QUADRUPED LOCOMOTION REFERENCE SYNTHESIS WITH CENTRAL PATTERN GENERATORS TUNED BY EVOLUTIONARY ALGORITHMS

by
ÖMER KEMAL ADAK

Submitted to the Graduate School of Engineering and Natural Sciences in partial fulfillment of the requirements for the degree of Master of Science

Sabanci University
January 2013

QUADRUPED LOCOMOTION REFERENCE SYNTHESIS WITH CENTRAL PATTERN GENERATORS TUNED BY EVOLUTIONARY ALGORITHMS

APPROVED BY:

Assoc. Prof. Dr. Kemalettin ERBATUR (Thesis Advisor) ....

Prof. Dr. Mustafa Ünel ....

Assoc. Prof. Dr. Serhat Yeşilyurt ....

Assoc. Prof. Dr. Ahmet Onat ....

Assist. Prof. Dr. Hakan ERDOĞAN ....

DATE OF APPROVAL: ......28.01.2013......

QUADRUPED LOCOMOTION REFERENCE SYNTHESIS WITH CENTRAL PATTERN GENERATORS TUNED BY EVOLUTIONARY ALGORITHMS

Ömer Kemal ADAK

Mechatronics Engineering, Ms. Thesis, 2013

Thesis Supervisor: Assoc. Prof. Dr. Kemalettin ERBATUR

Keywords: Quadruped robots, locomotion reference generation, Central Pattern Generation, gait tuning, genetic algorithms

ABSTRACT

With the recent advances in sensing, actuating and communication tecnologies and in theory for control and navigation; mobile robotic platforms are seen more promising than ever. This is so for many fields ranging from search and rescue in earthquake sites to military applications. Autonomous or teleoperated land vehicles make a major class of these mobile platforms. Legged robots, with their potential virtues in obstacle avoidance and cross-country capabilities stand out for applications on rugged terrain. In the nature, there are a lot of examples where four-legged anatomy embraces both speed and climbing characteristics. This thesis is on the locomotion reference generation of quadruped robots.

Reference generation plays a vital role for the success of the locomotion controller. It involves the timing of the steps and the selection of various spatial parameters. The generated references should be suitable to be followed. They should not be over-demanding to cause the robot fall by loosing its balance. Nature tells that the pattern of the steps, that is, the gait, also changes with the speed of locomotion. A well-planned reference generation algorithm should take gait transitions into account.

Central Pattern Generators (CPG) are biologically-inspired tools for legged-robot locomotion reference generation. They represent one of the main stream quadruped robot locomotion synthesis approaches, along with Zero Moment Point (ZMP) based techniques and trial–and–error methods.

CPGs stand out with their natural convenience for gait transitions. This is so because of the stable limit cycle behavior inhertent in their structure. However, the parameter selection and tuning of this type of reference generators is difficult. Often, trial–and–error iterations are employed to obtain suitable parameters.

The background of complicated dynamics and difficulties in reference generation makes automatic tuning of CPGs an interesting area of research. A natural command for a legged robot is the speed of its locomotion. When considered from kinematics point of view, there is no unique set of walking parameters which yield a given desired speed. However, some of the solutions can be more suitable for a stable walk, whereas others may lead to instability and cause robot to fall.

This thesis proposes a quadruped gait tuning method based on evolutionary methods. A velocity command is given as the input to the system. A CPG based reference generation method is employed. 3D full-dynamics locomotion simulations with a 16-degrees-of-freedom (DOF) quadruped robot model are performed to assess the fitness of artificial populations. The fitness is measured by three different cost functions. The first cost function measures the amount of support the simulated quadruped receives from torsional virtual springs and dampers opposing the changes in body orientation, whereas the second one is a measure of energy efficiency in the locomotion. The third cost function is a combination of the firs two. Tuning results with the three cost functions are obtained and compared. Cross-over and mutation mechanisms generate new populations. Simulation results verify the merits of the proposed reference generation and tuning method.

DÖRT BACAKLI ROBOTLAR İÇİN EVRİMSEL ALGORİTMALAR KULLANILARAK MERKEZİ ÖRÜNTÜ ÜRETİMİ YÖNTEMİYLE YÜRÜME REFERANSI AYARLANMASI

Ömer Kemal ADAK

Mekatronik Mühendisliği Programı, Master Tezi, 2013

Tez Danışmanı: Doç. Dr. Kemalettin ERBATUR

Anahtar Kelimeler: Dört bacaklı robotlar, dört bacaklı yürüme referansı oluşturulması, merkezi örüntü üretimi, genetik algoritma

ÖZET

Robotik biliminde gerçekleşen gelişmeler sonucunda mobil robotik platformları birçok açıdan bilim adamlarının ilgisini çekmeye başlamıştır. Mobil robotik platform teknolojisi arama kurtarma çalışmalarından askeri uygulamalara kadar geniş bir alanda kullanılmaktadır. Bacaklı robotlar sahip oldukları yüksek engel aşma kabiliyetleri ile engebeli arazilerde ön plana çıkmaktadırlar. Doğada, hızlı hareket etme ve tırmanma özelliklerine sahip çok sayıda dört bacaklı anatomi bulunmaktadır. Bu tezde de dört bacaklı robotların referans sentezi işlenmektedir.

Referans sentezi, bacaklı robotların hareket kontrolünde çok önemli bir rol oynamaktadır. Üretilen referans robot tarafından takip edilebilir olmalıdır. Üretilen referans robotu, düşmesine sebep olacak şekilde zorlamamalıdır. Dört bacaklı robotun hızının artması sonucunda hareket şeklini değiştirmesi daha verimli olabilmektedir. İyi planlanmış referans üretme yöntemi hareket geçişlerini de hesaba katmalıdır.

Merkezi Örüntü Üretimi, biyolojik yapılardan esinlenilerek oluşturulmuş bir referans sentezleme yöntemidir. Dört bacaklı robotlarda diğer referans sentezleme yöntemleri ise Sıfır Moment Noktası tabanlı teknikler ve deneme yanılma metotlarıdır.

Merkezi Örüntü Üretimi biyolojik yapısı sayesinde hareket geçişlerine çok uygun bir referans sentezleme yöntemidir. Yapılarındaki limit çevrim özelliği bu işleme olanak sağlamaktadır. Fakat hareket geçişlerini hesaba katan kararlı bir referans sentezi için uygun parametreler seçilmek zorundadır. Merkezi Örüntü Üretimi yönteminde uygun parametreleri elde etme işlemi zor bir görevdir. Genellikle deneme yanılma yöntemleri bu işlem için kullanılmaktadır.

Bu tezde dört bacaklı robotların evrimsel algoritmalar kullanılarak yürüme referansı ayarlanması sunulmuştur. Hız komutu sisteme girdi olarak verilmiştir. Merkezi Örüntü Üretimi yöntemi ile referans sentezi, 16 serbestlik dereceli bir dört bacaklı robot için, 3 boyutlu tam dinamikli benzetim ortamında gerçekleştirilmiştir. Hareket benzetimleri yapay bir nüfusa uygunluk değerleri biçilerek gerçekleştirilmiştir. Robotun vücut oryantasyon değişikliklerine karşı etki eden sanal torsiyonal süspansiyon sistemleri uygunluk değerlerini belirlemek için kullanılmıştır. Benzetim çalışmaları sonucunda üç farklı uygunluk fonksiyonuna göre uygun parametreler elde edilmiştir. Bunlardan birincisi robotun dengesini, ikincisi robotun enerji verimliliğini ölçmektedir. Son uygunluk fonksiyonu ise diğer iki uygunluk fonksiyonunun birleşimidir. Benzetim sonuçları sunulan referans üretme yöntemi ve parametre ayarlama metodunun yararlılığını göstermektedir.

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude for my thesis advisor Assoc. Prof. Kemalettin Erbatur. Throughout my Master of Science education, he always encourages and supports me to improve myself further. I could not measure the value of his enthusiasm and guidance in my master education and on this thesis.

I would also like to state my appreciation and regards to my thesis jury members Prof. Dr. Mustafa Ünel, Assoc. Prof. Dr. Serhat Yeşilyurt, Assoc. Prof. Dr. Ahmet Onat and Assist. Prof. Dr. Hakan Erdoğan, for pointing their valuable ideas.

My friends Hasan Beyazörtü, Mehmet Mutludoğan, Mustafa Şentürk, Mehmet Ali Öğütçü, Harun Serçe, Turgut Yılmaz and Tunç Akbaş deserve particular thanks for their invaluable support and friendship. I also particularly thank Tunç Akbaş and Eda Anlamlıer due to their support for my thesis.

I would like to thank; Iyad Hashlamon, Beşir Çelebi, Emre Özeren, Taygun Kekeç, Elif Çetinsoy, Sanem Evren, Soner Ulun, Mehmet Ali Güney, Alper Yıldırım, Mine Saraç, Selim Özel, Utku Seven, Kaan Can Fidan, Emre Eskimez, Mustafa Yalçın, Beste Bahçeci, and many friends from mechatronics laboratory.

Finally and most importantly, I want to express my gratefulness to my parents, Tufan Adak and Semra Adak for their invaluable love, caring and support throughout my life. This thesis is dedicated to my dear family.

QUADRUPED LOCOMOTION REFERENCE SYNTHESIS WITH CENTRAL PATTERN GENERATORS TUNED BY EVOLUTIONARY ALGORITHMS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1. INTRODUCTION

Legged robot research is a field which improved significantly in the last 25 years. On rough terrains, legged robots, especially the ones with the quadruped structure have advantages over other types of robotic land platforms. It is envisaged that quadruped robots will be entrusted with tasks such as search and rescue operations and military applications. The field appears to have a promising future.

Still, using quadruped robots in such assignments can be quite difficult. Robots of this type have many degrees of freedom (DoF) and their dynamics are highly non-linear and complex. They should be able to walk without falling on even and rough terrain in order to accomplish their tasks. These facts make the control of quadruped robots a tough problem. The gait planning must generate stable locomotion references in order to avoid falling. Stable locomotion reference synthesis is vital for balanced locomotion.

In general it is not straight forward to state whether a generated walking pattern is stable or not. There are cases where unstable locomotion doesn't reveal itself before the fall of the robot. In other words, it is hard to call a walk unstable before the robot falls. In order to avoid unstable locomotion, the reference generation method has to be based on well-defined stability criteria, or it should be adjusted based on experience or some learning and optimization process.

There are two main reference generation approaches used in the literature. The first one is the Zero Moment Point (ZMP) stability criterion related reference generation methodology. The other one is the Central Pattern Generator (CPG) based reference generation, which is a bio-inspired reference synthesis method.

The ZMP stability criterion is related to the contact forces between the robot feet and the ground. The Zero Moment Point is located where the moment due to contact forces equals zero in the ground level. For stability, this point must be in the support polygon defined by the locations of the robot's feet which touch the ground. The location of the ZMP depends on the trajectories of the links of the robot. The mapping between the link position and acceleration variables and the ZMP is quite complicated. Obtaining joint references for given foot stepping locations and a stable ZMP trajectory in compliance with the stepping points is impractical with this complicated relation. A simplified mapping, called the Linear Inverted Pendulum Model (LIPM) can be used though for calculation efficiency. The trajectory of the Center of Mass (CoM) of the robot can be obtained with the LIPM by a variety of methods [1, 2] from a given stable ZMP reference curve. Joint references are finally obtained by solving an inverse kinematics problem between the CoM and the pre-defined Cartesian foot trajectories.

Yoneda and Hirose proposed ZMP-based trajectory generation for a quadruped robot trot gait. Validity of this method was verified by experiments on quadruped robots TITAN IV and TITAN VI [3]. In Takeuchi's work, ZMP-based reference generation with receding horizon control, which provides real time optimization, was used and implemented on Mel Horse II quadruped robot [4, 5]. Pengfei et al. used ZMP principle on their wheeled foot quadruped robot, HITAN I [6]. Osumi et al. used the ZMP criterion with optimal control theory and they used Sony ERS-7 quadruped robot for their experiments [7]. Within the DARPA Learning Locomotion project, several researchers used their ZMP based reference generation methods on the quadruped robot Little Dog [8-11].

The use of Central Pattern Generators (CPG) is a biologically inspired approach for the generation of locomotion references. Joint references or toe references are generated using dynamic equations with stable limit cycles. The dynamic equations used for the reference generation purpose are classified as neural oscillators or non-linear oscillators. Suitable parameters of these oscillators can be found using trial and error, optimization or some learning methods. The output of the oscillator is used as the reference of one hip joint of the quadruped robot. The references of the other hip joints are obtained by adding phase difference to the output of this primary oscillator. The phase differences create the gait, that is, the locomotion pattern of the robot. This method carries out the reference generation of the quadruped robot using only one oscillator. Adding phase differences between references of the hip joints provides a certain degree of simplicity in gait transition, too. Gait transition is changing the gait from one type into another one (e.g. from crawl to trot) without interrupting locomotion.

In 1984 Katoh and Mori used reference generation with a dynamical system with a stable limit cycle for their simulated biped robot [12]. In 1987 Bay and Hemami used the Van Der Pol's oscillator as a dynamic system for reference generation in their biped simulations [13]. In 1991, Taga et al. used Matsuoka's neuron model [14, 15] as a dynamic system for trajectory generation for legged robots [16]. In 1994, Collins and Richmond used several types of neural and non-linear oscillators for generating references for a quadruped robot [17]. Kimura et al. were influenced from Taga's work and employed neural oscillators for quadruped locomotion. They also added reflex mechanisms to their reference generation method. With their CPG and reflex mechanism, they achieved to locomotion on regular and irregular terrains. They employed the Patrush and Tekken quadruped robots in the experiments [18, 19]. Ijspeert et al. also contributed on the development of the CPG. Their approaches are usually implemented on bipedal walking robots and swimming salamanders [20]. In 2001 Tsujita, Tsuchiya and Onat used non-linear oscillators for adaptive gait pattern control of their quadruped robot [21]. Tsujita et al. continued their works by adding a spine joint on their quadruped robot. They used non-linear oscillators with phase resetting [22]. Liu et al. used both Van Der Pol's and phase oscillators for their quadruped robot reference synthesis. They used oscillator outputs as joint references for the AIBO robot [23, 24]. Endo at al. used the CPG method in a different manner. They employed the output of the oscillator as a position reference for the tip of toe in the Cartesian task space. They generated joint references via inverse kinematics and implemented this technique on biped robot QRIO [25]. This approach simplifies the joint motion coordination process, since inverse kinematics is responsible with the computation of joint references. The remaining task is then to devise suitable Cartesian trajectories foot trajectories.

Genetic Algorithms (GA) are heuristic methods used to solve complex search and optimization problems [26]. They make use of the "survival of the fittest" principle and compare candidate solutions in regard to their fitness. Fitness can be defined as a measure of qualities of the solution, or an assessment of lack of deficiencies. Parameters representing a solution are coded into registers called "chromosomes" after the analogy with living beings. A set of random solutions - called a population - is created at first randomly. The solutions are called individuals of the population. The individuals are ranked according to their fitness values. A next generation of the population is created then from the first generation by chromosome cross-over and mutation processes. Chromosomes of fitter individuals are favored in this mechanism to carry their content into the next generation. New generations are further created iteratively. After a number of iterations, an individual with a desired value of fitness can be generated in this manner. This individual is the solution of the search problem.

In this study, a CPG based reference generation method is proposed to obtain stable locomotion for a quadruped robot. Following [25], oscillator outputs are exploited as Cartesian references. As mentioned before, this choice simplifies the computation of joint position references. Still, the generation of suitable Cartesian foot trajectories is a difficult task. The difficulty is more pronounced when considering a range of demanded locomotion speeds, in place of a single speed setting. The CPG oscillator parameter tuning process has to be repeated for every demanded speed value. This is where an automated tuning mechanism is of great significance. This thesis proposes the use of the above-mentioned genetic algorithm approach with fitness tests through full-dynamics simulations as an automatic tuning tool for the quadruped trot gait. Inspired by [27] and [28], virtual torsional springs and dampers are attached to the trunk frame center of the quadruped robot in order to maintain its balance during the simulations, even with unsuitable walking parameters. Three different fitness measures are employed. The exerted torques by the virtual springs and dampers are employed to design the first cost function to assess the stability of the generated walking gait. The second cost function is related to the locomotion energy efficiency, obtained from actuation power and robot velocity values recorded during the simulation. The third fitness function is a combination of the first two. The effect of the cost function choice on the tuning result is explored. The tuning results are tested via simulations without the virtual spring and damper support.

A 16-DoF quadruped model is used in the Newton-Euler based full-dynamics 3-D simulations.

The thesis is organized as follows. The second chapter introduces quadruped robotics terminology and surveys the timeline of the quadruped robot projects. CPG and GA techniques are briefly reviewed in the chapter, too. The simulation environment and the quadruped model are described in the third chapter. Chapter 4 discusses the proposed CPG system and elaborates on manual, trial-and-error based tuning. The GA tuning with the first cost function mentioned above is presented in Chapter 5. The GA based tuning results with the second and third cost functions follow this in Chapters 6 and 7, respectively. The last section presents conclusions and discusses future works.

Chapter 2

2. A SURVEY

2.1. Quadruped Gaits

Quadruped gaits can be categorized into three groups. These are walking gaits (including ambling), running or trotting gaits and leaping gaits [29]. This classification is made according to speed of the gaits. The quadruped gaits are ordered as the walk, trot, canter and gallop with respect to their speed [30]. Canter, bound and gallop are located in the leaping gaits. In some quadruped animals, the trot is completely replaced by the pace.

2.1.1. Walk

A walk gait, also known as a crawl gait, follows a "left front leg, right rear leg, right front leg and left rear leg" sequence. At the walk gait, the quadruped will always raise one foot when the other three feet are located on the ground. These three feet create a triple support phase on the ground. This phase provides a moment when weight is being transferred from one foot to another. A walking horse moves its head and neck in a slight up and down motion that helps maintain balance [30].

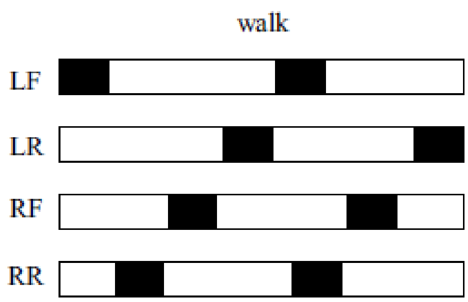


Fig 2.1: Walk gait pattern (black: stance phase, white: swing phase)

2.1.2. Trot

The trot is a two-beat diagonal gait of the quadruped where the diagonal pairs of legs move forward at the same time. At the trot gait, the quadruped will always raise two feet when the other two feet are on the ground. These two feet create double support phase. From the standpoint of the balance of the quadruped, the trot is a very stable gait and does not require the quadruped to make major balancing motions with its head and neck [30].

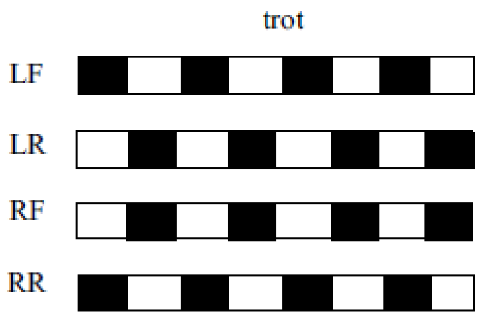


Fig 2.2: Trot gait pattern (black: stance phase, white: swing phase)

2.1.3. Pace

The pace is a lateral two-beat gait. In the pace, the two legs on the same side of the quadruped move forward together, unlike in the trot, where the two legs diagonally opposite from each other move forward together. In both the pace and the trot, two feet are always off the ground. The trot is much more common, but some quadrupeds naturally prefer to pace.

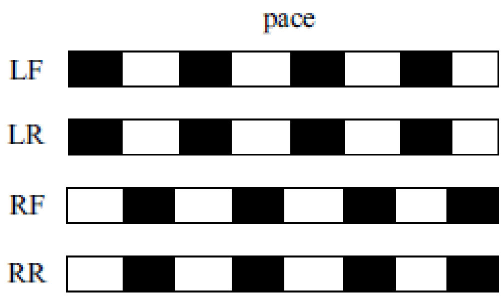


Fig 2.3: Pace gait pattern (black: stance phase, white: swing phase)

2.1.4. Canter

The canter is a bit faster than the average trot, but slower than the gallop. In the canter, one of the quadruped's rear legs, the left rear leg for example, moves the quadruped forward. During this beat, the quadruped is supported only on that single leg while the remaining three legs are moving forward. On the next beat the quadruped raises the right rear and left front legs while the other rear leg is still on the ground. On the third beat, the quadruped raises the right front leg while the diagonal pair is still in contact with the ground [30].

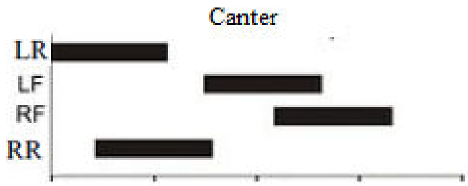


Fig 2.4: Canter gait pattern (black: stance phase, white: swing phase)

2.1.5. Bound

The bound is a controlled, two beat gait like trot and pace. In the bound, the two legs on the front or rear of the quadruped move forward together. The two legs on the rear side of the quadruped propel it forward. This propelling provides fast weight carrying for quadruped and quadruped can move faster.

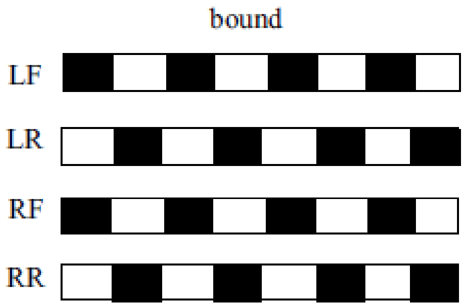


Fig 2.5: Bound gait pattern (black: stance phase, white: swing phase)

2.1.6. Gallop

In the gallop gait, the quadruped will strike off with its non-leading rear foot. The second stage of the canter becomes the second and third stages in the gallop. The inside hind foot hits the ground a split second before the outside front foot. Then both gaits end with the striking off of the leading leg.

