	0.020	0.025	0.030	0.035	0.040	0.045	0.050	0.055
0.675	29.2677	14.9340	10.0498	7.7058	6.3491	5.5109	4.9620	4.5771	**4.3112**	4.0891	
0.700	28.2762	14.3165	9.7094	7.5409	6.2100	5.4384	**4.9425**	**4.5583**	4.3958		
0.725	27.1517	13.6565	9.3537	7.2051	6.1657	5.4129	4.9732	4.6562			
0.750	26.1627	13.3115	9.0469	7.0747	6.0525	**5.3594**	5.0497				
0.775	25.1796	12.9967	8.8280	6.9937	**6.0182**	5.4750					
0.800	24.2676	12.3289	8.6624	**6.9352**	6.1848						
0.825	23.1485	11.9904	**8.4564**	7.0542							
0.850	22.3879	11.6679	8.5850	7.5094							
0.875	21.4573	**11.3867**	9.1215								
0.900	20.4646	11.9293									
0.925	**20.2677**										
0.950	22.5439										

Table 6.7: Comparison of the best *Ecpl.* values applicable to the Robot-EA308 for the constant stroke length wave gaits and constant protraction time phase modified wave gaits.

velocity (m/sec)	constant stroke wave gait best Ecpl.	constant prot. time phase mod. wave gait best Ecpl.	%decrease
0.005	34.7462	20.2677	41.6693
0.010	17.6491	11.3867	35.4828
0.015	12.4537	8.4564	32.0973
0.020	9.6448	6.9352	28.0939
0.025	7.7512	6.0182	22.3578
0.030	6.4314	5.3594	16.6682
0.035	5.5441	4.9425	10.8512
0.040	4.8767	4.5583	6.5290
0.045	4.3612	4.3112	1.1465
0.050	3.9365	3.9801	-1.1076
0.055	3.8629	3.7884	1.9286
0.060	3.8411	3.7643	1.9994
0.065	4.0392	3.8936	3.6047
0.070	4.4820	4.0720	9.1477
0.075	5.0559	4.0929	19.0471
0.080	4.8876	4.4724	8.4950
0.085	5.3381	4.8627	8.9058
0.090	5.0954	4.8913	4.0056

Table 6.8: Comparison of the stability margins for the best applicable *Ecpl.* cases for the constant stroke length wave gaits and constant protraction time phase modified wave gaits.

velocity (m/sec)	constant stroke wave gait stab. marg	constant prot. time phase mod. wave gait stab. marg	%decrease
0.005	0.1115	0.0896	19.6413
0.010	0.1083	0.0875	19.2059
0.015	0.1015	0.0845	16.7488
0.020	0.0802	0.0825	-2.8678
0.025	0.0802	0.0801	0.1247
0.030	0.0802	0.0775	3.3666
0.035	0.0802	0.0733	8.6035
0.040	0.0750	0.0717	4.4000
0.045	0.0694	0.0686	1.1527
0.050	0.0635	0.0600	5.5118
0.055	0.0570	0.0575	-0.8772
0.060	0.0500	0.0512	-2.4000
0.065	0.0424	0.0449	-5.8962
0.070	0.0341	0.0383	-12.3167
0.075	0.0250	0.0350	-40.0000
0.080	0.0250	0.0282	-12.8000
0.085	0.0150	0.0213	-42.0000
0.090	0.0150	0.0175	-16.6667

6.3. Heuristic Strategies to Find an Energy Efficient Standard Gait

In development of the strategies in this section the exceptional case that is not fitting to the general trend, the case of velocity *0.05m/s* in Table 6.7, will be ignored. This approach is acceptable since the violation of the general trend does not occur in a significant degree. The general conclusion is that, the gaits with minimum protraction time result in more efficient walk than the gaits with constant stroke length; moreover, in each case the walk is more efficient if the phase modified wave gait is used instead of the wave gait. The most energy efficient walk is achieved with phase modified wave gaits where protraction time is fixed to the minimum. The second and third strategies will handle this observation. But before that, the first strategy will deal with the relation between energy efficiency and velocity.

Strategy 1: If the velocity is free and the aim is to travel a given distance with minimum energy dissipation, choose the maximum possible velocity. On the other hand, if the velocity is free but the aim is to dissipate the minimum power, choose the minimum possible velocity.

Strategy 2: If the stroke length is allowed to be varied, choose the minimum possible protraction time to be applied.

Strategy 3: If it is allowed, switch to the phase modified wave gait in order to achieve more efficient walk for $\beta>2/3$.

Depending on the applicability of strategies two and three, one should make a search on a hypothetical table like Table 6.1, 6.2, 6.5, or 6.6. (It should be remembered that the upper parts ($\beta\leq 2/3$) of the Tables 6.2 and 6.6 are the same as the ones in Tables 6.1 and 6.5, respectively; therefore, these regions are not shown on Tables 6.2 and 6.6.) *Strategy 1* will determine the velocity. Then, the search will aim to find the best duty factor in the column of the hypothetical table for the given velocity. The following strategies for the search of the best duty factor will depend on the kind of the hypothetical table determined by the second and third strategies.

Strategy 4.i: If the constant stroke – wave gait is to be applied, the hypothetical table is like Table 6.1. For the given velocity, if the maximum applicable duty factor is less than two over three ($\beta\leq 2/3$), use that maximum as the duty factor. Otherwise, a search is required in the region $2/3<\beta\leq 1$. Since there is no regular behavior of the *Ecpl.* values in this region, all the possible duty factors should be checked, and the one resulting in the minimum *Epl.* (energy per length) should be chosen.

Strategy 4.ii: If the constant stroke – phase modified wave gait is to be applied, the hypothetical table is like Table 6.2, with the upper part given in Table 6.1. In this table the *Ecpl.* values continuously decrease with increasing duty factor; therefore, for the given velocity choose the maximum applicable duty factor.

Strategy 4.iii: If the constant protraction time – wave gait is to be applied, the hypothetical table is like Table 6.5. In this table the *Ecpl.* values have a regular behavior of first decreasing and then increasing with respect to the increasing duty factor. The minimum *Ecpl.* value is somewhere in between, but mostly close to the maximum applicable duty factor. Therefore, start from the maximum applicable duty factor and increase the duty factor till a local minimum of the *Epl.* value, which is also the global minimum of the associated column, is found. Choose the duty factor corresponding to this minimum *Epl.* value.

Strategy 4.iv: If the constant protraction time – phase modified wave gait is to be applied, the hypothetical table is like Table 6.6, with the upper part given in Table 6.5. In this table also the *Ecpl.* values have a regular behavior of first decreasing then increasing with respect to the increasing duty factor. The minimum *Ecpl.* value is somewhere in between, but mostly close to the maximum applicable duty factor. Therefore, start from the maximum applicable duty factor and increase the duty factor till a local minimum of the *Epl.* value, which is also the global minimum of the associated column, is found. Choose the duty factor corresponding to this minimum *Epl.* value.

6.4. Testing the Strategies on the Experimental Setup

Experimental studies are performed using the Robot-EA308. The energy consumption of the Robot-EA308 is recorded while walking with different gait plans and tabulated in tables. The mechanical six-legged system can be controlled with a computer program via the servo-motor controllers on the joints. The computer program is developed in the MATLAB and its SIMULINK environments, and communication is established via the serial port of the computer. Once the joint trajectories corresponding to a particular gait plan are developed in the computer program, they can be sent to the robot to achieve the desired walk. The servo-motors of the robot are fed by an approximately *7volt* DC current power supply. A data management system records the instantaneous voltage and current, and hence the dissipated power, in every *0.1sec*. The average energy per meter length of travel corresponding to the gait pattern is obtained by dividing the integral of the power by the covered distance. Therefore, for the case of the actual system, the data corresponds to actual dissipated energy per travelled length, which is denoted by *acEpl.* (actual energy per length) and has the unit of *joule/meter*. For each gait pattern, the robot is made to walk for six cycles of the period of the gait. In Figure 6.2 a photo of the overall experimental setup is given.

