LOCOMOTION TRAJECTORY GENERATION AND DYNAMIC CONTROL FOR BIPEDAL WALKING ROBOTS

YANG, LIN

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgments

I would like to express my sincere appreciations to my supervisors, Prof. Poo Aun Neow and Associate Prof. Chew Chee Meng, for their invaluable guidance, insightful comments, strong encouragements and personal concerns both academically and otherwise throughout the course of this work. In the course of this Ph.D study, I indeed have learnt and benefitted from their comments and critiques. I would also like to thank Prof. Teresa Zielinska from Warsaw University of Technology for her valuable assistance, comments and guidance on my research work and taking care of me when I spent time in her laboratory in Poland as part of the NUS-WUT collaboration.

I gratefully acknowledge the financial support provided by the National University of Singapore through the Research Scholarship without which it would have not been possible for me to work for my degree in NUS.

Last but certainly not the least, my thanks also to my friends and the officers in the Control and Mechatronics Laboratory for their support and encouragement. They have provided me with helpful comments, great friendship and a warm community during the past few years in NUS.

Finally, my deepest gratitude goes to my parents, for their encouragements, moral support and love that have given me strength throughout my life.

Summary

In this thesis, a general method for joint trajectory generation to achieve optimized stable locomotion for bipedal robots is first proposed and referred to as Genetic Algorithm Optimized Fourier Series Formulation (GAOFSF). This method is used to generate the basic motion patterns for joint motion coordination. Then, a soft motion control strategy which makes use of the reaction torques at the stance leg is proposed and investigated. Based on this motion control applied on the basic motion trajectories that the GAOFSF generated for walking on various terrains and for three dimensional walking motions, stable and robust limit cycle behaviors have been achieved. In achieving such a stable limit cycle behavior, the robot is also capable of overcoming certain perturbations and returning to the stable walking gait if the perturbations do not move it out of its stability region. Furthermore, a high-level motion adjustment agent based on the Truncated Fourier Series (TFS) formulation has been also developed to adjust the stride-frequency, step-length and walking posture in a very straightforward manner. Given these motion adjustment functionalities, human walking behaviors such as the rhythmic walking behavior and motion adaptation to the environment change can be achieved to a good extent. In addition, two motion-balance strategies based on the TFS formulation have been proposed and demonstrated to be able to achieve long-distance 3D human-like walking motions. From the results obtained, the damping behavior is found to be more important for motion balance as it can result in a smoother lateral behavior and naturally confine the motion into a sinusoidal profile. The entire bipedal walking control algorithm proposed in this thesis has shown to be general for different walking postures and for robots with different mechanical and geometrical properties.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Animal locomotion research has been around for more than one century. Bipedal locomotion is associated with animals that use exactly two limbs to achieve locomotion. As humans are bipedal, it is always desirable for us to replicate a machine with the same mobility as ourselves as such a machine will enjoy tremendous mobility advantage when placed in human environments. Such machines will be particularly useful in environments which pose great hazards for human beings. Research on bipedal walking control will provide greater insights to the biomechanics of both robots and humans and a better understanding of the limitations to walking in both humans and robots. A greater understanding of how humans walk will also aid the development of leg prostheses and help those who lost their lower extremities have a better chance to walk again.

However, it is a great challenge to build a bipedal robot that has agility and mobility similar to that of a human. There are several characteristics of bipedal walking robots that make them seemingly difficult to control:

- Non-linear dynamics.

- Multi-variable dynamics.
- Naturally unstable dynamics.
- Limited foot-ground interaction.
- Discretely changing dynamics.
- Subjective performance evaluation.

The first three of the above characteristics make synthesizing a controller using traditional linear control techniques difficult while the last three further move bipedal walking out of the range of traditional control techniques for which much have been developed.

A bipedal robot generally comprises multiple rigid links driven at its joints simultaneously. The system is a complex non-linear multiple-input multiple-output control system. In addition, the locomotion posture of the biped, unlike that for the quadruped or hexapod, has difficult stability issues since the biped is comparable to the inverted pendulum model.

The limited foot-ground reaction forces that can be generated is a distinctive feature of normal walking robots. This under-actuated joint is what makes the control of walking robots different from that of robotic arms fixed rigidly to the ground at their bases for which several traditional control methods are available. The torques that can be applied to the foot is limited as the foot will rotate over its toe or its heel if these are too large. Because of this, the extent of the control action which can occur during a walking stride is limited. In particular, the forward velocity of the robot cannot be quickly changed as this is limited to the reaction forces that the foot-ground interface can sustain.

The dynamics of a bipedal walker changes as it transitions from the single support phase to double support phase and back again. Since the continuity of the equations representing the dynamic motion can be broken by the foot-ground interactions at the instant of switching of these phases, determining the Lyupanov functions or applying other traditional control techniques poses a challenge.

Furthermore, the performance measure of a bipedal walker is not as well-defined as that of typical robotic systems. For example, the performance of an industrial robot arm is often measured by how well it can follow a given desired trajectory. In bipedal walking, due to the under-actuated joint at the foot, it may not be physically possible to control the biped to strictly follow the desired trajectory if large foot-ground reaction forces and torques are required. Because of this, many researcher simply use a performance measure based on a binary measure, whether a stable locomotion is achieved or whether the robot topples over while incorporating the dynamics errors.

Because bipedal walking is a challenging control problem, the approach for bipedal walking control usually has to be based on the specific physics of bipedal walking, rather than attempting to develop a general approach which is applicable to other classes of robots.

1.2 Objectives and Scope

In this thesis, the survey scope of the bipedal locomotion generation and control covers algorithms developed from static walking to dynamic walking.

Static walking refers to the walking motions for which the biped's vertically projected Center of Gravity (CoG) always lies within the footprint polygon, which refers to the boundary of the supporting foot during the single support phase or the smallest convex hull containing the two feet during the double support phase. With this constraint condition and for sufficiently slow walking motions, the biped is, at all instant of time, statically stable and the biped will be able to achieve stable walking without falling over. This type of walking is generally only applicable for robots with large footprints and only with slow walking speeds so that the dynamic forces do not affect the stability of the robot significantly.

Dynamic walking, on the other hand, does not require the vertically projected CoG to be always within the footprint polygon and also provides for more realistic, agile and faster walking motions similar to that in human walking. In this type of walking motion, the biped is almost always not statically stable and will topple over because of its momentum if all its joints should suddenly be frozen at any time. Instead of the CoG, the Zero Moment Point (ZMP) is a more important consideration in dynamic walking [19][20]. The ZMP is the point in the ground plane about which the momentum of all the forces applied on the foot or feet by ground reaction forces is zero. However, the ZMP does not have direct implications for walking stability. It only suggests that the prescribed motion will be physically possible if the ZMP lies within the footprint polygon at all times. Dynamic walking allows for larger step lengths, faster locomotion and greater efficiency than static walking. Unfortunately, the stability margin of dynamic walking is much harder to quantify.

Based on the survey, which will be detailed in Chapter 2, the objective of this thesis is then designed to synthesize and investigate a general bipedal walking motion control architecture based on a unified motion generator for different biped robots to achieve 2D and 3D dynamic walking. In addition to achieving stable walking, feedback of certain walking parameters is also incorporated to cater for real-time motion transitions and pattern regulations on level and multi-slope terrains. In the subsequent chapters, the walking task refers to the dynamic walking case unless otherwise specified. The control architecture developed is based on a divide-and-conquer approach in which the dynamic walking task is first decomposed into smaller subtasks. The Genetic Algorithm (GA) technique is first used to generate a suitable basic walking pattern and a learning method is subsequently applied to those subtasks that do not have simple solutions. In general, the characteristics desired of resulting algorithm include:

- Stability. The biped should not fall when challenged with disturbances from foot-ground interactions or other external forces from the environment.
- Versatility. Depending on the application, the biped should have be able to manoeuvre, vary its speed, and walk on rough-terrains.

- Generality. The algorithm should be applicable to bipeds with different geometrical and dynamic parameters.
- Naturalness. The biped should achieve more or less human-like natural motions.

The work presented in this thesis covers 2D rhythmic walking on level and multi-slope terrains and 3D rhythmic walking on level grounds.

1.3 Methodology

The key philosophy adopted in this thesis is to seek a simpler control algorithm that satisfies the specifications stated in the previous section. One of the ways to reduce the complexity of biped control is by task decomposition or the divide-and-conquer approach. For example, 3D bipedal walking can be broken down into motion controls in the transverse, sagittal and frontal planes (see Figure 1-1). Each of these can then be considered individually.

This thesis firstly proposed a simple mathematical model, referred to as the Truncated Fourier Series (TFS) model, to generate suitable basic walking patterns for different walking requirements. Here, the generated basic walking pattern does not mean the ideal pattern for the joint controllers to exactly follow. Rather, considering the fact that it is very difficult to achieve the high precision motion control, which represents the planned optimal pattern in a good accuracy, for biped systems, here the basic walking pattern therefore only means some motion pattern to coordinate and guide the robot motion into some physically stable and robust limit cycle behavior and excite more natural dynamics for the steady-state motion. For the sagittal plane motion control, key parameters such as the fundamental frequency, series amplitude and constant-shift contained in the TFS model are prepared for the composition of subtasks as: 1) stride-frequency adjustment; 2) step-length adjustment; and 3) walking environment adaptation. Through the use of feedback of walking state and a learning agent, the overall control algorithm for the sagittal plane motion adjusts the system towards achieving a stable rhythmic walking

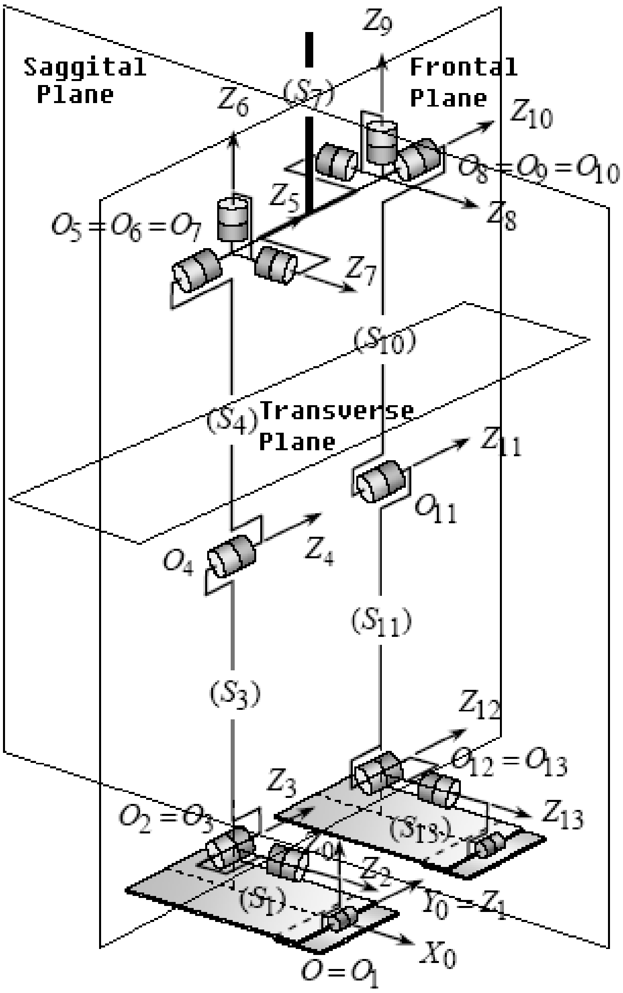


Figure 1.1: Robot motion plane and Degree of Freedom (DOF).

pattern for a range of perturbations due to the external environment. In the frontal plane, based on the TFS formulation, two force generators which balance the frontal plane motion have been proposed and compared. One generates the spring and damper forces concurrently and the other only generates the damper force. Through the application of reinforcement learning, both force generators are aimed to regulate online the lateral behavior and to achieve a stable rhythmic 3D walking motion.

1.4 Simulation Tool

In this thesis, the dynamic simulation tool Yobotics! has been used to test the developed walking algorithms and prove the motion stability throughout the work. Yobotics! is a commercial dynamic simulation package developed by Yobotics, Inc [1]. It is a full-featured software package intended the simulations of robots, biomechanical systems, and mechanical devices. It is based on Newtonian mechanics for interconnected rigid bodies.

The dynamic interaction between the biped and the terrain is established by specifying four ground contact points (two at the heel and two at the toe) beneath each of the feet. The ground contacts are modelled using three orthogonal spring-damper pairs. If a contact point is below the terrain surface, the contact model will be activated and appropriate contact force will be generated based on the parameters and the current deflection of the ground contact model. If a contact point is above the terrain surface, the contact force is zero. (Note: any ground contact will result in the contact point below the terrain surface, even just a very small value.)

Before a simulation is run, the user needs to add the control algorithm and joint controllers to the simulated robot. In the control algorithm, only information that is available to the physical robot is used. The body orientation in terms of the roll, pitch, and yaw angles and the respective angular velocities are assumed to be available. All the joint angles and angular velocities are also known. The contact points at the foot provide information about whether they are in contact with the ground or not.

The outputs of the control algorithm are all the joint torques applied to the simulated robot. Only the dynamics of the biped is taken into account in the simulations while that of the joint actuators are considered to be comparatively negligible and thus ignored in the simulation. That is, the actuators are considered to be perfect torque or force sources.

1.5 Thesis Contributions

The contributions of this thesis are summarized:

- A general motion pattern generator, GAOFSF, for bipedal walking control is developed. It is applicable for bipeds that have similar degrees-of-freedom but of different inertia and geometrical parameters.
- The objective functions of generating a basic walking pattern which can achieve stable walking with softer controllers have been studied. Guided by the generated basic walking pattern, and applying the lower control gains, the resulting motion converges to the steady-state walking smoothly.
- The GAOFSF generated pattern can easily guide the robot to walk in different stride-frequency, step-length and on undulating terrains.
- Successful applications of robot learning algorithms for perturbation adaptation and motion balance control.
- The synthesis of a general motion control architecture for 3D dynamic bipedal walking.

1.6 Thesis Organization

This thesis is organized as follows:

Chapter 2 gives a literature review of the bipedal locomotion research that is relevant to this thesis. It groups bipedal walking research into different categories and examples of each of the these are discussed.

In **Chapter 3** the proposed motion control architecture is presented together with a description of the methods or tools utilized to formulate the walking control algorithms contained in the proposed control architecture. It includes the Truncated Fourier Series (TFS) model which is used as the core walking pattern generator. It also introduces strategies for optimizing the walking pattern generated according to some desired characteristics through the use of Genetic Algorithm and Reinforcement Learning.

Chapter 4 develops a sagittal plane motion control algorithm based on the approach of coordinating the robot motion for stabilizing a basic walking pattern without having to critically depend upon adjusting the joint control gains, considering the difficulties of accurately tracking a planned bipedal motion as an under-actuated system with highly nonlinear dynamics. The motion stability of the sagittal motion control is achieved by the entrainment towards a stable limit cycle walking behavior when the biped is perturbed because of external factors but remains within a region of attraction. This range of attraction has been also investigated for various walking scenarios on different terrains and with different walking postures.

The motion adjustment modes contained in the TFS model for the sagittal plane motion is discussed in **Chapter 5**. A high-level motion supervision module is developed for rhythmic walking control and pattern transitions when there are external perturbations which include the foot-ground interaction, external force disturbances and changes in the terrain.

In **Chapter 6**, a TFS-based motion balance control strategy based on reinforcement learning to achieve stable walking is described. The results of the simulations for various walking examples demonstrating the successful application of this strategy for variable speed 3D walking motions are presented.

The same motion balance control strategy described in **Chapter 6** but enhanced with the introduction of damping is presented in **Chapter 7**. Here again the results of simulations for various walking examples for variable speed 3D locomotion are presented and comparisons made with that obtained in **Chapter 6** in which the spring effect is dominant.

Chapter 8 presents the conclusions for the work done here with some suggestions of areas for further development.

Chapter 2

Literature Review

For a biped robot to achieve a stable and feasible walking gait, the control algorithm needs to comply with the constraints of the bipedal system. One important constraint is the unpowered DOF between the foot and the ground [2]. This constraint, which depends entirely upon the nature of the foot-ground interface, limits the use of the much-studied trajectory tracking approaches used commonly in fixed-base manipulators and is one of the major reasons making the control of bipedal locomotion such a challenging research area.

Many algorithms have been proposed for the bipedal walking task [3]-[18]. As discussed in Chapter 1 bipedal locomotion is a complex problem with a wide range of issues that need to be investigated and in order for an autonomous bipedal robot to be developed which can achieve stable and natural locomotion. As a result, many research works are restricted to only certain aspects of a larger problem. For instance, some works concentrated on the area of mechanical analysis and design, some on various areas of control of the individual links or of the multiple-link mechanism, and some on energetics of biped locomotion. Other researchers further simplify matters by partitioning the biped gait and restricting their analysis either to the sagittal plane or the frontal plane [81] [83].

The various control approaches that have been adopted for dynamic bipedal walking can generally be classified into five basic categories: 1) ZMP-based; 2) model-based; 3) biologically inspired; 4) learning; and 5) divide-and-conquer. The classification is not fully restrictive. Some approach mainly belongs to a specific category but also interacts with the others to some extent.

2.1 ZMP-based

A popular approach used for joint trajectory planning for bipedal locomotion is based on the ZMP (Zero Moment Point) stability indicator [19][20]. ZMP was first introduced by Vukobratovic [2]. Based on this concept, Takanishi et al. conducted a series of work at Waseda University using a stabilization through trunk motion approach [21][22]. In this series of work, the control strategy is to confine the ZMP to be within the single-support or double-support footprint polygon so as to achieve stable locomotion. When the lower limbs move according to the prescribed trajectory, the error which resulted between the desired ZMP and the actual ZMP is to be minimized by adjustments to the body trunk's motion. This approach has been demonstrated to achieve successful stable walking on inclined terrains as well as on stairs. However, the algorithms derived are not applicable to a biped that does not have the extra waist joint on the body.

The humanoid robot (P2 and P3)[23] developed by Honda Motor Company, Limited, are state-of-the art 3D bipedal walking systems. The control approach used is based on playing back trajectory recordings of human walking on different terrains and then modify the joint trajectories through iterative parameter tunings and data adaptation according to the ZMP. Due to the fundamental differences between the robots and their human counterparts, for example, the actuator behaviors, inertias, and dimensions, such reverse engineering becomes rather computation intensive and tedious.

Many other typical works [24] [25] in this category approach the walking control by planning a bipedal walking motion whose ZMP is fully inside the supporting polygon and then to minimize the ZMP trajectory error through some strategies, such as a high tracking accuracy motion controller design based on a precise robot model to achieve the walking to be very close to the prescribed motion and then minimize the ZMP location error [25].

The advantage of the ZMP-based approaches is that robot stability is more clearly ensured based on a proven and sound dynamic basis. The main disadvantage of the ZMP-based control is that the resulting walking motions will be quite restricted. This is because in order to achieve the prescribed motion well, stiff motion controllers are required for good tracking accuracy. Then, robot motion will be rather sensitive to the perturbations, i.e. ground contact impact and terrain surface adjustment. As a result, motions need to be slowed and motion agility will be constrained to some extent. Besides, motion transition may need to be particularly planned to avoid any sudden change at the stance ankle joint torque resulting in motion instability.

2.2 Model-based

Typical model-based control algorithm synthesis is based on a mathematical model of the biped derived from an understanding of the underlying physics of the robot. The massless-leg model is the simplest model used in which the biped is assumed to be a point mass and considered as an inverted pendulum with discrete changes in its support links. This model is applicable only to a biped that has small leg inertia which can be considered as insignificant compared with that of the body, for example when the walking speed is slow and the dynamics of the legs can be neglected without much loss of accuracy.

Kajita et al. [26] derived a massless-leg model for a planar biped that follows a linear motion. During the single-support phase, the resulting motion of the model is treated as an inverted pendulum with a point mass and with a variable pendulum length. With this simplified model, the dynamic equations of the resulting linear motion can be easily solved analytically. Inverse kinematics is used to specify the desired joint trajectories and a simple control law is used at each joint for trajectory tracking. Although the walking stability of Kajita's work is also ensured by the ZMP location inside the supporting polygon, the various dynamic walking motions are all derived based on a massless-leg model. Therefore, Kajita's series of work is considered as a typical research in the model-based category.

When the leg inertia is not insignificant and cannot be ignored, this then needs to be considered in the dynamic model for the biped. One such model is the Acrobot model [27]. It is based on a double pendulum with no actuation between the ground and the base link corresponding to the stance leg. Although the Acrobot model has not been directly applied to the bipedal robot walking control, it is quite commonly used to characterize the single-support motion of the bipedal locomotion study.

In addition to the inverted pendulum model and the Acrobot model, linearization have been also used with respect to selected equilibrium points to simplify the multi-joint models. Mita et al proposed a control method for a planar seven-link biped using a linear optimal feedback regulator [11]. The model of the biped was linearized about a commanded posture. Then, linear state feedback control was used to stabilize the system to be not much deviated from the commanded posture. However, the work assumed that the biped had no reaction torque limitations at the stance ankle and the biped was given large feet so that the assumption was valid.

The advantage of model-based control approaches is that some analytical walking solutions can be obtained by simplifying to a dynamic model that can be solved by known analytical methods. However, the major disadvantage is that with the simplifying assumptions, the control strategy that is derived using this approach may not work well in actual implementation unless either the simplified dynamic model still represent the actual model to a good degree of accuracy or if the actual robot is fabricated according to the model used. For example, with the massless-leg model, the target robot for implementation will need to have very light legs.