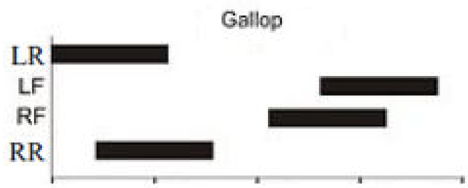


Fig 2.6: Gallop gait pattern (black: stance phase, white: swing phase)

2.2. Central Pattern Generation (CPG)

Animal and human locomotion is highly stable and adaptable. For example, when humans walk, there is no need to consider how high the foot should lift up and where to step. This is done subconsciously. While walking on a slippery ground, the step distance is shorter and the speed becomes lower. How could these reactions be possible without using brain? [66]

Shik et al. [31] demonstrated that decerebrate cats could be forced to walk on a treadmill by steady electrical stimulation to the midbrain region. Moreover, the animals can use different gaits depending on the stimulation strength and the speed of a treadmill. This shows that a complex type of behavior can be controlled by a simple signal, while it is not determined uniquely by the signal but is influenced by functional and environmental constraints.

Other biological studies of motion control show that the generation of periodic motions for both invertebrate and vertebrate animals are mainly generated at spinal cord by a combination of a central pattern generator (CPG) and reflexes receiving adjustment signals from a cerebrum, cerebellum and brain stem. [32-34]

CPGs can be basically defined as neural circuits that can produce rhythmic patterns of neural activity without receiving rhythmic inputs. The term central indicates that sensory feedback from the peripheral nervous system isn't necessary for generating rhythmic activities such as chewing, breathing and digesting. They make also a fundamental system for the locomotor neural circuits.

2.2.1. Motion Control Network of Vertebrates

Walking control structure of vertebrates is mainly divided into three parts. These parts are higher-level central nervous system (Brain), lower-level central nervous system (CPG) and feedback. The higher-level central nervous system specifies the initiation of motion, walking speed and walking direction. CPG controls extensor and flexors muscles and coordinates all joints. Using higher-level central nervous system and feedback are necessary for modulating the whole network for working properly.

Role of the musculo-skeletal system is actuating the muscles properly that creating a proper motion. It is connected with motoneurons and therefore, gets control signals from CPG. Sensorial information from the relations between the environment or some disturbance forces is used as feedback of the system, providing a fast action proceeding from the central pattern generator, which adapts the gait to the new situation.

The main reflexes in adaptive motion control can be divided two parts. These are spinal reflexes and vestibular reflexes. Spinal reflexes such as stretch reflex, flexor reflex and posture reflex are controlled by a spinal cord. The stretch reflex controls muscles' contraction degree, enhances strength of muscles and provides to locate limbs and spinal column in their correct positions. A flexor reflex is activated to contract limbs when receiving stimulation from skin or muscles. It helps animals to avoid obstacles. Posture reflexes can coordinate different muscles to maintain a proper posture. Vestibular reflexes are responsible for the body's balance [35].

Hence, the final motion control signals are the outcomes of the mutual interaction of central nervous system and reflexes system. Brain sends the order to start motion. Then a loop begins. CPG produces rhythmic signals and these signals are transmitted to Musculo-skeletal system by motoneurons. Finally, the body receives feedback information and reflexes to produce further movement commands. Then a new loop begins. There is a basic control system scheme of the vertebrate locomotion.

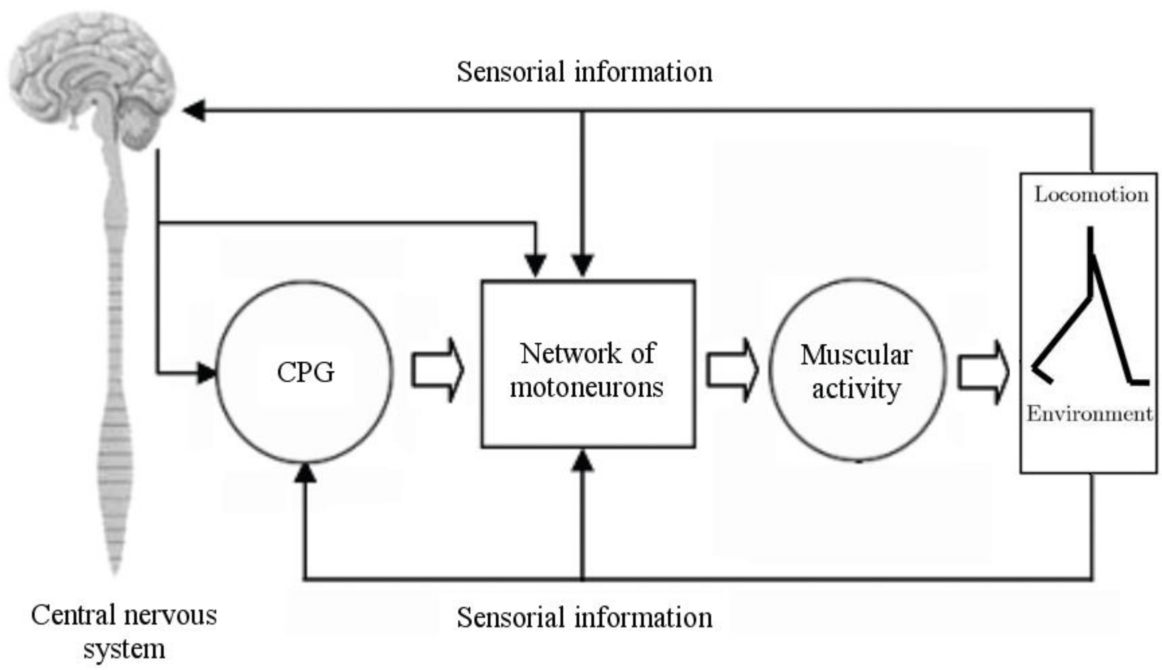


Fig 2.7: The basic control system of the animal locomotion

2.2.2. Implementation of CPG Based Motion Control in Legged Robots

From the engineering perspective, CPG neural circuits can be considered as distributed systems consisted of nonlinear coupled oscillators. Specific nonlinear oscillators, which consist of neuron models, are called neural oscillators. Rhythmic signals are generated through the phase coupling of oscillators and with the different phase relationships, different types of gaits can be produced by changing coupling methods of oscillators. In the literature, there are several types of oscillators used in the control of robot locomotion.

From the control perspective, motion control network of animals can be counted as a feedforward plus feedback control system. Higher-level central nervous system can be considered as a feedforward controller to send out initial values of the motion parameters when locomotion starts. Sensorial information from the environment and disturbances is used as feedbacks to accomplish the stability of the motion.

2.2.2.1. Types of the Oscillators Used in CPG Models

The oscillators are used in CPG models, should have stable limit cycles for producing rhythmic oscillations.

2.2.2.1.1 Neural Oscillators

2.2.2.1.1.1 Hodgkin-Huxley (H-H) Type Oscillators

H-H type oscillators are found by Hodgkin and Huxley [36, 37]. It uses a squid giant axon preparation to measure membrane potentials and ionic currents and models these currents with a four variable nonlinear system. But, the proposed H-H neuron model is too complicated and has many parameters. This causes more calculation costs. Because of that simplified H-H neuron model is used. The well-known FitzHugh-Nagumo model is one of a simplified H-H type model is defined by the equations [survey 49, 50];

$$\dot{x}_i = c\left(y_i + x_i + \frac{{x_i}^3}{3} + f_{ci}\right), \tag{2.1}$$

$$\dot{y}_i = -\frac{(by_i + x_i - a)}{c}, \tag{2.2}$$

where x_i is the membrane potential of the ith neuron, f_{ci} is the driving signal for neuron i, a, b and c are constants that don't correspond to any particular physiological parameters.

Morris and Lecar [38] also developed a simple H-H type neuron model called Morris-Lecar (M-L) model. Lakshmanan and Murali used a two variable first order autonomous system to build an H-H type neuron model [39]. The difference between this H-H type neuron models are the way of simulation of the neuron behaviors

2.2.2.1.1.2 Stein's Oscillator

Stein's model is defined by the following differential equations [40];

$$\dot{x}_i = a\left(-x_i + \frac{1}{1+e^{(-f_{ci}-by_i+bz_i)}}\right), \tag{2.3}$$

$$\dot{y}_i = x_i - py_i, \tag{2.4}$$

$$\dot{z}_i = x_i - qz_i, \tag{2.5}$$

where x_i is the membrane potential of the *i*th neuron, f_{ci} is the driving signal for neuron *i*, parameter a is a constant affecting the frequency of the oscillator *i*, *b* allows the model to adapt a change in stimulus, *q* and *p* control the rate of this adaptation.

Collins and Richmond [17] had used many non-linear oscillators to study gaits. Stein's model is one of them. Constructed by four coupled Stein oscillators, their CPG control network produced multiple phase-locked oscillation patterns. For quadrupedal robots, these patterns corresponded to several common gaits such as crawl, trot and bound.

2.2.2.1.1.3 Leaky-Integrator Oscillators

The most famous neuron oscillator model with an adaptation item was proposed by Matsuoka [15]. It is also the most used neural oscillator in the literature. A mutual inhibition network consisting of *n* neurons is represented by;

$$T_r\dot{u}_i + u_i = -\sum_{j=1}^{n} \omega_{ij} y_i - \beta v_i + s_i, \tag{2.6}$$

$$T_a\dot{v}_i + v_i = y_i, \tag{2.7}$$

$$y_i = g(u_i) \qquad \big(g(u_i) = max(u_i, 0)\big), \tag{2.8}$$

where u_i is the membrane potential of the ith neuron, $v_i$is a variable representing the degree of the adaptation in the ith neuron, T_r and T_a are the costants of rising time and adaptation time respectively, ω_{ij} is the weight of inhibitory synaptic connection from neuron j to the i, β is the parameter that determines the steady state-firing rate for a constant input and y_i is the output of the neuron.

Matsuoka [14, 15] analyzed mutually inhibiting neurons and found the conditions under which neurons generate oscillations. Because it simulates neuron behavior more precisely, Matsuoka's model has been widely applied locomotion control of legged robots. Taga et al. [16] proposed similar model for generating references for biped robot by using Matsuoka's neuron model. Kimura et al. [18, 19] used neural oscillators for generation references for quadruped robot based on Matsuoka and Taga. This CPG model consists of two mutually inhibiting neurons as shown in the figure 2.8. Each neuron is presented by the following nonlinear differential equations,

$$T_r \dot{u}_i^{\{e,f\}} = -u_i^{\{e,f\}} + \omega_{fe} y_i^{\{e,f\}} - \beta v_i^{\{e,f\}} + s_0 + Feed_i^{\{e,f\}} + \sum_{j=1}^{n} \omega_{ij} y_i^{\{e,f\}}, \quad (2.9)$$

$$T_a \dot{v}_i^{\{e,f\}} = -v_i^{\{e,f\}} + y_i^{\{e,f\}}, \quad (2.10)$$

$$y_i^{\{e,f\}} = max\left(u_i^{\{e,f\}}, 0\right), \quad (2.11)$$

$$y_i = -y_i^{\{e\}} + y_i^{\{f\}}, \quad (2.12)$$

where the suffix e, f and I denote an extensor neuron, a flexor neuron and ith neuron, respectively, $Feed_i$ represent the feedback signals from the robot such as joint angle, angular velocity, etc. $y_i^{\{e\}} and\ y_i^{\{f\}}$ are the outputs of extensor and flexor neurons, ω_{fe} is the connection weight, y_i is output signal of a CPG and the sign of y_i corresponds to the activity of a flexor or extensor neuron.

Neuronal oscillatory models have clear biological meanings, especially with the Matsuoka's model. This kind of models can easily couple the feedback information of environment and the higher level orders. In Matsuoka's model, sensory feedback can be integrated to the CPG network through the term $Feed_i$. The external input term s_0 simulates control signals from the higher level control nervous system. This provides the opportunity to obtain mutual entrainment between the CPG network and the mechanical body. Since so many parameters are involved in these models, selecting proper parameter is essential to the CPG inspired control methods.

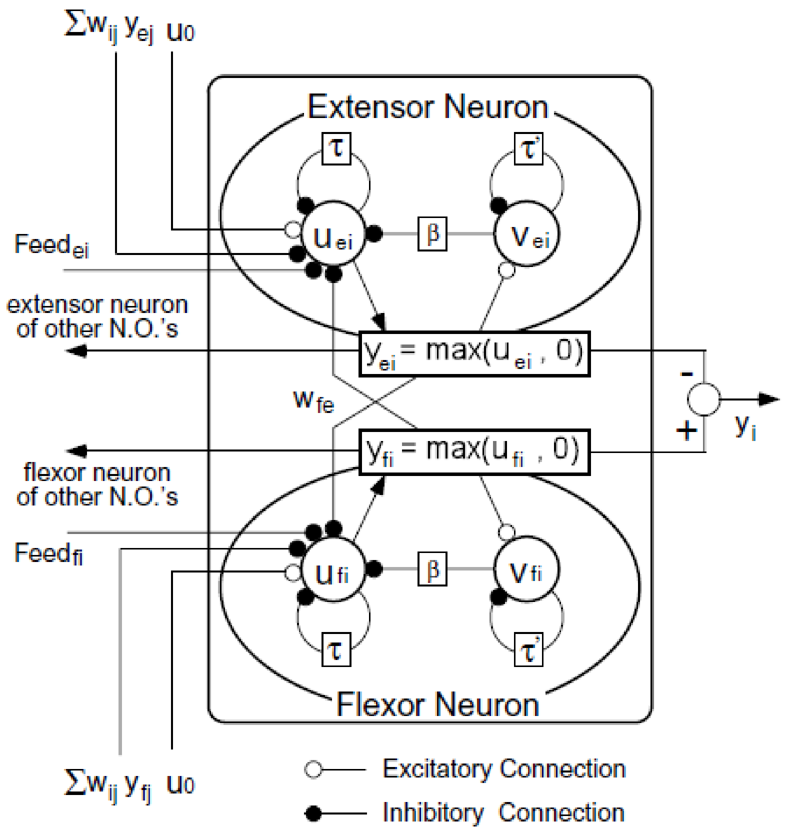


Fig 2.8: Neural oscillator as a model of a CPG.

2.2.2.1.2 Nonlinear Oscillators

In engineering applications, the main task of CPG is to produce periodic oscillations instead of simulating neuron behaviors. Some of the researchers prefer using nonlinear oscillators instead of neural oscillators to simulate CPG.

2.2.2.1.2.1 Phase Oscillators

Phase oscillator is a simple oscillator, whose radius is completely neglected while phase remains. Kuramoto's model is a typical example of this oscillator [41]. It consists of a population of N coupled phase oscillators and the phase oscillators are coupled through the sine function of their phase differences. The mathematical model is as follows,

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^{n} \lambda_{ij} \sin(\theta_j - \theta_i - \Delta\phi_{ij}), \tag{2.13}$$

$$\ddot{r}_i = \mu_i(R_i - r_i) - \frac{3}{2}\mu_i \dot{r}_i, \tag{2.14}$$

$$x_i = r_i(1 + \sin\theta_i), \tag{2.15}$$

where $\theta_i$is the phase of the *i*th oscillator. ω_i is frequency parameter. Each oscillator tries to run independently at its own frequency, while the coupling term λ_{ij} tends to synchronize it to all the others [41]. $\Delta\phi_{ij}$ denotes the desired phase shift between oscillators i and j. amplitude r_i will asymptotically converge to a positive constant R_i. This allows user to smoothly modulate amplitude of the oscillations, μ_i is a positive constant and x_i is the output of the oscillators.

Since it displays a lot of synchronization patterns, Kuramoto's modal has been well studied. A variety of more complex realistic models have been proposed from original Kuramoto's model [42]. When coupling is sufficiently weak, oscillators run incoherently. But after it goes beyond a certain threshold, it sets to limit cycle and collective synchronization emerges spontaneously. This synchronization property of the Kuramoto's model is the main reason why we use it as a CPG unit.

Based on Kuramoto's model, Conradt [43] built a distributed CPG network to control the motion of a serpentine robot and it showed various motion patterns under different environments.

2.2.2.1.2.2 Hopf Oscillator

Hopf oscillator can be described by,

$$\dot{x} = (\mu - r^2)x + \omega y, \tag{2.16}$$

$$\dot{y} = (\mu - r^2)y + \omega x, \tag{2.17}$$

where $r = \sqrt{x^2 + y^2}$, $\mu > 0$ determines the amplitude of the output signals, ω controls the frequency of the oscillator. The oscillator has a stable limit cycle with radius $\sqrt{\mu}$ and angular velocity ω rad/s.

Ijspert et al. [44] demonstrated a programmable CPG model based on Hopf oscillators and applied it to the locomotion control of humanoid robot and quadruped robots. This CPG network can learn arbitrary rhythmic teaching signals. Because of the dynamic characteristic of the programmable CPG network, the learning process was completely embedded to the system. Ijspert [45] validated this CPG control network with a humanoid robot HOAP-2 in simulations. Nicolas [46] studied coupling methods of Hopf oscillators and used two oscillators to control each degree of freedom to get various phase differences. Righetti et al. [47] constructed a locomotion controller with modified Hopf oscillators which could independently control the swing and stance phases of an oscillation. We can control the locomotion speed of robots using ascending phase of the oscillation to control the swing stage and the descending phases to control the stance stage of legs.

2.2.2.1.2.3 Van der Pol's and Rayleigh's Oscillator

Van der Pol's model (VDP) and Rayleigh's model are all relaxation oscillators that can produce various waveform signals. By adjusting the parameters of the nonlinear oscillators, self-sustained limit cycles can be generated.

Van der Pol's analysis of electronics and hear beat is generally credited as the original significant work for modeling biological phenomena with nonlinear oscillators [48]. The basic equation of the VDP oscillator is,

$$\ddot{x} + a(x^2 - p^2)\dot{x} + \omega^2 x = 0, \tag{2.18}$$

where x and $\dot{x}$ describe states of the system and $a > 0$ is the coefficient of the resistance. This resistance is negative for small amplitude of x, as given by $x^2 - p^2$ and is responsible for the generating of self-sustained oscillation.

Rayleigh investigated sound generating principles of the musical instruments and created Rayleigh model [49], which was very similar to VDP oscillator. The basic equation of Rayleigh's model is

$$\ddot{x} + a(\dot{x}^2 - p^2)\dot{x} + \omega^2 x = 0, \tag{2.19}$$

The parameters of Rayleigh oscillator are almost the same as the VDP oscillator's, except that the resistance is negative when amplitude of $\dot{x}$ is small.

As early as 1987, Bay and Hemami [13] made a CPG model with coupled VDP oscillators. By numerical simulations, they analyzed the property of the system and studied the ring and chain connection methods. Bay's early work verified that the VDP oscillator is a good model to simulate CPG. Zielinska [50] used coupled VDP to generate rhythm locomotion control signals for a two legged walking machine. Filho et al. [51] studied the behavior of hips and knees in locomotion and mutually coupled Rayleigh oscillators and VDP oscillators to generate control signals that were similar to human locomotion.

2.2.2.2. Construction of CPG Network for Quadruped Robots Motion Control

CPG network construction is essential part of the CPG based motion control of the quadruped robots. There is a relationship between each leg of the quadruped vertebrates. CPG network construction allows using this relationship for producing quadruped gaits. The most common use of CPG network is ring form of four coupled oscillators. Each oscillator oscillates in the same magnitude and same frequency with a fixed phase difference. By connecting the oscillator of the each leg, mutual entrainment between the four oscillators is maintained. This mutual entrainment between the oscillators of the each leg results in a gait.

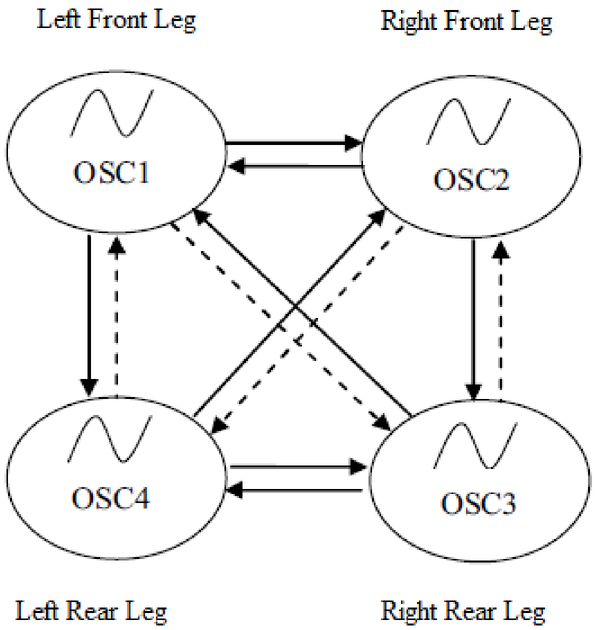


Fig 2.9: Mutual entrainment between the legs.

In the literature, output of the CPG network is used as a reference of the hip joint position, hip joint torques or tip of toe position. There is a connection between the joint of the individual legs. Their reference oscillations have same frequency. When output of the CPG network is used as a reference of the hip joint positions or hip joint torques, the joint references of other legs can be generated from the hip joint references in this relation. The drawback of this method is that there must be additional constraints for specifying the step height and step length. When the output of the CPG network is used as a position reference of the tip of a toe, the joints position references can be determined via inverse kinematic with respect to the body coordinate frame. With this method the parameters of the step height and step length can be determined without any additional constraints but using inverse kinematic can cause additional calculation cost.

2.3. Evolutionary Algorithms

Genetic algorithms belong to evolutionary class of artificial intelligence. It is an optimization technique that imitates natural selection. The process of natural selection and survival of the fittest are considered as important issues in evolution. Genetic Algorithm includes biological ideas such as population of chromosomes, separation of fittest, child production from parents with crossing-over and mutation of chromosomes for variation.

Biological terminology of genetic algorithms correspond to optimization terminology. Population can be taught as set of solution. A chromosome is a structure what represents solution. Generation is transition to a new set of solutions.