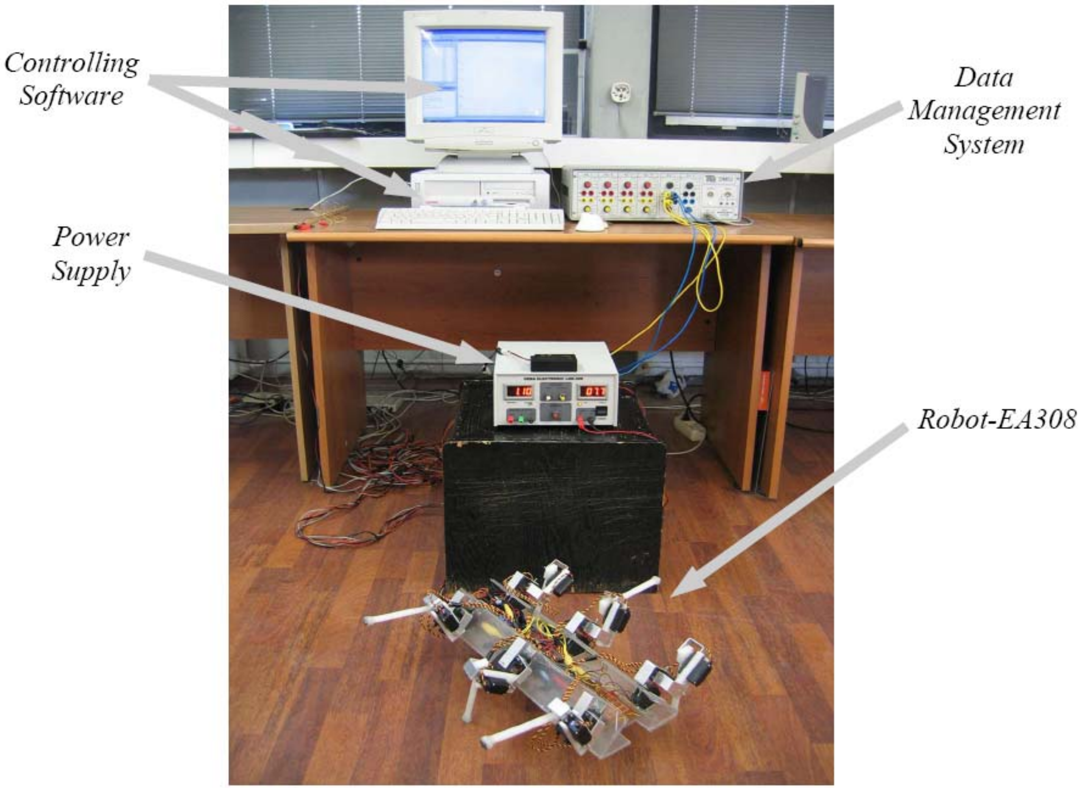


Figure 6.2: Experimental setup for recording energy dissipation of the Robot-EA308 while walking.

Table 6.9: The *acEpl*. values for different gait patterns for the Robot-EA308. The unit of *acEpl.* values is *joule/meter*. *(a)* Constant stroke – wave gaits; *(b)* constant stroke – phase modified wave gaits; *(c)* constant protraction time – wave gaits; *(d)* constant protraction time – phase modified wave gaits.

df / vel	0.010	0.020	0.030	0.040	0.050
0.500	1490.09	853.36	610.79	467.49	396.94
0.550	1430.44	785.86	590.04	453.39	384.81
0.600	1201.72	745.89	553.29	428.83	348.97
0.650	**1103.71**	675.56	538.98	416.32	337.85
0.700	1129.83	665.24	507.01	396.54	**330.29**
0.750	1109.78	647.92	**485.08**	**364.08**	**332.63**
0.800	1145.39	**635.32**	479.38	**391.13**	343.38
0.850	1134.51	**648.34**	**467.96**	407.02	357.50
0.900	1183.60	671.12			

(a)

df / vel	0.010	0.020	0.030	0.040	0.050
0.500	1474.00	743.05	495.53	399.41	**322.68**
0.550	1308.63	666.59	458.23	373.22	330.81
0.600	1271.52	625.99	449.99	**372.49**	333.07
0.650	1172.04	599.56	**425.47**	377.98	358.23
0.700	1135.96	594.91	460.12	397.70	
0.750	1042.00	**578.79**	478.65		
0.800	1008.77	613.54			
0.850	**963.23**				
0.900	1130.48				

(c)

df / vel	0.010	0.020	0.030	0.040	0.050
0.700	1093.86	649.57	498.13	**380.44**	320.91
0.750	1063.10	610.08	**459.80**	363.44	310.95
0.800	1042.13	**601.39**	450.06	359.54	297.55
0.850	1025.87	597.27	**443.40**	**355.38**	**293.43**
0.900	**1014.82**	**586.80**			

(b)

df / vel	0.010	0.020	0.030	0.040	0.050
0.700	1107.26	**559.05**	438.22	384.43	
0.750	1024.96	570.59	459.44		
0.800	998.12	594.11			
0.850	**957.08**				
0.900	1012.07				

(d)

In Table 6.9, the *acEpl.* values obtained for different gait patterns are given. The tables *(a), (b), (c),* and *(d)* correspond to the Tables 6.1, 6.2, 6.5, and 6.6 of the previous section, respectively. In table *(a)* the *acEpl.* values for constant stroke – wave gaits, in table *(b)* the *acEpl.* values for constant stroke – phase modified wave gaits are given. For both of these tables, the constant stroke length is taken to be *0.14m*. The regions with white background correspond to the ranges where protraction time is larger than *1.5sec.,* the minimum for a proper protraction of the actual robot. The regions with grey background correspond to the ranges where protraction time is less than the proper minimum, but still applicable for an acceptable walk. Therefore, they are included in the table to reveal the energy consumption profile, which is less affected by the protraction phases than the retraction. In table *(c)* the *acEpl.* values for constant protraction time – wave gaits, in table *(d)* the *acEpl.* values for constant protraction time – phase modified wave gaits are given. For these tables the constant protraction time is taken to be *1.5sec.,* the minimum for a proper protraction. The tables are restricted to the regions where the maximum stroke length is *0.14m.* (In actual robot, the stroke length is not possible to be exceeded without disturbing the walk.)

In all four tables, it is observed that the *acEpl.* values decrease with increasing velocity. The power consumption, which can be found in watts by multiplying the *acEpl.* values by the velocity, on the other hand, increases with increasing velocity. These confirm the observations made on the simulation data for the relation between energy consumption and velocity, and they justify *Strategy 1*. The *acEpl.* values in the constant stroke tables *(a)* and *(b)* are generally higher than the corresponding ones in the constant protraction time tables *(c)* and *(d)*. The exception is only in the velocity column of *0.01m/s* for the three duty factors of *0.6, 0.65,* and *0.7* between the tables *(a)* and *(c)*, and for the duty factor of *0.7* between the tables *(b)* and *(d).* But, even for these cases the violation is insignificant and can be ignored in the general trend. Therefore, the actual data support that more efficient walk is achieved with minimum protraction time; hence, *Strategy 2* is justified. All the *acEpl.* values in the phase modified wave gait tables *(b)* and *(d)* are lower than the corresponding ones in the wave gait tables *(a)* and *(c)*. This approves the observation that shifting to the phase modified wave gaits results in more energy efficient walk for the range $\beta > 2/3$; hence *Strategy 3* is justified.

In table *(a)*, all the *acEpl.* values in the region of $\beta \leq 2/3$, demonstrate a continuous decrease with increasing duty factor. In the region of $\beta > 2/3$, the *acEpl.* values demonstrate an irregular behavior with increases and decreases. These are in accordance with the observations on the simulation data of Table 6.1. The minimum of the columns occur in the region of $\beta > 2/3$ in all simulation and real data cases except for the *0.01m/s* velocity column of table *(a).* The minimum of this column is the minimum of the region $\beta \leq 2/3$, namely the one corresponding to the largest duty factor for $\beta \leq 2/3$. However, the difference between this minimum of the column *(1103.71 joule/m)* and the minimum of the region $\beta > 2/3$ *(1109.78 joule/m)* is so little that, this exception can be ignored in the general trend. Therefore, it can be concluded that the observations on table *(a)* justify *Strategy 4.i*.

Considering table *(b)* with the upper region ($\beta \leq 2/3$) of table *(a)*, the actual data supports the observation that for a given velocity, dissipated energy continuously decreases with increasing duty factor for constant stroke – phase modified wave gaits; hence the minimums of table *(b)* occur with maximum duty factor. This observation justifies *Strategy 4.ii*.

The table *(c)* for the constant protraction time – wave gaits, and table *(d)* for the constant protraction time – phase modified wave gaits (also considering the upper part of table *(c)* for the region $\beta \leq 2/3$), demonstrate that the *acEpl.* values have the regular behavior of decreasing and increasing with respect to the increasing duty factor, and the minimum *acEpl.* occurs somewhere close to the maximum duty factor. These are in accordance with the observations made on the simulation data of Tables 6.5 and 6.6; hence *Strategy 4.iii* and *Strategy 4.iv* are justified.

6.5. Conclusion

Despite the various literature on gait generation and stability analysis of gaits, there has been comparatively less research on energy efficiency of gait patterns. Wave gaits, which are proved to be the most stable and are most frequently applied in multi-legged robots, especially deserve to be analyzed from energy dissipation point of view. This chapter aims to fill this gap, and to derive some strategies in order to determine the best parameters of the wave gaits for minimum energy consumption. The chapter seems to be the first in establishing solid strategies to determine the parameters of the gaits, and justifying the strategies on realistic data obtained from a real experimental setup with a six-legged robot.