2.3 Biologically Inspired

Recently, the biologically inspired based walking control started to get more and more attentions. For example, one important biological concept, Central Pattern Generators (CPG)[28]-[33] has triggered many interests for locomotion control. Another typical biological inspired walking approach, passive walking control, is based on the observation that human beings do not need high muscle activities to walk. Note only approaches inspired by some proved biological findings and concepts are classified into the biologically inspired category. Approaches such as migrating the recorded human gaits to robot walking are not considered to be inclusive.

2.3.1 Central Pattern Generators (CPG)

CPG is defined based on the findings that certain legged animals seem to coordinate their muscles through some kind of central motion generator without using their brains. It was first proposed by Grillner [34] who found from experiments on cats that the spinal cord generates the required signal for the muscles to perform coordinated walking motion. The existence of a central pattern generator that is a network of neurons in the spinal cord was thus hypothesized.

The typical approach using the idea of CPG is the composition of a system of coupled nonlinear equations which can generate signals for the joint trajectories of bipeds. The biped is expected to achieve a stable limit cycle walking pattern with the use of these equations.

Started from Matsuoka's work about neuron oscillator for walking locomotion study [35], Taga has conducted a series of work [31] about a neural rhythm generator for the approximation of human locomotion. The neural rhythm generator was composed of artificial neural oscillators which received sensory information from the bipedal system and generated as output signals to the actuators in the system. Based on numerical simulations, a stable limit cycle behavior was entrained. Recently, the neuron oscillator based motion generator has been further investigated and successfully applied to adaptive dynamic walking of a quadruped robot on irregular terrain by Fukuoka et al[36].

Bay and Hemami[3] demonstrated that a system of coupled van der Pol oscillators could generate suitable periodic signals for bipedal locomotion. These generated periodic signals were applied to the walking task to produce rhythmic locomotion. However, in their analysis, the dynamics of the biped such as the force interactions between the support leg and the ground were not considered. The van der Pol oscillator based motion pattern generator has been further studied and explored by many researchers, i.e. Teresa [37][38].

One weakness of the reported works based on the CPG approach [28]-[31] is that the suitability for use of the coupled nonlinear equations for bipedal locomotion was based on the extent of the similarity of the generated joint trajectory signal profiles to that obtained from experiments on human gaits. The essence is the search for a set of coupled equations which can more or less mimic the joint trajectories profiles of human walking without any other consideration based on proven concepts of the physics, mechanics or dynamics of the robot and its motion. Therefore, it is difficult to find systematically a set of parameters that can enable entrainment of the overall system applicable for different walking situations. Even if a periodic stable walking behavior can be obtained, it is still difficult to predict the walking behavior when the robot is subjected to disturbances or changes in the locomotion because the causality between the parameters involved in the nonlinear equations and the resulting motion has not been clearly defined.

2.3.2 Passive Dynamics

The study of passive dynamics in walking provides an interesting natural dynamic model for the mechanics of human walking [39][40][41]. It was partly inspired by a bipedal toy that was capable of walking down a slope without any power source other than gravity. The toy rocked from left and right in a periodic motion. When a leg lifted off the ground at the end of a half-period motion, it would swing freely forward, acted on by gravitational forces, and arrived in a forward position to support the toy for the next half period of the motion. If the slope is within a certain range, a stable limit cycle walking motion can be achieved. When this occurs, the work done on the toy by the gravitational force will be equal to the energy loss in the biped.

Then, Goswami et al [42] and Thuilot et.al [43] studied nonlinear dynamics of a compass-like biped robot. The model included two variable length members with lumped masses representing the upper body and two limbs. The authors observed limit cycles as well as chaotic trajectories. They primarily focused on the following parameters: ground slope, mass distribution and limb length. Later, the work of Garcia et al [44] and Coleman et al [45] represent a step forward in the research of passively walking bipeds. Robotic models with rounded and point feet were used. Furthermore, kneed and straight-legged bipeds have been also considered in passive dynamics. They also showed the existence of walking gaits on arbitrarily small slopes.

Although passive walking has properties like being able to achieve a minimum energy gait without active control, it is rather sensitive to parameter variations [40] such as mass distribution and joint friction.

2.4 Learning

Learning is commonly applied to systems where known analytical approaches cannot be used and when the dynamic models cannot be accurately derived. In many cases, learning is also used to modify a nominal behavior that are generated based on a simplified model.

Benbrahim and Franklin [46] applied reinforcement learning for a planar biped to achieve dynamic walking. They adopted a ”melting pot” and modular approach in which a central controller used the experience of other peripheral controllers to learn an average control policy. The central controller was pre-trained to provide nominal trajectories to the joints. Peripheral controllers helped the central controller to adapt to any discrepancy during the motion. A dynamic model for the system was not required in the implementation. One disadvantage of this approach is that the nominal joint trajectories applicable for the central controller training may not be easily obtainable.

Russ et.al [47] developed a stochastic policy gradient reinforcement learning on a simple 3D biped robot to quickly and robustly obtain a feedback control policy. The robot was modelled after a passive walker to reduce the complexity level for the learning process. Then it allows to learn with only a single output which controlled a 9DOF system. Furthermore, by such a modelling of the robot, the motion can be formulated on the return map dynamics which dramatically increased the number of policies in the search space for the generation of stable walking. The learning algorithm worked well on simple robot, but whether it can also work well on more complicated robots is still left for exploration.

Chew [48][49] built up a general control architecture achieved by reinforcement learning, using the CMAC network as the function approximator. There is no joint trajectory pre-planned or pre-defined. The proposed motion control was to learn the walking stride and an offset value defined in balancing control with a local controller incorporated. The derived local controller was found to be effective for reducing the computation cost. The limitation of this control approach is that the resulting walking posture may not be very periodical although they are all feasible. Also, the strategy does not allow for the adjustments to the stride-frequency during motion because the local controller does not incorporate any parameter which can be varied with respect to time.

With the development of computation technologies, the Artificial Intelligence (AI) based computation methods become more and more popular but the common issues for the AI based techniques is the computation cost is high and the tolerance for the computation reliability is still rather limited. Currently learning is generally more reliable for small state-space tasks.

2.5 Divide-and-Conquer

Due to the complexity of the bipedal walking robots, many algorithms break the problem into smaller sub-problems that can be solved more easily. However, experience and intuition is usually required for such an approach, both in deciding how to break down the problem and how to solve the smaller sub-problems. Intuition can be obtained by observing the behavior of bipedal animals or by analyzing simple dynamic models, etc.

Pratt et al. [50][51] presented a control algorithm called ”Turkey Walking” based on a divide-and-conquer approach for the planar bipedal walking problem in a biped called ”Spring Turkey”. The walking cycle was first partitioned into two main phases: double support and single support. A simple finite-state machine was used to keep track of the currently active phase. In the double support phase, the task of the controller consisted of three sub-tasks: 1) body pitch control; 2) height control, and 3) forward speed control. In the single support phase, the task of the controller consisted of two sub-tasks: 1) body pitch control and 2) height control. The resulting algorithm was simple without the need to use dynamic equations.

Raibert’s control algorithms [52] for hopping and running machines also mostly utilized the divide-and-conquer approach. The control algorithm for a planar one-legged hopping machine was decomposed into: 1) the hopping motion (vertical), 2) the forward motion (horizontal), and 3) the body posture. These sub-tasks were considered separately and each was solved by using simple control algorithms. This resulted in a simple set of algorithms for the hopping task.

The divide-and-conquer approach has been proven to be simple and effective for practical implementations. However, not all bipedal systems can be easily be decomposed with the sub-tasks solved using simple, direct or analytic solutions. Often, when decomposed improperly, these sub-tasks may be coupled to and affect one another which resulted in the decomposed systems not accurately representing the total system.

2.6 Summary

This chapter provides a review of past works on dynamic bipedal locomotion, giving a classification of the research that has been done with details on each of these.

Motion feasibility is more ensured using the ZMP-based approach, given the motion controller can track the planned motion precisely. However, the pure ZMP-based approach walking control may be sensitive to the environment perturbations due to the applied stiff control gains. Therefore, the resulting motion is not very compliant to transitions.

The model-based approach can be an excellent approach if a simple and accurate-enough model can be used and for which well-developed solutions are available.

The biologically-inspired Central Pattern Generator (CPG) approach to joint trajectory generation has been studied and shown to be capable of generating periodic motions which can be used in bipedal locomotion. Its weakness is that the motions generated cannot be easily further developed or adjusted based on known the underlying physics and mechanics of the system in order to adapt to robots of different inertia or geometrical characteristics, to desired changes in stride frequency or length, to perturbations from the external environment. As such, their applications to the bipedal robot locomotion may be limited.

Studies in passive walking has given good insight to how robots can be made to walk like human beings without the need for active actuators or additional energy input other than that due to the potential energy due to gravity. Such walking style, unfortunately, is applicable only to walking down slopes of a limited range for robots of a certain structure. Still, the insights gained form such studies can greatly help the development of more energy- and effort-efficient bipedal locomotion.

With the rapid development of high speed computers and computation technologies, the learning approach has become a very promising area for further study, research and development. However, these approaches can become intractable if there are too many learning agents or when the bipedal walking task is not broken down into smaller and less-complex sub-tasks.

The divide-and-conquer approach has been demonstrated to be effective in breaking the complex walking problem into smaller and more manageable sub-tasks. These sub-tasks can then be more easily solved using established control techniques. Some sub-tasks, however, may still not have easy analytical or other solutions and will need some computational methods to achieve suitable solutions.

Based on the above literature survey, each method has shown some advantages but also posed some limitations. In this thesis, the proposed approach uses the divide-and-conquer approach but, in the implementation of the sub-tasks, it also makes use of the other control approaches mentioned earlier, including the use of the ZMP for motion stability considerations, the CPG concept for low-level motion convergence behavior and the learning approach to achieve good control of the locomotion without the need for rigorous analytical approaches with accurate dynamic models.

Chapter 3

Control Architecture and Algorithm Implementation Tools

In this chapter, a general view of the proposed walking motion control structure is first presented. Then, the core component of the motion control strategy, a basic motion pattern generation method, is then introduced. Here, the proposed motion generation method is named as Genetic Algorithm Optimized Fourier Series Formulation (GAOFSF) [53][54] and the mathematical formulation involved in the GAOFSF method is named as the Truncated Fourier Series (TFS) formulation. In addition to the introduction of the walking control architecture, necessary algorithm implementation tools such as the Genetic Algorithm (GA) and Reinforcement Learning (RL) are also presented. GA is used to search for an optimal motion pattern defined in the GAOFSF, and reinforcement learning is applied to the subtasks of motion adjustment on an as-needed basis. A reinforcement learning algorithm called Q-learning [55] is adopted, working with a function approximator called Cerebellar Model Articulation Controller (CMAC)[56] to generalize the learning experience for real-time correcting motions when perturbations occur.

3.1 Control Architecture

A 3D bipedal walking system can be very complex to analyze if it is not partitioned into smaller components or sub-tasks. It is difficult to apply a unified control algorithm for such a complex system. In this thesis, a divide-and-conquer approach is adopted in the formulation of the dynamic walking algorithm. By studying each subtasks separately, the problem becomes less complex. Appropriate control algorithms can then be applied to each of them. This section presents the framework for such a task decomposition approach. The following subsections define the subtasks considered in the three orthogonal motion planes and the whole control architecture that is composed.

3.1.1 Sagittal Plane

Sagittal plane motion is usually the largest during normal forward walking. The subtasks important for sagittal plane motion control are considered as: 1) maintaining body pitch [53], 2) maintaining desired walking speed (stride-frequency and step-length) [53], and 3) walking adaptation on uneven terrains [57][58][59]. It is not difficult to achieve the first subtask in the sagittal plane as it can be directly assigned to the joint control scheme. However, the second and the third subtasks become much more complex since they are directly associated with gait stabilization. Therefore, these two subtasks are particularly determined by the composition of a motion generator and control strategy which should take the flexibility and generality issues into account and achieve the environment entrained motions.

In this thesis, the total sagittal plane motion control is divided into two levels [60]. The low level control maintains the motion stability through a limit cycle behavior based on a generated basic walking pattern and the high level control modifies the basic walking pattern based on sensed dynamics and environment feedback. Therefore, in the low-level control part, conditions of the basic walking pattern to be generated to coordinate a walking motion and converge the motion into a feasible limit cycle behavior are investigated. Then, for the high-level motion adjustment part, key parameters contained in the TFS formulated motion generator are discussed on their particular applications for real-time gait adjustment, such as the stride-frequency, step-length and walking posture adjustments.

3.1.2 Frontal Plane

For normal walking along a straight path, frontal plane motion is smaller than sagittal plane motion. In the frontal plane, the dynamic walking task can be decomposed into: 1) maintaining the body roll angle [61] and 2) maintaining lateral balance [61].

The major control difficulty of the frontal motion control comes from the bigger acceleration component in changing the direction of the lateral motion. In the situation that only limited torque can be applied to the stance foot, such change of the motion direction makes the desired frontal motion trajectories less trackable. In addition to the trajectory tracking issue, ground contact behavior cannot be perfect as assumed. Due to the above difficulties, to achieve a long distance 3D walking locomotion, the motion control applied to the robot needs to incorporate a feedback loop. Therefore, the thesis explores two strategies using reinforcement learning algorithm for online dynamics compensation and the achievement of the prolong 3D walking motions.

Similar to the sagittal plane motion control, the motion control applied to the frontal plane motion is also a two-level based control. The low level control aims to maintain the basic motion pattern and the high level control executes adjustment when it is necessary.

3.1.3 Transverse Plane

For normal walking along a straight path, transverse plane motion is usually the simplest to control. In this case, the walking task can be decomposed into: 1) maintaining the body yaw angle and 2) maintaining the swing foot yaw angle. Both the body yaw angle

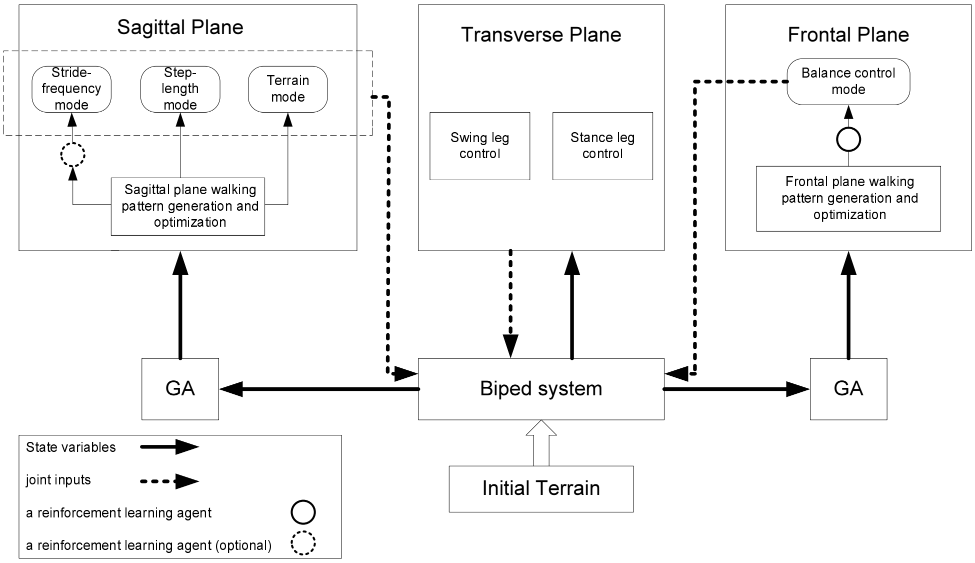


Figure 3.1: Proposed control architecture.

and the swing foot yaw angle can be set to zero if walking is facing forwards. Both subtasks can be easily achieved.

Based on the above illustrated sub-tasks of each orthogonal plane, the overall control architecture is summarized as shown in Figure 3.1.

3.2 GAOFSF Motion Generation Method

As mentioned, the sagittal motion is the major component of the entire walking control algorithm, and the basic walking pattern generation for the sagittal motion control is the most critical part for achieving the desired motion behavior and locomotion stability. Therefore, this section details the proposed motion generation method, the Genetic Algorithm Optimized Fourier Series Formulation (GAOFSF) approach [53]. It is aimed to be general for motion pattern generation according to user-defined performance indices. In this GAOFSF method, the Truncated Fourier Series (TFS) formulation is used to approximate the joint trajectories, and the GA is used to search for the optimal values of the parameters in these formulations describing the desired pattern. Furthermore, this

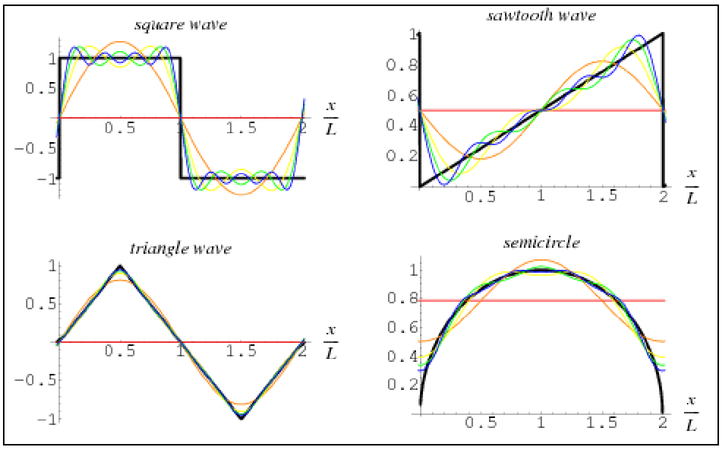


Figure 3.2: Examples of common function approximation using Fourier series [Quoted from Wolfram MathWorld].

GAOFSF approach is flexible for different walking patterns and general for robots of different mass and inertia properties. The following illustration of the GAOFSF approach starts from the basic mathematic formulation TFS.

3.2.1 Truncated Fourier Series Formulation

A Fourier series is an expansion of a periodic function $f(x)$ in terms of an infinite sum of sines and cosines *Wolfram MathWorld* [87]. It can make use of the orthogonality relationships of the sine and cosine functions to approximate any profile provided the order of Fourier series can be high enough. Figure 3.2 (Quoted from MathWorld) shows the way of approximations to common functions using Fourier series.

The Fourier series of a function is defined as:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos(nx) + \sum_{n=1}^{\infty} b_n sin(nx) \tag{3.1}$$

where $n = 1,2,3,...$. For a function $f(x)$ periodic on an interval $[-L,L]$ instead of $[-\pi,\pi]$, a simple change of variables can be used to transform the interval of integration from $[-\pi,\pi]$ to $[-L,L]$. Let

$$x \equiv \frac{\pi x'}{L} \tag{3.2}$$

$$dx = \frac{\pi dx'}{L} \tag{3.3}$$

Solving for x' gives $x' = Lx/\pi$, and plugging this in gives

$$f(x') = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos(\frac{n\pi x'}{L}) + \sum_{n=1}^{\infty} b_n sin(\frac{n\pi x'}{L}) \tag{3.4}$$

If a function is even so that $f(x) = f(-x)$, then $b_n = 0$ for all n. Similarly, if a function is odd so that $f(x) = -f(-x)$, then $a_n = 0$ for all n.

Since a stable walking locomotion has the following motion characteristics as:

1. Joint trajectories are periodical and the shapes, especially the hip joint trajectories resemble sinusoidal profiles [32] [33].

2. Any of the joint trajectories can be divided into phases such as positive and negative with respect to some off-set value, or constant and variable according to typical shapes from literatures [32] [33].

Therefore, given phase identifications, trajectory during each phase can be regarded as an odd function. Thus, a_n can be just assigned to be zero for all the cosine series. For those sine series start from an off-set value, it is defined as the Truncated Fourier Series formulation. In this thesis, this TFS formulation is used as the basic module for generating an optimal motion pattern according to the user defined objective functions. The applications for the generation of different basic walking patterns are presented in Chapter 4.

3.2.2 GAOFSF Motion Generator

For sagittal plane motion, the gait synthesis for this bipedal robot involves first determining the hip-pitch, $O_5 - Z_5$, and knee-pitch, $O_4 - Z_4$, joint trajectories for each of the legs referring to Figure 1.1. The ankle-pitch angle of the swing leg is then determined based on the condition that it is always parallel with the ground surface. The ankle-pitch joint trajectory of the stance leg is not planned as the foot is allowed to interact naturally with the ground in the motion controller that is presented in Chapter 4. The following subsection starts the desired motion generation from the elaboration of features of human gaits because one of the research objectives of this thesis is to achieve the human-like walking pattern.