Genetic algorithm has a characteristic different from the most of other commonly used optimization methods. First of all, genetic algorithms use codes instead of decision variables. It searches optimum value from a population. Genetic algorithms are a random process, it doesn't use any of the deterministic rules. Binary or symbolic coding is necessary for this type of optimization technique.

New population of the chromosomes can be generated different ways. Some of the elitist chromosomes can be added to the new generation or the chance of selection for each individual can be proportional to its fitness value where fitter solutions are more likely to be selected. Certain selection methods rate the fitness of each solution and select the best solutions. Other methods rate only a random sample of the population, as the former process may be very time-consuming.

The next step is to generate a next generation population of solutions from those selected through genetic operators: crossover and mutation. At the selection step, a random process is used that is biased toward the more fit members of the current population in order to select some of the members to become parents. After crossover, children, which don't comply with optimization constraints are discarded. The mutation process provides the algorithm with diversity. It makes escapes from local extremum points possible.

2.4. Example Quadruped Projects

The field of quadrupedal locomotion and quadruped walking robots is considered as a promising area. Quadrupeds have high mobility on natural terrains unlike wheeled robots which need a continuous support surface. Quadruped robots use discrete footholds for each foot. Therefore quadruped robots can walk on irregular terrain by changing their leg configuration in order to adapt themselves to surface irregularities. In addition, feet may contact the ground at properly selected points. Over soft surfaces, e.g. on sandy soil, the ability to use discrete footholds on the ground can also improve the energy consumption. This is because they deform the terrain less than wheeled or tracked vehicles [52]. Quadruped robots can be used for search and rescuing tasks, for carrying loads on irregular terrain, for military applications and in places where human beings can't live such as other planets or places with high radiation rates.

One of the first vehicles that was able to adopt different gaits was the General Electric quadruped (Figure 2.10), developed by R. Mosher and finished in 1968 [53]. This vehicle has a height of 3.3 m, a length of 3 m and a weight of 1400 kg. Each of the four legs has three DOFs. One of them is located in the knee and two of them are located in the hip joint. Each joint was actuated by a hydraulic cylinder and was propelled by a 68 kW internal combustion engine. Machine control was dependent on a well trained operator in order to function properly. The operator controlled the four legs through four joysticks and pedals that were hydraulically connected to the robot legs, with force reflection. The vehicle control was very important but it was very difficult because of the number of the DOF. Therefore few people were able to operate it. Although it demonstrated an ability to overcome obstacles and had good mobility on difficult terrain, it became clear that it needed a computer control system.

The other important past day quadruped robot is the Phoney Poney (Figure 2.10) was developed by McGhee and Frank [54] around the same time with General Electric quadruped. Phoney Poney, completed in 1966, was the first legged robot to move autonomously under computer control and with electrical actuation. Each leg had two DoF and each of its joints being actuated through an electrical motor. The joint coordination was performed through simple digital logic and presented two different gaits. Its main limitation was the fact that it only moved in a straight line, not being able to turn.

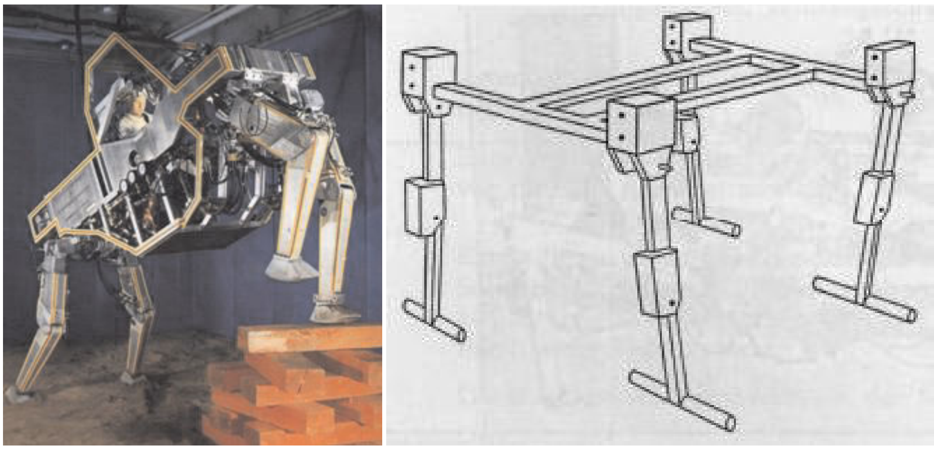


Figure 2.10. General Electric quadruped (left) and Phoney Poney quadruped (right).

Shigeo Hirose made great contributions for the development of legged robotic field. In the late 70s, he constructed KUMO I, which has a leg length of 1.5 meters and a weight of 14 kg. He also constructed PV-II, with a leg length of 0.9 meters and weight of 10 kg. After these studies, he developed the TITAN quadruped robot series in the Tokyo Institute of Technology [55].TITAN I - IV quadruped robots were constructed in the 80s and TITAN V - XI quadruped robots were developed in the 90s. TITAN VIII was constructed on 1996. It had three DOFs in each leg and walked in a walking posture jutting out its legs to each side [56].

Figure 2.11: Quadruped robots KUMO I (left) and PV II (right)

Four legged walking machines, Collie1 and Collie 2, were developed by H. Miura at the University of Tokyo in the middle of the 1980s. The machines had 3 degrees of freedom per leg. A potentiometer is mounted on each joint. DC servo motors are mounted on 2 joints around a pitch axis and 1 joint around a roll axis [57].

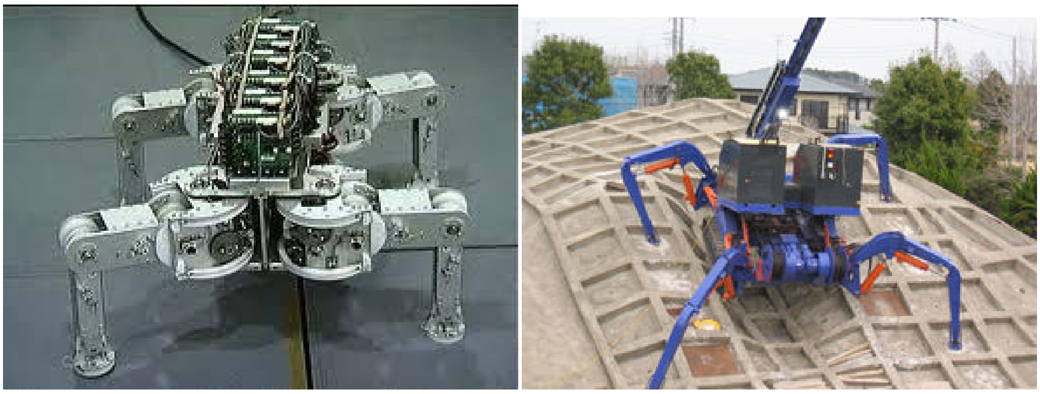


Figure 2.12: Quadruped robots TITAN VIII (left) and TITAN XI (right)

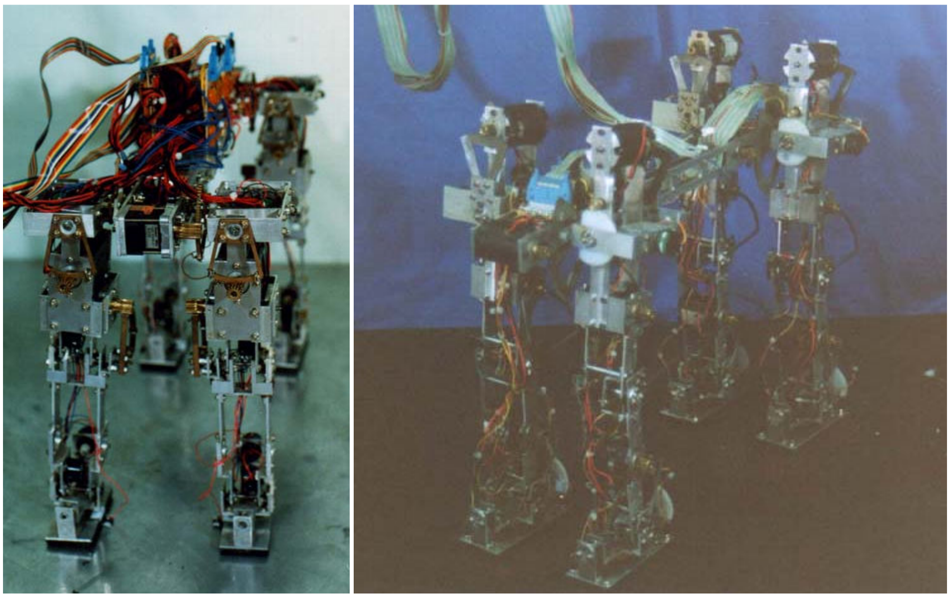


Figure 2.13: Quadruped robots developed by University of Tokyo: Collie 1-2

SCOUT, constructed at McGill University had only one degree of freedom per leg. Leg lengths and the body length was 20cm. The body width was 14.5cm at the front and 19cm at the back. The body's mass was 1.2kg First prototype SCOUT-1 is capable of walking, turning, and climbing steps, despite its mechanical simplicity [58].

The quadruped BISAM was built at University of Karlsruhe with three DOFs on each leg. It can rotate in the sagittal plane for mammalian gaits, and, in the transverse plane, for reptilian gaits. The articulation in the spine was investigated using a biologically inspired adaptive control concept [59].

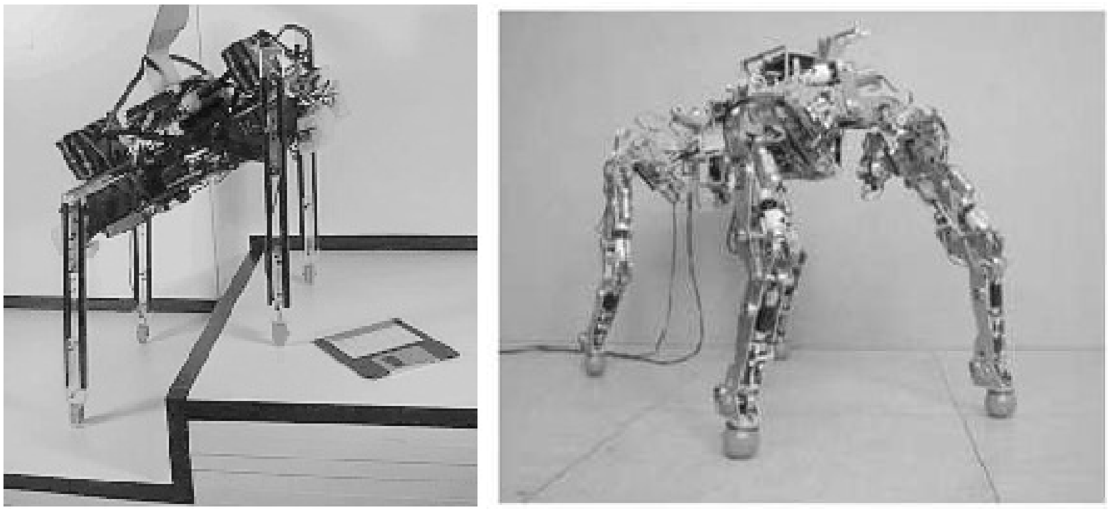


Figure 2.14: Quadruped robots Scout1 (left) and BISAM (right)

SILO4, constructed at Spanish Ministry of Science and Technology, had three active and one passive DOF on the each leg. Its construction purpose is mine-sweeping. The SILO4 was conceived as an indoor walking robot, but it can work in an outdoor environment under non extreme conditions. That means, for instance, that the robot can work on highly irregular terrain but not under rainy conditions [60].

AIBO, the quadruped pet-like robot, is developed by Sony. It has three DOFs on each leg with touch sensors and three acceleration sensors for posture control. It has three DOFS on its neck and has a camera and a microphone on its head. Aibo was used in Robocup tournaments for long years [61].

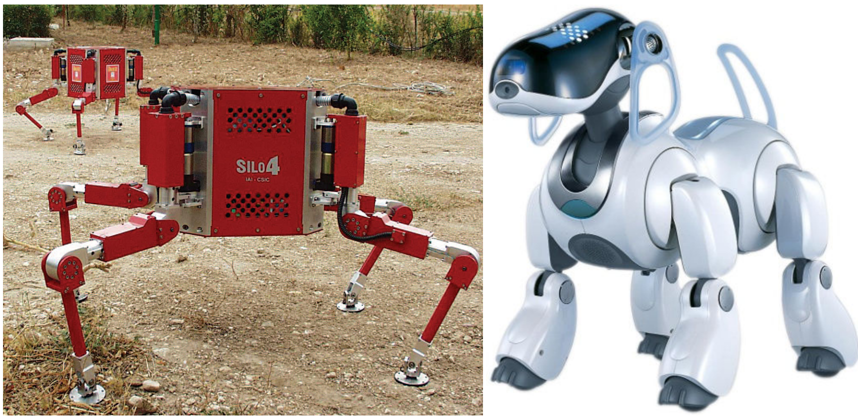


Figure 2.15: Quadruped robots SILO4 (left) and Aibo (right)

Kimura et al. constructed the quadruped robot Tekken at the University of Electro-Communications. The length of the body is 23 cm and a leg measures 20 cm. The weight of the robot is 3.1 kg. Each leg has a hip pitch joint, a hip yaw joint, a knee pitch joint, and an ankle pitch joint [19].

The quadruped robot HyQ was developed at the Italian Institute of Technology. It has three DOFs on each leg. It can move on both regular and irregular terrains. It has a hydraulic actuation system [62].

BigDog is a quadrupedal robot created in 2005. It is funded by the DARPA. It is designed to carry 150 kg load alongside a soldier at 6.4 km/h and traversing rough terrain at inclines up to 35 degrees [63].

LittleDog is a small quadruped robot developed for DARPA by Boston Dynamics for research. Unlike BigDog LittleDog is in use for a test robot for other institutions. [64].

Figure 2.16: Quadruped robots Little Dog (left) and BigDog (right)

The Cheetah is a quadruped robot that gallops at 45.06 km/h, which has a land speed record for legged robots since August 2012. The previous record was 21.08 km/h, set in 1989 at MIT. Cheetah development is funded by DARPA's Maximum Mobility and Manipulation program. This robot has an articulated back that flexes back and forth on each step.

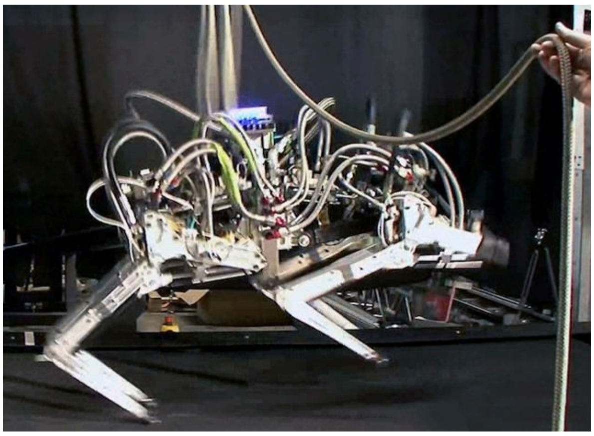


Figure 2.17: Cheetah Robot

Chapter 3

3. THE QUADRUPED SIMULATION MODEL

Quadruped robot model employed in this thesis has four DOFs at each leg. There are two DOFs at hip joint and one DOF at knee joint and ankle joint. Figure 3.1 presents the kinematic arrangement of the quadruped robot model. Design and drawing of the quadruped robot model is done with Solidworks. Link lengths, joint dimensions and their masses are shown in Table 3.1 . A drawing of the quadruped robot and its view in the animation environment can be seen in the Figure 3.2 .

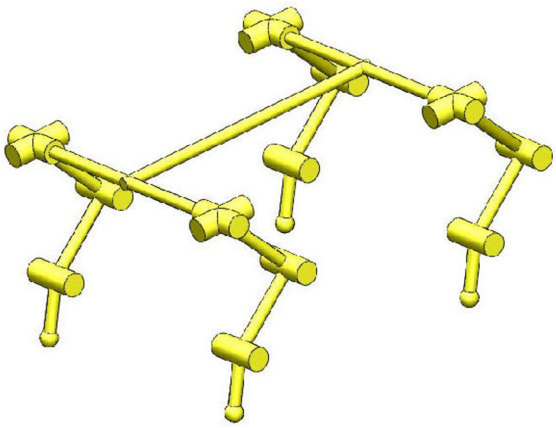


Figure 3.1: Kinematic arrangement of the quadruped robot model

Table 3.1: Link lengths, joint dimensions and their masses

Joint	Dimensions(LxWxH) [m]	Mass[kg]
Trunk	1.2x0.6x0.15	50
Thigh	0.28x0.05x0.1	4.8
Shank	0.27x0.05x0.1	3.85
Ankle	0.22x0.05x0.5	3.85

A Newton-Euler method based full-dynamics 3D simulation and animation environment is used for simulation studies. In this system, an adaptive penalty based method is employed to model the ground contact. The details of the simulation algorithm and the contact model can be found in [65].

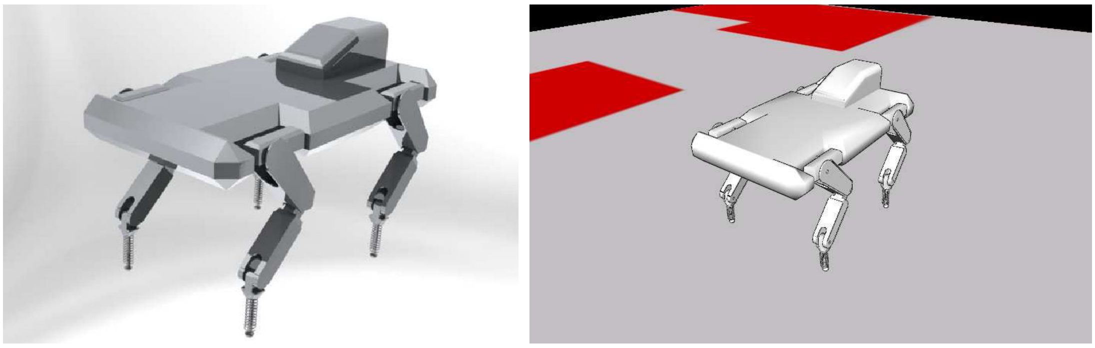


Figure 3.2: A drawing of the quadruped robot and animation view

3.1. Inverse Kinematic Calculation of the Quadruped Robot Model

Solving inverse kinematic problem is very important issue for legged robot locomotion. It is a key process of the reference generation. Position of the toe tip affects stability of the gait directly. Necessary joint trajectories are found from suitable tip of toe references via inverse kinematics. In the CPG based reference generation method used in this thesis, toe tip references are generated with a non-linear oscillator and joint references are found via inverse kinematics.

Position of the toe tip is identified with x_{target}, y_{target} and z_{target} parameters for position and μ_{target} parameter for orientation with respect to the hip coordinate frame.

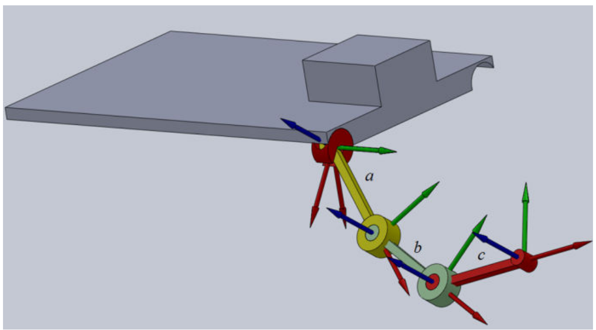


Figure 3.3: A model prepared for inverse kinematic calculation

Figure 3.3 presents coordinate frames of the each joint. x, y and z axes are represented with red, green and blue arrows, respectively. Since the hip joint has two DOFs, z axis of the hip1 joint and y axis of the hip2 joint are overlapped. Using the front view of the model, the joint angle q_1 of hip1 joint can be found easily with the atan2 function:

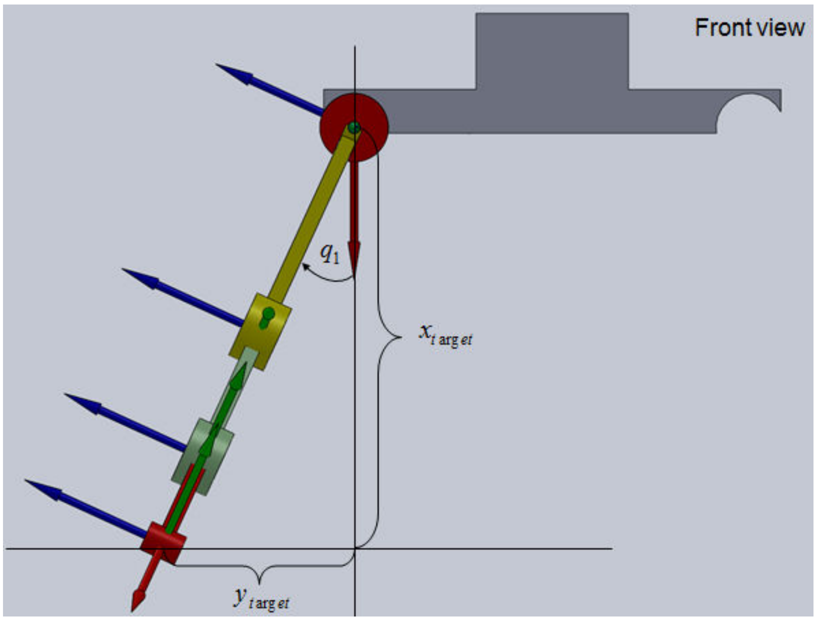


Figure 3.4: Front view of the model

Rest of the angles can be found from the planar view. But some rotation matrixes and positions should be obtained for this process. These can be found side view of the model.

$$q_1 = \text{atan2}\left(y_{target}, x_{target}\right), \tag{3.1}$$

$$R_0^4 = \begin{bmatrix} \cos(-\mu) & 0 & \sin(-\mu) \\ 0 & 1 & 0 \\ -\sin(-\mu) & 0 & \cos(-\mu) \end{bmatrix} \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 \\ \sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