The analysis of wave gaits on the simulation data pointed to a modification of the ipsilateral phase difference of the wave gaits for a more efficient walk. The resultant gaits with this modification are still periodic, regular, symmetric, and constant phase increment; but differently from the wave gaits, the increment of the constant ipsilateral phase difference is fixed to *1/3* for large duty factors. The loss in stability due to this modification is analyzed; and, it is concluded that the loss in stability is insignificant. The effect of the duty factor, stroke length, phase difference between ipsilateral legs, protraction time, retraction time, and velocity on energy consumption are analyzed. In general, the dissipated energy increases with increasing stroke length and protraction time and with decreasing duty factor and velocity (hence, retraction time). However, since all these parameters are dependent on each other, it is not meaningful to speak about the effects of the independent changes of these parameters. Rather, the observations in particular mode of walking is more useful in determining the best parameters. The approach here is based on determining the duty factor and velocity, for the given mode of walk. If the mode is free, the phase modified wave gait with minimum protraction time is the best for an efficient walk. The strategies are developed to minimize the search effort for determining the best duty factor and velocity.

The energy analysis here is performed in the vicinity of the wave gaits without sacrificing the stability in a significant degree. However, there can be situations that more sacrifice from stability is tolerable for more energy efficiency. Namely, just the achievement of static stability may be enough, which is possible when the gait deviates more from the wave gait. The search of more efficient but just statically stable gaits, which are not necessarily in the vicinity of the optimal stable wave gaits, remains open as a future work based on this chapter.

CHAPTER 7

SUMMARY AND CONCLUSION: CONTRIBUTIONS

In this thesis, basically five pieces of work, performed for the three-joint six-legged Robot-EA308, are presented in Chapters 2-6. In these work, the aim was to provide contribution to understanding and application of six-legged walking, with using an actual robot. In each chapter, a particular side of six-legged walking is handled. The contributions of each chapter are intended both for researchers interested in multi-legged walking in general, and also for the researchers who are interested in particular topics that are in relation to this. Theoretical analysis of gaits, real-time application of reinforcement learning to robots, trajectory optimization and controller design for robot manipulators, kinematic-dynamic formulation of parallel systems with changing configuration, force distribution in multi-linked systems, and energy-efficient walking can be cited in this regard. In the following brief summaries of the contributions of the work presented in Chapters 2-6 are given.

In *Chapter 2* an analytical study of stability of "standard gaits" is presented. The mathematical tools and definitions of terms for such analytical study are provided:

1) A *systematic of set relations* – including the standard, periodic, regular, and wave gaits – is developed under the general set of "orderly gaits".
2) A systematic approach of *analysis of standard gaits* is presented with the stated theorems and their proofs.
3) The *stability margin of "standard gaits"* is calculated.
4) A *proof of the optimum stability of wave gaits within standard gaits* is performed by narrowing down to the specific from the general.
5) *Proofs* are provided *for the intuitively accepted propositions* about duty factor limits, stability guarantee with the rule of neighborhood, superior stability of rear-to-front wave gaits, and maximum phase increment.

In *Chapter 3* a free gait generation with reinforcement learning is developed and applied to the actual Robot-EA308:

1) A *discrete model of stepping* is introduced. Using this model, a *free gait generation algorithm*, which guarantees static stability in all instances, is developed.

2) The *central pattern generation* and *reflex models* for multi-legged walking are *conciliated* by combining the developed free gait generation with *reinforcement learning*.

3) *Continuous improvement* of stability, *smooth passages* between differently commanded speeds, and *adaptation* to the case of a rear leg deficiency in real-time walking are achieved.

4) The algorithm is successfully *applied to the actual Robot-EA308*. Continuous improvement of stability for a given speed, smooth passages between different speed commands, and learning of five-legged walking in case of a rear leg deficiency are successfully realized on the robot *in real-time*.

5) The difference between the experiments with the actual robot and simulations, due to the irregular behavior of some standing states, is delineated as the problem of *"on-the-border" situations*. The effects of on-the-border situations in the real-time applications are highlighted.

In *Chapter 4* a trajectory optimization and controller design is performed for the protraction movement of a three-joint robot leg:

1) The *trajectory optimization* is performed by making use of the gradient-based *optimal control method*.

2) The *dynamic equations* of the three-joint leg system are obtained by Lagrange formalism. The *Hamiltonian formulation* is given for this system.

3) The optimization is performed by making use of a *dense discretization*, rather than a parametrized trajectory. In this way the search space is not restricted.

4) In order to come over the problem of sticking to local optimums, the approach named "*optimization with objective function modification*" is developed and applied.

5) The results of optimization with objective function modification for *79 initial-final tip point cases* are used to construct *an interpolating network*, to be used to generate the guiding trajectories for the controller.

6) The idea of "*multi agent system based fuzzy controller design for manipulators*" is *improved and applied* for the controller design of the three-joint leg system.

7) The performance of the controller is demonstrated with the actual Robot-EA308

In *Chapter 5*, the force-moment distribution of a six-legged robot is handled as a "torque distribution" problem, rather than the conventional approach of tip-point "force distribution":

1) A *full kinematic and dynamic formulation of a six-legged robot* is performed, considering both retracting and protracting legs, and without neglecting any leg-link masses.

2) The derivation of equations is performed by taking the *joint-torques*, rather than the tip-point forces, as the *primary variables*.

3) The *torque distribution problem* is formulated as a quadratic programming problem, and *a compact matrix representation* is provided for this purpose, with the joint-torques being the primary variables.

4) The *quadratic programming problem* is solved by using *Lemke's Complementary Pivoting Algorithm*. The problem of cycling of pivoting is addressed with practical solution methods.

5) The *results* of "torque distribution" and "force distribution" are *compared* for power consumption, with the cases of stationary standing and tripod gait walking.

6) It is demonstrated that the *"torque distribution" is more efficient*, because it makes use of the friction and interaction forces to minimize the joint-torqes on the supporting legs.

In *Chapter 6*, the wave gaits are analyzed from the point of view of energy consumption:

1) The *energy consumption data* of four modes of walking are generated with simulation, and *tabulated* in a descriptive way.

2) Due to the analysis of conventional wave gaits with constant stroke, a modification of wave gaits is proposed with the name of "*phase modified wave gaits*".

3) Using the formulas derived in Chapter 2, *the stability margins* of phase modified wave gaits and conventional wave gaits are *calculated* and compared. It is demonstrated that the loss of stability by passing to the phase modified wave gaits is insignificant.

4) It is demonstrated that, for the general-purpose robots with similar dimensions and facilities as the Robot-EA308, the *gaits with higher speeds and lesser protraction time are more efficient*. Besides, *phase modified wave gaits are more efficient* that the conventional wave gaits.

5) *Strategies are established* in order to adopt the best mode of walking, and to minimize the search effort for the best duty factor and velocity for the given conditions of the robot and the task.

6) The *strategies are justified* on realistic data obtained by *using the actual Robot-EA308*.

The results of the five pieces of work point to some possible directions of future work. Some of these are already mentioned in the concluding sections of the chapters; however, they will be reemphasized here for convenience of the concluding chapter.

Chapter 2 reveals that, a systematic approach based on relations of different clusters of gaits is useful for the analysis of gaits. The proposed relations under the set of "orderly gaits" provided an easier comprehension of gait classification, and made it easier to establish a systematic approach.

Considering the fact that, analytical work of gaits is a tedious work, such systematization of relations is very important. The analytical work on gaits can be furthered by detailing the sets under the orderly gaits, by analyzing their stability properties, and by establishing relations between the different clusters. This should not be considered only as a matter of proving the optimal stability of wave gaits, but also as a research of the unexplored gait models, which can result in better performance for different preoccupations.

The basic difficulty of application of reinforcement learning to real world problems is mentioned as the necessity of large number of trials; therefore, it is frequently suggested that reinforcement learning should be biased to speed up the learning, for example by equipping it with a priori knowledge of the programmer. In the application in Chapter 3, this was performed by using the free state generation algorithm, which guarantees stability for six-legs. In case of rear-leg deficiency, it is only with this bias that the robot can learn to walk. A branch of future work might be to improve this bias by making use of the experiences of the robot in the free state generation algorithm. The learning scheme presented here is based on memorizing the stable transitions. Namely, the unstable transitions are not memorized in a set of bad experiences. The learning algorithm can be made faster by memorizing the bad experiences as well. However, since the number of bad experiences generally far exceeds the number of useful experiences, a problem of memory storage and processing might occur during real-time applications. To determine the bad experiences to be memorized, to incorporate bad experiences with the learning scheme, and to utilize those can be considered as a future work in order to speed up the learning. In this way learning of five-legged walking only with a few-falls can be possible. The problem of “on-the-border” situations, faced in the learning of five-legged walking with the real robot, is also a subject of future work considering the real-world applications of learning with binary reinforcement signal.