For a deeper insight into human motion properties, 3D *VICON* motion registration system and *POLYGON* human motion analysis software were used to record and process human gaits. Figure 3.3 shows the hip and knee trajectories for a 1.83m tall person (74kg weight). The reference frames are the same as what is defined for the biped model, as shown in Figure 1.1 (Coordinates $O_3 - Z_3$, $O_4 - Z_4$ and $O_5 - Z_5$ without considering the toe joint $O_1 - Z_1$ at one side). The gait analysis software verified the obtained data comparing it with norms stored. From the gaits recorded by *VICON*, considering the fact that human body is physically different from the robot's rigid links, Figure 3.4 is derived in a general form applicable to different robots by capturing the main features of Figure 3.3. The trajectories for both legs are identical in shape but are shifted in time relative to each other by half of the walking period. For example, θ_{lh} for the left hip is identical to that for the right hip, except that θ_{lh} is time shifted by $(t_6 - t_0)/2$ w.r.t. θ_{rh}. It is also noted that the joint angle trajectories can be referenced by ”offsets”. The values of the defined offsets actually influence the biped's posture during walking. c_h denotes the hip angle offset. This is the value of both hip joint angles at the point they become equal, or at which the two thighs cross each other. c_k denotes the knee angle offset. This is the value of the knee angle when the knee is locked during part of the support phase. t_1 and t_2 denote the start and end time, respectively, of the lock phase. Such a lock phase is comparable to the physical pattern of the stance leg lifting up the

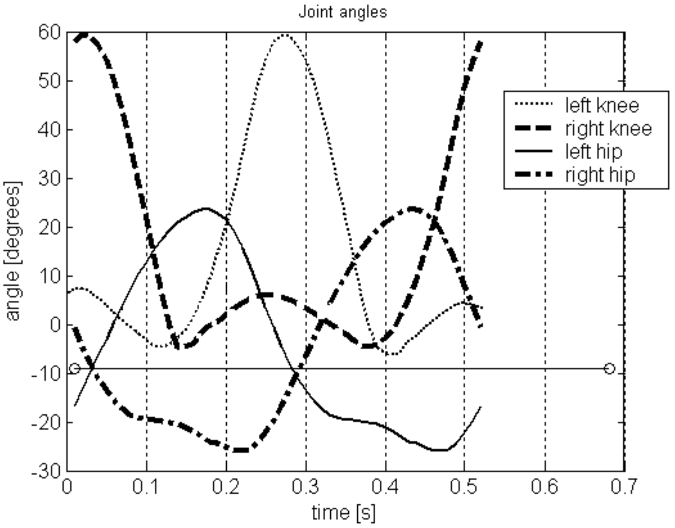


Figure 3.3: Human gaits recorded by VICON motion registration system.

body and then straightening, corresponding to $t = 0.14s$ to $0.4s$ in Figure 3.3.

Consider first the hip angle trajectories. They can be divided into an upper portion, θ_h^+, for which $\theta_h \geq c_h$, and a lower portion, θ_h^-, for which $\theta_h < c_h$. Thus, referring to Figure 3.3 for the two portions of the walking cycle, the hip joint angles for the two legs are given by

$$\begin{cases} t \in [t_0, t_3) \;\; \theta_{rh} = \theta_h^-(t) \;\; \theta_{lh} = \theta_h^+(t) \\ t \in [t_3, t_6] \;\; \theta_{rh} = \theta_h^+(t - t_3) \;\; \theta_{lh} = \theta_h^-(t - t_3) \end{cases} \tag{3.5}$$

where θ_{rh} and θ_{lh} are the right and the left hip joint angles, respectively.

Similarly, the right knee joint angle trajectory for different portions of the walking cycle is given by

$$\begin{cases} t \in [t_0, t_4) \;\; \theta_{rk} = \theta_{k1}(t + t_6 - t_5) \\ t \in [t_4, t_5) \;\; \theta_{rk} = \theta_{k2}(t) \\ t \in [t_5, t_6) \;\; \theta_{rk} = \theta_{k1}(t - t_5) \end{cases} \tag{3.6}$$

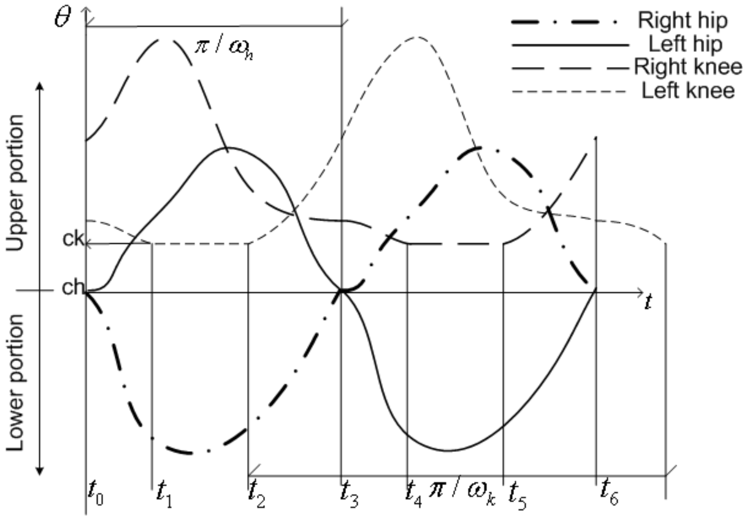


Figure 3.4: Uniform gaits elaborated from human gaits features.
Right support: $[t2,t5)$; Right swing: $[t0,t2)$ and $[t5,t6]$;
Left support: $[t0,t2)$ and $[t5,t6]$; Left swing: $[t2,t5)$.

where θ_{k1} is the knee joint trajectory from the beginning of swing phase, denoted by t_5 for the right knee in Figure 3.4, to the instant in the support phase when the knee joint is locked, denoted by t_4 in Figure 3.4. θ_{k2} is the locked knee joint angle.

Similarly, referring to Figure 3.4, the joint angle for the left knee is given by

$$\begin{cases} t \in [t_0,t_1) & \theta_{lk} = \theta_{k1}(t+t_6-t_2) \\ t \in [t_1,t_2) & \theta_{lk} = \theta_{k2}(t) \\ t \in [t_2,t_6) & \theta_{lk} = \theta_{k1}(t-t_2) \end{cases} \tag{3.7}$$

where t_1 is the instant when the stance knee is locked and t_2 is the instant when the walking phases of the two legs are switched.

Joint Trajectory Representation using the TFS

As discussed above, the Fourier Series of a periodic function of time $f(t)$ can be written as

$$f(t) = \frac{1}{2}a_0 + \sum_{i=1}^{\infty} a_i sin(\frac{\pi i}{T}t) + \sum_{i=1}^{\infty} b_i cos(\frac{\pi i}{T}t) \tag{3.8}$$

where a_i and b_i are constant coefficients and T is the period. The fundamental frequency is given by $\omega_1 = \frac{\pi}{T}$.

As mentioned in the previous section, all the joint trajectories during a gait cycle can be divided into two portions. Each portion can be viewed as an odd function output according to the intersection with the angle axis. Therefore the sine series in the Fourier series function, in Equation (3.4), is simplified to be the Truncated Fourier Series (TFS) used to model each portion as:

$$f(t) = \sum_{i=1}^{n} a_i sin(i\omega t) + c_f, \quad \omega = \frac{\pi}{T} \tag{3.9}$$

where a_i, and c_f are constants to be determined and ω is the fundamental frequency determined by the desired period of the gait. The parameter n, which determines the number of terms in the Fourier series, is chosen as a trade-off between the accuracy of the approximation required and the computational load. The formulation as shown in Equation (3.9) is used for the joint angles $\theta_h^+(t)$, $\theta_h^-(t)$ and $\theta_{k1}(t)$ as given in Equations (3.5) to (3.7). Although the function is periodic, only the first half of the period is needed for the joint angles. It is noted here that since the shapes of the upper and the lower portions are not symmetrical about the $(0, c_h)$ point, even if the full Fourier series is used, this cannot automatically give an equal time period for the upper and the lower portions of a walking motion. The use of the full Fourier series, as with other approximation functions, will therefore also require an additional mathematical constraint to fix the profile so that the upper and the lower portions intersect at the points $(0, c_h)$ and (T_s, c_h), T_s being the step period. As can be seen from the foregoing, the use of the TFS allows for a reduced series with fewer parameters for the same approximation accuracy, and with fewer constraints required. This significantly reduces the subsequent computational load in the search for feasible and optimal solutions using GA.

Using Equation (3.9) and based on Figure 3.3, the TFS for the hip pitch angles are then formulated as:

$$\theta_{rh}, \theta_{lh} = \begin{cases} \theta_h^+ = \sum_{i=1}^n R \cdot A_i sin i \omega_h (t - t_h^+) + c_h \\ \theta_h^- = \sum_{i=1}^n R \cdot B_i sin i \omega_h (t - t_h^-) + c_h \end{cases} \tag{3.10}$$

where $\omega_h = \frac{\pi}{t_3 - t_0} = \frac{\pi}{t_6 - t_3}$, A_i and B_i are constant coefficients, θ_h^+ and θ_h^- the upper and the lower portion respectively of the hip joint angle trajectory, and t_h^+ and t_h^- are time-shift values according to Equation (3.5). R is an amplitude scaling parameter used for changing the step-length. Initially, R is set to be 1.

Similarly, the trajectories for the knee joint angles are expressed as:

$$\theta_{rk}, \theta_{lk} = \begin{cases} \theta_{k1} = \sum_{i=1}^n R \cdot C_i sin i \omega_k (t - t_k) + c_k \\ \theta_{k2} = c_k \geq 0 \end{cases} \tag{3.11}$$

where $\omega_k = \frac{\pi}{(t_6 - t_2) + (t_1 - t_0)}$, C_i are constant coefficients and t_k is the time-shift.

Compared with other trajectory synthesization approaches[22] [77] [78], the advantages of using the TFS to synthesize the walking gait generator for bipedal robots are as follows:

- With only a few terms in the series, it can represent quite accurately the shapes of the required joint trajectories for human walking inspired biped robot gait, for which the upper and the lower portions are not symmetrical but individually are similar to half a sinusoid.
- Each TFS used here is a simple expression that makes no mathematical assumption when considered as half of an odd function. The gait period is included

directly. Other functions, such as the Spline, Gaussian or the full Fourier Series Function will add additional constraints to regulate the motion period.

- Motion planning is directly conducted in joint space. Thus, the resulting body motion is a direct representation for a robot CG motion, i.e., motion smoothness. Also, mathematical singularity is not an issue needs to be considered in the motion generation as it avoids the inverse-kinematics in motion planning.
- Key parameters of the TFS can be easily adjusted on-line during walking to change the walking gait, either of the desired pace, or in response to terrain irregularities. (This will be discussed in Chapter 5).

Gait Synthesis

In the gait synthesis, a suitable set of coefficients A_i, B_i, C_i and parameters c_h, c_k, t_1, t_2 (as indicated in Figure 3.4) need to be obtained using the GA [62]. t_1 is the instant when the knee of the stance foot starts to lock and t_2 is the instant when the walking phases of the two legs are switched.

Through experimentation, it was found that a value for n in both Equation (3.10) and (3.11) as low as 5 can give good performance. This value was thus used. In this case, there will be altogether 19 parameters to be determined for the sagittal plane motion. For the GA, these parameters are coded in a chromosome which has 19 genes. The format of the chromosome is set as $x = [A_i, B_i, C_i, c_h, c_k, t_1, t_2]$. Here, the real number chromosome representation was adopted which, from experiments performed, is more efficient than the other representations. It takes about 1 minute on a Pentium IV 2.8 GHz computer to come up with the solution fulfilling all motion requirements.

The key component of the GA is the formulation of the fitness function. GA optimization is based on the minimization or maximization of the fitness which conducts a trade-off among multiple motion behaviors important for stable walking. The objective function is (to be minimized) as formulated in the form:

$$f = \sum_{i}^{n} w_i f_i \tag{3.12}$$

where $w_i, i = 1, 2, ...n$ are the weighting factors, and $f_i, i = 1, 2, ...n$ are the performance indices for optimization. In this thesis, two motion optimization strategies have been applied: 1) motion symmetry optimization, programme code Op_1 and 2) motion robustness optimization, programme code Op_2. The reasons for the derived objective functions are presented in the later sections where the corresponding optimization is used. Here, only the objective and constraint functions are listed.

For optimization programme code Op_1, the following motion objective functions are derived for motion symmetry optimization:

f_1 = the absolute difference between the positive and negative angular momentum generated by the stance ankle joint torque. Here, positive is for the stance ankle joint torque larger than τ_0 which occurs at the center line position $X = 0$, and vice versa for negative,

$$f_1 = ||S^+| - |S^-|| \;\; where\ S^- = \int_0^{t^-} \tau_{13} dt\ and\ S^+ = \int_0^{t^+} \tau_{13} dt \tag{3.13}$$

where τ_{13} is the stance ankle joint torque.

f_2 = the distance between the points $(t_m, 0)$ and $(T_s/2, 0)$ where $(t_m, 0)$ is the intersection point of the line: $\tau_{13} = \tau_{13}(min) + (|\tau_{13}(max)| - |\tau_{13}(min)|)/T_s t$ with the time axis, and $(T_s/2, 0)$ indicates the half step-period.

$$f_2 = |t_m - T_s/2| \tag{3.14}$$

Here the stance ankle joint torque is computed by:

$$\tau_{13} = mgx_G - \dot{H}_{oy} = \sum_{i=1}^{n} (m_i x_{G_i} g - (\dot{H}_{G_i y} + m_i (z_{G_i} \ddot{x}_{G_i} - x_{G_i} \ddot{z}_{G_i}))) \tag{3.15}$$

where x_{G_i}, y_{G_i}, and z_{G_i} are the coordinates of the centroid of link i; m_i the mass of link i; $\dot{H}_{G_i}$, the rate of angular momentum about point G_i.

f_3 = standard deviation of the trunk velocity from the desired average velocity over a walking cycle,

$$f_3 = \sqrt{\frac{(v_1 - V_{B1})^2 + (v_2 - V_{B1})^2 + ...(v_{m_1} - V_{B1})^2}{m_1}} + \sqrt{\frac{(v_{m_{1+1}} - V_{B2})^2 + (v_{m_{1+2}} - V_{B2})^2 + ...(v_{m_2} - V_{B2})^2}{m_2 - m_1}} \tag{3.16}$$

where v_i , $i = 1, 2, ...m_2$, are the trunk's velocity at the i^{th} sampling instant and V_{B1} and V_{B2} are the desired average walking speed over the first and the second half of a step referring to the center line $X = 0$, respectively. For the generation of dynamically symmetrical walking patterns, $V_{B1} = V_{B2}$, but for the asymmetrical walking patterns, V_{B1} and V_{B2} will be assigned to be different.

For optimization programme code Op_2, the performance optimization indices are listed as follows:

f_1 = standard deviation of the trunk velocity from the desired average velocity over a walking cycle, same as the above f_3 by setting $V_{B1} = V_{B2}$.

f_2 = the body CG displacement before and after crossing the center line $X = 0$ is being comparable.

$$f_2 = ||X_{cg}^+| - |X_{cg}^-|| \tag{3.17}$$

f_3 = minimization of the distance between the average ZMP location P_m over a walking step and the center P_c of the stance foot-print, formulated as

$$f_3 = |P_m - P_c| \tag{3.18}$$

If the frontal plane motion is incorporated, one more motion performance objective function will be added: f_4 to minimize the angular momentum generated by the rolling joint of the stance foot.

$$f_4 = Min(\int_0^T \tau_{a_r} dt) \tag{3.19}$$

where τ_{a_r} is the rolling joint torque of the stance ankle.

In addition to the above objective functions to be minimized, there are five penalty functions used as motion constraints to ensure a valid walking posture. Motion constraints work differently from the previous motion objectives. For a motion constraint, there is no penalty if the constraint is not violated. A penalty is only imposed when the motion violates the constraint. The five motion constraint penalty functions are described below.

For optimization code Op_1:

s_1: constrains the motion symmetry to be within a certain level, formulated as:

$$s_1 = max((f_1 - a_{max}), 0) + max((a_{min} - f_1), 0) + max((f_2 - b_{max}), 0) + max((b_{min} - f_2), 0) \tag{3.20}$$

where a_{max}, a_{min} and b_{max}, a_{min} describe the symmetry level that is considered in the GA. For the symmetrical motion generation, $a_{min} = 0$, $b_{min} = 0$.

For optimization code Op_2:

s_1: constrains the average ZMP location P_m to be always inside the supporting polygon of the stance foot:

$$s_1 = max((|OP_m - OO'| - L_f/2), 0) \tag{3.21}$$

where OO' is the length from the center of the supporting polygon to the ZMP coordinate, L_f the foot-length of the robot.

The following s_2 to s_5 are the common motion constraints for both codes Op_1 and Op_2.

s_2: constrains the swing height to be above a specified minimum value except for the phase-switch moment so that the swing foot does not hit the ground prematurely or drag on the ground. Let:

m be the number of the sampling instances within one walking step.

d_1 and d_2 be the sums of the differences between the desired swing height and the actual height for one walking step.

H_r and H_l be the length projections of the right and the left leg respectively onto the vertical plane with respect to the terrain surface.

H_{min} be the minimum swing height. With this constraint, although the swing height cannot be always above the minimum height, the problem of the swing foot dragging on the ground for a long time will be avoided.

```
$for\ i = 1 : m$                                                    (3.22)
            $if(t(i) \leq t_2)$
            $d_1 = d_1 + max((H_r - H_l - H_{min}), 0);$
            $elseif(t(i) > t_2)$
            $d_2 = d_2 + max((H_l - H_r - H_{min}), 0);$
            $end$
            $s_2 = d_1 + d_2;$
      $end$
```

s_3: constrains the swing foot's velocity to be always positive except during the short time period before or after the touchdown moment,

$$s_3 = max(-v_f, 0) \tag{3.23}$$

where v_f is the swing foot velocity and the max function is taken over the sampling instances for the walking step except for the ones before or after the touchdown moment.

s_4: constrains the deviation of the step length from the desired to be within a small specified value,

$$s_4 = |L - L_0| \tag{3.24}$$

where L and L_0 are the actual and the desired step length, respectively.

s_5: constrains the deviation of the touch-down instant of the swing foot, t_d, from the planned phase switching time, t_2, to be within a small specified value,

$$s_5 = |t_2 - t_d| \tag{3.25}$$

Based on these constraints, the penalty function is defined as

$$P = \sum_{i=1}^{5} p_i s_i \tag{3.26}$$

where $p_i, i = 1, 2, ...5$, are assigned penalty weighting factors.

Using Equations (3.12) and (3.26), the fitness function for the GA algorithm is established as

$$fitness = \begin{cases} 0, & if\ \sum_{i=1}^{n}|A_i| = 0\ or\ \sum_{i=1}^{n}|B_i| = 0\ or\ \sum_{i=1}^{n}|C_i| = 0 \\ C_{max} - f - P \end{cases} \tag{3.27}$$

The first expression of Equation (3.27) is used to avoid the possible situation of standing still which can give a very good performance on stance foot stable on the ground. The parameter C_{max} is chosen to be as small as possible in order to have a better differentiation among various possible solutions. However, it should also be such that the fitness value for most, if not all, possible solutions are positive, (because the selection method used in this thesis is based on probabilities, not ranking). Suitable values are chosen by trial and error.

In selecting the values for the weight factors w_i and p_i, consideration was given to balance all objectives and constraints, usually through estimating f_i and s_i values of a motion with all the objectives and constraints achieved at the average level and then estimating the weight factors to make them about equally important.

3.3 Implementation Tools

In order to achieve the proposed motion control architecture, some computation methods are needed to get the numerical solutions because the analytical solution of a biped system is very difficult to derive currently. In this thesis, Genetical Algorithm (GA)[62] and Reinforcement Learning (RL)[53][54] are used on a necessary basis. In this section, important computing methods that have been used in the proposed bipedal walking control algorithm are described.

3.3.1 Genetic Algorithm

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics and considered as a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology[62].

Genetic algorithm search the solution space of a function through the use of simulated evolution, i.e., the survival of the fittest strategy. In general, the fittest individuals of any population tend to reproduce and survive to the next generation, thus improving successive generations. Genetic algorithms have been shown to be effective in solving linear or nonlinear problems by exploring all regions of the state space and exploiting promising areas through mutation, crossover, and selection operations applied to individuals in the population [62]. A genetic algorithm (GA) is summarized as follows:

(1) Supply a solution P_0 of N individuals and respective function values.

(2) $i \leftarrow 1$.

(3) $P_i' \leftarrow$ selection function (P_{i-1}).

(4) $P_i \leftarrow$ crossover and mutation functions P_i'.

(5) evaluate (P_i).

(6) $i \leftarrow i+1$.

(7) Repeat step 3 until termination.

(8) Pick up the best solution found.

The use of Genetic Algorithm requires the determination of six fundamental issues: chromosome representation, selection function, the genetic operators making up the reproduction function, the creation of the initial population, termination criteria, and the evaluation function.

Chromosome Representation

For any GA, a chromosome representation is needed to describe each individual in the population of interest. The representation determines how the problem is structured in the GA and also determines the genetic operators that are used. Each individual or chromosome is made up of a sequence of genes from a certain alphabet. An alphabet could consist of binary digits (0 and 1), floating point numbers, integers, symbols (i.e. A, B, C, D), matrices, etc. One useful representation of an individual or chromosome for function optimization involves genes or variables from an alphabet of floating point numbers with values within the variables upper and lower bounds. Michalewicz [62] has done extensive experimentation comparing real-valued and binary GAs. The results show that the real-valued GA is an order of magnitude more efficient in terms of CPU time. Also, a real-valued representation moves the problem closer to the problem representation which offers higher precision with more consistent results across replications.

Selection Function

The selection of individuals to produce successive generations plays an extremely important role in a genetic algorithm. A probabilistic selection is performed based upon the individual's fitness such that the better individual in the population can be selected more than once with all individuals in the population having a chance of being selected to reproduce into the next generation [62]. There are also several other schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, elitist models, and ranking methods [65] [66]. In the work presented in this thesis, the probabilistic selection method is used for the selection function.