The distance d_0^3 is obtained as;

$$d_0^3 = d_0^4 - c\begin{bmatrix} \cos(\mu)\cos(q_1) \\ \cos(q_1) \\ \cos(\mu) \end{bmatrix} = \begin{bmatrix} x_{target} \\ y_{target} \\ z_{target} \end{bmatrix} - c\begin{bmatrix} \cos(\mu)\cos(q_1) \\ \cos(q_1) \\ \cos(\mu) \end{bmatrix} \equiv \begin{bmatrix} x_{ankle} \\ y_{ankle} \\ z_{ankle} \end{bmatrix}. \tag{3.3}$$

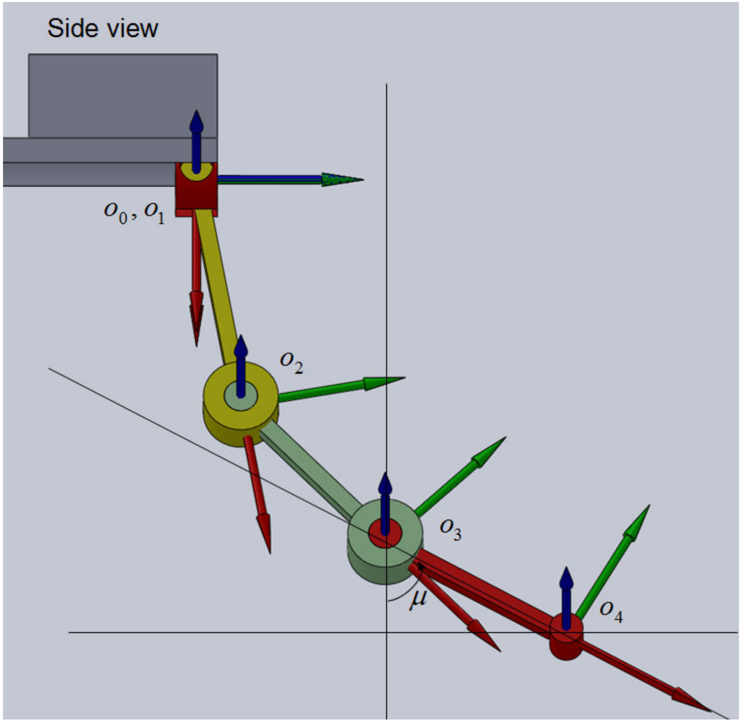


Figure 3.5: Side view of the model

After these calculations, ankle origin can be expressed with respect to the hip 2 frame. Rest of the angles can be found like solving inverse kinematic of planar arm robot. R_1^0 rotation matrix can be found with the 90 degrees twist parameter α_1.

$$R_0^1 = \begin{bmatrix} c_{q_1} & 0 & s_{q_1} \\ s_{q_1} & 0 & -c_{q_1} \\ 0 & 1 & 0 \end{bmatrix}. \tag{3.4}$$

Hence;

$$\begin{bmatrix} \overline{x}_{ankle} \\ \overline{y}_{ankle} \\ \overline{z}_{ankle} \end{bmatrix} \equiv d_1^3 = R_1^0 d_0^3 = \begin{bmatrix} c_{q_1} & s_{q_1} & 0 \\ 0 & 0 & 1 \\ s_{q_1} & -c_{q_1} & 0 \end{bmatrix} \begin{bmatrix} x_{ankle} \\ y_{ankle} \\ z_{ankle} \end{bmatrix}, \tag{3.5}$$

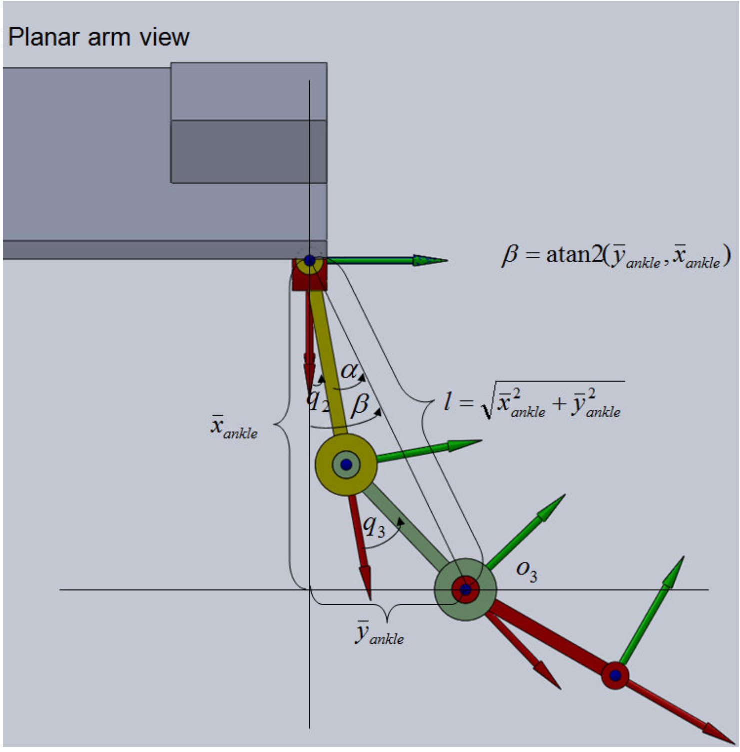


Figure 3.6: Planar view of the model

$\sin q_3$ and $\cos q_3$ can be found geometrically from the planar view of the model. Using atan2 q_2 and q_3 angles can be obtained.

$$l^2 = a^2 + b^2 + 2ab\cos(q_3), \tag{3.6}$$

$$\cos(q_3) = \frac{l^2 - a^2 - b^2}{2ab}, \tag{3.7}$$

and

$$\sin(q_3) = \pm\sqrt{1 - \cos(q_3)^2}. \tag{3.8}$$

Using the atan2 function q_3 is obtained as,

$$q_3 = \text{atan2}(\sin(q_3), \cos(q_3)). \tag{3.9}$$

Using atan2 function for finding the angle α of Figure 3.6 as

$$\alpha = \text{atan2}(b\sin(q_3), a + b\cos(q_3)), \tag{3.10}$$

q_2 is obtained from,

$$q_2 = \beta - \alpha, \tag{3.11}$$

where β is an angle shown in Figure 3.6.

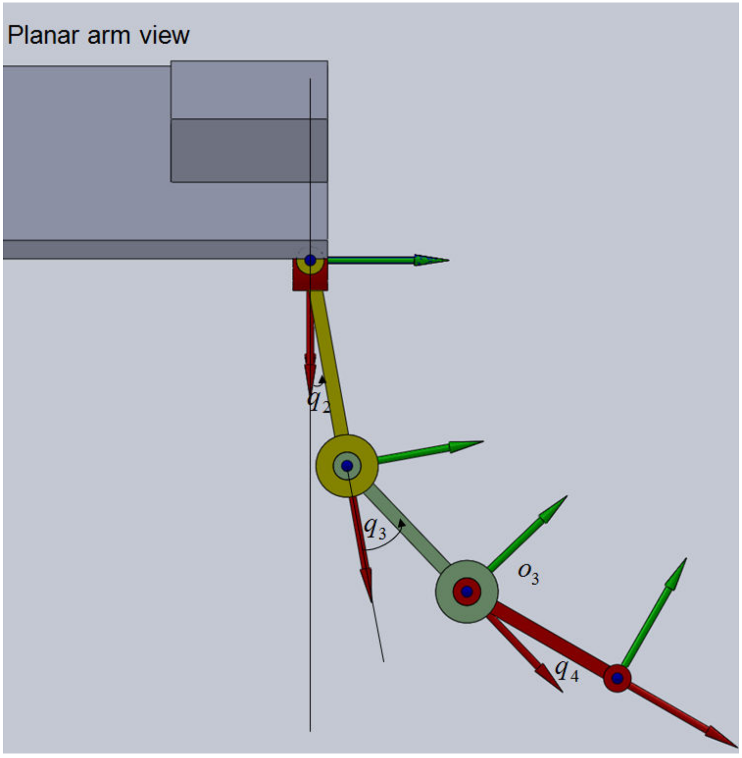


Figure 3.7: Planar view of the model

The rotation matrix between the first and fourth frame is given at the below:

$$R_1^4 = \begin{bmatrix} \cos(q_2+q_3+q_4) & -\sin(q_2+q_3+q_4) & 0 \\ \sin(q_2+q_3+q_4) & \cos(q_2+q_3+q_4) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.12}$$

Defining the angle γ as,

$$\gamma \equiv q_2 + q_3 + q_4, \tag{3.13}$$

and computing this angle using atan2 function,

$$\gamma = \text{atan2}(x_{4_z}, c_{q_1} x_{4_x} + s_{q_1} x_{4_y}). \tag{3.14}$$

the last joint angle is computed:

$$q_4 = \gamma - q_2 - q_3. \tag{3.15}$$

3.2. Dynamics of the Quadruped Robot Model

The quadruped robot used in this thesis has 4 degrees of freedom in each leg, as explained in the previous chapter. Its dynamics can be described by the following equation,

$$\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{pmatrix} \dot{v}_B \\ \dot{\omega}_B \\ \ddot{\theta} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \begin{pmatrix} u_{E_1} \\ u_{E_2} \\ u_{E_3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \tau \end{pmatrix} \tag{3.16}$$

where H_{ij} for $(i, j) \in \{1,2,3\}$ are sub-matrices of the robot inertia matrix. v_B is the linear velocity of the robot body coordinate frame center with respect to a fixed world coordinate frame, ω_B is the angular velocity of the robot body coordinate frame with respect to a fixed world coordinate frame, and θ is the vector of joint movements of the quadruped. The vector formed by augmenting b_1, b_2, and b_3 is termed as the bias vector in this dynamics equation. u_{E_1} is the net force effect and u_{E_2} is the net torque effect of the reaction forces on the robot body. u_{E_3} represents the effect of reaction forces on the robot joints. Reactive forces are generated by environmental interaction. τ is the generalized joint control vector, typically consisting of joint actuation torques for a robot with revolute joints. H_{11}, H_{12}, H_{21}, and H_{22} are 3×3 matrices. For a 16 DoF robot with 4 DoF at each leg, as described in the previous chapter, H_{13} is 3×16, H_{23} is 3×16, H_{31} is 16×3, H_{32} is 16×3, and H_{33} is 16×16.

3.3. The Ground Contact Model

In this section the ground contact model is described. A penalty based approach is used with fictitious spring and dampers for modeling contact forces. Consider a box that is positioned over a flat surface. The important parameters for the contact model are the mass of the box, the initial height and spring and damper parameters. Let us assume that the box is equipped with imaginary spring and dampers at its four corners as shown in Fig. 3.8 and that the spring and damper penalty plan uses the following rule in the vertical direction.

$$f_{z_i} = \begin{cases} -K_z z_i - B_z v_{z_i} & \text{if } z_i \le 0 \\ 0 & \text{if } z_i > 0 \end{cases} \tag{3.17}$$

Here, i is an index number indicating the specific corner on which the force f_{z_i} is exerted by the ground in the z -direction. K_z is a spring constant and B_z is a viscous friction coefficient. z_i is the z coordinate of the corner i measured from the ground level.

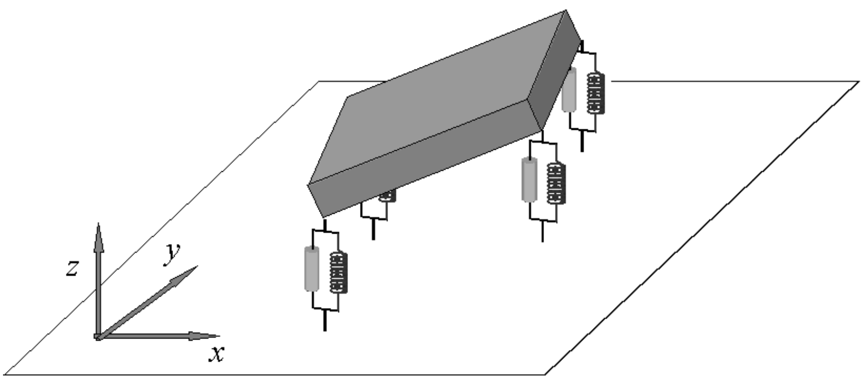


Figure 3.8: The vertical spring and dampers

When a corner of the box is in contact with the ground, or has moved into the ground, a horizontal friction effect should be considered too. This friction is actually the force behind horizontal motion of the legged robots. It can be generated by the following expression.

$$f_{x,y_i} = \begin{cases} \dfrac{-f_{z_i} B_{x,y} \begin{bmatrix} v_{x_i} & v_{y_i} \end{bmatrix}^T}{\left\| \begin{bmatrix} v_{x_i} & v_{y_i} \end{bmatrix}^T \right\| + \varepsilon} & \text{if } z_i \leq 0 \\ 0 & \text{if } z_i > 0 \end{cases} \tag{3.18}$$

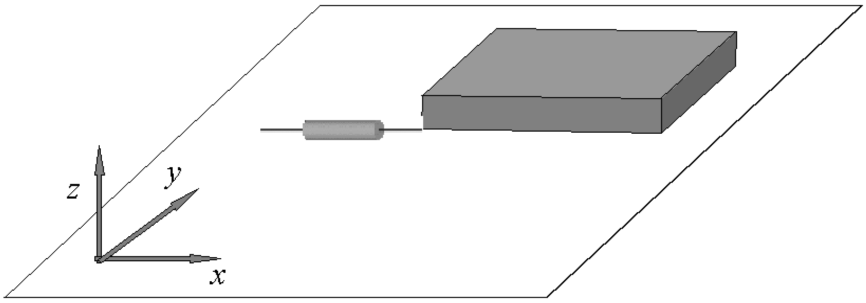


Figure 3.9: Horizontal friction element

Here, $f_{x.y_i}$ is the horizontal two dimensional force vector acting on the corner i which is in contact with the ground, $B_{x,y}$ is a friction coefficient, and v_{x_i} and v_{y_i} are the velocities of the corner i in the x and y directions, respectively. ε is a very small constant used to avoid overflow in software implementation. It can be observed that the expression above generates a force proportional to the vertical force acting on the corner in a direction opposite to the motion of the corner. As a modification of this horizontal force computation plan, a spring can be added to the system as follows.

$$f_{x,y_i} = \begin{cases} \dfrac{-f_{z_i} B_{x,y} \begin{bmatrix} v_{x_i} & v_{y_i} \end{bmatrix}^T}{\left\| \begin{bmatrix} v_{x_i} & v_{y_i} \end{bmatrix}^T \right\| + \varepsilon} - K_{x,y} \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \end{bmatrix}^T & \text{if } z_i \leq 0 \\ 0 & \text{if } z_i > 0 \end{cases} \tag{3.19}$$

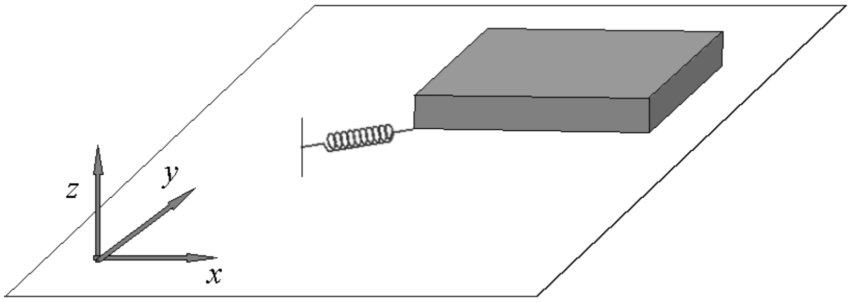


Figure 3.10: The horizontal anti-slip spring element

Here, $K_{x,y}$ is a spring coefficient acting in the horizontal direction. $\bar{x}_i$ and $\bar{y}_i$ are the amounts of slip in the two horizontal directions, computed as $\bar{x}_i = x_i - \hat{x}_i$ and $\bar{y}_i = y_i - \hat{y}_i$, respectively. $[\hat{x}_i \quad \hat{y}_i]^T$ is the position of the corner i on the horizontal plane at the time when the contact is established, and it is updated at every such time point in the simulation. All of the contact modeling parameters are constants, except K_z, which is adapted during the simulation with a energy minimization based optimization technique [65].

Chapter 4

4. APPLICATION OF NONLINEAR OSCILLATOR BASED CPG ON QUADRUPED LOCOMOTION WITH MANUAL TUNING

In this chapter the CPG based reference generation is applied on quadruped robot in the simulation environment which is described at the previous chapter. A Non-linear type phase oscillator is used for the reference generation process. Parameters of the phase oscillator are tuned by manually. Trial and error iterations used for obtaining suitable parameters. Initial conditions of parameters are specified from the other CPG based reference generation researches on quadruped robots. The output of the oscillators is specified as a position reference for a toe tip at the task space. Joint references of the each leg are determined by using inverse kinematics.

4.1. Construction of the CPG Based Reference Generation Network

The phase oscillator is a simple structured one. The radius is completely neglected, only the phase is retained at the output of the phase oscillator. The Kuramoto model [Liu reference 22-23] is used as a phase oscillator which is defined by the equations;

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^{n} \lambda_{ij} \sin\left(\theta_j - \theta_i - \Delta\phi_{ij}\right), \tag{4.1}$$

$$\ddot{r}_i = \mu_i (R_i - r_i) - \frac{3}{2}\mu_i \dot{r}_i, \tag{4.2}$$

$$x_i = r_i (1 + \sin\theta_i), \tag{4.3}$$

where θ_i is the phase of the *i*th oscillator. ω_i is the frequency parameter. Each oscillator tries to run independently at its own frequency, while the coupling term λ_{ij} tends to synchronize it to all the others [41]. $\Delta\phi_{ij}$ denotes the desired phase shift between oscillators i and j. The amplitude r_i will asymptotically converge to a positive constant R_i. This allows the user to smoothly modulate the amplitude of the oscillations. μ_i is a positive constant and x_i denotes the outputs of the oscillators.

Such a CPG model has stable limit cycle behavior and the output forms can be adjusted by explicit parameters. For example, the amplitude and frequency can be modulated easily as shown in Figure 4.1.

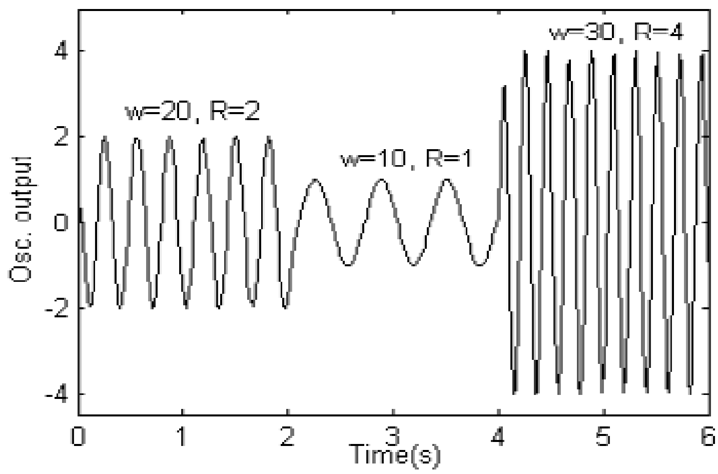


Fig. 4.1: Change in frequency and amplitude of the oscillator

Mammal quadruped gait types are discussed in Chapter 2. In this chapter walk (crawl) and trot are used for simulations. These gaits are shown.

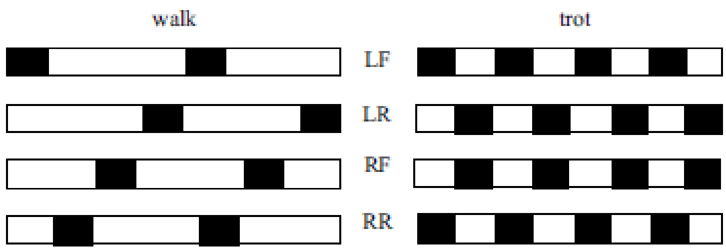


Fig. 4.2: The phase of typical gaits (black: stance phase, white: swing phase)

The CPG is constructed as a network of four coupled phase oscillators. The oscillators control toe tips. By connecting the oscillator of each leg, oscillators are mutually entrained. They become coupled and oscillate with the same period and with a fixed phase difference. This mutual entrainment between the oscillators of the legs results a gait. And by changing the phase difference between the oscillators; walk and trot gaits are obtained.

The connection between the oscillators is described by a connection matrix. The phase differences between neighbor oscillators are set negative for the descending connections and positive for the ascending connections.

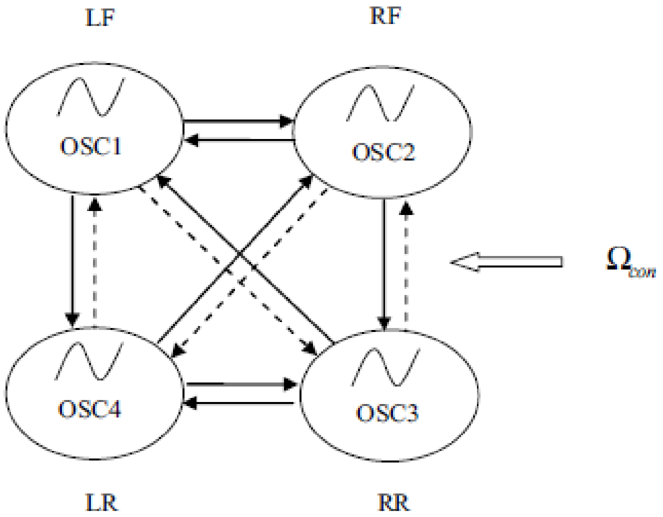
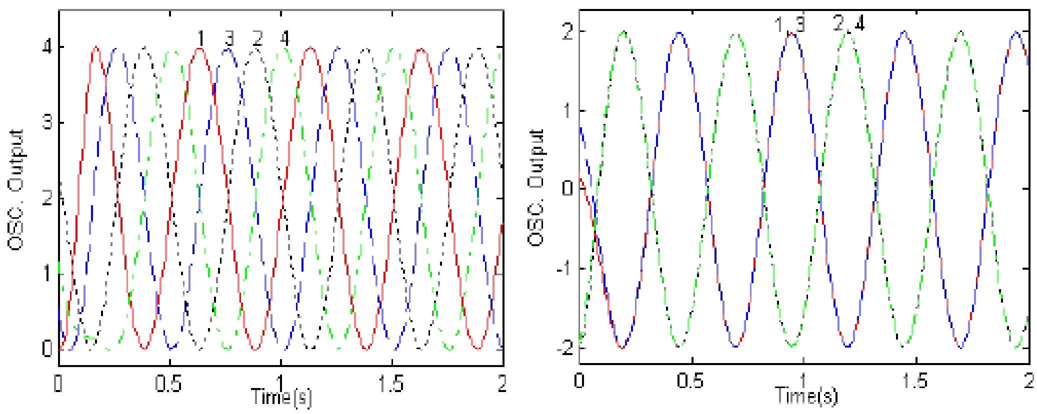


Fig. 4.3: Locomotion network and connection matrix.



$$\Omega_{con.walk} = \begin{bmatrix} 0 & \pi & \pi/2 & 3\pi/2 \\ -\pi & 0 & -\pi/2 & \pi/2 \\ -\pi/2 & \pi/2 & 0 & \pi \\ -3\pi/2 & -\pi/2 & -\pi & 0 \end{bmatrix}$$

(a) waking pattern

$$\Omega_{con.trot} = \begin{bmatrix} 0 & \pi & 0 & \pi \\ \pi & 0 & \pi & 0 \\ 0 & \pi & 0 & \pi \\ \pi & 0 & \pi & 0 \end{bmatrix}$$

(b) trotting pattern

Fig. 4.4: Walk and trot gaits patterns and the connection matrixes

4.2. Simulation Results of Crawl and Trot Gaits

Each leg oscillator has two parts at the Cartesian coordinate frame. Tip of toe references are generated with respect to x axis and z axis. There is no varying reference with respect to y axis. Generated tip of toe references are 2D references.