In Chapter 4, “optimization with modified objective function” is proposed. The main drawback of this approach is that it necessitates a long time for termination. An intelligent and continuous modifier was suggested in order to speed up the process. This can be achieved by modifying the objective function according to the values of different segments of the nominal objective function, and some other gradient information. Developing such an intelligent optimal control algorithm might be a future work on the proposed optimization scheme. In achieving this, the main problem can be expected to be that the algorithm will be stuck in new kind of local optimums that are not experienced before. The main task will be to handle such sticking to the new experienced local optimums and to improve the algorithm as far as possible. The future work for the approach of “multi agent system based fuzzy controller design for robot manipulators” might be to develop a systematic to write the rules for the joints of a manipulator considering different expectations, such as minimum time, minimum distance, or minimum power dissipation. The proposed control scheme should also be tested with more complex manipulators of six-degree of freedom.

The basic future work for the “torque distribution” proposed in Chapter 5 might be to develop a joint-torque controlled walking machine and to apply the scheme on an actual system. In force-moment distribution, it can be the case that objectives other than power consumption are considered. For example, when the friction coefficient is not known, although it results in a very inefficient distribution, it might be desirable to apply the “force distribution” scheme in order to minimize the horizontal components of tip-point forces. For such a case, the future work for Chapter 5 might be to develop a structure, which can shift between different distribution schemes, or come to a compromise between the two, according to the environmental conditions. Considering the torque distribution scheme, a system that estimates the friction coefficient would be very useful. Developing such a system, which makes use of the previous experiences, might be another branch of future work.

The energy analysis in Chapter 6 pointed to a modification in wave gaits, which have optimum stability. The modification is an example that a kind of gaits, which does not have optimum stability, might be desirable from the point of view of other objectives. This situation strengthens the importance of the future work of development of set relations of gaits and examination of those, as mentioned above in relation to Chapter 2. As mentioned before, the analysis of Chapter 6 is performed only in the vicinity of the wave gaits. The aim in doing this was not to sacrifice the stability in significant degrees. However, with strong machines walking with very small stability margins can be possible. This allows the gaits to deviate more from the wave gaits. The search and analysis of more efficient, though less stable, gaits in the unexplored regions can be a future work based on Chapter 6.

REFERENCES

[1] Barreto, J.P, Trigo, A., Menezes, P., Dias, J., and Almeida, A.T. (1998), "FBD-The free body diagram method. Kinematic and dynamic modeling of a six leg robot." In *Proceedings of 5th International Workshop on Advanced Motion Control*, AMC'98-Coimbra, Portugal, (29 June-1 July, 1998): 423-428.

[2] Bazaara, M.S., Jarvis, J.J., Sherali, H.D. (1990). *Linear Programming and Network Flows*, John Willey & Sons, Inc., New York, p. 169.

[3] Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. (1993), *Nonlinear Programming Theory and Applications*, John Willey & Sons, Inc., New York.

[4] Beji, L. and Pascal, M. (1999), "The kinematics and the full minimal dynamic model of a 6-DOF parallel robot manipulator." *Nonlinear Dynamics*, 18: 229-356.

[5] Bertsekas, D.P. (1999), *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts.

[6] Bobrow, J.E., Martin, B., Sohl, G., Wang, E.C., Park, F.C, Kim, K. (2001), "Optimal robot motions for physical criteria." *Journal of Robotic Systems*, 18 (12):785-792.

[7] Carvalho, J.C.M and Ceccarelli, M. (2001), "A closed form formulation for the inverse dynamics of a casino parallel manipulator." *Multibody System Dynamics*, 5: 185-210.

[8] Celaya, E. and Porta, J.M. (1998), "A control structure for the locomotion of a legged robot on difficult terrain". *IEEE Robotics and Automation Magazine*, 5 (2): 43-51.

[9] Cheng, F.T. and Orin, D.E. (1990), "Efficient algorithm for optimal force distribution – The Compact -Dual LP Method". *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 2: 178-187.

[10] Cheng, F.T., Orin, D.E. (1991), "Efficient formulation of the force distribution equations for simple closed-chain robotic mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1):25-32.

[11] Chettibi, T., Lehtihet, H.E., Haddad, M., Hanchi, S. (2004), "Minimum cost trajectory planning for industrial robots". *European Journal of Mechanics A/Solids*, 23: 703-715.

[12] Clark, J.E., Cham, J.G., Bailey, S.A., Froehlich, E.M., Nahata, P.K., Full, R.J., Cutkosky, M.R. (2001). "Biomimetic Design and Fabrication of a Hexapodal Running Robot," In *Proceedings of the 2001 ICRA IEEE International Conference on Robotics and Automation*, Volume 4: 3643 - 3649

[13] Cruse, H. (1979), "A new model describing the coordination pattern of the legs of a walking stick insect." *Biological Cybernetics*, 32: 107-113.

[14] Cruse, H. (1990), "What mechanisms coordinate leg movement in walking arthropods." *Trends in Neurosciences*, 1990, 13: 15-21.

[15] Cruse, H., Kindermann, T., Schumm, M., Dean, J. and Schmitz, J. (1998), "Walknet-a biologically inspired network to control six legged walking." *Neural Networks*, 11: 1435-1447.

[16] Cruse, H., Clarac, F, Chasserat, C. (1983), "The control of walking movements in the leg of the rock lobster." *Biological Cybernetics*, 47: 87-94.

[17] Cymbalyuk, G.S., Borisyuk, R.M., Müeller-Wilm, U., Cruse, H. (1998), "Oscillatory network controlling six-legged locomotion. Optimization of model parameters." *Neural Networks*, 1998, 11: 1449-1460.

[18] Dantzig, G.B. (1963), *Linear Programming and Extensions,* Princeton University Press, Princeton, New Jersey.

[19] Desoer, C.A. (1970), *Notes for a Second Course on Linear Systems*, Van Nostrand, Reinhold Company, New York.

[20] Donner, M.D. (1987). *Real Time Control of Walking*, Boston: Birkhäuser, pp. 7-16.

[21] Dürr, V., Schmitz, J., Cruse, H. (2004), "Behaviour-based modeling of hexapod locomotion: linking biology and technical application". *Arthropod Structure & Development*, 33: 237-250.

[22] Erden, M.S., Leblebicioğlu, K, (2004), "Fuzzy controller design for a three joint robot leg in protraction phase - an optimal behavior inspired fuzzy controller design." In *Proceedings of the First International Conference On Informatics In Control, Automation And Robotics*, Setúbal, Portugal, Vol. 2: 302-306.

[23] Erden, M.S., Leblebicioğlu, K. (2005), "Multi legged walking in robotics and dynamic gait pattern generation for a six-legged robot with reinforcement learning." book chapter in *Mobile Robots: New Research*, Nova Publishers (in press, ISBN: 1-59454-359-3).

[24] Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2004), "Multi-agent system based fuzzy controller design with genetic tuning for a service mobile manipulator robot in the hand-over task." *Journal of Intelligent and Robotic Systems*, 38: 287-306.

[25] Espenschied, K.S., Quinn, R.D., Beer, R.D., Chiel, H.J. (1996), "Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot". *Robotics and Autonomous Systems*, 18:59-64.

[26] Ferrell, C. (1995), "A comparison of three insect-inspired locomotion controllers". *Robotics and Autonomous Systems*, 16:135-159.

[27] Frangos, C., Yavin, Y. (2001), "Control of a three-link manipulator with inequality constraints on the trajectories of its joints". *Computers and Mathematics with Applications*, 41: 1562-1574.

[28] Fu, K.S., Gonzalez, R.C., and Lee, C.S.G. (1987), *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill International Editions, Singapore.

[29] Galvez, J.A., Gonzalez de Santos, P., Pfeiffer, F. (2001). "Intrinsic tactile sensing for the optimization of force distribution in a pipe crawling robot," *IEEE/ASMA Transactions on Mechatronics*, 6(1): 26-35.

[30] Galvez, J.A., Estremera, J., Gonzales de Santos, P. (2003), "A new legged-robot configuration for research in force distribution." *Mechatronics*, 13: 907-932.

[31] Garcia, E., Galves, J.A., and Santos, P.G. (2003), "On finding the relevant dynamics for model-based controlling walking robots." *Journal of Robotic and Intelligent Systems*, 37: 375-398.

[32] Garg, D.P., Kumar, M. (2002), "Optimization techniques applied to multiple manipulators for path planning and torque minimization." *Engineering Applications of Intelligence*, 15: 241-252.