Genetic Operators

Genetic Operators provide the basic search mechanism of the GA. The operators are used to create new solutions based on existing solutions in the population. There are two basic types of operators: crossover and mutation. Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution. The application of these two basic types of operation and their derivatives depends on the chromosome representation that is used.

Let $\bar{X}$ and $\bar{Y}$ be two m-dimensional row vectors denoting individuals (parents) from the population. For real-valued representation, the following operators have been used: uniform mutation, non-uniform mutation, multi-non-uniform mutation, boundary mutation, simple crossover, arithmetic crossover, and heuristic crossover in this thesis.

Uniform mutation randomly selects one variable j and sets it equal to an uniform random number $U(a_i, b_i)$:

$$x'_i = \begin{cases} U(a_i, b_i), & if\ i = j \\ x_i, & otherwise \end{cases} \tag{3.28}$$

Boundary mutation randomly selects one variable j and sets it equal to either its lower or upper bound, where $r = U(0,1)$:

$$x'_i = \begin{cases} a_i, & if\ i = j, r < 0.5 \\ a_i, & if\ i = j, r \geq 0.5 \\ x_i, & otherwise \end{cases} \tag{3.29}$$

Non-uniform mutation randomly selects one variable j and sets it equal to an non-uniform random number:

$$x_i' = \begin{cases} x_i + (b_i - x_i) f(G), & if\ r_1 < 0.5 \\ x_i - (x_i + a_i) f(G), & if\ r_2 \geq 0.5 \\ x_i, & otherwise \end{cases} \tag{3.30}$$

where $f(G) = (r_2(1 - \frac{G}{G_{max}}))^b$; r_1, r_2: a uniform random number between $(0,1)$; G: the current generation; G_{max}: the maximum number of generations; b: a shape parameter.

The multi-non-uniform mutation operator just applies the non-uniform operator to all of the variables in the parent $\overline{X}$.

Simple crossover generates a random number r from a uniform distribution from 1 to m and creates two new individuals ($\overline{X}'$ and $\overline{Y}'$)[62]:

$$x_i' = \begin{cases} x_i, & if\ i < r \\ y_i, & otherwise \end{cases} \tag{3.31}$$

$$y_i' = \begin{cases} y_i, & if\ i < r \\ x_i, & otherwise \end{cases} \tag{3.32}$$

Arithmetic crossover [62] produces two complimentary linear combinations of the parents, where $r = U(0,1)$:

$$\overline{X}' = r\overline{X} + (1-r)\overline{Y} \tag{3.33}$$

$$\overline{Y}' = (1-r)\overline{X} + r\overline{Y} \tag{3.34}$$

Heuristic crossover [62] produces an linear extrapolation of the two individuals. This is the only operator that utilizes the fitness information. A new individual $\overline{X}'$ is created using Equation (3.36), where $r = U(0,1)$ and $\overline{X}$ is better than $\overline{Y}$ in terms of fitness. If

$\bar{X}'$ is infeasible, i.e., feasibility equals zero as given by Equation (3.38), then generate a new random number r and create a new solution using Equation (3.36), otherwise stop. To ensure halting, after t failures, let the children equal the parents and stop.

$$\bar{X}' = \bar{X} + r(\bar{X} - \bar{Y}) \tag{3.35}$$

$$\bar{Y}' = \bar{X} \tag{3.36}$$

$$feasibility = \begin{cases} 1, & if\ x_i' \geq a_i, x_i' \leq b_i, \forall i \\ 0, & otherwise \end{cases} \tag{3.37}$$

Initialization, Termination, and Fitness Function

The GA must be provided an initial population as indicated in step (1) mentioned above. The most common method is to randomly generate solutions for the entire population. However, since GAs can iteratively improve existing solutions, the beginning population can be seeded with potentially good solutions, with the remainder of the population being randomly generated solutions.

The GA moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specific maximum number of generations. The maximum number of generations can be tuned through a few trials. Usually the convergence rate appears to be quite stable is the setting of GA is not dramatically changed. The algorithm can also be terminated due to a lack of improvement in the best solution over a specified number of generations. Alternatively, a target value for the evaluation measure can be established based on some arbitrarily ”acceptable” threshold. The termination condition for the work in this thesis is simply based on a specific maximum number of generations.

Fitness functions of many forms can be used in a GA, subject to the minimal requirement that the function can map the population into a partially ordered set. As stated, the evaluation function is independent of the GA.

3.3.2 Reinforcement Learning

This subsection describes a learning paradigm known as reinforcement learning [63] [64]. Reinforcement learning is a class of learning algorithms in which an agent learns to achieve a goal through trial-and-error interactions with the environment. The learning agent learns only from reward information, and it is not told how to achieve the task. From failure experience, the learning agent reinforces its knowledge so that success can be attained in future trials.

Most reinforcement learning analyses assume that the learning problems can be posed as Markov decision processes (MDPs) with finite states and finite action sets. MDP is a discrete-time dynamic system in which the state transition depends only on the present state i and the action u taken. That is, if an action $u \in U(i)$ is chosen in state i, the system will go to the next state j with probability $P_r(j|i,u)$. For each transition, say from i to j due to the action u, an immediate reward $r(i,u,j)$ is received.

A MDP problem can be classified into either a finite horizon problem or an infinite horizon problem [68]. In the former case, there is a finite number of sets, whereas there is an infinite number of sets in the latter case. In this thesis, the bipedal walking task is posed as a discounted problem. This is a class of the infinite horizon problem which aims to maximize the discounted return R_t associated to t the stage:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \tag{3.38}$$

where r is the reward received, $\gamma \in [0,1)$ is called the discount factor which indicates how much emphasis is put on the future reward, and the subscript denotes the stage number.

The action u is selected based on a policy π which is a function of the present state i. The aim of reinforcement learning techniques is to learn a policy that can achieve a given goal. When applied to control problems, the agent indirectly learns a policy by estimating a value function called Q-factors that is a function of the state and the action

[63]. For MDPs, the Q-factor of a state-action pair under policy $\pi(Q^{\pi}(i,u))$ represents the expected return R_t when the action u is taken at the state i, thereafter, following the policy π. For a discounted problem, $Q^{\pi}(i,u)$ at t the stage can be expressed as follows:

$$Q^{\pi}(i,u) = E^{\pi}[R_t | i_t = i, u_t = u] = E^{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+1+k} | i_t = i, u_t = u] \tag{3.39}$$

where $\gamma \in [0,1)$ is a discount factor, r_t denotes the immediate reward received after the $(t-1)th$ stage.

In most problems, the aim is to obtain optimal Q-factors $Q^{*}(i,u)$ which are defined as follows:

$$Q * (i,u) = maxQ^{\pi}(i,u), \forall_i \in U(i) \tag{3.40}$$

$Q^{*}(i,u)$gives the expected return when the agent takes the action u in the state i and adopts an optimal policy π^{*} thereafter. Based on $Q^{*}(i,u)$, an optimal policy π^{*} can easily be derived by simply taking any action u that maximizes $Q^{*}(i,u)$ over $U(i)$.

The major issue in reinforcement learning algorithm is how to efficiently find the optimal Q-factors [69]. In many practical problems, the environment models is usually unknown in advance. In the following subsection, a reinforcement learning algorithm called Q-learning which does not need the environment model is presented. In this thesis, it is used in all the reinforcement learning problems. Q-learning algorithm is a model free approach in that it does not try to learn the environment model. It simply tries to iteratively improve the estimation of the optimal Q-factors from the immediate reward received for each action taken. And based on these estimated optimal Q-factors, an optimal policy can be gradually derived.

There are several algorithms have been proposed for the model-free reinforcement learning (RL) approach. One popular algorithm that is applicable to such an environment is the Q-learning algorithm developed by Watkins [55]. The following subsection briefly introduced the Q-learning algorithm.

Q-learning

Let's first assume that the environment is a fully observable Markov decision process that has finite state and action sets. Also, the environment is assumed to be stationary; that is, the state transition probabilities are not correlated with the stage number. The Q-learning algorithm recursively estimates the optimal Q-factors $Q^*(i,u)$ from experience obtained at every state. The experiences are in the form of immediate reward sequence, $r(i_t,u_t,i_{t+1})$ $(t = 0,1,2,...)$. The aim is to obtain the optimal Q-factors for all state-action pairs based on which an optimal policy can be derived.

The Q-learning algorithm is analogous to the value iteration method of dynamic programming. However, it does not ”sweep” through all the subsequent possible states when updating the estimate of the optimal Q-factor $Q(i,u)$ of present state-action pair. For the discounted problem, the single-step sample update equation for $Q(i,u)$ is given as follows:

$$\begin{aligned} Q_{t+1}(i_t,u_t) \longleftarrow\ & Q_t(i_t,u_t) + \alpha_t(i_t,u_t)[r(i_t,u_t,i_{t+1}) \\ & + \gamma max_{u\in U(i_{t+1})} Q_t(i_{t+1},u) - Q_t(i_t,u_t)] \end{aligned} \tag{3.41}$$

where the subscripts indicate the stage number; $r(i_t,u_t,i_{t+1})$ denotes the immediate reward received due to the action u_t taken which causes the transition from state i_t to i_{t+1}; $\alpha \in [0,1)$ denotes the step-size parameter for the update; $\gamma \in [0,1)$ denotes the discount rate. Equation (3.40) updates $Q(i_t,u_t)$ based on the immediate reward $r(i_t,u_t,i_{t+1})$ and the maximum value of $Q(i_{t+1},u)$ over all $u \in U(i_{t+1})$. This form of update is also called the bootstrapping [63].

For finite state and action problem, $Q(i,u)$ can be represented by a lookup table. The table is initialized with some values (e.g., zeros) before the beginning of the learning process. The Q-learning algorithm is summarized as in Figure 3.5. The convergence of Q-learning has been proved by several research works [70] [71] and defined as [55]:

Theorem: For a stationary Markov decision process with finite action and state spaces, and bounded rewards $r(i_t, u_t, i_{t+1})$. If the update Equation (3.42) is used and $\alpha_t \in [0,1)$ satisfies the following criteria: $\Sigma_{t=1}^{\infty} \alpha_t(i,u) = \infty$ and $\Sigma_{t=1}^{\infty} [\alpha_t(i,u)]^2 = \infty$, $\forall (i,u)$; then $Q_t(i,u) \longrightarrow Q^*(i,u)$ as $t \longrightarrow \infty$ with probability 1, $\forall (i,u)$.

However, the above theoretical convergence criteria is hard to be accomplished in practical implementation. Especially for large state and action spaces, the convergence rate for the Q-factors may be very slow because each state-action pair needs to be visited infinitely often in order to satisfy the convergence conditions. Nevertheless, in practice Q-learning works fine if the goal of the learning problem can be achieved without having a precise estimate of the optimal Q-factor for each state-action pair.

In addition to the exploitation process presented above, reinforcement learning also focus on on-line performance, which involves finding a balance between exploitation (of current knowledge) and exploration (of uncharted territory). One method that allows a structured exploration is called the ε-greedy method [63]. However, it may not be feasible for control applications in which most states have failure actions to explore randomly over the action set. For the bipedal walking example, the biped may have to fall frequently if the exploration mode is not turned off or somehow limited. Therefore, the exploration method used in the work presented here is called passive exploration [63] which means initializing Q-factors for all state-action pairs to optimistic value. At any given state, the learning agent selects an action that has the maximum Q-factor. If a failure is encountered or a penalty is given, the learning agent then downgrades the particular Q-factor of the state-action pair and selects other actions.

Furthermore, similar to neural networks, reinforcement learning also involves the credit assignment during learning. In the Q-learning algorithm, the discount factor γ determines the credit assignment structure. When γ is zero, only immediate reward is considered. Such credit structure is considered myopic. That is any rewards generated by subsequent states will not have any influence on the return computed at the present state. Usually, if a reinforcement learning problem can be posed such that $\gamma = 0$ works fine, it is much simpler and can be solved more quickly. When γ is large (close to one), the future rewards have significant contribution to the return computation for the present action. The agent is then said to be farsighted since it looks far ahead of time when evaluating a state-action pair.

Normally, reinforcement learning works with a function approximator for representing Q-factors $Q(i,u)$. The purposes of using a function approximator are as follows [63][72]:

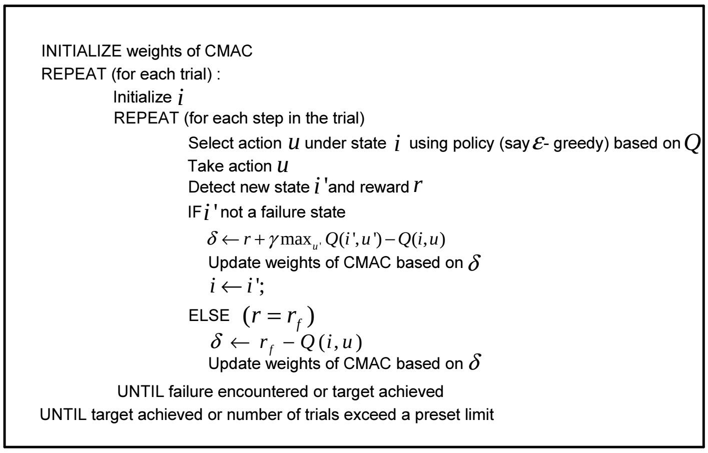
1) To reduce memory requirement; this is especially critical for high-dimensional and continuous state-action space.

2) To enable generalization; generalization enables estimation of the Q-factors in those locations which are not explored.

Gaussian Radial Basis Function (GRBF) networks and Cerebellar Model Articulation Controller (CMAC) are common function approximators used in reinforcement learning algorithms. Both have local generalization property. Compared with CMAC, GRBF Network is more computationally costly. Therefore, the CMAC function approximator is incorporated for all the reinforcement learning problems derived in this thesis. The Q-learning algorithm using CMAC as the Q-factor function approximator is summarized as in Figure 3.5, and the following subsection describes the working mechanism of the CMAC.

Cerebellar Model Articulation Controller

Cerebellar Model Articulation Control (CMAC) is a mathematical model proposed by Albus [56] for efficient computation in manipulator control. It is based on the neurophysiological theory of the cerebellum. It has been adopted as a type of neural networks for supervised learning.



```
INITIALIZE weights of CMAC
REPEAT (for each trial) :
        Initialize $i$
        REPEAT (for each step in the trial)
                Select action $u$ under state $i$ using policy (say $\varepsilon$- greedy) based on $Q$
                Take action $u$
                Detect new state $i'$ and reward $r$
                IF $i'$ not a failure state
                   $\delta \leftarrow r + \gamma \max_{u'} Q(i',u') - Q(i,u)$
                   Update weights of CMAC based on $\delta$
                   $i \leftarrow i'$;
                ELSE $(r = r_f)$
                   $\delta \leftarrow r_f - Q(i,u)$
                   Update weights of CMAC based on $\delta$
        UNTIL failure encountered or target achieved
UNTIL target achieved or number of trials exceed a preset limit
```

Figure 3.5: Q-learning algorithm using CMAC to represent Q-factors.

The Albus's CMAC implementation mainly consists of an addressing scheme followed by a single-layered network. The addressing scheme maps the input vector to a proper subset with C elements of M weights of the single-layered network. By summing up the weights addressed, an output is calculated. The addressing scheme is usually static throughout the training process. A supervised training algorithm is applied to adjust the weights of the single-layered network.

The addressing scheme consists of C layers of partitioning of the input space. Each segment of a partitioning layer represents a receptive field. The partitioning layers are offset from one another. For a given input state, only one receptive field is activated for each partitioning layer. The total number of the receptive fields is thus equal to the total number C of the partitioning layers.

The shape of the receptive field adopted in this thesis is rectangular type, which is the same as the original Albus's CMAC implementation. This can be illustrated by considering a three-dimensional and continuous input space (x_1, x_2, x_3). Partition the state space uniformly to create grid-like receptive fields. Each of the receptive fields is linked to an index. This is like a lookup table if there is only one partitioning layer. If other layers (layer two and layer three) of receptive fields is overlapped and offset from the base layer as shown in Figure 3.6, a state in the input space will activate the corresponding

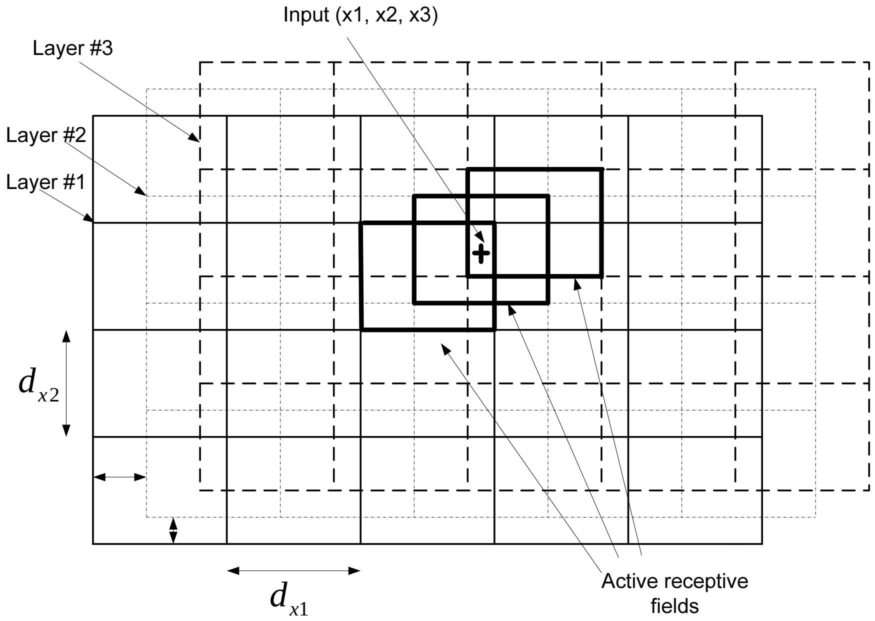


Figure 3.6: An addressing scheme for a three-dimensional input CMAC implementation.

three receptive fields.

The widths (d_{x1} and d_{x2}) of the rectangular receptive fields are selected to match the desired generalization widths. The wider the receptive field, the greater is the generalization width. The number of partitioning layers corresponds to the desired accuracy in approximating a function. More layers means finer discretization. However, having more layers also results in greater computational cost.

Once the indices from the addressing scheme are generated, the output y of CMAC can be obtained by summing C weights ($\omega^i : i = 1,2,3...C$) linked to the indices:

$$y = \Sigma_{i=1}^{C} \omega^i \tag{3.42}$$

If the desired output y_d is given, these weights can be adjusted as follows, which is similar to the LMS (Widrow-Hoff) algorithm [73]:

$$\delta\omega^i = \alpha \frac{y_d - y}{C}, \; i = 1,2,3...C \tag{3.43}$$

where $\delta\omega^i$ is the adjustment to ω^i and α is the training rate. That is, the adjustment is the same for all the weights. If α is large, the adjustment will also be large. Usually α is small as large α may lead to instability in the overall Q-learning algorithm that uses CMAC to approximate the Q-factors.

3.4 Summary

In this chapter, a general control architecture is synthesized for dynamic bipedal walking based on the divide-and conquer framework discussed earlier. The dynamic walking task is partitioned into the three orthogonal planes: frontal, sagittal and transverse. In the sagittal and frontal motion planes, the motion control is further split into two levels: a low-level motion controller and a high-level motion monitor. The low-level motion controller is designed to achieve the walking stability and the high-level motion monitor is aimed to execute certain motion adjustment strategies, which are difficult to be derived by an analytical solution, in real-time to prevent motion from falling. Besides, the core motion generation method GAOFSF is introduced as well as all the necessary computation tools used for algorithm implementations.

The following chapters present the algorithms for low-level, high-level, sagittal plane and frontal plane motion control strategies for the achievement of the dynamically stable and smooth bipedal walking motions.

Chapter 4

Sagittal Plane Walking Algorithm

This chapter presents the algorithm for sagittal plane walking control. The major objective of this algorithm is based on a soft motion controller to achieve motion convergence towards a stable limit cycle walking behavior which is physically achievable and robust. With the achievement of such a limit cycle behavior under the soft motion control, the robot will then be able to smoothly damp the perturbations from environment to some extent, and self-converge the motion to the steady-state walking pattern.

4.1 Motion Control Strategy

In this section, the motion control law that is used in the sagittal plane motion control is discussed. For bipedal walking control, the challenges mainly come from the following two aspects:

(1) The root coordinate that describes the robot motion is attached with the body link. However, the robot body usually is not actively actuated. Therefore, if the conventional PD position control is applied to the two legs, it is difficult to maintain the robot body to be just upright stably. Besides, the pitching body motion will also cause the swing foot not to land much as planned. Then the motion will be more perturbed by the ground

impact effect.