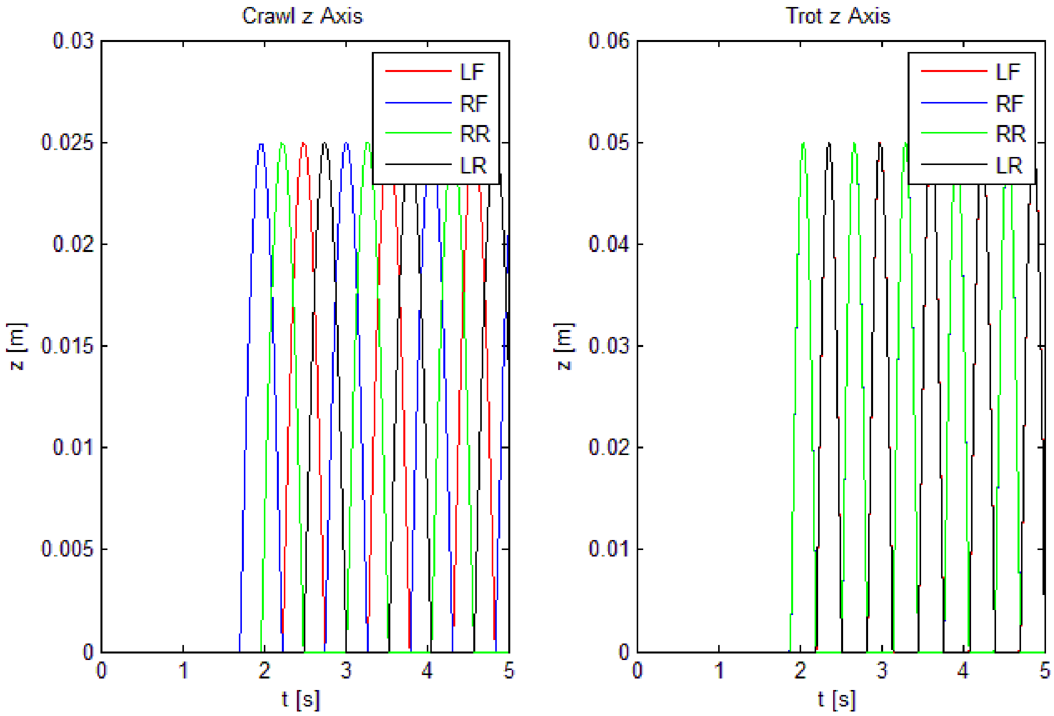


Fig. 4.5: Tip of to trajectories with respect to z axis

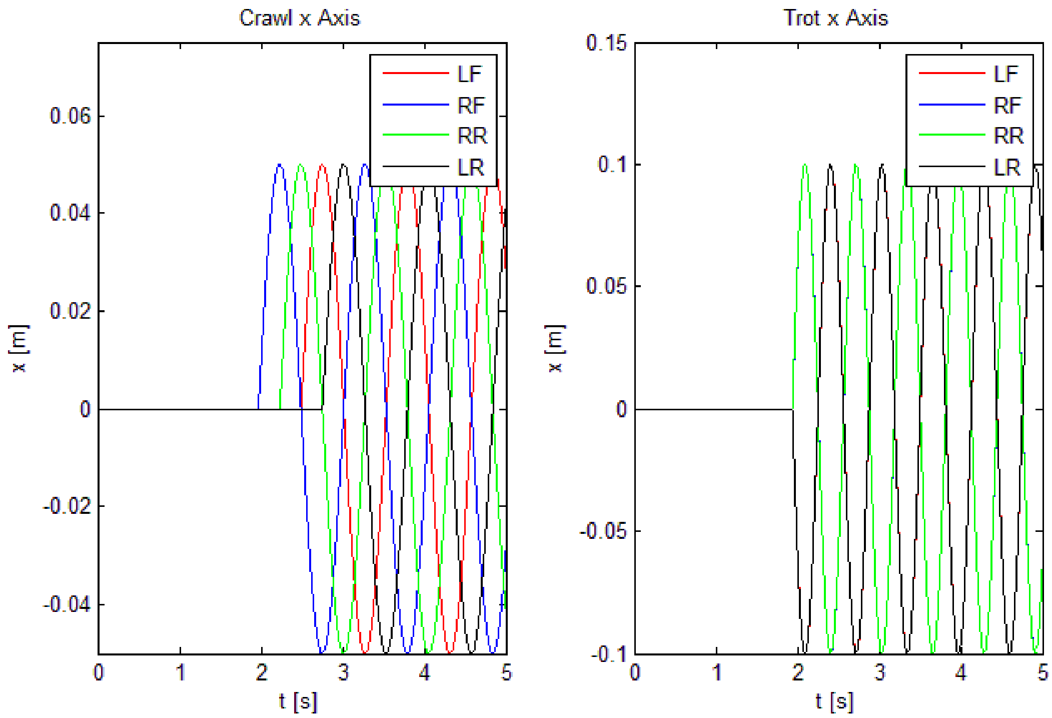


Fig. 4.6: Tip of to trajectories with respect to x axis

There is a phase difference between the outputs of the z axis and x axis. When the robot leg is in swing phase, it should move forward with respect to the x axis and when it is in stance phase, it should move backward with respect to the x axis. Otherwise, it is difficult for the quadruped robot to move forward. This situation can be achieved when the phase difference value equals to $\frac{\pi}{2}$.

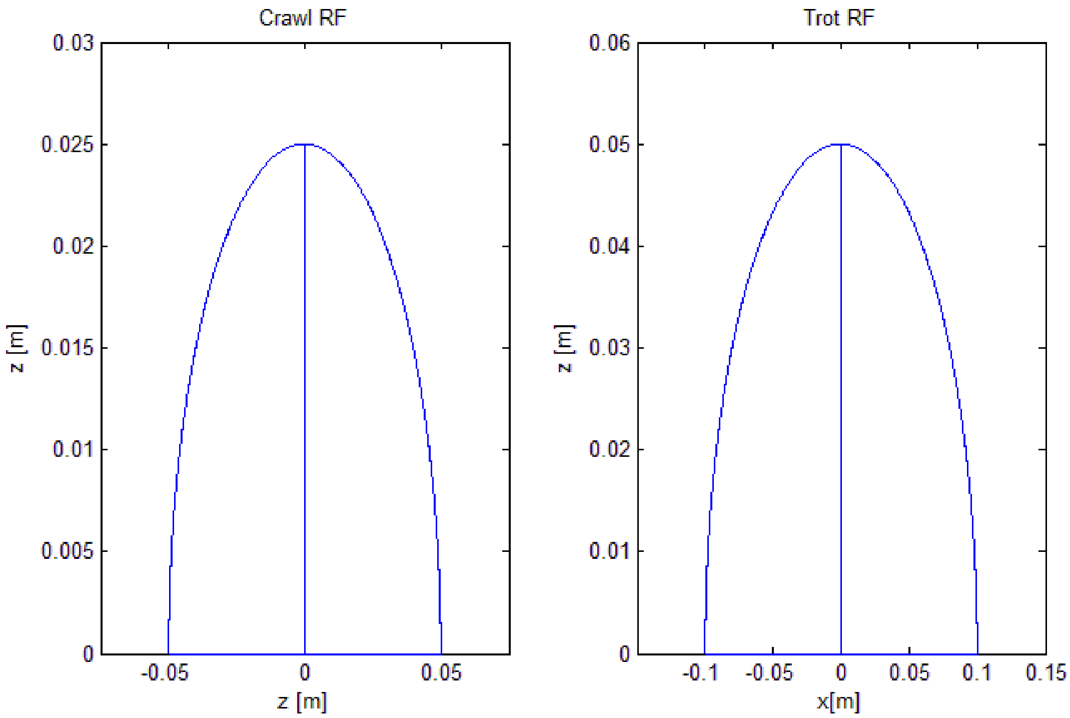


Fig. 4.7: Tip of toe trajectory with respect to Cartesian coordinate frame

As a result of trial and error iterations, the step height is selected as 0.025 m, the step size is selected as 0.1 m and frequency is selected as 0.96 s^{-1} for the crawl gait. The step height is 0.05 m, the step size is 0.2 m and the frequency is 1.6 s^{-1} for trot gait.

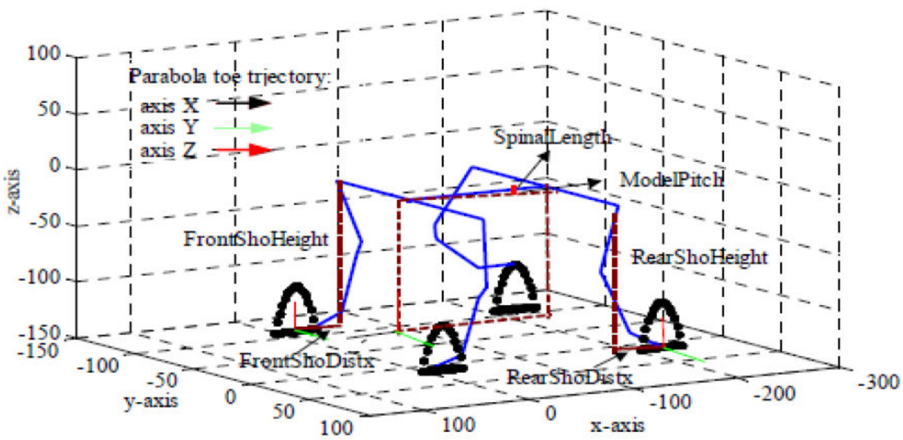


Fig. 4:8 CPG based reference generation for quadruped robot on the task space

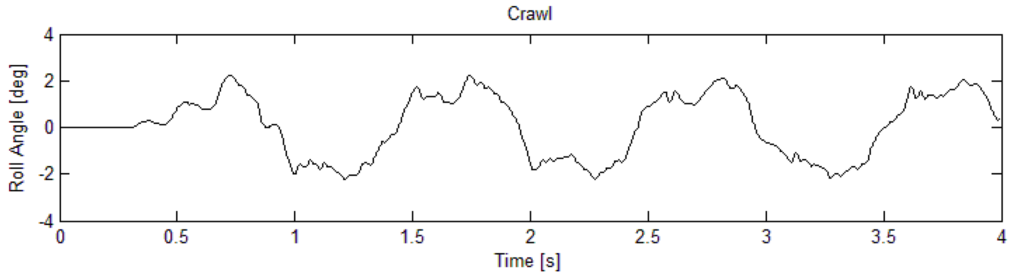


Fig. 4.9: Roll and pitch angles of the body during the crawl gait

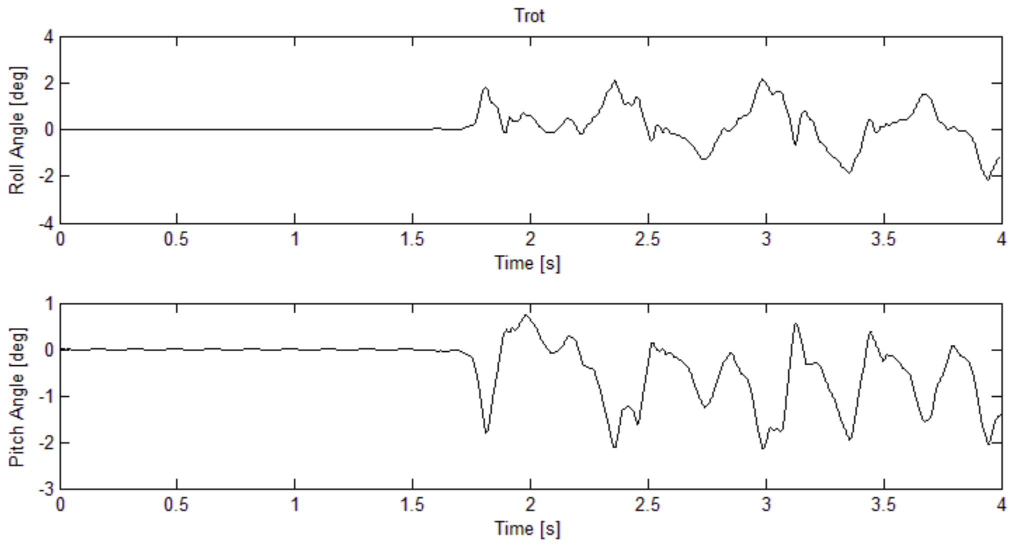


Fig. 4.10: Roll and pitch angles of the body during the trot gait

4.3 Gait Transition with Manual Tuning

Quadruped walking animals can easily change their gait patterns to suite to the environment, but shifting locomotion patterns is a difficult task for a legged-robot. The CPG based reference generation provides an opportunity for gait transition. Two methods are usually used to realize the gait transitions for a CPG driven system: One is changing the driving signal another is changing coupling configuration [19].

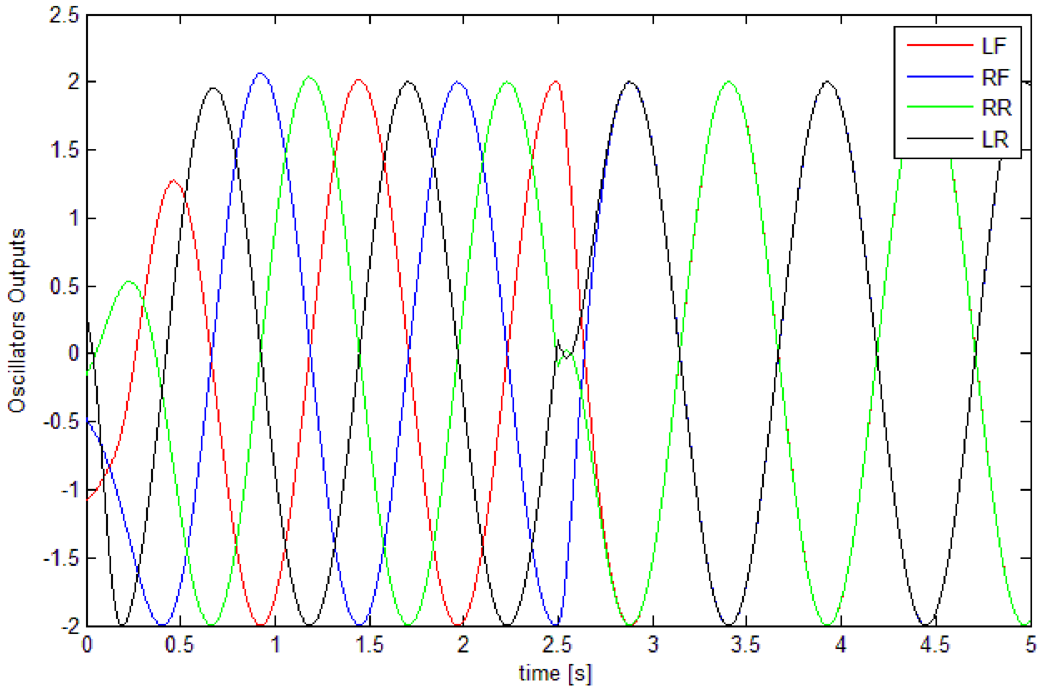


Fig. 4.11: Crawl to trot gait pattern

4.4 Simulation Results of Gait Transition

In this study, gait transition is realized by switching the coupling configuration which defines a connection matrix. Manually tuned parameters are used in the gait transition process. Step height and step size parameters are selected the same for crawl and trot gaits in the gait transition process. Gait transition takes places at the fourth second during the simulation.

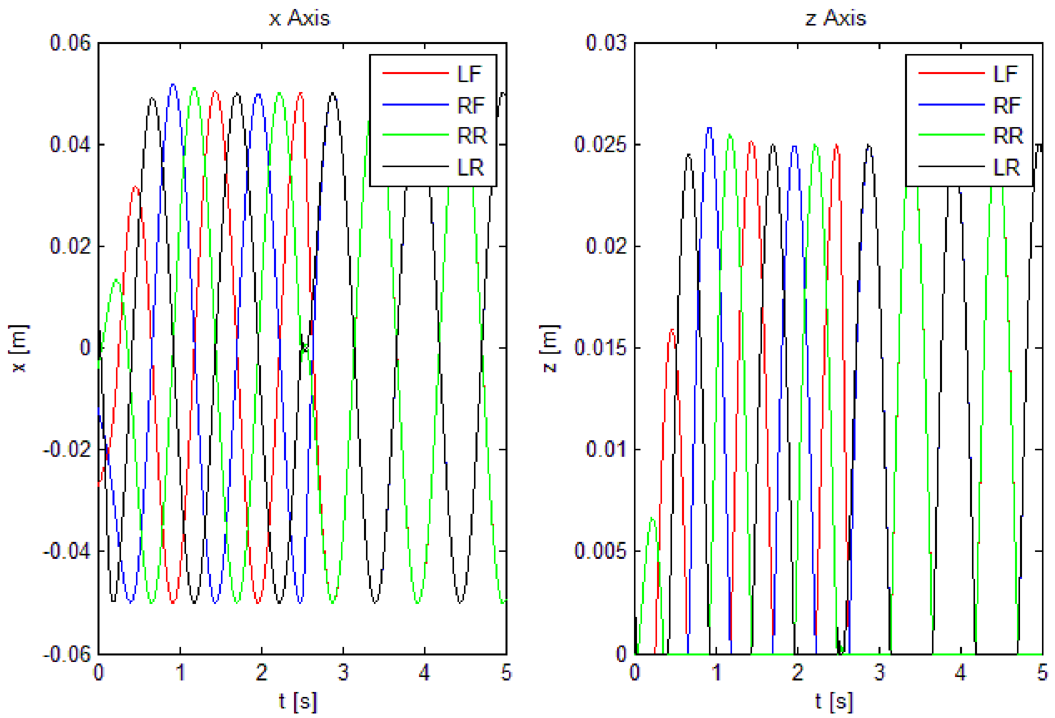


Fig. 4.12: Tip of to trajectories with respect to x axis and z axis

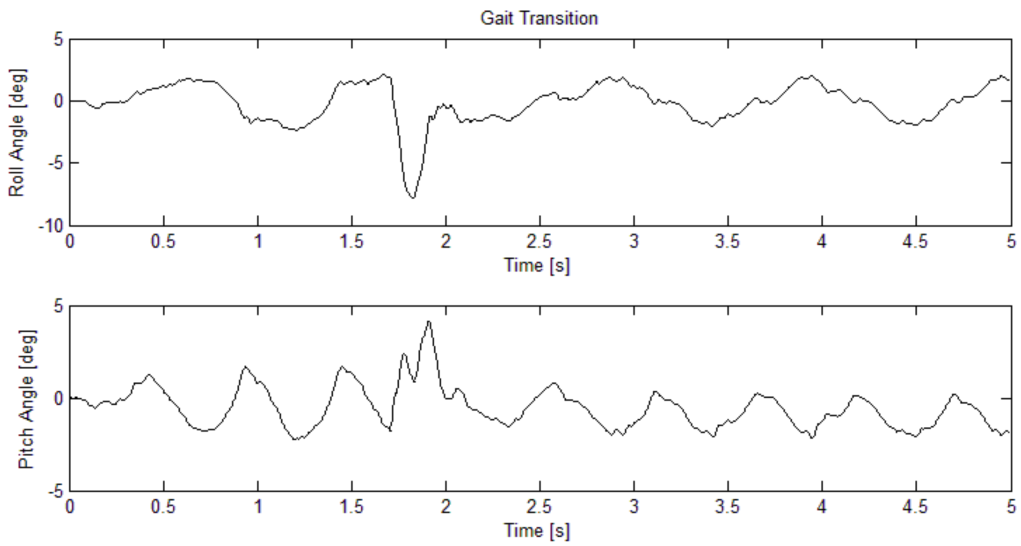


Fig. 4.13: Roll and pitch angles of the body

4.5 Discussion

3D dynamic simulations are realized for a quadruped robot. CPG based reference generation is used for gait planning. Crawl and trot gait simulations are done individually and crawl to trot gait transition is realized, phase oscillator, which is a type of non-linear oscillator, is used for CPG network. Thanks to the stable limit cycle behavior, stable gait patterns are obtained. Until the output of the oscillator reaches the limit cycle from the initial conditions, the reference of the toe tip is kept steady at zero at crawl and trot gait simulations. This method isn't used for crawl to trot gait transition simulation. This is an order to observe the robustness of the reference generation method before reaching the limit cycle.