[33] Gill, P.E, Murray, W., Wright, M.H. (1981). *Practical Optimization*, Academic Press Inc., London.

[34] Goldfarb, D., and Idnani, A. (1983), "A numerically stable dual method for solving strictly convex quadratic programs". *Mathematical Programming*, 27: 1-33.

[35] Gonzalez de Santos, P., Garcia, E., Cobano, J.A., Guardabrazo, T. (2004), "Using walking robots for humanitarian de-mining tasks". In *Proceedings 35th International Symposium on Robotics (ISR-2004)*, Paris, France, March 2004.

[36] Hardarson, F. (1998), "Locomotion for difficult terrain." *Technical Report TRITA-MMK 1998:3*, Dept. of Machine Design, Royal Institute of Technology, S-100 44 Stockholm, Sweden, April 1998. ISSN 1400-1179, ISRN KTH/MMK-98/9-SE.

[37] Huang, Q.J. and Nomani, K. (2003), "Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control." *Mechatronics*, 13: 773-790.

[38] Huber, M. and Grupen R.A. (1997), "A feedback control structure for online-learning tasks". *Robotics and Autonomous Systems*, 22: 303-315.

[39] Hughes, G.M. (1965), "Locomotion: Terrestrial." In *The Physiology of Insecta*. Editor: M. Rockstein, pp 227-254.

[40] Hung, M.H., Orin, D.E., Waldron, K.J. (2000), "Efficient formulation of the force distribution equations for general tree-structured robotic mechanisms with a mobile base," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 30(4): 529-538.

[41] Ilg, W. and Berns, K. (1995), "A learning architecture based on reinforcement learning for adaptive control of the walking machine LAURON". *Robotics and Autonomous Systems*, 15: 321-334.

[42] Ilg, W., Bernes, K., Mühlfriedel, T., Dillman, R. (1997), "Hybrid learning concepts based on self-organizing neural networks for adaptive control of walking machines". *Robotics and Autonomous Systems*, 22: 317-327.

[43] Inagaki, K, Kobayashi, H. (1994), "Adaptive wave gait for hexapod synchronized walking." In *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, Vol.2: 1326-1331.

[44] Inagaki, K. (1997), "Gait study for hexapod walking with disabled leg." In *Proceedings of the 1997 IEEE/RSJ - International Conference on Intelligent Robots and Systems (IROS'97)*, Vol.1: 408-413.

[45] Inagaki, S., Yuasa, H. and Tamio, A. (2003), "CPG model for autonomous decentralized multi-legged robot system–generation and transition of oscillation patterns and dynamics of oscillators." *Robotics and Autonomous Systems*, 44:171-179.

[46] Ji, P. and Wu, H. (2003), "Kinematic analysis of an offset 3-UPU translational parallel robotic manipulator." *Robotics and Autonomous Systems*, 42: 117-123.

[47] Jiang W.Y., Liu, A.M., Howard, D. (2004). "Optimization of legged robot locomotion by control of foot-force distribution," *Transactions of the Institute of Measurement and Control*, 26(4): 311-323.

[48] Kar, D.C, Issac, K.K, and Jayarajan, K. (2001), "Minimum energy force distribution for a walking robot". *Journal of Robotic Systems*, 18(2): 47-54.

[49] Kar, D.C, Issac, K.K, and Jayarajan, K. (2003), "Gaits and energetics in terrestrial legged locomotion". *Mechanisms and Machine Theory*, 38(2): 355-366.

[50] Karalarli, E., Erkmen, A.M., Erkmen, I. (2004), "Intelligent gait synthesizer for hexapod walking rescue robots". In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004, pp. 2177-2182.

[51] Kimura, H., Yamashita, T., and Kobayashi, S (2001), "Reinforcement learning of walking behavior for a four-legged robot". In *Proceedings of the 40th IEEE Conference on Decisions and Control*, Oriando, Florida USA, December 2001.

[52] Kirchner, F. (1998), "Q-learning of complex behaviours on a six-legged walking machine". *Robotics and Autonomous Systems*, 25: 253-262.

[53] Kirk, Donald E (1970). *Optimal Control Theory – An Introduction*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

[54] Klaassen, B., Linnemann, R., Spenneberg, D. and Kirchner, F. (2002), "Biomimetic walking robot SCORPION: Control and modeling." *Robotics and Autonomous Systems*, 41:69-76.

[55] Klein, C.A. and Kittivatcharapong, S. (1990), "Optimal force distribution for the legs of a walking machine with friction cone constraints". *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 1: 73-85.

[56] Kumar, V.R., Waldron, K.J. (1988). "Force distribution in closed kinematic chains," *IEEE Journal of Robotics and Automation*, 4(6): 657-665.

[57] Lasa, M. de. (2000), “Dynamic compliant walking of scout II quadruped.” *Master Thesis*, submitted to Department of Electrical and Computer Engineering, McGill University, Montréal, Canada.

[58] Lin, B.S. and Song, S.M. (2001), “Dynamic modeling, stability and energy efficiency of a quadrupedal walking machine.” *Journal of Robotic Systems*, 18 (11): 657-670.

[59] Liu, G., Lou, Y., and Li, Z. (2003), “Singularities of parallel manipulators: A geometric treatment.” *IEEE Transactions on Robotics and Automation*, 19 (4): 1623-1627.

[60] Liu, J.F., Abdel-Malek, K. (2000), “Robust control of planar dual-arm cooperative manipulators.” *Robotics and Computer-Integrated Manufacturing*, 16(2-3): 109-120.

[61] Mahajan, A. and Figueroa, F. (1997), “Four-legged intelligent mobile autonomous robot.” *Robotics and Computer Integrated Manufacturing*, 13(1): 51-61.

[62] Marhefka, D.W. and Orin, D.E. (1997), “Gait planning for energy efficiency in walking machines”. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April.

[63] Marhefka, D.W. and Orin, D.E. (1998), “Quadratic optimization of force distribution in walking machines,” In *Proc. of the 1998 IEEE International Conference on Robotics and Automation*, Lueven, Belgium, pp. 477-483, May.

[64] Martins-Filho, L.S. and Prajoux, R. (2000), “Locomotion control of a four-legged robot embedding real-time reasoning in the force distribution.” *Robotics and Autonomous Systems*, 32: 219-235.

[65] Muraro, A., Chevallereau, C., Aoustin, Y. (2003) “Optimal trajectory for quadruped robot with *trot, amble* and *curvet* gaits for two energetic criteria.” *Multibody System Dynamics*, 9: 39-62.

[66] Murty, K.G. (1983), *Linear Programming*, John Willey & Sons, Inc., New York.

[67] Nishii, J. (1998), “Gait pattern and energetic cost in hexapods.” *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 20(5): 2430-2433.

[68] Nahon, M.A and Angeles, J. (1992), “Real-time force optimization in parallel kinematic chains under inequality constraints”. *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 4: 439-450.

[69] Noble, B. and Daniel, J.W. (1977). *Applied Linear Algebra*, Prentice –Hall Inc., New Jersey, pp. 179-184.

[70] Ota, Y., Inagaki, Y., Yoneda, K., Hirose, S. (1998), “Research on a six-legged walking robot with parallel mechanism,” In *Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada, October, 241:248.

[71] Pal, P. K., Mahadev, V, Jayarajan, K. (1994), “Gait generation for a six-legged walking machine through graph search”. In *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2: 1332-1337.

[72] Pearson, K. (1976), “The control of walking.” *Scientific American*, 235: 72-86.

[73] Pfeiffer, F., Rossmann, T., Löffler, K. (2000), “Control of a tube crawling machine”. In *Proceedings of the 2nd International Conference on Control of Oscillations and Chaos*, 5-7 July 2000, St. Petersburg, Volume 3: 586-591.

[74] Pfeiffer, F., Weidermann, J., Danowski, P. (1991), “Dynamics of the walking stick insect.” *IEEE Control Systems Magazine*, 11 (2): 9-13.

[75] Pfeiffer, F., Eltze, J., Weidemann, H.J. (1995), “Six-legged technical walking considering biological principles.” *Robotics and Autonomous Systems*, 14:223-232.

[76] Porta, J.M. and Celaya, E. (2001), “Efficient gait generation using reinforcement learning.” In *Proceedings of the 4th International Conference on Climbing and Walking Robots, Clawar*, Karlsruhe, Germany, Professional Engineering Publishing, ISBN 1-86058-365-2: 411-418.

[77] Porta, J.M. and Celaya, E. (2004), “Reactive free gait generation to follow arbitrary trajectories with a hexapod robot.” *Robotics and Autonomous Systems*, 47: 187-201.

[78] Pratihar, D. K., Deb, K., Ghosh, A. (2002), “Optimal path and gait generations simultaneously of a six-legged robot using GA-fuzzy approach.” *Robotics and Autonomous Systems*, 41: 1-20.