(2) Stable motion with the ZMP always inside the supporting polygon must involve a larger acceleration component especially for faster but also robust walking motions, referring to Equation (4.1) for constraining the ZMP location to be close to the center of the supporting foot-print [74]-[79]. However, joint controllers may only follow such a motion better by applying stiff joint control gains, but for bipedal locomotion control, the ground perturbation always exists and disturbs the motion. Given a stiff motion controller, the limited actuated stance foot may quickly flip due to the environment disturbances and then cause the instability during transitions.

$$ZMP_x = \frac{mgx_G - \dot{H}_{oy}}{m(g+\ddot{z}_G)} = \frac{\sum_{i=1}^{n}(m_i x_{G_i} g - (\dot{H}_{G_i y} + m_i(z_{G_i}\ddot{x}_{G_i} - x_{G_i}\ddot{z}_{G_i})))}{mg + \sum_i^n m_i \ddot{z}_{Gi}} \tag{4.1}$$

where x_{G_i}, y_{G_i}, and z_{G_i} are the coordinates of the centroid of link i; m_i the mass of link i; and $\dot{H}_{G_i}$, the rate of angular momentum at point G_i which is computed as:

$$\dot{H}_{G_i} = R_i(I_{G_i}\dot{\omega}_i - (I_{G_i}\omega_i) \times \omega_i) \tag{4.2}$$

where R_i is the rotation matrix associated with the ith solid, I_{G_i} its inertia matrix, ω_i its rate of rotation, and $\dot{\omega}_i$ its angular acceleration.

To solve the first issue discussed above, the joint control strategies for the stance and swing legs have been separately considered.

The motion control for the stance leg is formulated based on the reaction torques, as formulated in Equation (4.2). The coordinates are defined in Figure 4.1.

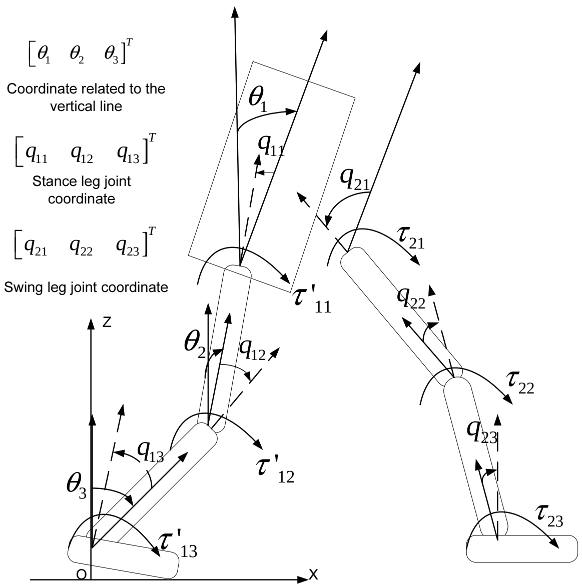

$[\theta_1 \ \theta_2 \ \theta_3]^T$
Coordinate related to the vertical line
$[q_{11} \ q_{12} \ q_{13}]^T$
Stance leg joint coordinate
$[q_{21} \ q_{22} \ q_{23}]^T$
Swing leg joint coordinate
z
x
O
θ_1
q_{11}
q_{21}
τ_{21}
τ'_{11}
q_{22}
θ_2
q_{12}
τ_{22}
q_{13}
θ_3
τ'_{12}
q_{23}
τ'_{13}
τ_{23}

Figure 4.1: Joint coordinate describing the robot motion.

$$\begin{cases} \tau'_{11} = K_{p1}(\hat{\theta}_1 - \theta_1) + K_{d1}(\dot{\hat{\theta}}_1 - \dot{\theta}_1) - \tau'_{21} \\ \tau'_{12} = K_{p2}(\hat{\theta}_2 - \theta_2) + K_{d2}(\dot{\hat{\theta}}_2 - \dot{\theta}_2) \\ \quad\;\; = K_{p2}(\theta_1 + \hat{q}_{11} - (\theta_1 + q_{11})) + K_{d2}(\dot{\theta}_1 + \dot{\hat{q}}_{11} - (\dot{\theta}_1 + \dot{q}_{11})) \\ \quad\;\; = K_{p2}(\hat{q}_{11} - q_{11}) + K_{d2}(\dot{\hat{q}}_{11} - \dot{q}_{11}) \\ \tau'_{13} = K_{p3}(\hat{\theta}_3 - \theta_3) + K_{d3}(\dot{\hat{\theta}}_3 - \dot{\theta}_3) \\ \quad\;\; = K_{p3}(\theta_2 + \hat{q}_{12} - (\theta_2 + q_{12})) + K_{d2}(\dot{\theta}_2 + \dot{\hat{q}}_{12} - (\dot{\theta}_2 + \dot{q}_{12})) \\ \quad\;\; = K_{p3}(\hat{q}_{12} - q_{12}) + K_{d3}(\dot{\hat{q}}_{12} - \dot{q}_{12}) \end{cases} \tag{4.3}$$

The definition of all the symbols are indicated in Figure 4.1. It can be seen that the motion control for the stance leg is based on the world coordinate $\theta_1, \theta_2, \theta_3$. This is because the robot walking posture, including the upright body, is to be maintained with respect to the world coordinate. Besides, using such a joint control, body pitch motion can be directly controlled and the motion planning for the stance ankle joint can be omitted as the stance ankle joint will be allowed to naturally interact with the ground. Furthermore, it should be also noted that the posture is maintained sequentially, because the set-point is given as $\hat{\theta}_{i+1} = \theta_i + \hat{q}_{1i}$ instead of $\hat{\theta}_{i+1} = \hat{\theta}_i + \hat{q}_{1i}$. Therefore, given the

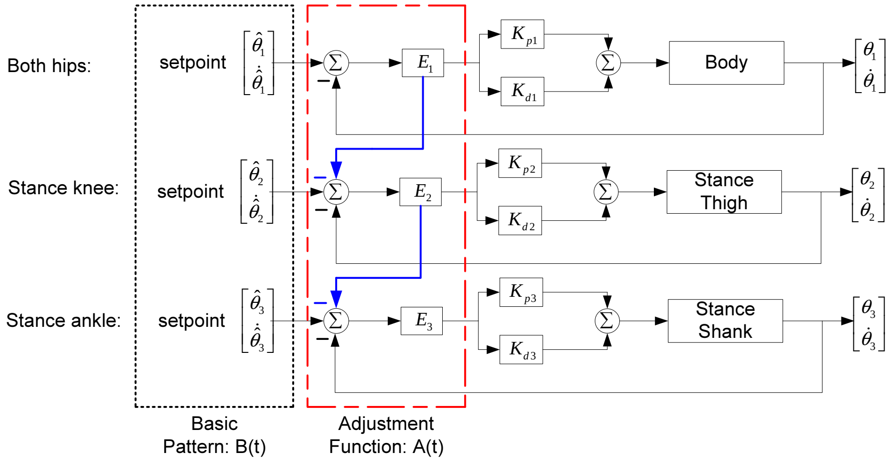


Figure 4.2: Control Block-diagram for stance leg control.

body can be controlled to be upright, the whole walking posture can be still achieved but the stance motion control becomes rather analogous to be a single link model actuated at the bottom. Then, the entrainment of a walking limit cycle behavior, if existed, is supposed to be unique.

Compared to tracking a statically planned basic walking pattern, e.g. $\theta_{i+1} = \hat{\theta}_i + \hat{q}_{1i}$, the difference is there will be two feedback pathways included in between the hip-and-knee and the knee-and-ankle, as shown in Figure 4.2.

Here, the motion control for the swing leg is just the conventional PD position control, as formulated in Equation (4.3). The joint control block diagram for the swing leg motion is shown in Figure 4.3.

$$\begin{cases} \tau_{21} = k_{p1}(\hat{q}_{21} - q_{21}) + k_{d1}(\dot{\hat{q}}_{21} - \dot{q}_{21}) \\ \tau_{22} = k_{p2}(\hat{q}_{22} - q_{22}) + k_{d2}(\dot{\hat{q}}_{22} - \dot{q}_{22}) \\ \tau_{23} = k_{p3}(\hat{q}_{23} - q_{23}) + k_{d3}(\dot{\hat{q}}_{23} - \dot{q}_{23}) \end{cases} \tag{4.4}$$

To solve the second issue about motion sensitivity to the environment changes as listed above, instead of tracking a prescribed motion as precisely as possible using very stiff control gains, the motion control strategy proposed in this thesis is developed based on

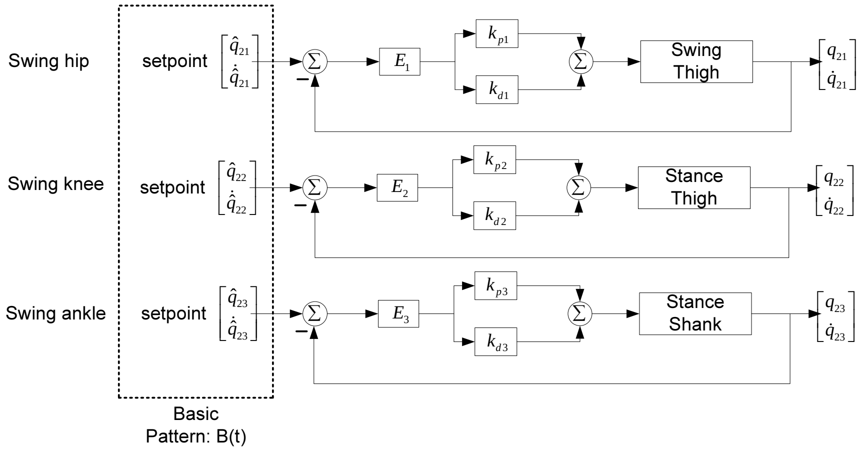

Swing hip
setpoint
$\begin{bmatrix}\hat{q}_{21}\\ \dot{\hat{q}}_{21}\end{bmatrix}$
E_1
k_{p1}
k_{d1}
Swing Thigh
$\begin{bmatrix}q_{21}\\ \dot{q}_{21}\end{bmatrix}$
Swing knee
setpoint
$\begin{bmatrix}\hat{q}_{22}\\ \dot{\hat{q}}_{22}\end{bmatrix}$
E_2
k_{p2}
k_{d2}
Stance Thigh
$\begin{bmatrix}q_{22}\\ \dot{q}_{22}\end{bmatrix}$
Swing ankle
setpoint
$\begin{bmatrix}\hat{q}_{23}\\ \dot{\hat{q}}_{23}\end{bmatrix}$
E_3
k_{p3}
k_{d3}
Stance Shank
$\begin{bmatrix}q_{23}\\ \dot{q}_{23}\end{bmatrix}$
Basic
Pattern: B(t)

Figure 4.3: Control Block-diagram for swing leg control.

the logic: guided by a basic walking pattern $B(t)$ and applying a soft motion controller with low control gains, the actual robot motion can converge to an ”optimum” steady-state walking motion. Here, the low control gains are roughly tuned to only achieve the end body CG magnitude (referring to the center line $X = 0$), which in reverse is also the initial CG magnitude for the subsequent step, given some initial push or velocity that can drive the motion forward. Based on this motion control logic, the objective functions for generating a basic walking pattern that can guide the actual motion to smoothly converge to the steady-state walking become rather important to study first.

Inspired by the characters of human walking gaits, such as the walking phases in the sagittal plane are roughly symmetrical according to the center line $X = 0$ and the walking motion is rather smooth, a uniform basic walking pattern (velocity is planned to be uniform) that has the symmetrical walking phases according to the center line $X = 0$ ($|X_{cg-}| \approx |X_{cg+}|$) is selected to be the first attempt for the study. This walking pattern is defined as the dynamically symmetrical basic walking pattern. Intuitively, when the step-frequency is close to the robot’s natural frequency, it will not be challenging to achieve the desired end-step stance phase using low control gains guided by this dynamically symmetrical basic walking pattern. This is because the pattern does not involve any major system CG accelerations and the angular momentum caused by gravity force is roughly compensated by the symmetrical magnitudes of the robot CG.

Table 4.1: Geometrical and inertial properties of NUSBIP-I.

Link	Mass (*kg*)	Length (*m*)	Moment of inertia (kgm^2)
Trunk	12	0.5	0.97
Thigh	6	0.3	0.04546
Shank	5	0.3	0.038
Foot	1	forward=0.15;backward=0.1	0.03012

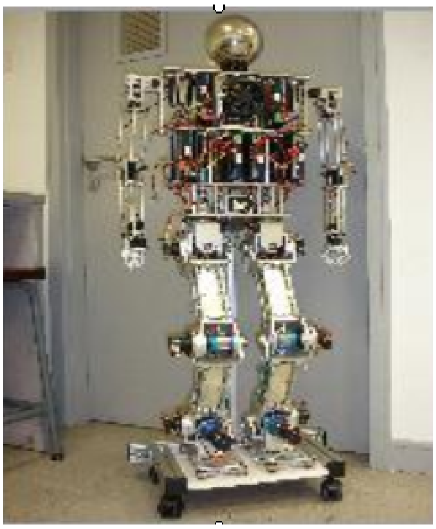


Figure 4.4: Simulated Robot NUSBIP-I.

Note in this thesis, all the generated motion patterns are targeted to be human-like as such patterns are more challenging at the stability issue. The GAOFSF method presented in Chapter 3 is used for generating different desirable patterns according to the user-defined performances. The target robot is modelled according to the mechanical properties of robot NUSBIP-I, as shown in Figure 4.4. This robot was developed by the Control and Mechatronics lab, National University of Singapore. The mechanical properties measured for the sagittal plane are shown in Table. 4.1.

4.2 Walking Guided by Dynamically Symmetrical Basic Walking Pattern

In this section, the defined symmetrical basic walking pattern, $B_{sym}(t)$ is first generated to test the proposed control strategy. Walking specifications for this dynamically symmetrical walking pattern B_{sym} are given as: step-length $0.33m$ and the average walking speed $0.45m/s$ on the flat-terrain. The walking speed is set based on the estimation of the natural frequency of the modelled robot. Here, the natural frequency for the NUSBIP-I robot is obtained roughly at $1.2987Hz$. Therefore, the step-period will be at $0.77s$. Given the step-length is $0.33m$, the walking speed is thus specified at $0.45m/s$. (The estimation of the natural frequency is conducted in the simulation environment by making all the joints rigid and then push the robot to fall. The time period that the robot CG bypasses the desired CG magnitude for a step motion is estimated as the natural step-period which will lead to the estimation of the natural frequency of this particular robot). Note, even if this step-length is not achievable in the actual motion control, the step-length can be always reduced by tuning the scaling parameter R in Equation (3.1) and (3.2).

To achieve a natural human-like gait but not taking the small stance knee angle change as significant, the knee joint angle offset value in the TFS formulation was set at $c_k = 0$. Table 4.2 gives the set-up of the GAOFSF including the objective functions, constraint functions, the necessary parameters for the GA initialization and the weights for the objective and penalty functions referring to Chapter 3.

After the GA evolutionary computation, the fitness function for this solution was found to be 1309 as shown in Figure 4.5, indicating a good optimization performance, as C_{max} value is just set at 1400. The chromosome solutions, x_{sym}, obtained for the flat-terrain symmetrical walking is:

Table 4.2: GAOFSF Set-up for symmetrical motion pattern generation (flat-terrain)

Description	Remark
Objective components	Code Op_1: f_1, f_2, f_3
Constraint components	Code Op_1: $s_1 - s_5$
Chromosome representation	real-valued GA
Initial population M	150
Generation number T	250
Crossover operators	heuristic crossover simple crossover arthritic crossover
Mutation operators	multi-non-uniform mutation Uniform mutation Boundary mutation
Weights for objectives	$w_i = [10, 10, 10]$
Weights for constraints	$p_i = [60, 20, 20, 20, 70]$
Chromosome format	$[A_i, B_i, C_i, c_h, c_k, t_1, t_2]$

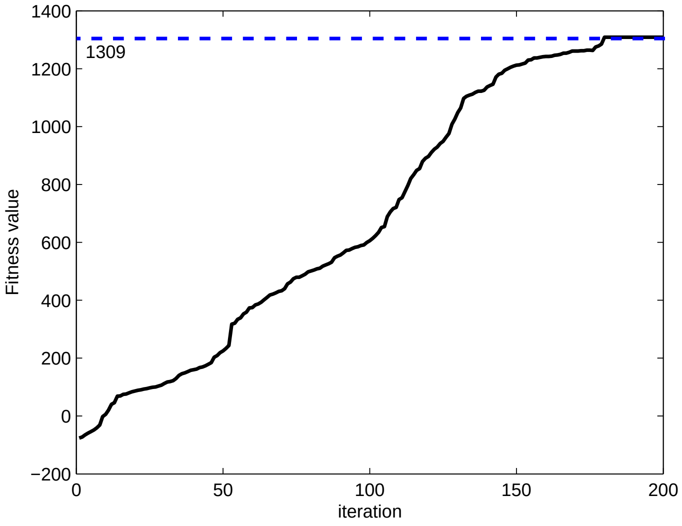


Figure 4.5: GA fitness profile of the symmetrical walking pattern generation.

$$x_{sym} = [0.277\ -0.087\ 0.022\ -0.008\ -0.000\ -0.397\ -0.118\ -0.024\ -0.017\ -0.006\ 0.457\ 0.200\ -0.038\ -0.077\ -0.046\ -0.036\ 0.000\ 0.050\ 0.440] \tag{4.5}$$

with the stride-frequency assigned to be $\omega_h = 4.36 rad/s$ based on the estimation of robot natural frequency.

From the solution x_{sym} obtained, the corresponding hip and knee joint angle trajectories, generated using the TFS, are shown in Figure 4.6(a). Figure 4.6(b) shows the trajectory of the ZMP over one walking cycle. The periodical two steps in the generated walking cycle shows the constraint s_5 for the touchdown moment $t_d \doteq t_2$ is satisfied by the GA. Then, it can be noted that the ZMP is outside the supporting polygon $[-0.1, 0.15]m$ indicated by the horizontal lines but the shape of the ZMP trajectory is rather symmetrical for the positive and negative walking phases referring to the center line $X = 0$. The stance ankle joint torque τ_{13} usually has a similar trajectory shape compared to the ZMP trajectory as the vertical acceleration of walking is very minor. As shown in Figure 4.6(c), the stance ankle joint torque τ_{13} has been optimized to be also symmetrical according to the center line. Referring to Figure 4.6(d), uniform walking velocity of $0.45m/s$ has been achieved well via the estimation of the inclination of the position-time plot. Figure 4.6(e) shows the landing velocity of the swing foot. The touchdown strike velocity $[v_x, v_z]$ is found to be around [0.15, -0.2] m/s and the swing foot velocity profile (Figure 4.6(e)) shows the velocity has been constrained before landing. Although the touchdown impact is not particularly optimized, the touchdown impact can be naturally constrained when motion is optimized to be rather symmetrical. The resulting ground impact is found to be quite small from the subsequent dynamic simulations to be discussed in a later section. Figure 4.6(f) shows the stick diagram of the final generated motion pattern which has a rather uniform stick density. The step length is found to be equal to $0.328m$, very close to the target set.

The above results show that the basic walking pattern $B_{sym}(t)$ generated by the GAOFSF method satisfies the walking specifications and the motion requirement as being dynamically symmetrical, although the ZMP is outside the supporting polygon. Since the ZMP is no longer always inside the supporting polygon, conventional PD position control with stiff and critical control gains to track the planned motion precisely will result in quick rotations of the stance foot. Instead, using the proposed motion control strategy, given various initial walking velocities, Figure 4.7 shows the motions, through their forward walking velocities, have all converged to the same limit walking cycle pattern after a certain period of motion transition. Besides, the achieved steady-state walking velocity is also in a rather symmetrical and smooth ”U” shape. The highest walking velocity V_{max} happens at the touchdown moment while the V_{min} occurs in the middle of a step, roughly at the $X = 0$ position. Therefore, in the motion stick-diagram of the achieved stable walking(Figure 4.8), the stick-density is symmetrically distributed with the highest stick-density in the middle but the lowest stick-density at the touchdown moment.