If it is observed in the simulations that stable and balanced locomotion is obtained using the CPG based reference generation application with manual tuning. Although trot gait's velocity is more than crawl gait's velocity, it is seen from body angles that, the trot gait is more balanced than crawl gait. Genetic Algorithm tuning is applied at the next Chapter for achieving better results.

Chapter 5

5. GAIT TUNING VIA GENETIC ALGORITHMS WITH BALANCE BASED FITNESS FUNCTION

Evolutionary algorithms can be described as a generic-population based meta-heuristic optimization methods which use the principles of the biological evolution process such as selection, reproduction, recombination and mutation. Genetic algorithms belong to evolutionary class of the artificial intelligence techniques. The process of natural selection and the survival of the fittest are considered important elements in evolution. They are proven to be robust for search and optimization problems. In addition a genetic algorithm uses randomized operators instead of deterministic rules which make it suitable for the proposed problem. In this chapter, GAs are employed for the parameter tuning of a CPG-based quadruped trot gait.

5.1. Problem Definition

There are many parameters involved in the reference generation process for a quadruped robot. Proper tuning of these parameters is necessary to improve the locomotion performance and balance of the robot. The walking speed is one of the key features. It is a function of several reference generation parameters. The average velocity is considered in this thesis as a command variable and related reference parameters are tuned to generate a suitable reference with this locomotion velocity. The average velocity is defined as follows:

$$v_{average} = B\omega \tag{5.1}$$

where B is the step size and ω is the step frequency. The genetic algorithm is implemented for tuning step height and step frequency for a given commanded average velocity value. The step size l_s is then determined automatically according to (5.1).

5.2. The Setting of Chromosome

After the input parameters are defined, the genetic algorithm operates through chromosomes, formed strings of joined parameter values, to mimic the natural evolution. A chromosome can be formed by the use of the binary alphabet for the string. For this problem each of the genes (step height and step frequency) are represented by a 5-bit binary number.

Since the GA parameters will have physical values the first set of genes are generated within a physically reasonable interval. The intervals of the step height h_s and the step frequency ω_s are set as follows for the first generation:

$$
\begin{aligned}
&0.03\, m < h_s < 0.1\, m \\
&0.8\, s^{-1} < \omega_s < 2.5\, s^{-1} \\
&0.01\, m < l_s < 0.25\, m
\end{aligned}
\tag{5.2}
$$

5.3. Simulation Scenario

A Newton-Euler method based full-dynamics 3D simulation and animation environment as described in Chapter 3 is used for simulation studies to observe the results of the GA implementation. In Figure - , a view of the mentioned animation is presented.

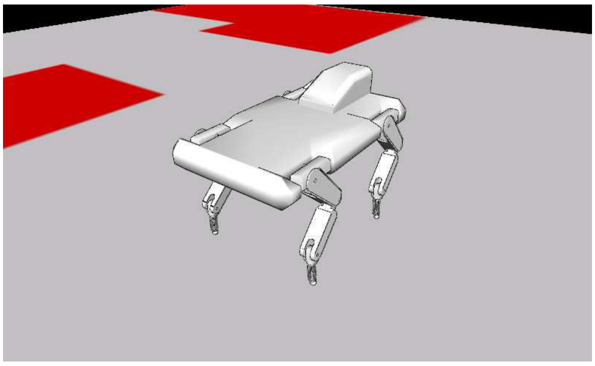


Figure 5.1: The animation environment

5.4. Virtual Walking Aid and Balanced Based Fitness Function

5.4.1. Virtual Walking Aid

In the case of an inadequate parameter selection, the robot will lose its balance and fall in an unrecoverable manner. It is observed that with the ranges of the parameters used in GA, many individuals cannot maintain the stable gait. After the start of the simulation if the value of the parameter is unfeasible, the robot falls before completing the first step. However, in order to assess the feasibility of all individuals (naturally falling ones and naturally not falling ones) in the same standard way, a sufficient number of steps have to be completed by the quadruped robot without falling down. Therefore, a mechanism is required to keep the robot moving even when the generated parameters are not suitable for stable gait.

The body orientation with respect to a fixed coordinate frame is one of the measures of quadruped robots balance. This orientation can be represented by a set of roll, pitch and yaw angles. In Figure - these angles are denoted by α , β and γ respectively. In this work virtual torsional spring-damper systems are employed to resist the deviations of these angles from zero and to pull them to the vicinity of zero. By doing so, it is aimed to keep robot stable even for parameter sets unsuitable for stable gait. The value of the spring constant employed is 250 N/m and the damper coefficient is set to be 1500 Ns/m by trial and error. The adjusted coefficients leave a reasonable motion space for the robot trunk.

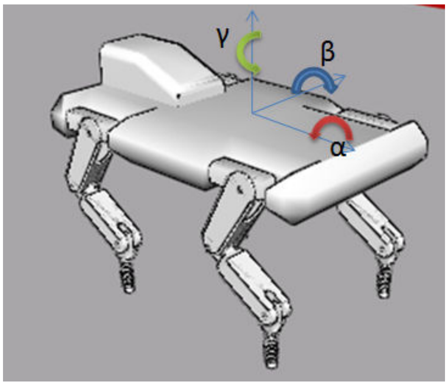


Figure 5.2: Virtual torsional spring-damper systems attached to trunk of the robot

The effect of the proposed spring-damper system can be described by the equations below;

$$
\begin{aligned}
u_{sd\,roll} &= K_{spring}.\alpha + K_{damper}.\dot{\alpha}, \\
u_{sd\,pitch} &= K_{spring}.\beta + K_{damper}.\dot{\beta}, \qquad (5.3) \\
u_{sd\,yaw} &= K_{spring}.\gamma + K_{damper}.\dot{\gamma},
\end{aligned}
$$

where $u_{sd\,roll}$, $u_{sd\,pitch}$ and $u_{sd\,yaw}$ denote the supporting torques around the *x, y* and *z* axes, respectively.

Maintaining the balance of the simulated robot by virtual torsional springs is inspired from [27,28] where a ZMP based biped gait was tuned.

5.4.2. The Fitness Function

Given a particular chromosome, the fitness function returns a single numerical value which corresponds to a performance measure of the represented individual.

The generated torques from torsional spring-damper systems can be considered as a stability measure during the locomotion of the quadruped robot. As the angular deflection around the trunk coordinate frame decreases, the resistance of the torsional spring-damper system will decrease as well. A formulation of the torsional spring damper resistance can be applied as a performance measure for GA. For this purpose the following fitness function is defined.

$$f_B = \frac{\Sigma(u_{sd\,roll}+u_{sd\,pitch}+u_{sd\,yaw})}{t_s}, \tag{5.4}$$

where t_s is the simulation duration for fix steps of the robot.

The fittest individuals of the population are chosen using the minimum values of fitness function because the support of external support torques will decrease as fitness function decreases. Since the running time of simulation increases with the number of steps in locomotion, the minimum required step number is determined as six.

5.5. The Selection of the Next Generation

The *survivor of the fittest* process composes the core structure of the GA. During the reproduction phase of the GA, certain individuals are selected from the population according to their fitness values. This process, producing offspring individuals, creates the next generation.

Individuals chosen for this process are called parents. Usually the parents are selected randomly using a scheme which favors the more fit individuals. After the selection process, their chromosomes are recombined. In traditional GA, crossover and mutation are the two typical mechanisms.

In this implementation the parents are chosen directly through the fitness value. The crossover operators are used on the best parents from the candidate pool. Numbers of the best parents are obtained by crossover ratio. The mutation operators are used on randomly selected parents from the candidate pool Also, to reduce the probability of divergence, two of *elite* (the fittest) members of each population are transferred to the next one.

5.5.1. The Cross-over Mechanism

Crossover takes two individuals, and cuts their chromosome strings at some randomly chosen position. The produced segments are named as *tails* and *heads*. These segments are swapped over to produce two new full length chromosomes. This is known as single point crossover. Figure 5.3 shows a crossover sample for given parents.

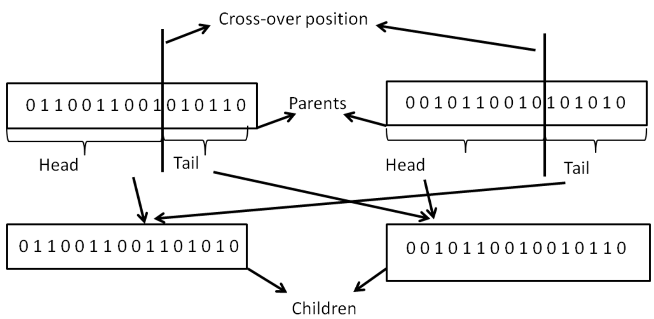


Figure 5.3: A sample cross-over

5.5.2. The Mutation Mechanism

After the crossover process, mutation is applied to randomly chosen individuals. It alters a randomly chosen gene. This mechanism provides a small amount of random search, and helps ensure that no point in the search space has a zero probability of being examined. In Figure 5.4 a mutation scheme of an individual is presented.

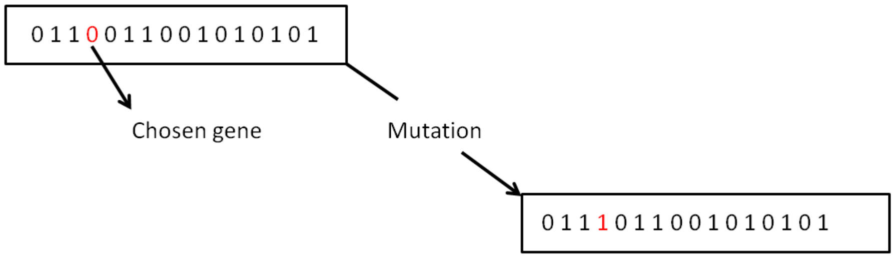


Figure 5.4: Mutation scheme

5.5.3. Overall Reproduction Process

In order to complete the definition of the process, certain parameters have to be determined. The number of individuals within a population and the number of maximum iterations has to be set. Also the amounts of population selected for crossover and mutation are parameterized beforehand together with the amounts of elite members within the population.

Table 5.1: The parameters of GA

The amount of population chosen for Cross-over	%45
The amount of individuals exposed to Mutation	%10
Amount of Elite individuals	%10
Population	20
Number of maximum iterations	20

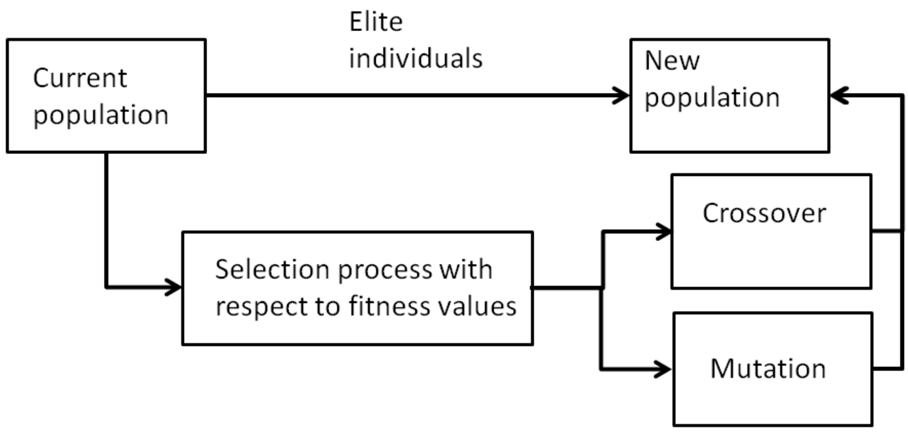


Figure 5.5: Reproduction Scheme

5.6. Outcome of The Tuning Process

The GA is implemented for the 16-DoF quadruped robot model of Chapter 3. The GA is applied for the given $v_{average}$ values on trot gait. In order to examine the effectiveness of the tuning and the fitness function, nine different successive $v_{average}$ values are selected. The chosen $v_{average}$ values and the resulting reference generation parameters used in tuning process are given in Table 5.2.

Table 5.2: The parameters of GA tuning with respect to chosen velocities

Velocity (m/s)	Step frequency (s^{-1})	Step height (m)	Step size (m)
0.0125	0.8	0.03	0.0156
0.025	1.1299	0.0424	0.0222
0.05	0.8	0.06	0.0626
0.1	1.1111	0.04	0.09
0.2	0.8696	0.08	0.23
0.3	1.1111	0.09	0.27
0.4	1.0526	0.07	0.38
0.5	2.451	0.0941	0.204
0.6	2.5	0.05	0.24

The mean of fitness function values over the individuals is used to observe the convergence of the GA process. Figure 5.6.a, Figure 5.6.b and Figure 5.6.c present this information for average velocity values given in Table 5.2.

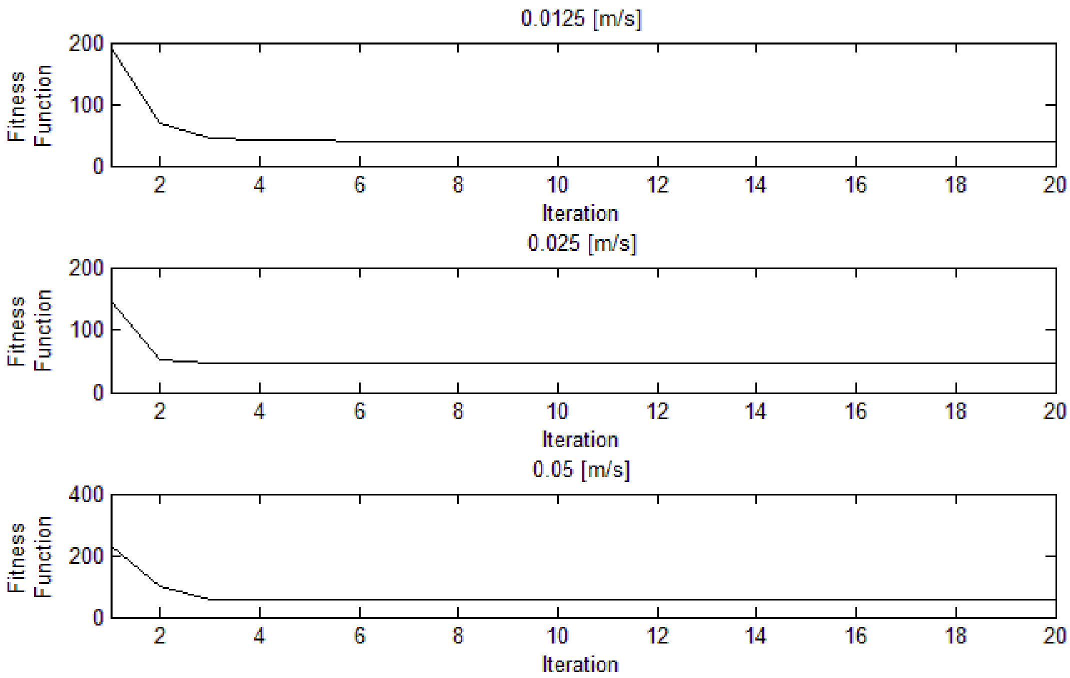


Figure 5.6.a: The mean values of fitness function with slow velocities

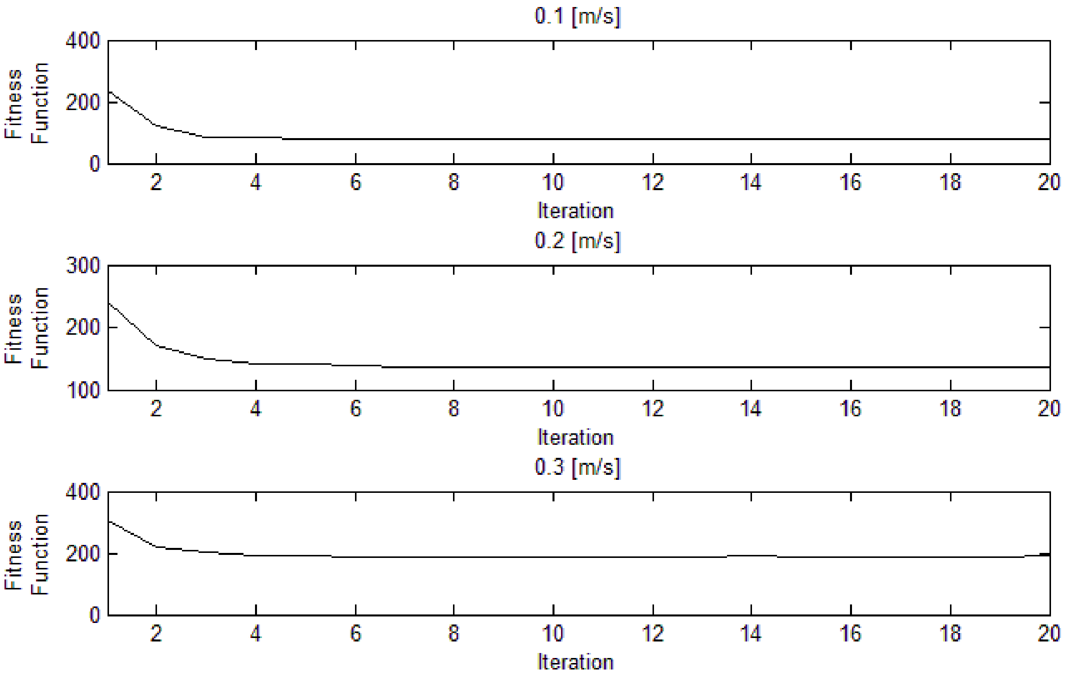


Figure 5.6.b: The mean values of fitness function with medium velocities

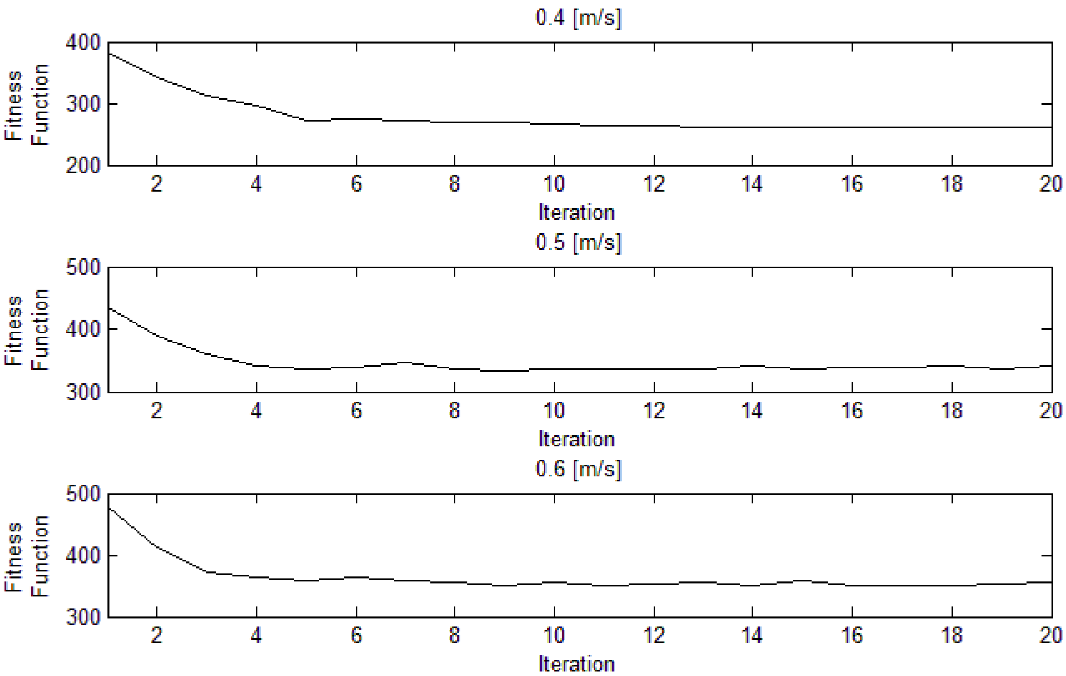


Figure 5.6.c: The mean values of fitness function with high velocities

The results indicate a proper operation of GA implementation since the average values of the individuals' fitness functions are converging to a minimum value between the average velocities 0.0125 m/s and 0.6 m/s.

The roll, pitch and yaw angles of the body are indicators of the stability of the gait. Body angles of the average velocity 0.3 m/s and 0.6 m/s are presented at the figure 5.7.a and figure 5.7.b.