[79] Preumont, A., Alexandre, P., Doroftei, I., Goffin, F. (1997), “A conceptual walking vehicle for planetory exploration”. *Mechatronics*, 7(3):287-296.

[80] Preumont, A., Alexandre, P., Ghuys, D. (1991), “Gait analysis and implementation of a six leg walking machine.” In *Proceedings of the Fifth International Conference on Advanced Robotics (91 ICAR – IEEE)*, Vol.2: 941-945.

[81] Sabes, P.N. (2001), “Linear Algebraic Equations, SVD, and the Pseudo-Inverse. [Lecture Notes]”, October 18, http://www.keck.ucsf.edu/~sabes/Docs/SVDnotes.pdf, last accessed in July, 2006.

[82] Saramago, S.F.P., Stefen Jr., V. (1998), “Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system”. *Mech. Mach. Theory*, 33(7): 883-894.

[83] Song, S.M. and Choi, B.S. (1990), “The optimally stable ranges of 2*n*-legged wave gaits”. *IEEE Transactions on System, Man, and Cybernetics*, 20(4): 888-902.

[84] Song, S.M. and Waldron, K.J. (1987), “An analytical approach for gait study and its applications on wave gaits”. *The International Journal of Robotics Research*, 6(2): 60-71.

[85] Svinin, M.M., K. Yamada, K. Ueda (2001), “Emergent synthesis of motion patterns for locomotion robots”. *Artificial Intelligence in Engineering*, 15:353-363.

[86] Wilson, D. (1966), “Insect walking.” *Annual Review of Entomology*, 11.

[87] Ye, T.Y. (2003), “Gait planning and transitions of walking robots on smooth and rough terrains.” *Ph.D. Thesis*, Mechanical Engineering in the Graduate College of the University of Illinois at Chicago.

[88] Zhao, Y.S., Lu, L., DU, Y.H., and Huang, Z. (2000), “Dynamic performance analysis of six-legged walking machines.” *Mechanism and Machine Theory*, 35: 155-163.

APPENDIX

A. Inertia Matrices for the Body Frame and the Three-Joint Legs of the Six-Legged Robot-EA308

The rotational inertia of a link around a given point is a tensor, which can be represented as a 3×3 matrix with respect to any reference frame. In derivation of dynamical equations, it is common that the inertia tensor is calculated around the center of mass of the link. The representation of the inertia tensor of the link with respect to a frame fixed the link is a constant matrix, which can be calculated by the below formulas:

$$\{\check{J}_k\}^{(k)} = \hat{J}_k^{(k)} = \hat{J}_k = \begin{bmatrix} J_{k11} & J_{k12} & J_{k13} \\ J_{k21} & J_{k22} & J_{k23} \\ J_{k31} & J_{k32} & J_{k33} \end{bmatrix}$$

$$J_{kij} = \int_B [r_k^2 \delta_{ij} - x_{ki} x_{kj}]\, dm \qquad r_k^2 = |\vec{r}_k|^2 = x_{k1}^2 + x_{k2}^2 + x_{k3}^2 \qquad \delta_{ij} = \left\{ \begin{matrix} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{matrix} \right\}$$

The representation of the inertia tensor of the link, with respect to any other reference frame can be found as:

$$\{\check{J}_k\}^{(j)} = \hat{J}_k^{(j)} = \hat{C}^{(j,k)} \hat{J}_k^{(k)} \hat{C}^{(k,j)}$$

Body Frame:

The calculation of the inertia matrix for the body frame is performed with respect to the configuration in Figure A.1. The body of the robot, which is composed of the plastic frame and the six first joint motors, as seen in Figure 1.1, is approximated as a rigid rectangular frame. The values of the parameters in Figure A.1 are given in Table A.1. The center of mass of the body frame is assumed to coincide with the origin of the body reference frame. The resultant inertia matrix of the robot body with respect to the body reference frame is given below Table A.1.

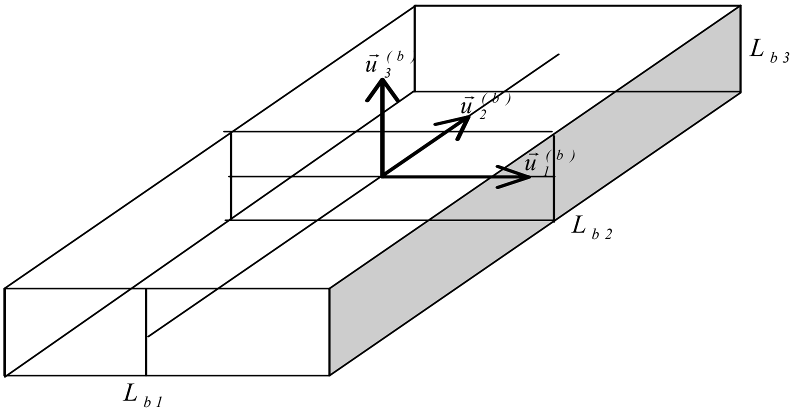


Figure A.1: Robot body frame and reference frame configuration for the calculation of inertia matrix.

Table A.1: Parameters for inertia matrix calculation of the body frame. (These parameters belong to the Robot-EA308)

	***m_b* (g)**	***L_{b1}* (cm)**	***L_{b2}* (cm)**	***L_{b3}* (cm)**
Body Frame	992	16.5	44	3

$$\{\check{J}_b\}^{(b)} = \hat{J}_b = \begin{bmatrix} \frac{1}{12} m_b (L_{b2}^2 + L_{b3}^2) & 0 & 0 \\ 0 & \frac{1}{12} m_b (L_{b1}^2 + L_{b3}^2) & 0 \\ 0 & 0 & \frac{1}{12} m_b (L_{b1}^2 + L_{b2}^2) \end{bmatrix}$$

Links of the Legs:

The calculations of the inertia matrices for the three links of a leg are performed with respect to the sample configuration in Figure A.2. The values of the parameters in this figure are given in Table A.2 for the three links. These are approximated values for the three-joint six-legged Robot-EA308. The center of mass of the *kth* link ($O_{cm,k}$) is assumed to be in the middle of the line connecting the origins of *k-1st* and *kth* reference frames, shown in Figure 4.1 ($c_k = a_k/2$). The inertia matrix of the *kth* link is calculated around this center of mass, and represented with respect to the *kth* reference frame fixed to the link.

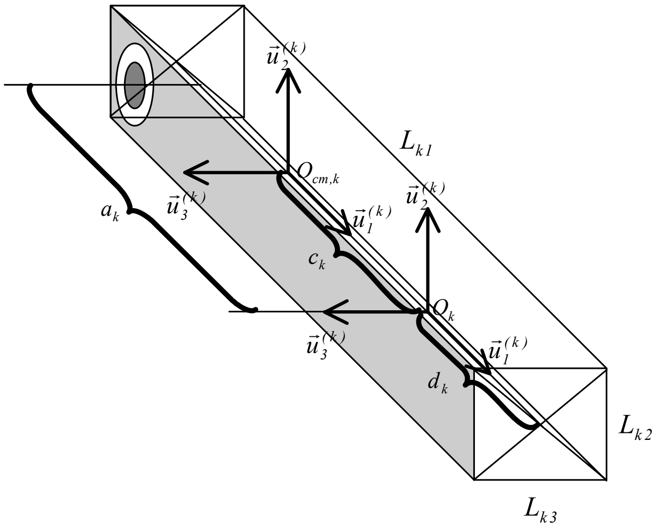


Figure A.2: Sample leg-link and the reference frame configuration for the calculation of the related inertia matrix.

Table A.2: Parameters for the inertia matrix calculation of leg-links. (These parameters belong to the legs of the Robot-EA308).