In addition, it can be also observed that during transitions a lower initial velocity resulted

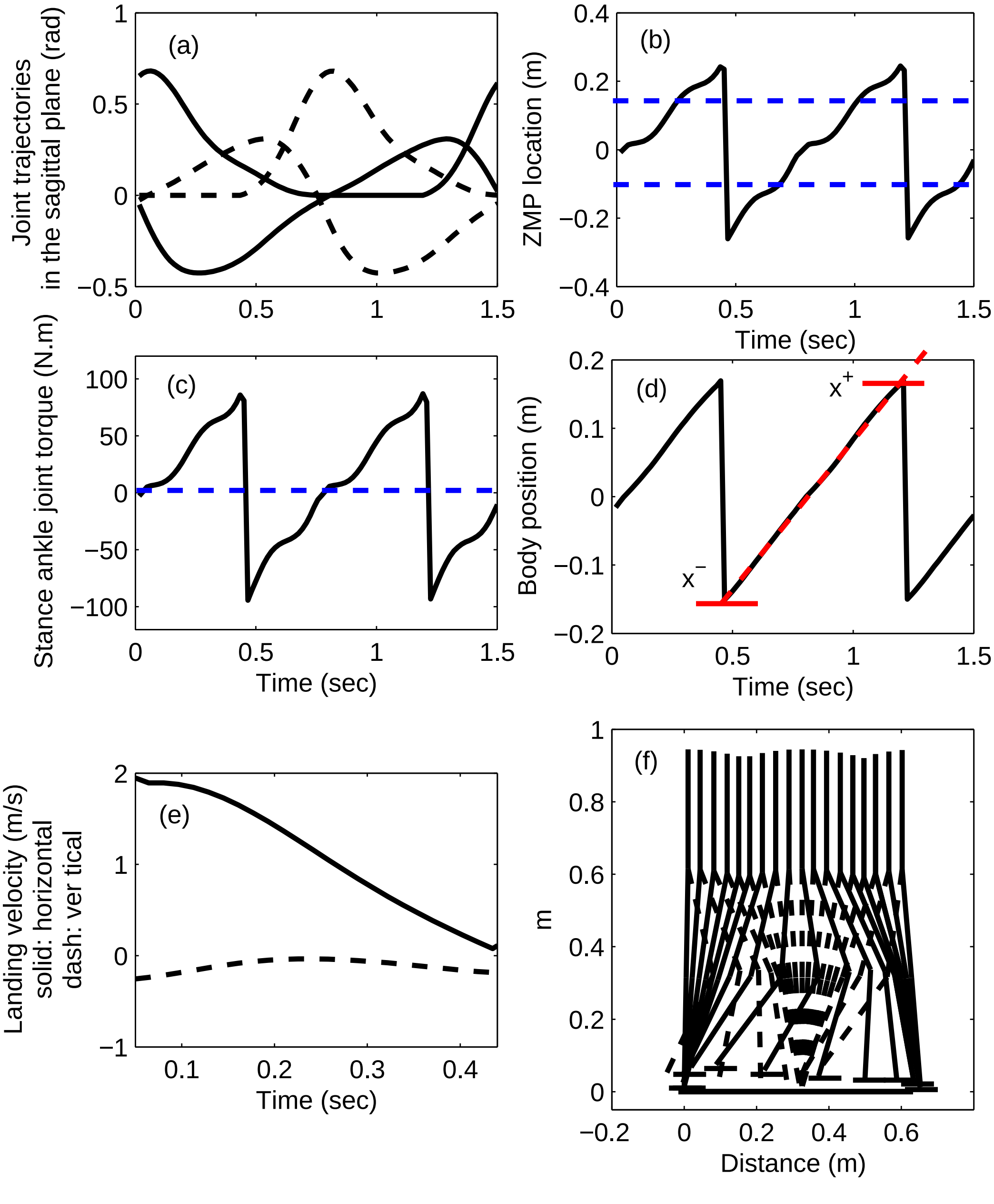


Figure 4.6: Generated joint angle trajectories of the symmetrical walking pattern $B(t)$.

in a higher initial velocity for the subsequent step, but a higher initial velocity instead resulted in a lower initial velocity for the following step, as shown in Figure 4.7(a) and Figure 4.7(c), respectively. Only when the initial velocity is assigned to be similar as the highest velocity appeared in the steady-state, the transition becomes not obvious, as shown in Figure 4.7(b). Note, the difference to the motion steady-state was always being reduced until eliminated. This indicates the motion convergence has been achieved by the proposed motion control strategy. Furthermore, for each step in the steady-state, the initial velocity of a step V_{0i} is the same as the end velocity V_{ti} of a step, yielding a one-step period limit cycle behavior. This means the left stance step is exactly the same as the right stance step.

Figure 4.9 shows the detailed data of the resulting walking dynamics. It can be seen that the actual motion trajectories still resembled the shape of the basic walking pattern $B_{sym}(t)$ but with some position deviation during a step, as shown in Figure 4.9 [d] and [e]. Nevertheless, the deviation of the CG magnitude at the touchdown moment was almost recovered to be zero, as indicated by the dash-dot lines in Figure 4.9 [d] and [e]. This shows the planned step-magnitude can be still maintained well by the applied low control gains. From all the results obtained, the achieved walking is found to be quite desirable in the sense of motion smoothness and stability.

To sum up, the achieved stable walking guided by the symmetrical basic walking pattern $B_{sym}(t)$ has the following characteristics:

- The walking steady-state is shown of one-step period limit cycle behavior which means $V_{0i} \doteq V_{ti} = V_{max}$ with the desired step-magnitude maintained. V_{0i} and V_{ti} are the initial and end velocities of a step in the steady-state motion. V_{max}, the maximum walking velocity which occurs at the touch-down moment in the steady-state.
- V_{min} is in the middle of a step. This indicates the velocity profile is also rather symmetrical in the motion steady-state.
- The highest stick-density is shown in the middle of the step but the lowest stick-

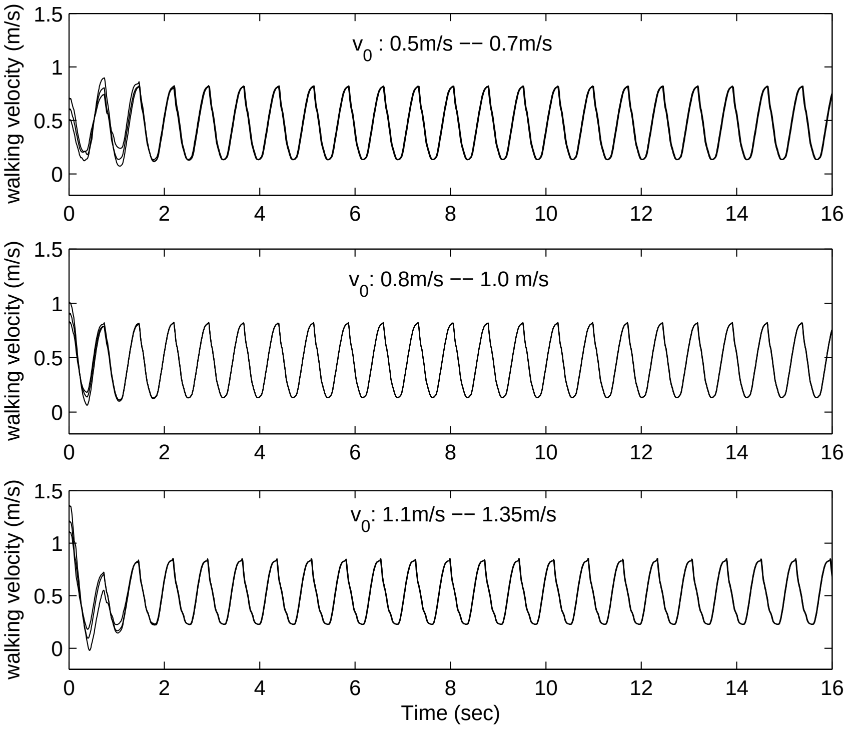


Figure 4.7: Walking velocity of motions started by different initial velocity v_0.

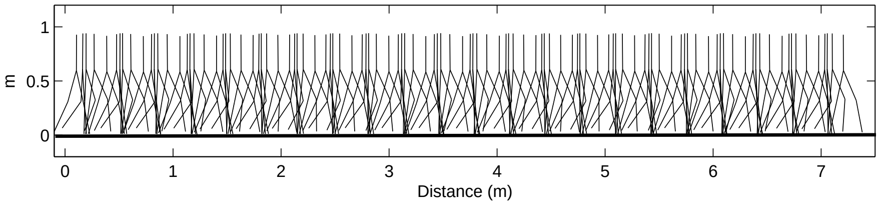


Figure 4.8: Stick-diagram of the dynamic walking controlled by basic walking pattern B_{sym}.

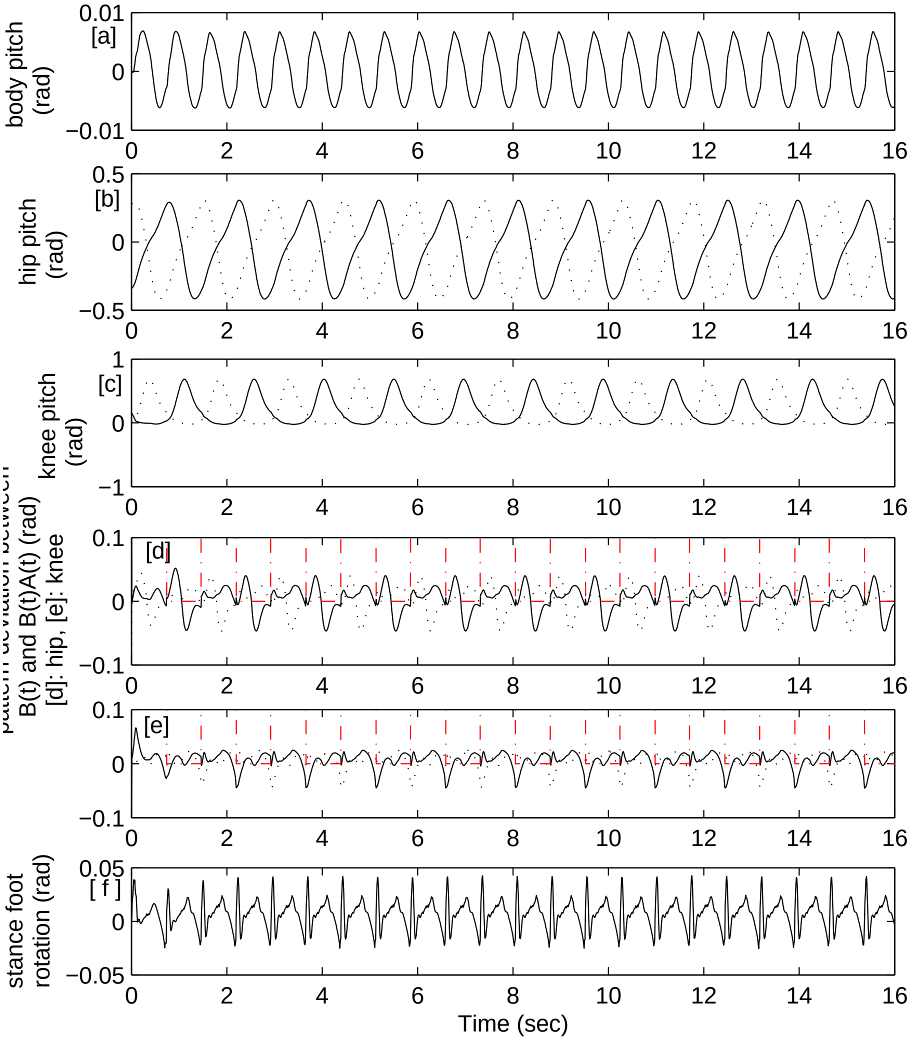


Figure 4.9: Resulting dynamics of the dynamic walking controlled by basic walking pattern B_{sym}.

density occurs at the touchdown moment, showing the actual ZMP location is in a sub area of the planned ZMP location in the basic walking pattern $B(t)$.

- Control convergence behavior is found to be: compared to the V_{max}, a higher initial velocity → a lower initial velocity for the subsequent step; a lower initial velocity → a higher initial velocity for the subsequent step; and the difference to the steady-state motion is quickly reduced to be zero, while the step-magnitude is well maintained in transitions.

Based upon the desirable walking behaviors obtained from the walking control guided by the symmetrical basic walking pattern, in order to further understand the correlation between the basic walking pattern and the resulting steady-state walking, the study of objective functions for the generation of a desirable basic walking pattern is further extended to the dynamically asymmetrical basic walking patterns.

4.3 Walking Guided by Dynamically Asymmetrical Basic Walking Patterns

Similar to the symmetrical motion pattern generation, the asymmetrical basic walking patterns are also generated by the GAOFSF method.

Since the asymmetrical pattern can be referred to position asymmetry or velocity asymmetry, the dynamically asymmetrical motions are further divided into the following two categories:

Category 1: velocity is still maintained to be uniform, which can be considered as symmetrical according to the center line $X = 0$, but the end configuration of a step is less symmetrical.

Category 2: the end configuration of a step is still maintained to be rather symmetrical, but the average walking velocity before and after crossing the center line $X = 0$ is less

Table 4.3: GAOFSF Set-up for asymmetrical motion pattern generation, Category 1 (flat-terrain)

Description	Remark
Objective components	Code Op_1: f_1, f_2, f_3
Constraint components	Code Op_1: $s_1 - s_5$
Equivalent level	$a_{min} = 10, b_{min} = 6$
Chromosome representation	real-valued GA
Initial population M	150
Generation number T	250
Crossover operators	heuristic crossover simple crossover arthritic crossover
Mutation operators	multi-non-uniform mutation Uniform mutation Boundary mutation
Weights for objectives	$w_i = [10, 10, 10]$
Weights for constraints	$p_i = [60, 20, 20, 20, 70]$
Chromosome format	$[A_i, B_i, C_i, c_h, c_k, t_1, t_2]$

equivalent.

4.3.1 Walking Results of Category 1

Walking specifications for this B_{asym} Category 1 are: similar step-length as that of the B_{sym}, uniform walking velocity, but the stance ankle joint torque during the positive and negative phases are not supposed to be in an equivalent level to get some dynamics asymmetry. Therefore, non-zero values of a_{min} and b_{min} are assigned to s_1 to specify the motion asymmetry, referring to Chapter 3. Table 4.3 gives the set-up of the GAOFSF for the generation of asymmetrical motion patterns that belong to the Category 1.

The following two representative walking solutions are then generated:

$$\begin{aligned} x_{asym1} = [&0.262\ -0.062\ 0.013\ -0.005\ 0.000\ -0.388\ -0.047 \\ &0.047\ 0.004\ -0.003\ 0.516\ 0.082\ -0.055\ 0.008 \\ &-0.018\ -0.016\ 0.001\ 0.07\ 0.470] \end{aligned} \tag{4.6}$$

$$\begin{aligned} x_{asym2} = [&0.255\ -0.074\ 0.019\ -0.003\ 0.000\ -0.379\ -0.031 \\ &0.050\ 0.014\ 0.000\ 0.468\ 0.102\ -0.014\ -0.026 \\ &-0.035\ -0.010\ 0.000\ 0.062\ 0.480] \end{aligned} \tag{4.7}$$

Figure 4.10 shows the generation results of the above two solutions x_{asym1}, x_{asym2} in terms of the ZMP trajectory, stance ankle torque trajectory and the body position w.r.t the base frame. It can be observed that the ZMP trajectories of both solutions are outside the supporting polygon, and the needed stance ankle joint torque is in a similar shape of the ZMP trajectories, showing the vertical motion is minor. From the ZMP or the stance ankle joint torque trajectory, it can be seen that the trajectories are a bit shifted up according to the center line compared to the symmetrical walking pattern, as shown in Figure 4.10(a) and (b). Figure 4.10(e) and (f) show that the walking velocity of the two solutions are both optimized to be uniform over the walking cycle, but the end configurations are less symmetrical as $|x^-|$ is not this equal to $|x^+|$, where $|x^-|$ and $|x^+|$ are the initial and end body CG position of a step motion w.r.t the base frame.

For the generated motion pattern x_{asym1}, given different initial velocities v_0 ($0.4m/s$ $-$ $-1.2m/s$), the motion responses are shown in Figure 4.11. It can be seen that the limit

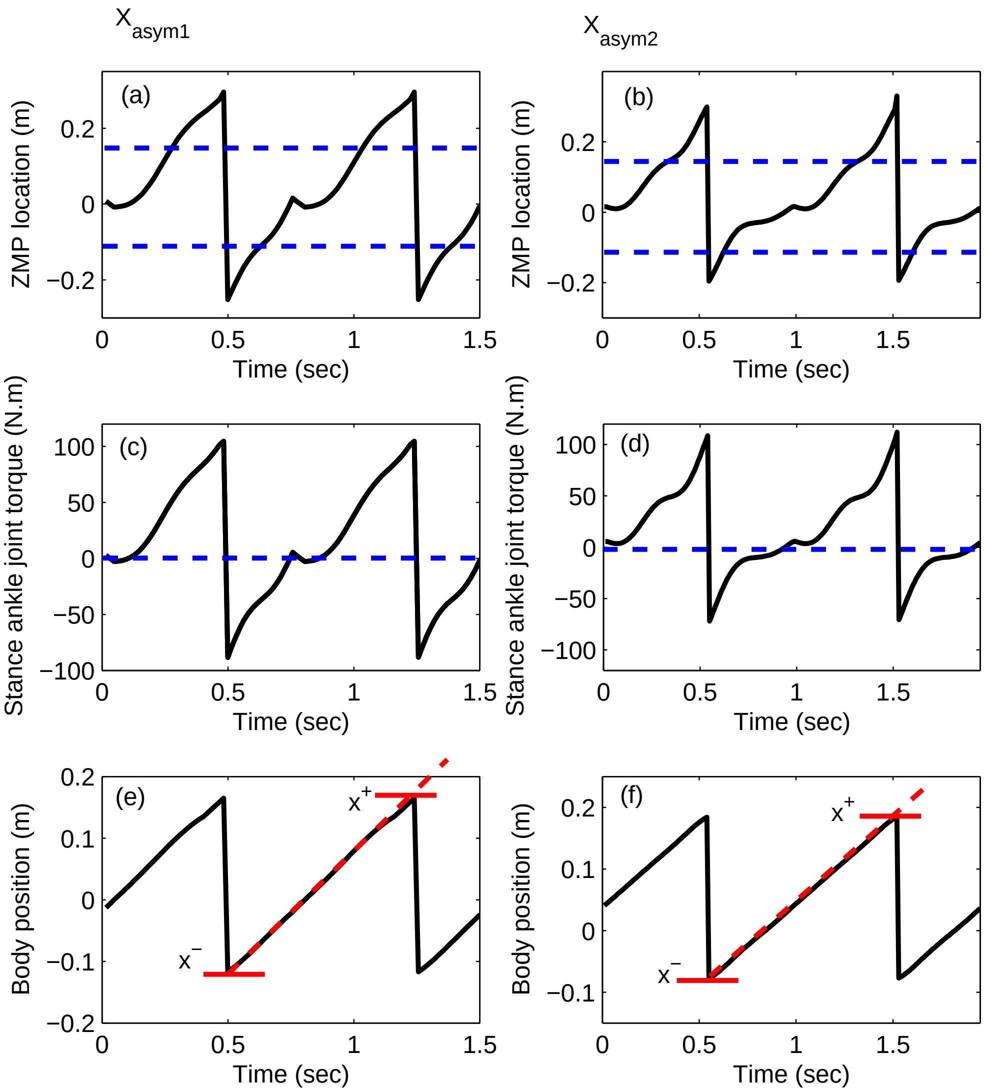


Figure 4.10: Motion generation result of the two solutions belonging to B_{asym} Category 1.

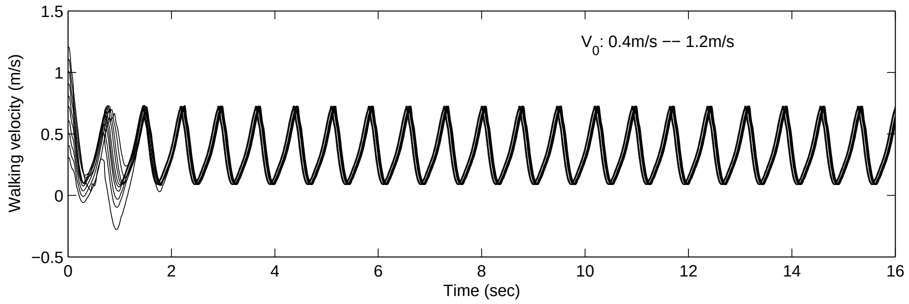


Figure 4.11: Walking velocity profile of motions excited by different initial velocity v_0. (Solution X_{asym1}).

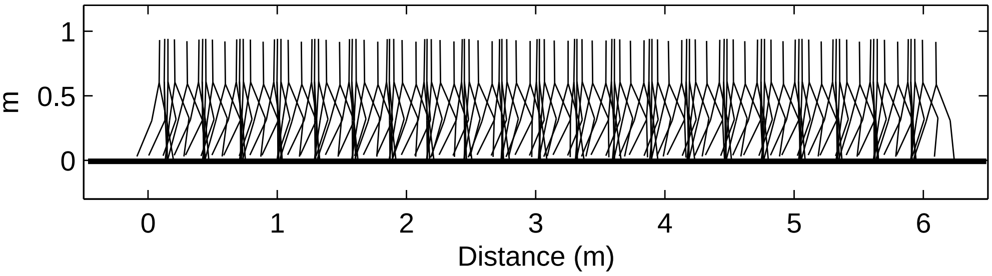


Figure 4.12: Stick-diagram of the dynamic walking controlled by pattern X_{asym1}.

cycle behavior can be still achieved when motion is initialized by a range of initial velocities v_0. Also, the resulting V_{0i} is almost the same as V_{ti}, being the maximum velocity V_{max}. Such behavior is the same as the symmetrical walking motion, having the one-step period limit cycle behavior. However, compared to the results obtained from the symmetrical walking pattern (Figure 4.7), it can be observed that the minimum velocity V_{min} in Figure 4.11 is a bit biased, not having the V_{min} in the middle of a step-period. Nevertheless, the minimum walking velocity still occurs at the center line $X = 0$ although the resulting motion is a bit forward compared to the previous symmetrical walking motion, as shown in the stick-diagram, Figure 4.12.

Furthermore, from the range of the given initial walking velocities v_0 ($0.4m/s - -1.2m/s$), compared to that of the symmetrical pattern ($0.5m/s - -1.35m/s$), it can be seen that walking motion x_{asym1} is more tolerant to the smaller initial walking velocities, but less for the larger initial walking velocities. This means the center of the actual ZMP location of the resulting limit cycle pattern was shifted to be slightly forward. Thus, it becomes more tolerant for the lower initial velocities. Referring to Figure 4.10(a), the ZMP of the basic pattern x_{asym1} is also shifted in the same direction, compared with Figure 4.6(b). These results show that the ZMP of the resulting limit cycle pattern is correlated with the ZMP planned in the basic walking pattern $B(t)$. This will be particularly discussed in the later section. Figure 4.13 shows the actual dynamics data. It can be still seen that the step-magnitude is maintainable as indicated in Figure 4.13 (d)(e) by dash lines.