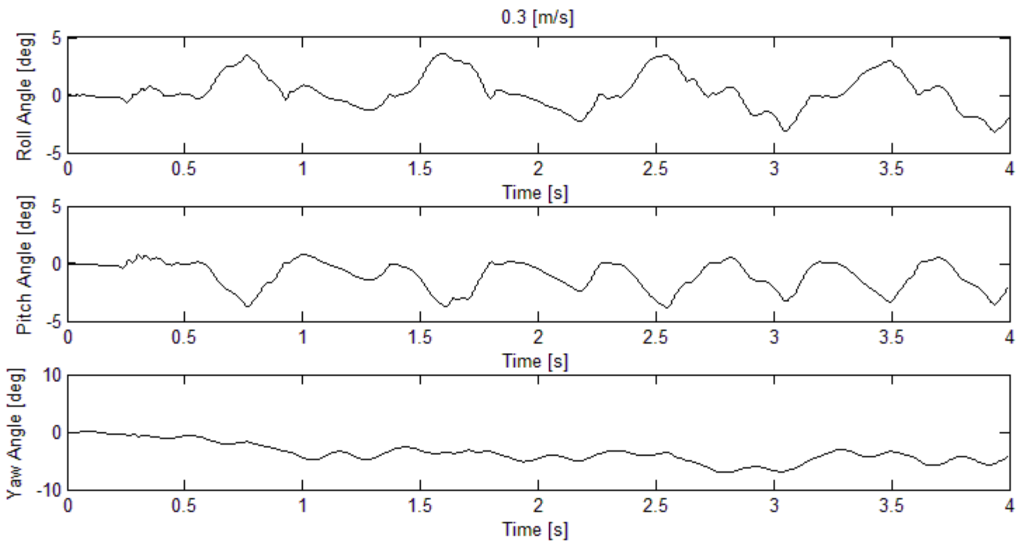


Figure 5.7.a: The roll, pitch and yaw angles of the body with 0.3 m/s average velocity

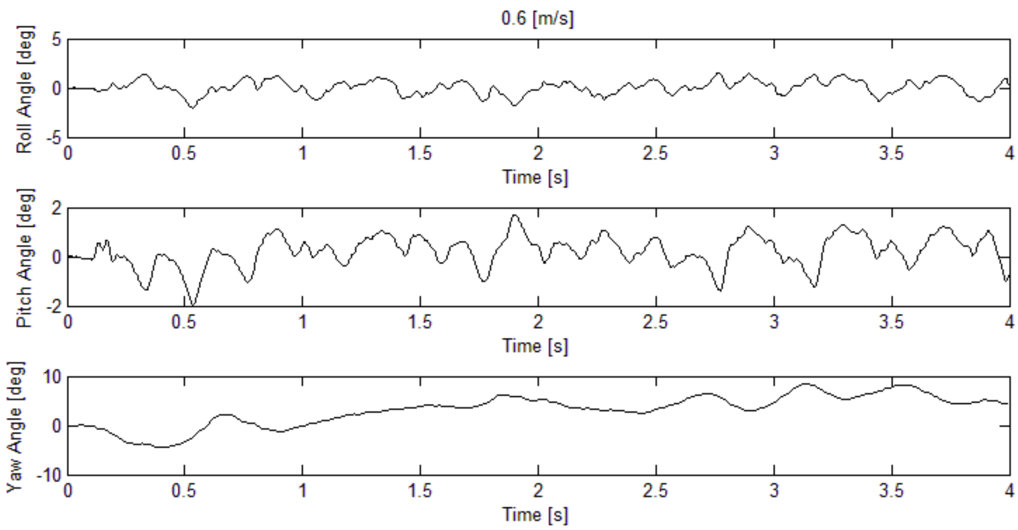


Figure 5.7.b: The roll, pitch and yaw angles of the body with 0.6 m/s average velocity

5.7. Discussion

CPG based reference generation is realized for the quadruped robot model with the GA tuning. The average of the generated torques from torsional spring-damper systems over the simulation duration constitute the fitness function of GA. GA tuning is applied to CPG based reference generation method with nine different average velocities. Adding two fittest individuals directly to the next generation and crossing over between nine elitist individuals provide fast convergences to optimum parameters. The mutation process provides to find global extremum.

Outcomes of GA tuning provide interesting information. When the average velocity is larger than 0.4 m/s, step sizes become smaller, and stepping period decreases for moving fast speed. Stability of the motion can be observed from angles of the body.

Chapter 6

6. GAIT TUNING VIA GENETIC ALGORITHM WITH ENERGY EFFICIENCY BASED FITNESS FUNCTION

In this chapter a different fitness function is applied on same process. Energy efficiency is as important as balance. Energy efficient locomotion increases locomotion time on the outdoor duties. The fitness function for energy efficiency is defined:

$$f_E = \frac{\bar{P}}{\bar{V}}, \tag{6.1}$$

where $\bar{P}$ stands for the average sum of absolute joints actuator powers, and $\bar{V}$ is the average velocity of the robot body during the simulation.

6.1. Outcome of The Tuning Process

The GA is applied for the given $v_{average}$ values. In order to examine the effectiveness of the tuning and the fitness function, the same nine $v_{average}$ command values as in the previous chapter are used. The resulting reference generation parameters used in tuning process are given in Table 6.1.

Table 6.1: The parameters of GA tuning with respect to chosen velocities

velocity(m/s)	step frequency (s^{-1})	step height (m)	step size (m)
0.0125	0.8696	0.03	0.0144
0.025	0.8	0.03	0.0312
0.05	0.9091	0.03	0.055
0.1	0.9542	0.0324	0.1048
0.2	0.9524	0.03	0.21
0.3	2.5	0.03	0.12
0.4	2.5	0.04	0.16
0.5	2.5	0.06	0.2
0.6	2.5	0.03	0.24

The mean of fitness the function values are used to assess the convergence of the GA process. Figure 6.1.a, Figure 6.2.b and Figure 6.2.c present this information for average velocity values given in Table 6.1.

The results indicate the proper operation of GA implementation since the average values of the individuals' fitness functions are converging to a minimum value for all commanded average velocities.

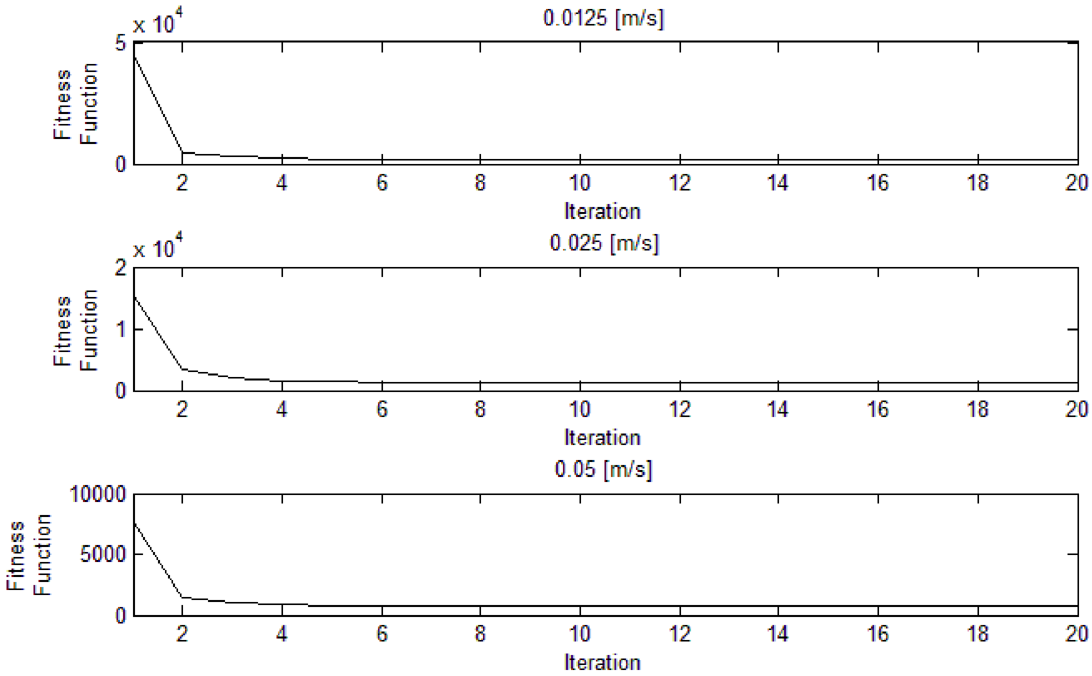


Figure 6.1.a: The mean values of fitness function with slow velocities

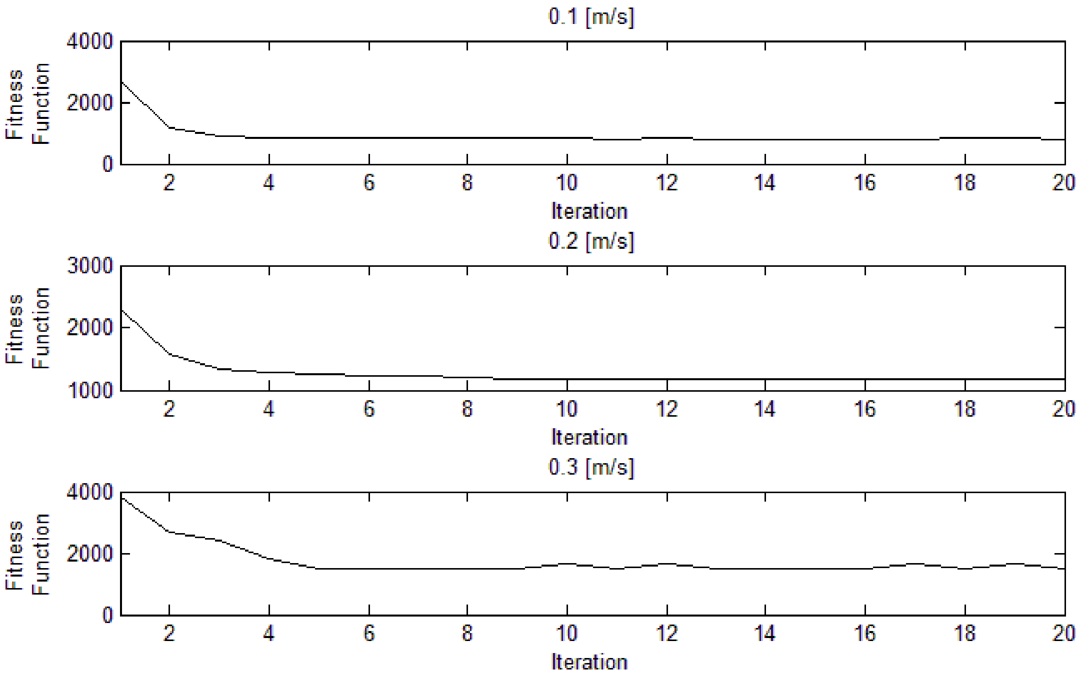


Figure 6.1.b: The mean values of fitness function with medium velocities

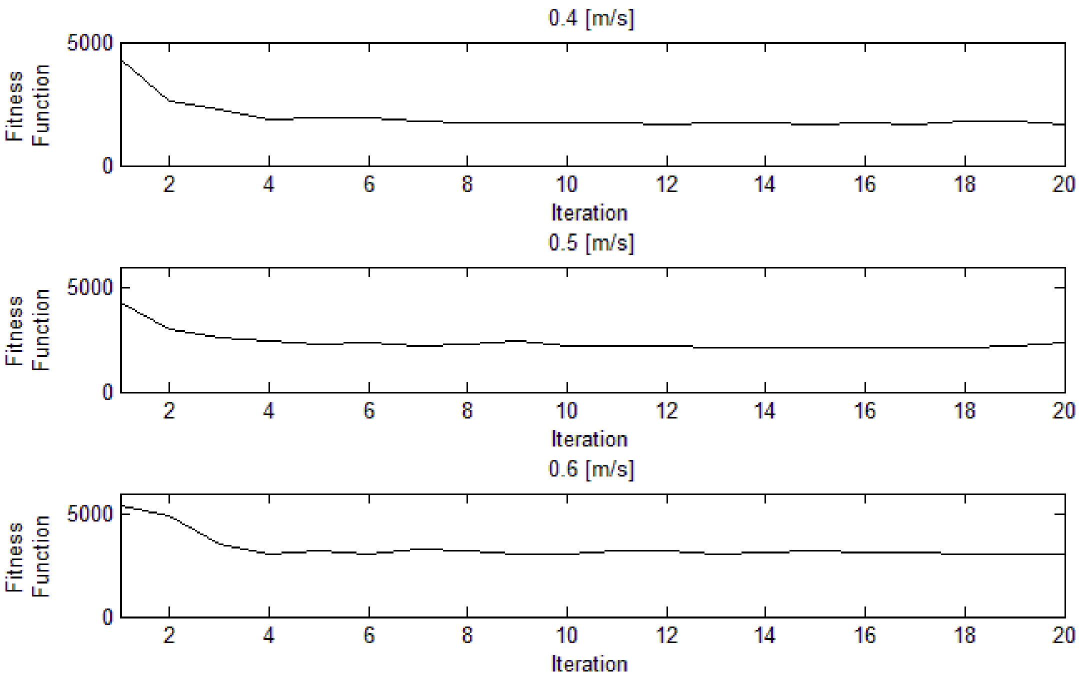


Figure 6.1.c: The mean values of fitness function with high velocities

As in the case of the balance based tuning, the results indicate that the GA reaches convergences. The average values of the individuals' fitness functions are converging to a minimum value.

The roll, pitch and yaw angles of the body recorded. Body orientation angles with the average velocity 0.3 m/s and 0.6 m/s are presented at the Figures 6.2.a, 6.2.b.

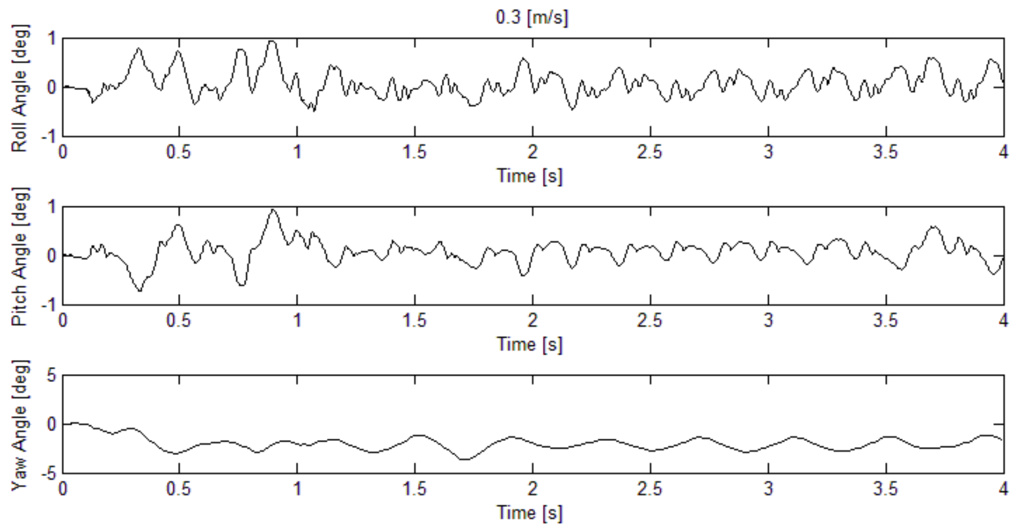


Figure 6.2.a: The roll, pitch and yaw angles of the body with 0.3 m/s average velocity

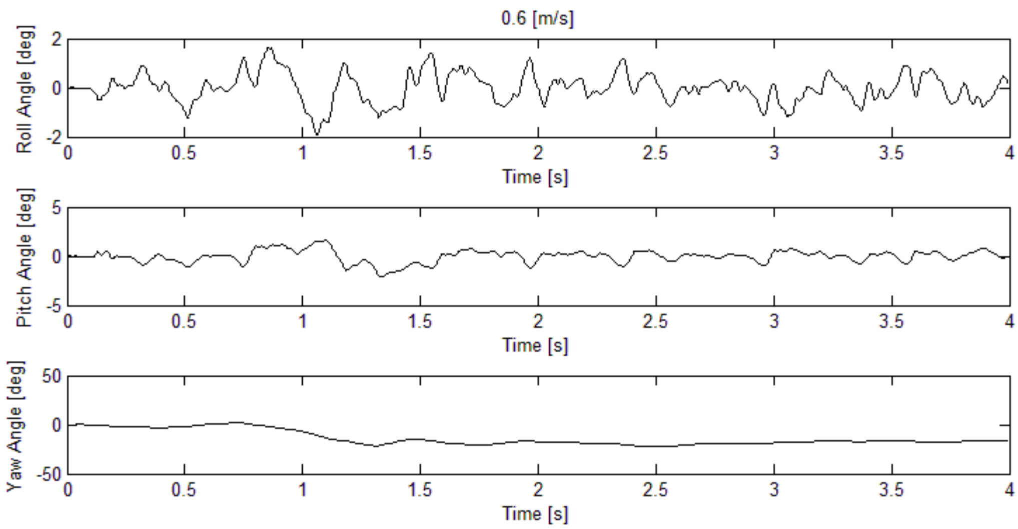


Figure 6.2.b: The roll, pitch and yaw angles of the body with 0.6 m/s average velocity

6.2. Discussion

The parameters of the CPG based reference for a quadruped gait are obtained via GA. AN energy efficiency based fitness function, which is defined in Section 6.1, is used for the optimization process.

In the energy efficient locomotion, the step height is usually obtained as 0.03 m which is the lower bound of the range allowed for this parameter. From this outcome, we may make the statement that using low step height values is more energy efficient for locomotion. This observation is quite intuitive. In energy efficient locomotion, short and fast steps are preferred rather than long and slow steps, which were favored with the balance-based fitness function. The roll, pitch and yaw angle curves obtained in this chapter and in Chapter 5 are similar. It can generally be stated that moving waddlingly isn't energy efficient. Smooth locomotion is not only more balanced but also energy efficient. With this point of view, balanced locomotion and energy efficient locomotion are quite similar.

Chapter 7

7 GAIT TUNING VIA GENETIC ALGORITHM WITH WEIGHTED FITNESS FUNCTION

In this chapter a different fitness function is applied on same process. This fitness function based on both a energy efficiency and balance. Balance is important for stable gait and energy efficiency is an asset for increasing locomotion time. A new fitness function is generated for serving the purpose of multi-objective tuning. The new fitness function is a weighted sum of the two fitness functions employed in the previous chapters:

$$f_e = w_1 f_B + w_2 f_E. \tag{7.1}$$

The values of the weights w_1 and w_2 can be needed to stress one of the tuning variable more than the others. In this work both gains are taken equal to 1.

7.1. Outcome of The Tuning Process

The same nine velocity command values as in Chapter 5 and 6 are used. The $v_{average}$ values and the resulting reference generation parameters obtained in tuning process are given in Table 7.1.

Table 7.1: The parameters of GA tuning with respect to chosen velocities

velocity(m/s)	step frequency (s^{-1})	step height (m)	step size (m)
0.0125	0.8	0.03	0.0156
0.025	0.9091	0.03	0.0276
0.05	0.9091	0.03	0.055
0.1	0.8696	0.03	0.0114
0.2	1.1111	0.03	0.18
0.3	2.5	0.04	0.12
0.4	2.5	0.04	0.16
0.5	2.5	0.04	0.2
0.6	2.5	0.04	0.24

The mean values of fitness function values of the individuals over the generations are recorded and plotted as a measure of convergence. Figure 7.1.a, Figure 7.2.b and Figure 7.2.c present the fittest value convergence for the set of velocity commands considered.

Again, as in the previous two Chapters, convergence of fittest values is observed for all velocities.

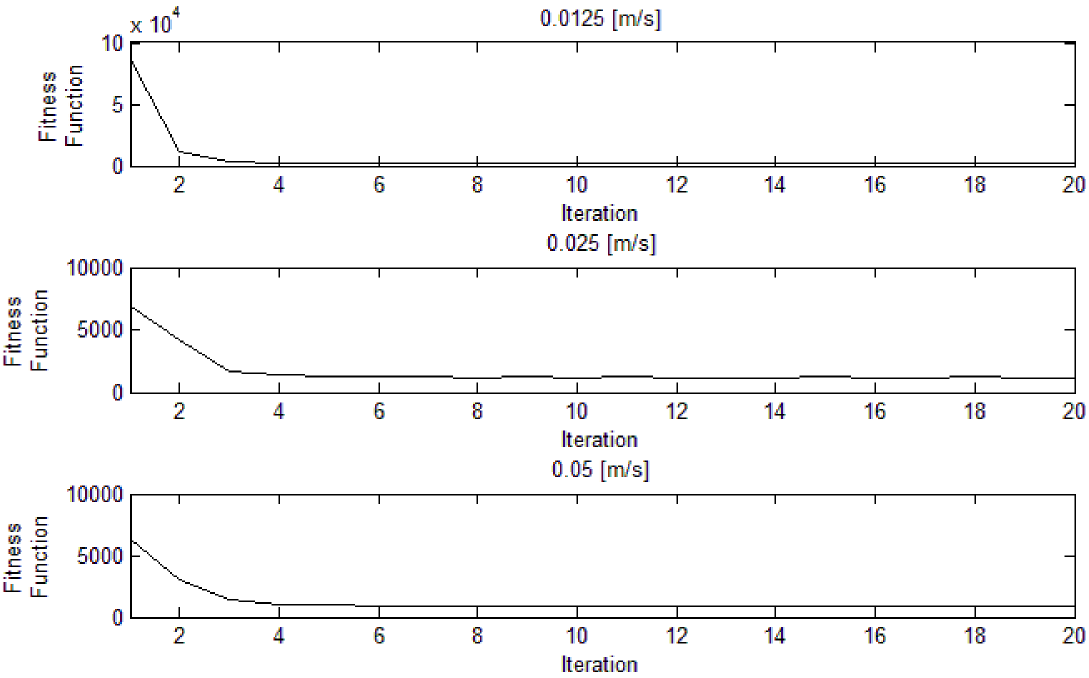


Figure 7.1.a: The mean values of fitness function with slow velocities

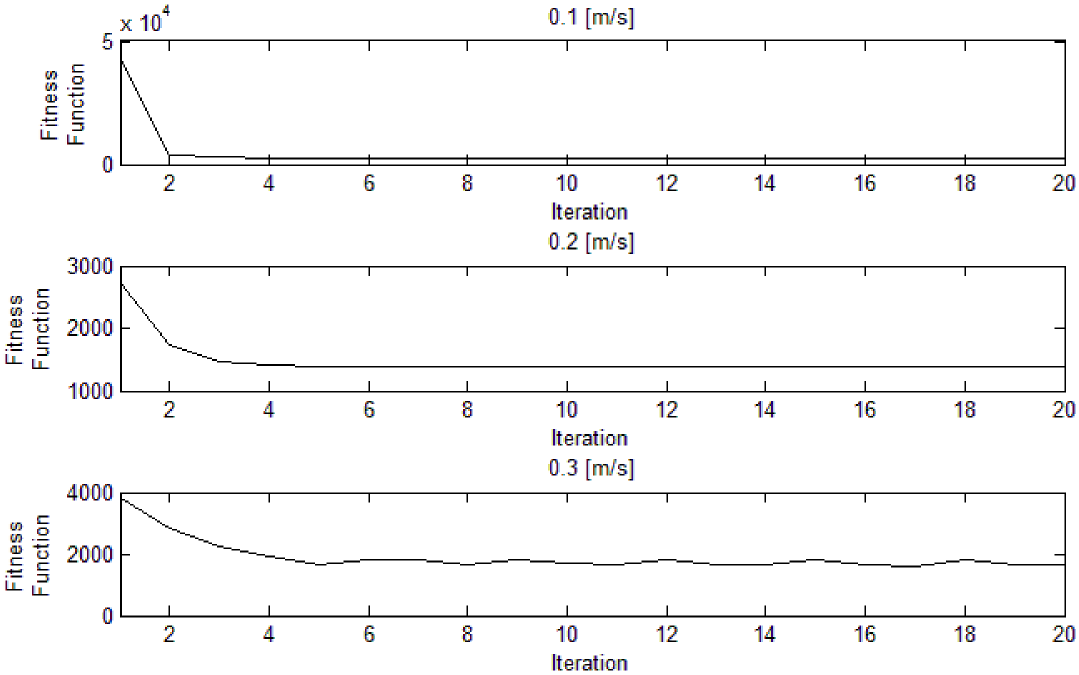


Figure 7.1.b: The mean values of fitness function with medium velocities

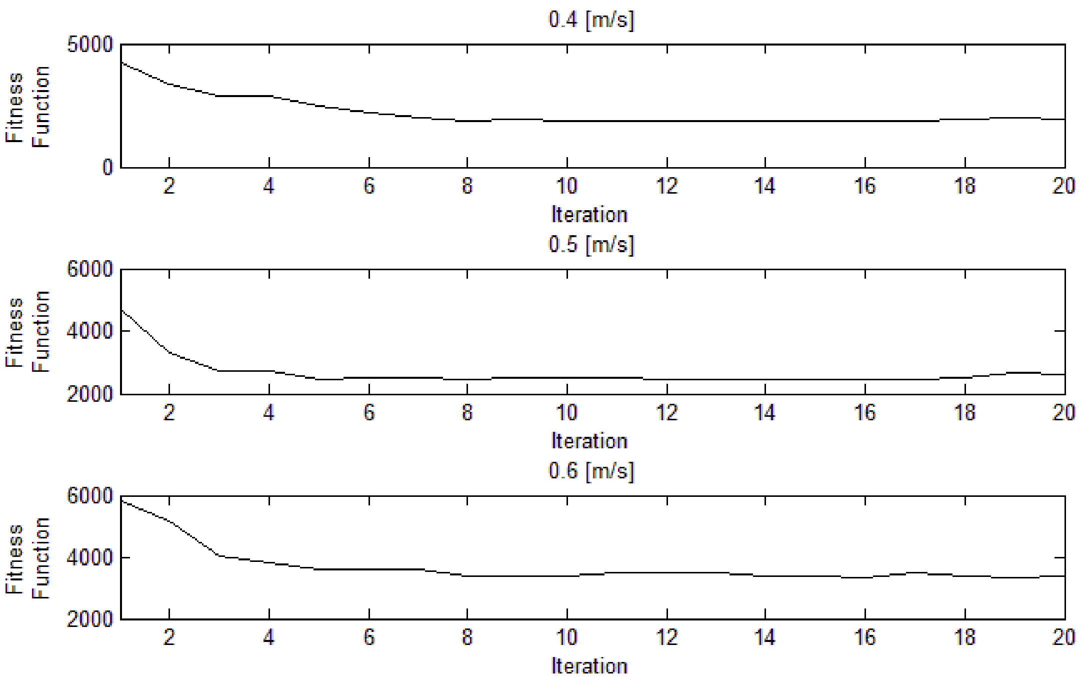


Figure 7.1.c: The mean values of fitness function with high velocities

The roll, pitch and yaw angles of the body for velocities, 0.3 m/s and 0.6 m/s are presented at the figure 7.2.a and figure 7.2.b, respectively.