Link k	***m (g)***	***L_{k1} (cm)***	***L_{k2} (cm)***	***L_{k3} (cm)***	***d_k (cm)***	***$c_k=a_k/2$ (cm)***
k=1	67	4.3	2	3.8	1	0.75
k=2	88	6	2	3.8	1	2.625
k=3	18	14.2	0.5	1	0	6.625

The entries of the inertia matrices are calculated as follows (*k=1, 2, 3)*:

$$J_{k1}=m_k\left(\frac{L_{k2}^2+L_{k3}^2}{12}\right)$$

$$J_{k2}=m_k\left(\frac{3(d_k+c_k)^2-3(d_k+c_k)L_{k1}+L_{k1}^2}{3}+\frac{L_{k3}^2}{12}\right)$$

$$J_{k3}=m_k\left(\frac{3(d_k+c_k)^2-3(d_k+c_k)L_{k1}+L_{k1}^2}{3}+\frac{L_{k2}^2}{12}\right)$$

$$\hat{J}_k=\begin{bmatrix} J_{k1} & 0 & 0 \\ 0 & J_{k2} & 0 \\ 0 & 0 & J_{k3} \end{bmatrix}$$

B. Entries of the Matrices in the Equation of the Dynamic Motion of a Three-Joint Leg

$$\hat{M}(\bar{q})\ddot{q}+\bar{C}(q,\dot{q})+\bar{G}(q)=\bar{Q}$$

In the formulas the variables represented by a_i stand for the length of the *ith* link (namely, the length of the portion of the link between the origins of *i-1st* and *ith* reference frames). The variables represented by θ_{ij} mean the sum of the *ith* and *jth* joint angles *(*$\theta_{ij}=\theta_i+\theta_j$*)*. *C* and *S* correspond to cosine and sinus functions, respectively. The variables represented by c_i designate the position of the center of gravity of the *ith* link: *(*$-c_i,0,0$*)* represents the position of the center of gravity of the *ith* link with respect to the *ith* reference frame. Namely, the term *(*a_i-c_i*)* designates the distance of the center of gravity of the *ith* link from the *ith* joint. For the Robot-EA308, the center of mass of the *ith* link is assumed to be in the middle of the line connecting the origins of *i-1st* and *ith* reference frames ($c_i=a_i/2$).

The mass (inertia) matrix $\hat{M}$ is in the form of

$$\hat{M}=\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}.$$

The entries are calculated as follows:

$$\begin{aligned}
m_{11}=&(m_2+m_3)a_1^2+m_1(a_1-c_1)^2+m_2(a_2-c_2)^2(C\theta_2)^2+m_3a_2^2(C\theta_2)^2+m_3(a_3-c_3)^2(C\theta_{23})^2\\
&+2m_2a_1(a_2-c_2)C\theta_2+2m_3a_1a_2C\theta_2+2m_3a_1(a_3-c_3)C\theta_{23}+2m_3a_2(a_3-c_3)C\theta_2C\theta_{23}\\
&+J_{12}+J_{21}(S\theta_2)^2+J_{22}(C\theta_2)^2+J_{31}(S\theta_{23})^2+J_{32}(C\theta_{23})^2\\
m_{22}=&m_2(a_2-c_2)^2+m_3a_2^2+m_3(a_3-c_3)^2+m_3 2a_2(a_3-c_3)C\theta_3+J_{23}+J_{33}\\
m_{33}=&m_3(a_3-c_3)^2+J_{33}\\
m_{23}=&m_{32}=m_3a_2(a_3-c_3)C\theta_3+m_3(a_3-c_3)^2+J_{33}\\
m_{12}=&m_{21}=m_{13}=m_{31}=0
\end{aligned}$$

$\overline{C}$ is the vector of coriolis, centrifugal, and gyroscopic terms. It is in the following form:

$$\bar{C}(q,\dot{q})=\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}=\sum_{i=1}^{3}\sum_{j=1}^{3}c_{kij}\dot{q}_i\dot{q}_j$$

The c_{kij} variables are calculated as in the following formulas:

$$c_{111}=0$$

$$\begin{aligned}c_{112}=&-2(a_2-c_2)^2 m_2 C\theta_2 S\theta_2-a_2^2 2C\theta_2 S\theta_2 m_3-(a_3-c_3)^2 2C\theta_{23} S\theta_{23} m_3\\&-2a_1(a_2-c_2)S\theta_2 m_2-2a_1 a_2 S\theta_2 m_3-2a_1(a_3-c_3)S\theta_{23} m_3\\&-2a_2(a_3-c_3)(S\theta_2 C\theta_{23}+C\theta_2 S\theta_{23})m_3+2J_{21} S\theta_2 C\theta_2-2J_{22} C\theta_2 S\theta_2\\&+J_{31} 2S\theta_{23} C\theta_{23}-2J_{32} C\theta_{23} S\theta_{23}\end{aligned}$$

$$\begin{aligned}c_{113}=&-(a_3-c_3)^2 2C\theta_{23} S\theta_{23} m_3-2a_1(a_3-c_3)S\theta_{23} m_3-2a_2(a_3-c_3)C\theta_2 S\theta_{23} m_3\\&+2J_{31} S\theta_{23} C\theta_{23}-2J_{32} C\theta_{23} S\theta_{23}\end{aligned}$$

$$c_{121}=0$$

$$c_{122}=0$$

$$c_{123}=0$$

$$c_{131}=0$$

$$c_{132}=0$$

$$c_{133}=0$$

$$c_{211}=-(1/2)c_{112}$$

$$c_{212}=0$$

$$c_{213}=0$$

$$c_{221}=0$$

$$c_{222}=0$$

$$c_{223}=-m_3 2a_2(a_3-c_3)S\theta_3$$

$$c_{231}=0$$

$$c_{232}=0$$

$$c_{233}=-m_3 a_2(a_3-c_3)S\theta_3$$

$$c_{311}=-(1/2)c_{113}$$

$$c_{312}=0$$

$$c_{313}=0$$

$$c_{321}=0$$

$$c_{322}=(1/2)m_3 2a_2(a_3-c_3)S\theta_3$$

$$c_{323}=-m_3 a_2(a_3-c_3)S\theta_3+(1/2)m_3 a_2(a_3-c_3)S\theta_3$$

$$c_{331}=0$$

$$c_{332}=(1/2)m_3 a_2(a_3-c_3)S\theta_3$$

$$c_{333}=0$$

G is the vector of gravitational terms. Its entries are given as follows:

$$\bar{G}=\begin{bmatrix}G_1\\G_2\\G_3\end{bmatrix}\qquad\begin{aligned}G_1=&\,gS\Psi[m_1(-(a_1-c_1)C\theta_1)+m_2(-a_1 C\theta_1-(a_2-c_2)C\theta_1 C\theta_2)]\\&+gS\Psi[m_3(-a_1 C\theta_1-a_2 C\theta_1 C\theta_2-(a_3-c_3)C\theta_1 C\theta_{23})]\\G_2=&\,m_2 g[S\Psi(a_2-c_2)S\theta_1 S\theta_2+C\Psi(a_2-c_2)C\theta_2]\\&+m_3 g[S\Psi(a_2 S\theta_1 S\theta_2+(a_3-c_3)S\theta_1 S\theta_{23})+C\Psi(a_2 C\theta_2+(a_3-c_3)C\theta_{23})]\\G_3=&\,m_3 g[S\Psi(a_3-c_3)S\theta_1 S\theta_{23}+C\Psi(a_3-c_3)C\theta_{23}]\end{aligned}$$

C. Joint Related Position Vectors

The vectors pointing from the origin of the zeroth reference frame to the joints (to the origins of the link reference frames) represented in the zeroth reference frame are as follows:

$$\vec{r}^{(0)}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \vec{r}^{(0)}_{0,1} = \begin{bmatrix} a_1 C\theta_1 \\ a_1 S\theta_1 \\ 0 \end{bmatrix} \quad \vec{r}^{(0)}_{0,2} = \begin{bmatrix} a_1 C\theta_1 + a_2 C\theta_1 C\theta_2 \\ a_1 S\theta_1 + a_2 S\theta_1 C\theta_2 \\ a_2 S\theta_2 \end{bmatrix}$$

The vector pointing from the origin of the zeroth reference frame to the tip point position represented in the zeroth reference frame is as follows:

$$\vec{P}^{(0)}_{tip} = \begin{bmatrix} a_1 C\theta_1 + a_2 C\theta_1 C\theta_2 + a_3 C\theta_1 C\theta_{23} \\ a_1 S\theta_1 + a_2 S\theta_1 C\theta_2 + a_3 S\theta_1 C\theta_{23} \\ a_2 S\theta_2 + a_3 S\theta_{23} \end{bmatrix}$$

D. MATLAB Fuzzy System File for the Joint Acceleration Controller

dth: Joint angle velocity (normalized rad/sec).
alpha: *Pseudo_error* (normalized rad.m)
output: Joint angle acceleration (normalized rad/s^2)