Similarly, solution x_{asym2} shows the maximum walking velocity V_{max} also occurs at the touchdown moment as desired and the minimum walking V_{min} still occurs close to the center line $X = 0$, as shown by the stick-density in Figure 4.15. Also, the adjustment range of the initial walking velocity v_0 assigned is observed to be further biased as $v_0 : 0.3m/s - -1.1m/s$, as shown in Figure 4.14.

From the results obtained from this walking under the dynamically asymmetrical basic pattern $B_{asym}(t)$ Category 1, the resulting walking pattern has the characteristics summarized as:

- The steady-state is shown of one-step period limit cycle as: $V_{0i} \doteq V_{ti} = V_{max}$ with the desired step-magnitude maintainable.
- V_{min} still occurs close to the center line position $X = 0$ although the position configuration is no longer symmetrical.
- The center of the actual ZMP location is associated with that of the planned ZMP location in the basic walking pattern $B(t)$.
- Control convergence behavior is still found to be: compared to the V_{max}, a higher initial velocity → a lower initial velocity for the subsequent step; a lower initial

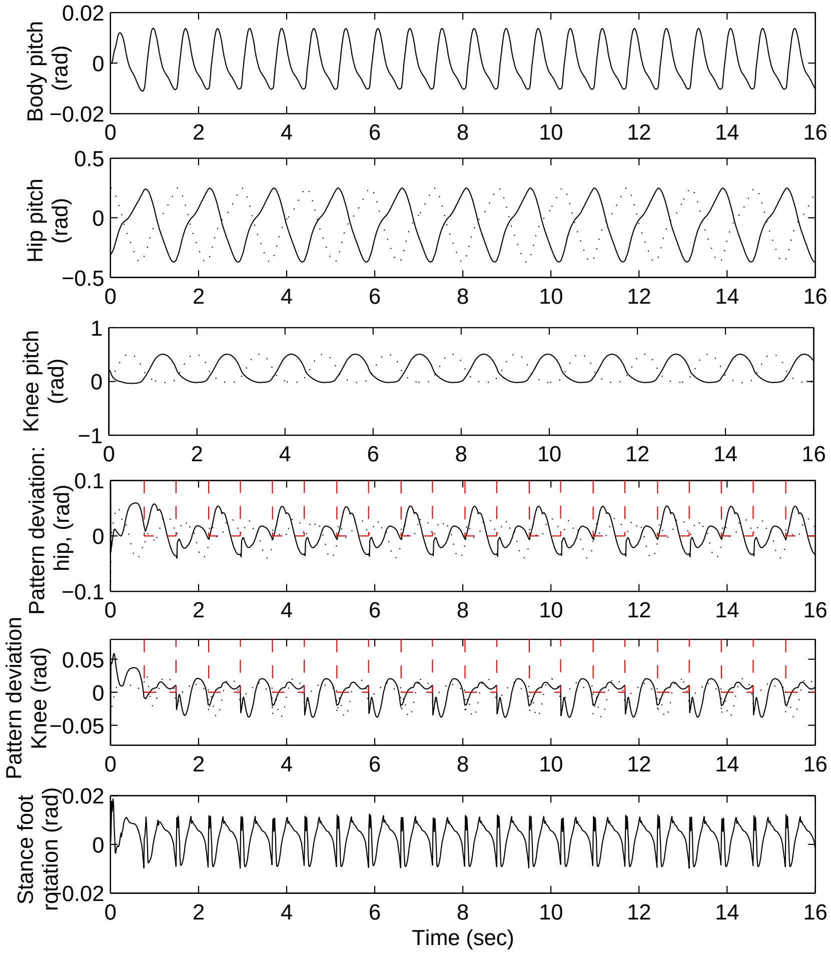


Figure 4.13: Resulting dynamics of the dynamic walking X_{asym1}.

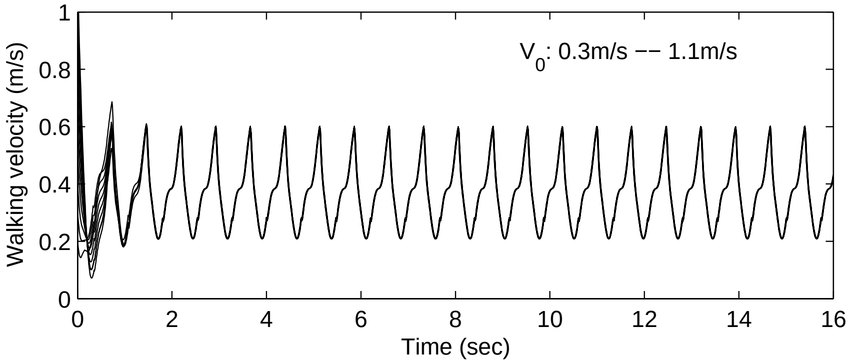


Figure 4.14: Walking velocity profile of motions under X_{asym2}.

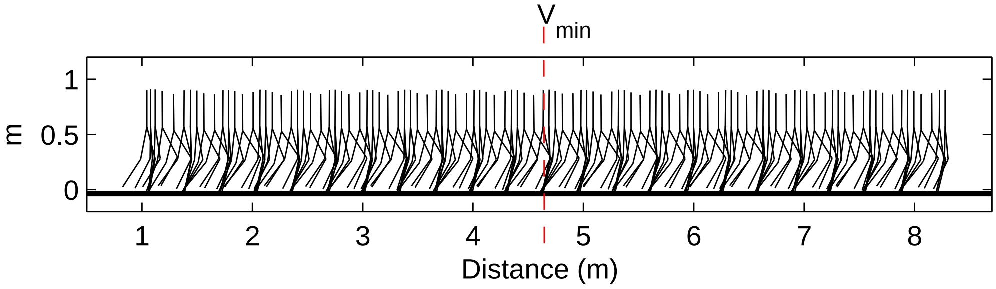


Figure 4.15: Stick-diagram of the motion under X_{asym2}.

velocity → a higher initial velocity for the subsequent step; and the difference to the steady-state motion was quickly reduced and eliminated eventually, while the desired step-magnitude is maintainable in transitions.

4.3.2 Walking Results of Category 2

For B_{asym} Category 2, average walking velocity V_{B1} and V_{B2} are given to the f_3 (Op_1 code) in the GAOFSF for generating asymmetrical walking velocity during the negative and positive walking phases, referring to Chapter 3. Table 4.4 gives the set-up of the GAOFSF.

Also two representative walking solutions in this category have been generated:

$$x_{asym3} = [0.259 \; -0.05 \; -0.013 \; 0.016 \; -0.007 \; -0.448 \; -0.173 \\ -0.037 \; 0.010 \; 0.014 \; 0.382 \; 0.180 \; 0.079 \; -0.034 \\ -0.019 \; -0.021 \; 0.000 \; 0.056 \; 0.420] \tag{4.8}$$

Table 4.4: GAOFSF Set-up for asymmetrical motion pattern generation, Category 2 (flat-terrain)

Description	Remark
Objective components	Code Op_1: f_1, f_2, f_3 + Code Op_2: f_5
Constraint components	Code Op_1: $s_1 - s_5$
Equivalent level	$a_{min} = 10, b_{min} = 6$
Average walking velocity	$\bar{v_1} = 3.5m/s, \bar{v_2} = 4.5m/s$
Chromosome representation	real-valued GA
Initial population M	150
Generation number T	250
Crossover operators	heuristic crossover simple crossover arthritic crossover
Mutation operators	multi-non-uniform mutation Uniform mutation Boundary mutation
Weights for objectives	$w_i = [10, 10, 10, 10]$
Weights for constraints	$p_i = [60, 20, 20, 20, 70]$
Chromosome format	$[A_i, B_i, C_i, c_h, c_k, t_1, t_2]$

$$
\begin{aligned}
x_{asym4} = [&0.297\ 0.074\ \ 0.006\ 0.009\ \ -0.008\ \ -0.474\ \ -0.116 \\
&0.045\ 0.030\ 0.011\ \ 0.675\ \ 0.176\ \ -0.065\ \ -0.012 \\
&-0.033\ \ -0.015\ \ 0.001\ \ 0.050\ \ 0.440]
\end{aligned}
\tag{4.9}
$$

The planned ZMP and the needed stance ankle joint torque are shown in Figure 4.16 for the above two solutions. The ZMP trajectories of both patterns get outside the supporting polygon. Also, from the ZMP or the stance ankle joint torque, it can be observed that neither of the two motions is dynamically symmetrical referring to the center line $(X = 0)$, and the first motion is further more asymmetrical in dynamics. Figure 4.16(e) and (f) show that the end configuration of the above two patterns are rather symmetrical as $|x^-| \doteq |x^+|$, however, the inclinations of the body position plots show that the velocity pattern during one-step is less symmetrical. x_{asym3} is even more obvious in the velocity asymmetry, as shown in Figure 4.16(e).

Applying the proposed motion control strategy, given various initial walking velocities v_0 to the pattern x_{asym3}, Figure 4.17 shows the velocity profiles during motion transitions and the resulting limit cycle behavior. It can be seen that the controller still can take this level of motion asymmetry and achieve the limit cycle behavior. Generally, the walking velocity is still first reduced and then increased, and the resulting limit cycles are almost the same to each other although they are excited by different initial velocities. Furthermore, it can be observed that the first step getting into the limit cycle behavior can be changed by giving different initial velocities. This is because different initial velocity may lead a different leg first to get into any one step in the cyclic pattern. Nevertheless, the walking results further confirm the existence of the limit cycle behavior using the proposed motion control strategy.

Figure 4.18 shows the stick-diagram of the resulting walking motion. As expected, the

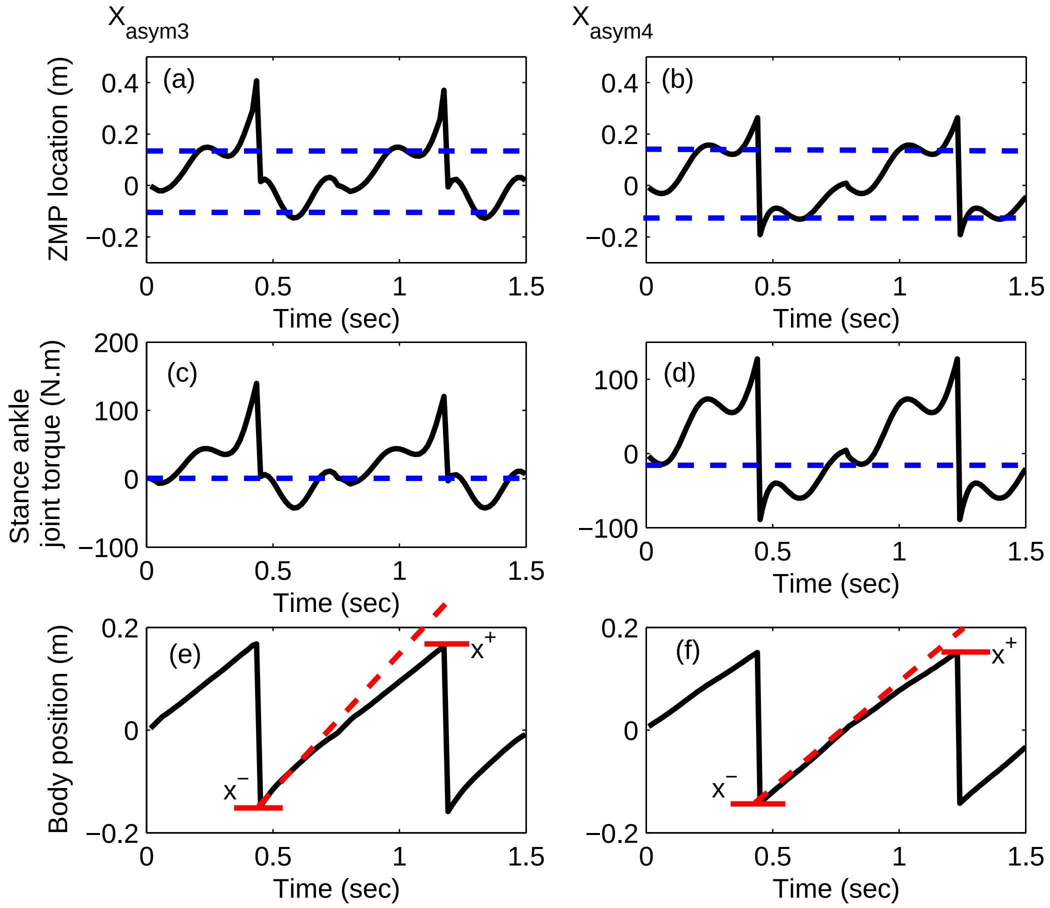


Figure 4.16: Pattern generation results for B_{asym} Category 2.

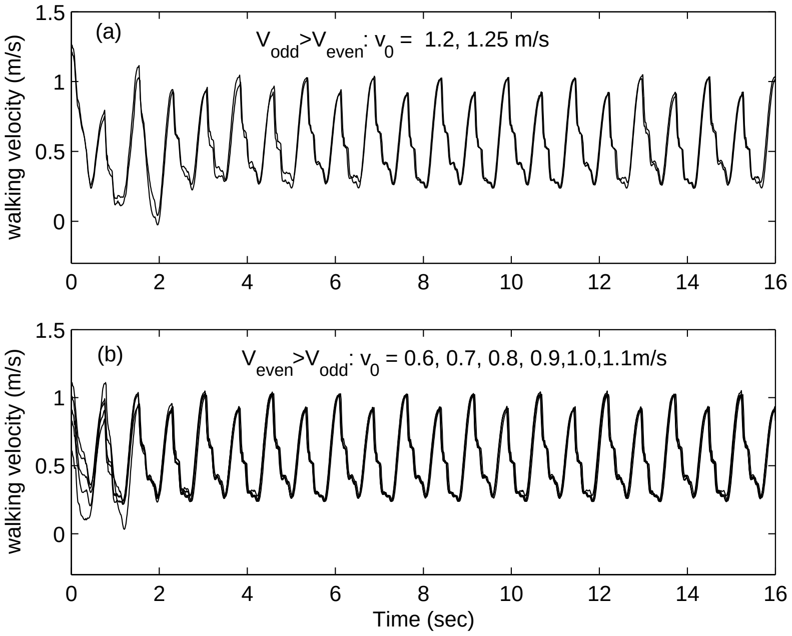


Figure 4.17: Walking velocity profile of motions under different initial velocity v_0.

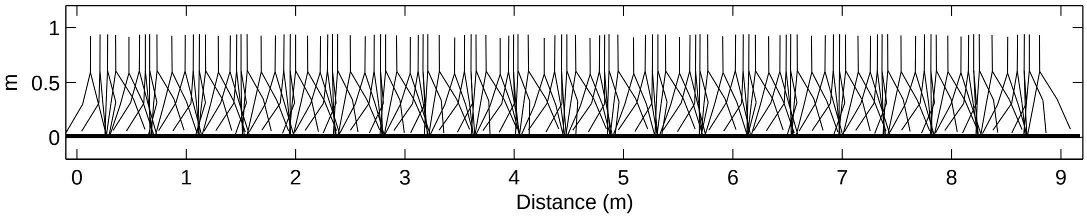


Figure 4.18: Stick-diagram of the dynamic walking. Solution X_{asym3}.

real motion is still shrunk inside the planned motion observed from the density of the stick-diagram. This indicates that the walking velocity is generally lower in the middle of a step but higher near the touch-down phase. Figure 4.19 gives the detailed dynamics of this walking motion. The step-magnitude was still maintained well, as errors were reduced at the touchdown moment, as indicated in Figure 4.19(d) and (e).

However, compared with the results obtained from B_{sym} and B_{asym} in Category 1, the primary difference is the limit cycle composed under this x_{asym3} is shown to be of two-step period, being $V_{0i} = V_{0i+1}$, $V_{ti} = V_{ti+1}$ but $V_{0i} \neq V_{ti}$, as shown in Figure 4.17. Further

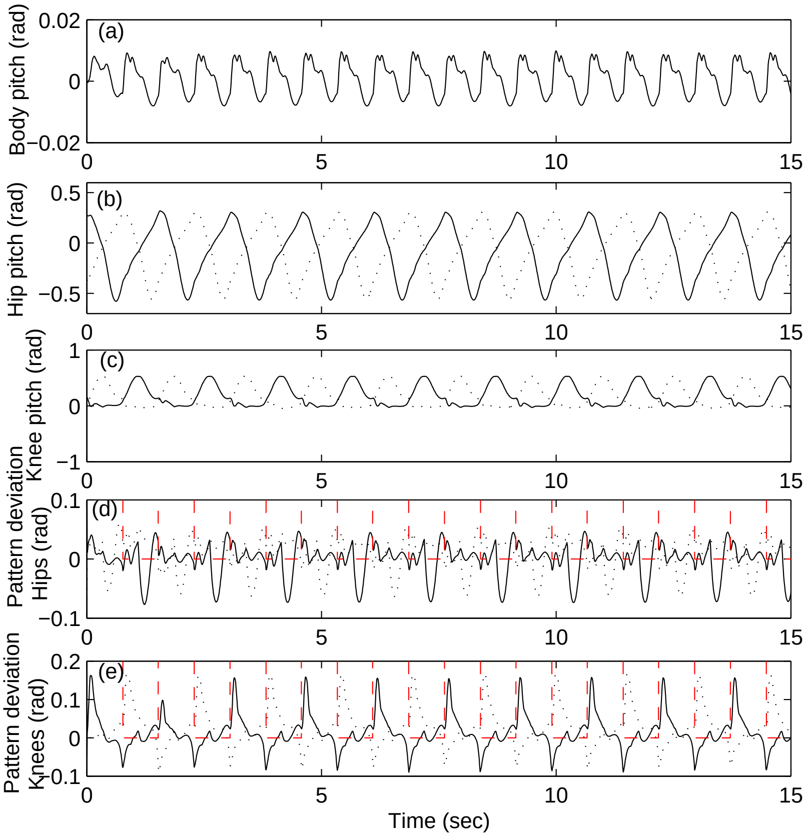


Figure 4.19: Resulting dynamics of the dynamics walking, solution X_{asym3}.

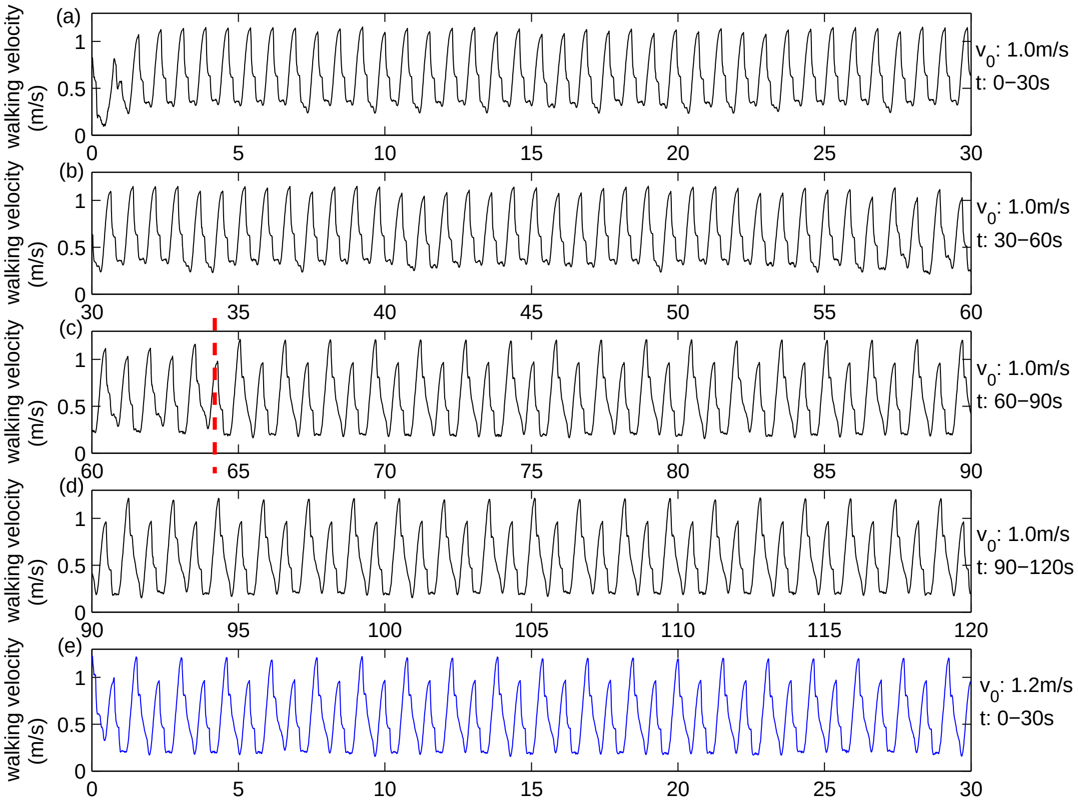


Figure 4.20: Walking velocity profile of motions with further bigger velocity asymmetry under different initial walking velocity v_0.

increasing the magnitude of this locomotion for 1.1 times, the difference $\Delta = |v_{0i} - v_{ti}|$ is also accordingly increased, as shown in Figure 4.20. Due to the higher asymmetrical behavior, the adjustable range of v_0 that can result in the limit cycle becomes more restricted. Note v_0 range is reduced from $(0.6, 1.25)m/s$ to $(0.8, 1.2)m/s$. The first four rows of Figure 4.20 show that when $v_0 = 1.0m/s$, the robot took a long time, as indicated about 65s marked by the dash line, to get into the limit walking cycle pattern. Given another initial velocity $v_0 = 1.2m/s$, walking quickly converged to the limit cycle pattern, as shown in the last row of Figure 4.20. Although the transition periods for the two cases are quite different, the final steady-state walking motions of both cases appeared to be exactly the same. The reason for the long period transition found in walking excited by $v_0 = 1.0m/s$ will be explained in the later section.