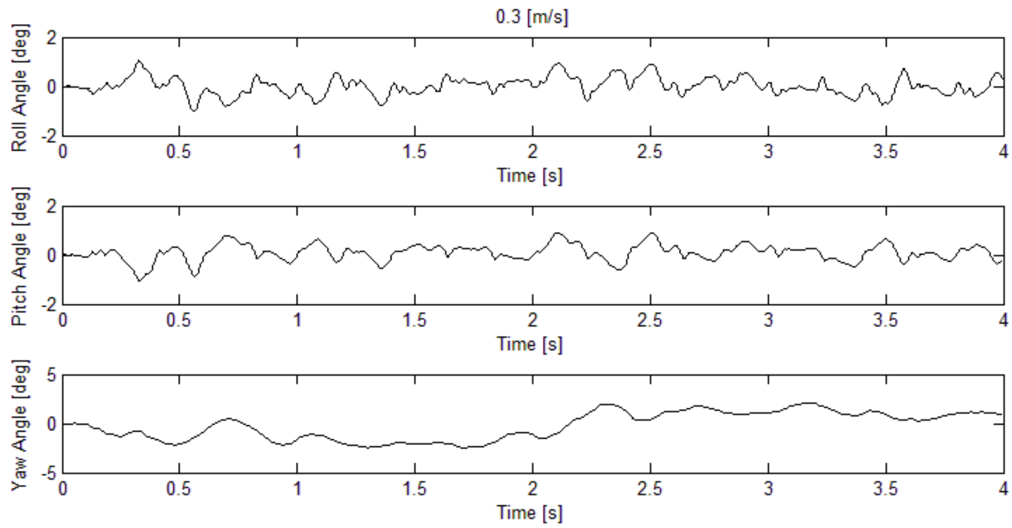


Figure 7.2.a: The roll, pitch and yaw angles of the body with 0.3 m/s average velocity

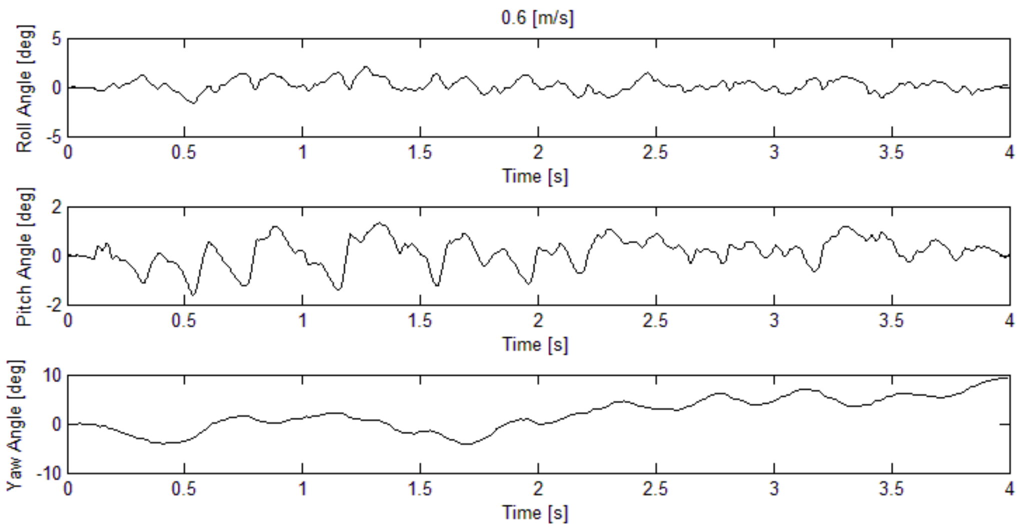


Figure 7.2.b: The roll, pitch and yaw angles of the body with 0.6 m/s average velocity

7.2. Discussion

The fitness function is further modified in this chapter. The new fitness function is the weighted sum of the two fitness functions employed in the previous two chapters. Both balanced and energy efficient locomotion reference parameters are obtained with this fitness function. With these parameters both balanced and energy efficient motion is expected. Virtual torsional spring-damper mechanism support body of the robot. Under these circumstances, optimal energy efficient parameters can be unstable when virtual torsional spring-damper mechanism is removed. In order to prohibit this situation weighted fitness function is applied.

Simulation results give small step height values like in the previous chapter. Using small step height values are profitable for legged robot locomotion. But they hinder to make long steps. Therefore, in order to reach higher motion speeds, the stepping frequency is increased.

Chapter 8

8. CONCLUSION

A CPG-based quadruped robot locomotion reference generation technique is presented in this thesis. Parameters of the references are obtained with evolutionary algorithms. A non-linear type phase oscillator is used for the CPG network. The main target is finding suitable locomotion parameters for a demanded locomotion speed.

Newton-Euler based full-dynamics 3D simulations are carried out with a quadruped robot model which has four DOFs at each leg. Toe tip references of the legs are generated for stable locomotion by using phase oscillator. Joint trajectories are obtained via inverse kinematics.

3D dynamic simulations are performed for crawl gait, trot gait and crawl to trot gait transition with manual tuning. It is observed from the simulation results that the trot gait is smoother than crawl gait. Gait transition is managed successfully. Managing gait transitions is more straight-forward with the CPG than other reference generation methods. This is the advantage of the CPG based reference generation.

CPG based reference generation is also realized with GA tuning. A velocity command is given as the input to the system. The fitness is measured by three different cost functions. The first cost function measures the amount of support the simulated quadruped receives from torsional virtual springs and dampers opposing the changes in body orientation, whereas the second one is a measure of energy efficiency in the locomotion. The third cost function is a combination of the first two. Tuning results with the three cost functions are obtained and compared. Simulation results verify the success of the proposed reference generation and tuning method.

The simulation results indicate that balanced locomotion and energy efficient locomotion are quite similar. Smooth locomotion is both more balanced and energy efficient.

The proposed method is promising for the following future studies:

a) CPG based reference generation with quadruped gait transition tuned by evolutionary algorithms.
b) Hybrid CPG-ZMP based reference generation for legged robots tuned by evolutionary algorithms.
c) CPG based reference generation with reflex mechanism for the motion on rough terrain
d) Experimental verification of the proposed techniques is also considered as a future work.

REFERENCES

[1] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K. and Hirukawa, H., "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point", *Proceedings of the 2003 IEEE International Conference on Robotics& Automation*, pp. 1620-1626, Taipei, Taiwan, September 2003.

[2] Erbatur, K. and Kurt, O., "Natural ZMP Trajectories for Biped Robot Reference Generation", IEEE Transactions on Industrial Electronics, vol. 56, no. 3, pp. 835-845, March 2009

[3] Yoneda, K. and Hirose S., "Dynamic and Static Fusion Gait of a Quadruped Walking Vehicle on a Winding Path", *Proceedings of the 1992 IEEE International Conference on Robotics & Automation*, pp. 143-148, Nice, France, May 1992.

[4] Takeuchi, H., "Development of MEL HORSE", *In Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, pp 3165–3171, Seoul, Korea, May 2001

[5] Takeuchi, H., "Real Time Optimization for Robot Control Using Receding Horizon Control with Equal Constraint", Journal of Robotic Systemsvol 20, no 1, pp 3-13, 2003

[6] Pengfei, W., Bo, H. and Lining, S., "Bionic Robot Based on Moving ZMP", *Proceedings of the 1992 IEEE International Conference on Mechatronics & Automation*, pp. 1935-1940, Niagara Falls, Canada, July 2005.

[7] Osumi, H., Kamiya S., Kato, H. and Umeda, K. "Time Optimal Control for Quadruped Walking Robots", *Proceedings of the 2006 IEEE International Conference on Robotics & Automation*, pp. 1102-1108, Orlando, Florida, May 2006.

[8] Shkolnik, A. and Tedrake R., "Inverse Kinematics for a Point-Foot Quadruped Robot with Dynamic Redundancy Resolution", *Proceedings of the 2007 IEEE International Conference on Robotics & Automation*, pp. 4331-4336, Roma, Italy, April 2007.

[9] K. Byl, A. Shkolnik, S. Prentice, N. Roy, and R. Tedrake, "Reliable dynamic motions for a stiff quadruped," in *Experimental Robotics*, 2009, pp. 319–328.

[10] Kalakrishnan, M., Buchli J., Pastor, P., Mistry, M. and Schaal, S."Fast, Robust Quadruped Locomotion over Challenging Terrain", *Proceedings of the 2006 IEEE International Conference on Robotics & Automation*, pp. 1102-1108, Orlando, Florida, May 2006.

[11] Kalakrishnan, M., Buchli J., Pastor, P., Mistry, M. and Schaal, S." Learning, Planning and Control for Quadruped Locomotion over Challenging Terrain", The International Journal of Robotics Research, vol. 30, no. 2, pp. 236-258, November 2011

[12] R. Katoh and M. Mori, "Control method of biped locomotion giving asymptotic stability of trajectory," Automatica, vol. 20, no. 4, pp. 405-414, 1984.

[13] J. Bay and H. Hemami, "Modeling of a Neural Pattern Generator with Coupled Nonlinear Oscillators" IEEE Transactions on Biomedical Engineering, vol. 34, no. 4, pp. 297-306, April 1987.

[14] K. Matsuoka, "Sustained Oscillations Generated bu Mutually Inhibiting Neurons with Adaptation" Biological Cybernetics, vol: 52 pp: 367-376, 1985

[15] K. Matsuoka, "Mechanism of Frequency and Pattern Control in the Neural Rhythm Generators," Biological Cybernetics, vol: 56 pp: 345-353, 1987

[16] G. Taga, Y. Yamaguehi, H. Shimizu "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," Biological Cybernetics, vol.65, pp.147-159, 1991

[17] J. J. Collins and S: A: Richmond, "Hard-Wired Central Pattern Generators for Quadrupedal Locomotion," Biological Cybernetics, vol.71, pp.375-385, 1994

[18] H. Kimura, S. Akiyama, and K. Sakurama "Realization of Dynamic Walking and Running of the Quadruped Using Neural Oscillator," Autonomous Robots, vol.7, pp.247-258, 1999

[19] Y. Fukuoka, H. Kimura, and A. Cohen "Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts," International Journal of Robotic Research, vol.22, no:3-4 pp.187-202, 2003

[20] A. J. Ijspeert, A. Crespi, D. Ryczko and J. Cabelguen "From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model," Science, vol.315, no:5817 pp.1416-1420, 2007

[21] K. Tsujita, K. Tsuchiya, and A. Onat, "Adaptive gait pattern control of a quadruped locomotion robot," *Proc. IEEE/RSJ Int. Conf. on Inteligent. Robots Syst*em, pp. 2318–2325, 2001.

[22] S. Aoi, T. Yamashita, A. Ichikawa and K. Tsujita, "Hysteresis in Gait Transition İnduced by Changing Waist Joint Stiffness of a Quadruped Robot Driven by Nonlinear Oscillators

with Phase Resetting," *Proc. IEEE/RSJ Int. Conf. on Inteligent. Robots Syst*em, pp. 1915–1920, 2010.

[23] C. Liu, Y. Chen, J. Zhang and Q. Chen, "CPG Driven Locomotion Control of Quadruped Robot," *Proc. IEEE Int. Conf. on Systems, Man, and Cyberbetics*, pp. 2368–2373, 2009.

[24] C. Liu, Q. Chen and J. Zhang "Coupled Van Der Pol Oscillators Utilised a Central Pattern Generators for Quadruped Locomotion," *Proc. IEEE Int. Conf. on Control and Decision*, pp. 3677–3682, 2009.

[25] G. Endo, J. Nakanishi, J. Morimoto and G. Cheng "Experimental Studies of a Neural Oscillator for Biped Locomotion with QRIO," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 596–602, Barcelona, Spain, April 2005.

[26] D.E. Goldberg, "Genetic Algorithm in Search Optimization, and Machine Learning.", Addison Wesley, 1989.

[27] Bebek, O., K. Erbatur, "A Fuzzy System for Gait Adaptation of Biped Walking Robots", *Proceedings of the IEEE Conference on Control Applications*, 2003

[28] Akbas, T. "Bipedal Humanoid Robot Walking Reference Tuning by the Use of Evolutionary Algortihms" Master Thesis, İstanbul, Sabanci Universtiy, 2012

[29] David, T., and Roberts, M. "Understanding Balance: The Mechanics of Posture and Locomotion", Nelson Thornes, 1995.

[30] Harris, Susan E. "Horse Gaits, Balance and Movement" New York: Howell Book House 1993

[31] Shik, M. L., Severin, F. V., Orlovskii, G. N. "Control of Walking and Running by Means of Electrical Stimulation of the Mid-Brain" Biophysics, 1966

[32] Grillner, S. "Control of Locomotion in Bipeds, Tetrapods and Fish." Handbook of Physiology, American Physiology Society, pp: 1179-1236, 1981

[33] Stent, G. S., Kristan, W. B., Friesen, W. D. "Neuronal Generation of the Leech Swimmimng Movement," Science, pp:1348-1357, 1978

[34] Bachanam, J. T.,Grillner, S. "Newly Identified Glutamate Interneurons and Their Role in Locomotion in Lamprey Spinal Cord." Science, vol.315, pp: 312-314, 1987

[35] Zhang X L. "Biological-inspired rhythmic motion and environmental adaptability for quadruped robot". Dissertation for the Doctoral Degree. Beijing: Tsinghua University, 2004

[36] Hodgkin A L, Huxley A F. "Currents carried by sodium and potassium ions through the membran of the giant axon of Loligo," J Pjysiol, vol:116, pp: 449–472, 1952

[37] Hodgkin A L, Huxley A F. "A quantitative description of membrane and its application to conduction and excitation in nevre," J Pjysiol, vol: 117 pp: 500–544,1952

[38] Morris C, Lecar H. "Voltage oscillations in the barnacle giant muscle fiber," Biophys J, vol: 35 pp: 193–213, 1981

[39] Lakshmananm M, Murali K. "Chaos in Nonlinear Oscillators: Controlling and Synchronization," Singapore: World Scientific Press, 1996.

[40] Stein R B, Leung K V, Mangeron D, et al. "Improved neuronal models for studying neural networks," Kybernetik, vol:15 pp: 1–9, 1974

[41] Acebron J A, Bonilla L L, Vicente C J P, et al. "The Kuramoto model: A simple paradigm for synchronization phenomena," Rev Mod Phys, vol:77 pp: 137–185, 2005

[42] Nakamura Y, Tominaga F, Munakata T. "Clustering behavior of time-delayed nearest-neighbor coupling oscillators," Phys Rev, E 49 pp: 4849–4856 1994

[43] Conradt J. "Distributed central pattern generator control for a serpentine robot," *In: Proc of the International Conference on Artificial Neural Networks (ICANN)*, pp: 338–341, Istanbul, Turkey, 2003.

[44] Righetti L, Buchli J, Ijspeert "A. Dynamic hebbian learning for adaptive frequency oscillators," Phys D, 2005

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

[46] Nicolas J. "Artificial evolution of controllers based on nonlinear oscillators for bipedal locomotion," Master's thesis, EPFL, Switzerland, 2005-2006

[47] Righetti L, Ijspeert A J. "Pattern generators with sensory feedback for the control of quadruped locomotion," *In: IEEE International Conference on Robotics and Automation*, pp: 819–824, 2008.

[48] Van der Pol B, Van der Mark J. "The heartbeat considered as a relaxation oscillation, and an electrical model of the heart," Philosoph Mag, vol:6 no:38, p: 763–775 1928

[49] Rayleigh J W S, Lindsay R B. "The Theory of Sound," Vol: 1 New York: Dover Publications, 1945

[50] Zielinska T. “Coupled oscillators utilized as gait rhythm generators of a two-legged walking machine,” Biological Cybernetics, vol:74, pp: 263–273,1996

[51] de Pina Filho A C, Dutra M S, Raptopoulos L S C. “Modeling of a bipedal robot using mutually coupled Rayleigh oscillators,” Biological Cybernetics, vol:92, pp: 1–7 2005

[52] Silva M, Machado T. “ A Historical Perspective of Legged Robots” Journal of Vibration and Control, vol: 13, no: 9-10, pp: 1447-1486, 2007

[53] Mosher R. S., “Test and Evaluation of a Versatile Walking Truck,” *Proceedings of the Off-Road Mobility Research Symposium, International Society for Terrain Vehicle Systems*, pp: 359-379, Washington, May 1968

[54] McGhee R. B. “Finite State Control of Quadruped Locomotion,” *in Second International Symposium on Extrenal Control of Extremities*, 1966

[55] S. Hirose. “A study of design and control of a quadruped walking vehicle,” International Journal of Robotics Research, 3(2), 1984.

[56] K. Arikawa, S. Hirose “ Development of Quadruped Walking Robot *TITAN-VIII*” *In: IEEE International Conference on Intelligent Robots and Systems*, pp: 208–214, 1996.

[57] H. Miura, I. Shimoyama, M. Mitsuishi, and H. Kimura. “Dynamical walk of quadruped robot,” MIT Press, 1985.

[58] M. Buehler, R. Battaglia, A. Cocosco, G. Hawker, J. Sarkis, K. Yamazaki “SCOUT: A Simple Quadruped that Walks, Climbs and Runs” *In: IEEE International Conference on Robotics and Automation*, pp:1707–1712, Leuven, Belgium, May 2006

[59] K. Berns, W. Ilg, M. Deck and R. Dillmann “The Mammalian-Like Quadrupedal Walking Machine BISAM” *International Workshop on Advanced Motion Control,* pp: 429-433 1998

[60] P. G. De Santos, J. A. Galvez, J. Estremera and E. Garcia “SILO4 A True Walking Robot for the Comparative Study of Walking Machine techniques” IEEE Robotics & Automation Magazine, pp: 23-32, December 2003

[61] H. Kitano, M. Fujita, S. Zrehen and K. Kageyama “Sony Legged Robot for Robocup Challenge” *In: IEEE International Conference on Robotics and Automation*, pp:2605–2612, Leuven, Belgium, May 1998

[62] Semini C. “HyQ – Design and Development of a Hydraulically Actuated Quadruped Robot”. Dissertation for the Doctoral Degree, University of Genoa, 2010

[63] M. Raibert, K. Blankespoor, G. Nelson, R. Playter and the Bigdog Team "BigDog, the Rough-Terrain Quadruped Robot" *Proceedings of the International Federation of Automatic Control,* Seoul, Korea, July 2008

[64] Greenemeier, L. "DARPA Pushes Machine Learning with Legged LittleDog Robot" Scientific American, April 15, 2008

[65] Erbatur, K. and Kawamura, A., "A New Penalty based Contact Modeling and Dynamics Simulation Method as Applied to Biped Walking Robots," *Proceedings of the 2003 FIRA World Congress*, Vienna, Austria, October 2003.

[66] Q. Wu, C. Liu, J. Zhang and Q. Chen, "Survey of Locomotion Control of Legged Robots Inspired by Biological Concept" Science in China Press, August 3, 2009.