.fis file:

```
[System]
Name='input_acc_director_u123'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=13
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='dth'
Range=[-1 1]
NumMFs=5
MF1='NB':'trimf',[-1.5 -1 -0.5]
MF2='NS':'trimf',[-1 -0.5 0]
MF3='Z':'trimf',[-0.5 0 0.5]
MF4='PS':'trimf',[0 0.5 1]
MF5='PB':'trimf',[0.5 1 1.5]

[Input2]
Name='alpha'
```

```
Range=[-1 1]
NumMFs=5
MF1='NB':'trimf',[-1.5 -1 -0.5]
MF2='NS':'trimf',[-1 -0.5 0]
MF3='Z':'trimf',[-0.5 0 0.5]
MF4='PS':'trimf',[0 0.5 1]
MF5='PB':'trimf',[0.5 1 1.5]

[Output1]
Name='output1'
Range=[-1 1]
NumMFs=7
MF1='NB':'trimf',[-1.333 -1 -0.6666]
MF2='NM':'trimf',[-1 -0.6666 -0.3334]
MF3='NS':'trimf',[-0.6666 -0.3334 0]
MF4='Z':'trimf',[-0.3334 0 0.3334]
MF5='PS':'trimf',[0 0.3334 0.6666]
MF6='PM':'trimf',[0.3334 0.6666 1]
MF7='PB':'trimf',[0.6666 1 1.334]

[Rules]
0 1, 3 (0.03) : 1
0 1, 4 (0.3) : 1
0 2, 3 (0.03) : 1
0 2, 4 (0.3) : 1
0 3, 4 (0.01) : 1
0 4, 5 (0.03) : 1
0 4, 4 (0.3) : 1
0 5, 5 (0.03) : 1
0 5, 4 (0.3) : 1
1 3, 7 (1) : 1
2 3, 7 (1) : 1
3 3, 4 (0) : 1
4 3, 1 (1) : 1
5 3, 1 (1) : 1
```

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Erden, Mustafa Suphi
Nationality: Turkish (TC)
Date and Place of Birth: 21 October 1977, Ankara
Marital Status: Single
Phone: +90 542 631 96 49
email: suphi@metu.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
MS	METU Electrical and Electronics Engineering	2001
BS	METU Electrical and Electronics Engineering	1999
High School	İzmir Scientific High School, İzmir	1995

WORK EXPERIENCE

Year	**Place**	**Enrollment**
1997-2003	METU Department of Electrical and Electronics Engineering	Research Assistant
2003 May-November	ETA (Electronic Design Industry & Trade Inc.), METU Techno-polis	Computer Programming (Development of an Emulative Simulator)
1998 July	Fethiye Telekom, Fethiye, Muğla	Intern Engineering Student
1997 June	Arcelik Refrigerator Plant, Eskişehir	Intern Engineering Student

FOREIGN LANGUAGES

Advanced English, Beginner German

HOBBIES

Sociology, Philosophy, Classical Guitar, Bağlama.

PUBLICATIONS

Master Thesis:

Erden, Mustafa Suphi, *MS Thesis – Conflict Resolution in Air Traffic Management*, METU, Department of Electrical and Electronics Engineering, September 2001.

International Journals:

1) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2004), "Multi-agent system based fuzzy controller design with genetic tuning for a service mobile manipulator robot in the hand-over task." *Journal of Intelligent and Robotic Systems*, 38: 287-306.

2) Erden, M.S. (2004), “The exchange of Greek and Turkish populations in the 1920s and its socio-economic impacts on life in Anatolia.” *Crime, Law & Social Change* 41: 261–282, 2004.

<u>Book Chapter:</u>

Erden, M.S., Leblebicioğlu, K. (2005), “Multi legged walking in robotics and dynamic gait pattern generation for a six-legged robot with reinforcement learning.” Book chapter in *Mobile Robots: New Research*, Nova Publishers, New York, (ISBN: 1-59454-359-3).

<u>International Conference Papers:</u>

1) Erden, M.S., Leblebicioğlu, K (2005), “Dynamic gait pattern generation with reinforcement learning.” *Presented in the congress of* IFAC 2005, July 4-8, Prague.

2) Erden, M.S., Leblebicioğlu, K (2004), “Fuzzy controller design for a three joint robot leg in protraction phase - an optimal behavior inspired fuzzy controller design.” In *Proceedings of the First International Conference On Informatics In Control, Automation And Robotics*, Setúbal, Portugal, Vol. 2: 302-306.

3) Erden, M.S., Leblebicioğlu, K (2004), “Conflict resolution for free flight considering degree of danger and concession.” In *Proceedings of the First International Conference On Informatics In Control, Automation And Robotics*, Setúbal, Portugal, Vol. 1: 185:192.

4) Erden, M.S., Leblebicioğlu, K (2004), “A lateral director autopilot design for conflict resolution algorithms.” In *Proceedings of the First International Conference On Informatics In Control, Automation And Robotics*, Setúbal, Portugal, Vol. 2: 330-334.

5) Erden, M.S. (2003), “Positivistic tendencies due to engineering education.”, In *Proceedins of ACE2003- 6th IFAC Symposium on Advances in Control Education, Oulu, Finland*, 389-394.

6) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2002), “Conflict resolution by negotiation.” In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 21-26 July 2002, Vol. L: 471-476.

7) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2002), “Multi-agent system based fuzzy controller design with genetic tuning for a service mobile manipulator robot in the hand-over task.” In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 21-26 July 2002, Vol. K: 503-508.

<u>Under Review Papers Submitted to International Journals:</u>

1) Erden, M.S. and Leblebicioğlu, K. “Protraction of a three joint robot leg: Trajectory optimization and controller design.” (Under Review).

2) Erden, M.S. and Leblebicioğlu, K. “ ‘Torque distribution’ in a six-legged robot.” (Under Review).

3) Erden, M.S. and Leblebicioğlu, K. “Analysis of wave gaits for energy efficiency.” (Under Review).

4) Erden, M.S. and Leblebicioğlu, K. “Stability analysis of ‘standard gaits’.” (Under Review).

5) Erden, M.S. and Leblebicioğlu, K. “Free gait generation with reinforcement learning for a six-legged robot.” (Under Review).

Turkish Conference Papers:

1) Erden, M.S., Leblebicioğlu, K (2006), “Altı-bacaklı robotta burgu kuvvetlerinin dağıtımı (Turque distribution for a six-legged robot).” *14. IEEE Sinyal İşleme ve Uygulamaları Kurultayı CD’si*, Antalya.

2) Erden, M.S., Leblebicioğlu, K (2006), “Dalgalı yürüyüşlerin enerji verimliliği analizi (Analysis of wave gaits for energy efficiency).” *14. IEEE Sinyal İşleme ve Uygulamaları Kurultayı CD’si*, Antalya.

3) Erden, M.S., Leblebicioğlu, K (2005), “Üç eklemli robot bacağı için gezinge eniyilemesi (Trajectory optimization for a three-joint robot leg).” *TOK2005, Ulusal Otomatik Kontrol Konferansı Bildiriler Kitabı*, İstanbul, 243-246.

4) Erden, M.S., Leblebicioğlu, K (2004), “Üç eklemli bir robot bacağının ileri adım atması için optimal davranış esinli bulanık kontrolcu tasarımı (Optimal behaviour inspired fuzzy controller design for a three-joint robot leg in protraction phase).” *12. IEEE Sinyal İşleme ve Uygulamaları Kurultayı CD’si*, Kuşadası.

5) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2003), “Pekiştirmeli eğitim kullanarak yürüme planı öğrenimi (Gait-pattern generation with reinforcement learning).” *11. IEEE Sinyal İşleme ve Uygulamaları Kurultayı Bildiriler Kitabı*, İstanbul, 671-674.

6) Erden, M.S., Önder, M., Leblebicioğlu, K., Halıcı, U. (2003), “Görüntülerde uzam ve zaman bilgilerini kullanarak hareket bölütlemesi (Motion segmentation with using spatial and temporal information).” *11. IEEE Sinyal İşleme ve Uygulamaları Kurultayı Bildiriler Kitabı*, İstanbul, 331-334.

7) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2001), “Çok ajanlı sistem yaklaşımıyla hava trafiği kontrolü (Air-traffic control with multi-agent system perspective).” *9. IEEE Sinyal İşleme ve Uygulamaları Kurultayı Bildiriler Kitabı*, Gazimagosa, Vol II: 427-432.

8) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2001), “Çok Ajanlı Sistem Yaklaşımıyla Genetik Ayarlamalı Bulanık Kontrol Kullanarak İki Boyutta Nesne Teslimi Sistemi için Robot Kontrolü (Multi-agent system based fuzzy controller design with genetic tuning for a service mobile manipulator robot in the hand-over task).” *9. IEEE Sinyal İşleme ve Uygulamaları Kurultayı Bildiriler Kitabı*, Gazimagosa,Vol II: 717-722.

9) Erden, M.S., Leblebicioğlu, K., Halıcı, U. (2001), “Hava trafik kontrolünde pazarlık kullanarak çelişki çözme uygulaması (Application of negatiation for conflict resolution in air-traffic control).” *TOK2001 Otomatik Kontrol Ulusal Toplantısı Bildiriler Kitabı*, Uludağ-Bursa, 215-218.

10) Tuncay, S., Erden, M.S., Leblebicioğlu, K., Halıcı, U., Özgen C. (2000), “Genetik algoritma kullanarak otomatik Petri-Ağı çatımı (Automatic Petri-Net construction using genetic algorithms)”, *TOK2000 Otomatik Kontrol Ulusal Toplantısı Bildiriler Kitabı*, Ankara, 202-206.