Such a two-step period limit cycle behavior is also confirmed by the pattern x_{asym4}. Furthermore, it can be seen that: with the walking velocity being less asymmetrical, the

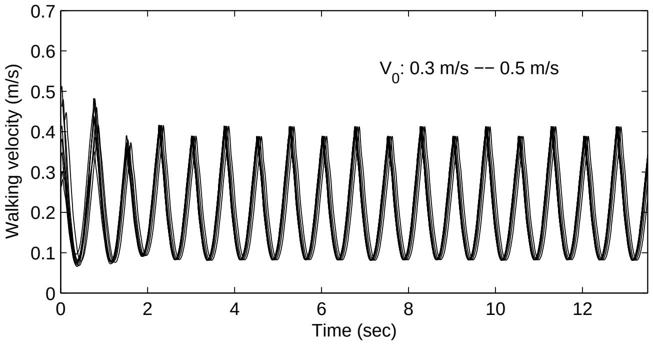


Figure 4.21: Walking velocity profile of motions under different initial walking velocity v_0, Solution X_{asym4}.

resulting difference between the two steps in a walking cycle is also reduced, referring to Figure 4.17 and Figure 4.21.

Based on the results obtained from this walking under the dynamically asymmetrical basic pattern $B_{asym}(t)$ Category 2, the resulting walking pattern has the following characteristics:

- $V_{0i} \neq V_{ti}$ but the step-magnitude can be still well maintained, yielding a two-step period limit cycle behavior.
- With the difference between V_{B1} and V_{B2} getting smaller, the difference between the two steps in the limit cycle pattern is also accordingly reduced.
- The stick-density is generally higher in the middle of a step, but lower for the swing-in and swing-out phases, as shown in Figure 4.18. This indicates the actual ZMP location has been also shrunk compared to the ZMP location of the planned basic walking pattern.
- Control convergence behavior is still found to be: with the step-magnitude being well maintained in transitions, compared with the initial and end walking veloci-

ties in one step motion in the steady-state, a higher initial velocity → a lower end velocity, which is also the initial velocity for the subsequent step; a lower initial velocity → a higher end velocity instead; and the difference to the steady-state was eventually eliminated.

4.4 Analysis of The Limit Cycle Patterns

Based on the obtained dynamic simulation results, it can be clearly seen that the proposed motion control is able to achieve the desired stable limit cycle walking behavior excited by quite a large range of initial velocities (or pushes).

However, two types of limit cycle behaviors have been found when walking velocity in the basic walking pattern is changed from uniform ($\bar{v_1} \doteq \bar{v_2} \doteq const$) to be less equivalent ($\bar{v_1} \neq \bar{v_2}$). The limit cycle behavior of the walking motion under the basic walking patterns with the uniform walking velocity all appear to be of one-step period. This means the left stance step is the same as the right stance step. However, when the walking velocity in the basic walking pattern becomes rather inequivalent before and after the position $X = 0$, the resulting limit cycle behavior is shown to be of two-step period. This means the walking cycle is repeated but the left-stance step is a bit different from the right-stance step. In addition, with the difference $\Delta = |\bar{v_1} - \bar{v_2}|$ getting smaller, the difference of the two steps in a walking cycle was also accordingly reduced. Therefore, uniform walking velocity is shown to be very correlated with the achievement of the one-step period limit cycle walking behavior. The following gives a possible reason why the basic walking patterns with uniform walking velocity achieved the one-step limit cycle behavior.

Thanks to the following three characters of the generated basic walking pattern: 1) walking velocity is uniform, 2) the gravitational momentum is about compensated referring to the center line $X = 0$ over a step-period, and 3) stride-frequency is selected to be close to the natural frequency, the following control behaviors have been achieved:

1. Even if the low and non-critical control gains are applied, still a rather wide range of initial conditions can result in the robot reaching the desired end configuration of the planned step motion.

2. Given the low control gains, more natural dynamics [80] can be excited, resulting in the the forward velocity profile smoothly reduced during the swing-in phase and then increased during the swing-out phase. Such a ”U” shape velocity profile is desirable for walking locomotion stability, as it indicates the resulting ZMP location is shrunk inside the planned ZMP area.

3. Compared to the steady-state motion, the higher initial velocities cause the CG magnitude overlap with the planned motion a bit forward than the center line $X = 0$ and vice versa for the assigned lower initial velocities. Therefore, a motion with the center line position $X = 0$ fully responded (V_{min} occurs at the position $X = 0$) is found to be included in the range that the step-magnitude is maintainable.

Based on the above control behaviors contained in the soft motion control, the following explains why the resulting motion is eventually stabilized at a motion that the desired step-magnitude is maintained and the velocity appears to be the ”U” shape with the minimum walking at the center line position $X = 0$.

Referring to a basic walking pattern with the uniform walking velocity, it can be noted that the basic walking pattern automatically entrains the sub-basic walking patterns $B(t)_{sub}$ inside the $B(t)$ with the same dynamics correlation before and after the center line $X = 0$, as illustrated in Figure 4.22(a). (Line pairs $OA - OB$, $OC - OD$, $OE - OF$ before and after the center line $X = 0$ (OO') indicate the sub-basic walking patterns). Then, the motion, which can have the CG position overlap with the prescribed motion at the center line $X = 0$ and achieve the desired step-magnitude, can be considered as a compound motion based on a linear motion base (the smallest sub-basic walking pattern $OE - OF$ in Figure 4.22(a)) adding equivalent acceleration modifications to the linear base (*abc* in Figure 4.22(b)) by all the sub-basic walking patterns. In Figure 4.22(b), slope *ab* and *bc* represent the accelerations a_{L1} and a_{L2} in the linear base for slowing

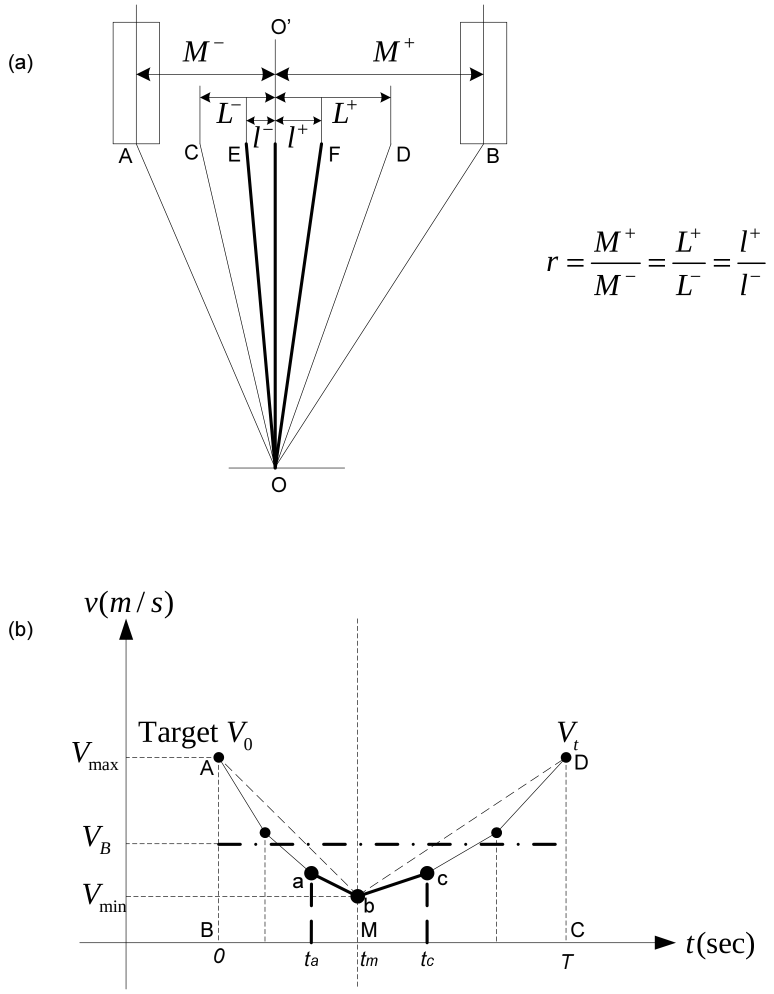


Figure 4.22: The resulting motion compounded on the linear base.

down and speeding up before and after the moment t_m. Therefore, the correlation between the positive and negative accelerations in each sub-basic walking pattern then is considered to be fixed for the resulting motion, as illustrated in Figure 4.22(b).

Based on Figure 4.22(b), the relationship of the accelerations a_{L1} and a_{L2} in the linear base can then be derived as Equations (4.10) to (4.13):

$$\because l^+/l^- = S_{cbMf}/S_{abME} = M^+/M^- = S_{ABMb}/S_{DCMb} \tag{4.10}$$

$$\therefore \frac{V_{min}(t_c - t_m) + \frac{1}{2}a_{L2}(t_c - t_m)^2}{V_{min}(t_m - t_a) + \frac{1}{2}a_{L1}(t_m - t_a)^2} = \frac{V_B(T - t_m)}{V_B t_m} \tag{4.11}$$

$$\because t_c - t_m = k(T - t_m) \;\; and \;\; t_m - t_a = kt_m \tag{4.12}$$

$$\therefore \frac{a_{L1}}{a_{L2}} = \frac{T - t_m}{t_m} \tag{4.13}$$

Hence, incorporating all the equivalent acceleration modifications to the linear motion base, the relationship between the actual average accelerations $\bar{a_1}$ and $\bar{a_2}$ for slowing down and speeding up will be also maintained as:

$$\frac{\bar{a_1}}{\bar{a_2}} \doteq \frac{T - t_m}{t_m} \tag{4.14}$$

resulting in the same initial and end walking walking velocity of the steady-state motion pattern. Thus,

$$V_{0i} = V_{ti} \tag{4.15}$$

Furthermore, the ratio r of the positive and negative magnitude of all the actual sub-motions, including the linear base, will be the same as $r = M^+/M^-$ illustrated in Figure 4.22(a).

Based on the above features of the motion that the center line $X = 0$ was responded, it can be seen that this motion is rather correlated with the basic motion pattern in the aspects: 1) the same average position displacement and the same average walking velocity; 2) the same dynamics correlation before and after the $X = 0$ position, as the magnitude ratio is always being r and the corresponding velocity ratio is always being 1. Therefore, this motion is supposed to be the least deviated motion under the set of low control gains. Since the general control behavior of the proposed motion control law is still to resemble the planned motion pattern through achieving the desired relative joint angles, all the other motions aside from the least deviated motion pattern will then converge to the least deviated motion pattern. Because the desired step-magnitude can be well maintained for a wide range of initial conditions around the initial condition of the steady-state motion, the velocity response to the least deviated motion pattern is then sketched as shown in Figure 4.23(a), where the lines AB and CD denote the least deviated motion pattern, and the dash arrows indicate the change of the starting velocity when motion is in transitions to the stable limit cycle pattern.

Referring to the convergence behavior, given an initial walking velocity which is in the range that the step-magnitude is still maintainable, the resulting motion will just follow the illustrated velocity response behavior, transiting to the least deviated motion pattern which has the end dynamics of a step motion equal to the initial's. This explains why the appeared limit cycle pattern under the basic walking patterns with uniform walking velocity are all of the one-step period type and the minimum walking velocity V_{min} roughly occurs at the center line position $X = 0$, referring to the simulation results of B_{sym} and B_{asym}, Category 1.

Following the same logic, it can also explain why the two-step limit cycle pattern occurs when the walking velocity in the basic walking pattern is less symmetrical (inequivalent) according to the center line $X = 0$.

In this work, the velocity of the negative and positive phases for the asymmetrical walking patterns B_{asym} Category 2 are planned to be close to V_{B1} and V_{B2}, respectively. Then the basic walking pattern $B(t)$ also entrains the approximated sub-basic walking patterns according to the center line $X = 0$. Then, the pattern that $X = 0$ position is roughly responded can then be identified as the least deviated motion pattern under the current gain setting.

Since the positive and negative step-magnitude are about the same for motions in Category 2, area S_{ABMO} and S_{CDEM} in the velocity-time plot will then be equal to each other. With $V_{B1} > V_{B2}$, $t_m < (T - t_m)$, the initial velocity of the least deviated motion pattern V_0 shall then be higher than its end velocity V_t, as $V_0 > V_t$ illustrated in Figure 4.24.

Based on the motion convergence behaviors observed from the above study about basic walking patterns, it can be derived that in Figure 4.23(b)

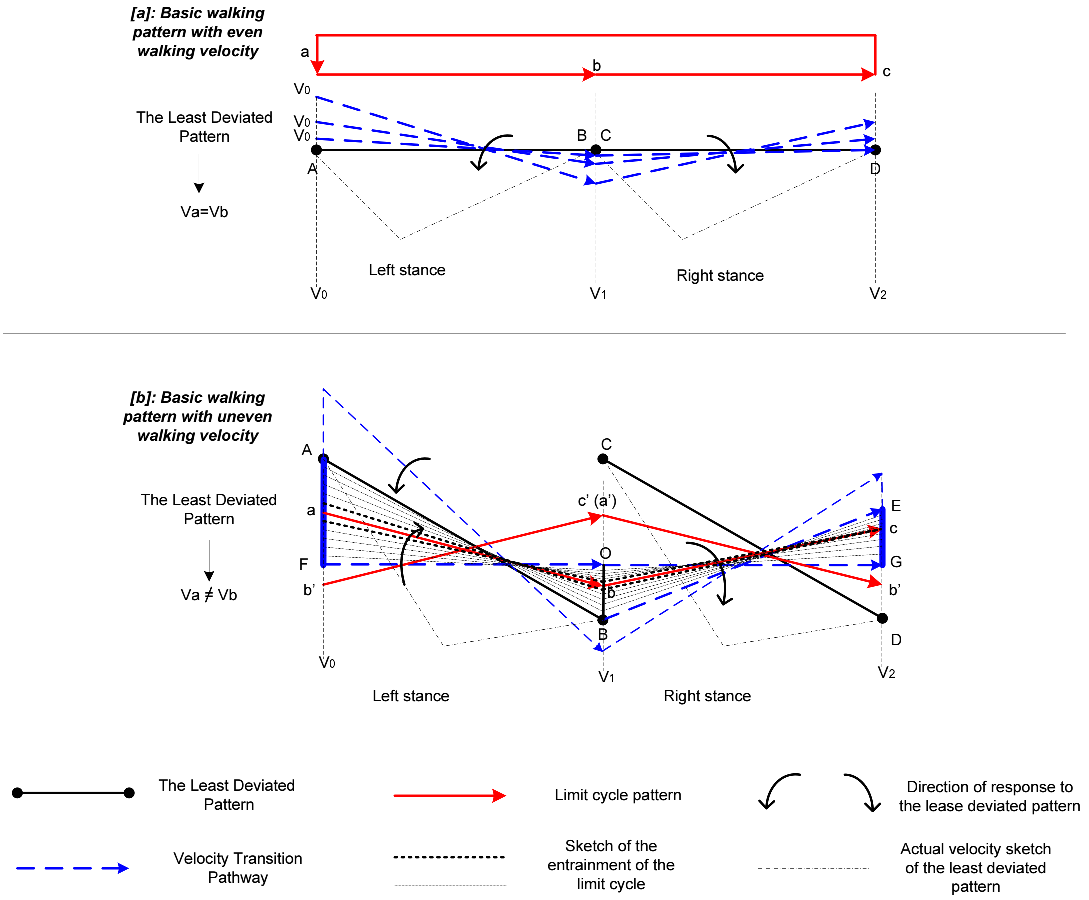

[a]: Basic walking pattern with even walking velocity
The Least Deviated Pattern
Va=Vb
a
b
c
V0
V0
V0
B
C
A
D
Left stance
Right stance
V0
V1
V2
[b]: Basic walking pattern with uneven walking velocity
The Least Deviated Pattern
Va ≠ Vb
A
C
a
c' (a')
E
c
F
O
G
b'
b
b'
B
D
V0
V1
V2
Left stance
Right stance
The Least Deviated Pattern
Limit cycle pattern
Direction of response to the lease deviated pattern
Velocity Transition Pathway
Sketch of the entrainment of the limit cycle
Actual velocity sketch of the least deviated pattern

Figure 4.23: Explanations for the resulting one-step limit cycle and two-step limit cycle patterns. *With the step-magnitude being maintained, the higher initial velocity* v_{0i} *of a step results in the end velocity* v_{ti} *lower than that of the steady-state motion; the lower initial velocity* v_{0i} *results in the end velocity* v_{ti} *higher than that of the steady-state motion; the difference to the steady-state motion converges to zero eventually.*

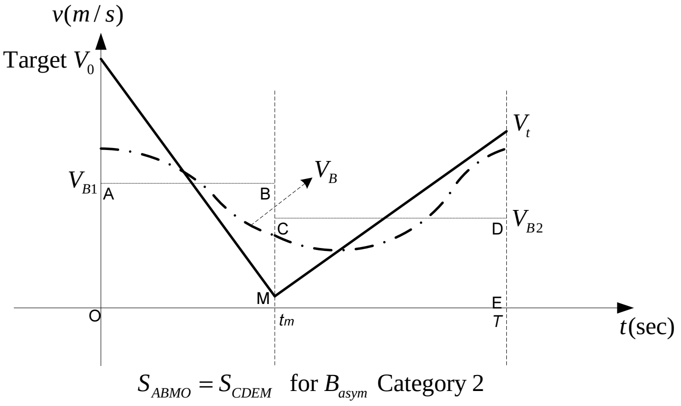


Figure 4.24: Walking velocity sketch for the basic walking patterns with uneven walking velocity $V_{B1} \neq V_{B2}$.

(1) if $v_0 > V_A$, then $v_1 < V_B$ and $|v_1 - V_B| < |v_0 - V_A|$

(2) if $v_1 < V_C$, then $v_2 > V_D$ and $|v_2 - V_D| < |v_1 - V_B|$

(Note, for $v_0 < V_A$, it is just the same as (2) states).

Therefore, when the assigned initial walking velocity is higher than V_A which results in the least deviated motion pattern, the pathway of the velocity transition will be just similar to what is shown in Figure 4.23(a), having $v_2 < v_0$. This is because when $v_0 > V_A$, the two steps in a walking cycle respond to the least deviated pattern in different directions as indicated in Figure 4.23(b). Therefore, there is no conflict in between, same as the walking guided by a basic walking pattern with uniform walking velocity. Taking one walking cycle as a whole, the highest velocity v_2 of the right stance step would be lower than the initial velocity v_0 of the left stance step. Therefore, there must be a moment that the initial velocity of the left stance step is equal or smaller than V_A, just the same as assigning an initial velocity $v_0 < V_A$.

When v_0 is assigned to be smaller than V_A as $v_0 \leq V_A$, the direction of the left stance step responds to the least deviated motion pattern will be changed, as indicated in Figure 4.23(b), and cause some mutual inhibition between the two steps in a walking cycle. Since the inclination of AB and BE are in different directions, there must be an initial velocity v_0 yielding $v_0 = V_F = V_O = V_G$, as indicated in Figure 4.23(b). Also, referring to the (2) above, the length of AF shall be longer than the length of EG. Therefore, there must be a limit cycle entrained in the area of $AFGE$, and the entrained limit cycle can be shown as either abc or FOG, as indicated in Figure 4.23(b).

However, compared to Figure 4.23(a), it can be noted that $AF > OB$ (referring to the (2) above) for walking guided by a velocity asymmetrical walking pattern, but $AF = OB = 0$ for walking guided by a uniform walking velocity pattern. Therefore, if the difference $\Delta = V_{B1} - V_{B2}$ is more observable, the entrained limit cycle is believed to be shown of the two-step period type provided the desired step-magnitude is still maintainable, as represented by abc in Figure 4.23(b).

Furthermore, with the velocity difference $V_{B1} - V_{B2}$ getting increased or decreased, the sketch of the two-step limit cycle would be also roughly scaled. This explains the trend found in the simulations, including those with the uniform walking velocity $V_{B1} \doteq V_{B2}$: the more V_{B1} is deviated from V_{B2} planned in the basic walking pattern, the more velocity difference would be shown in the two steps in the entrained two-step limit cycle pattern provided the step-magnitude is maintainable.

In addition, as seen from the sketch of the two-step period limit cycle, Figure 4.23(b), the resulting two-step period limit cycle is like a stretched limit cycle of the one-step period limit cycle. Therefore, the area in between the stretched limit cycle (abc and $b'c'b'$) is also a transitional region although there is a state with $V_F = V_O = V_G$ included. Back to the results obtained from x_{asym3}, this explains why the higher velocity $v_0 :> 1.1m/s$ got the highest velocity V_{max} occur to the right leg (odd-step) but the lower velocity v_0 : $0.6m/s - -1.1m/s$ got the highest velocity V_{max} occur to the left leg (even-step) instead. Furthermore, Figure 4.20 shows with the pattern being proportionally enlarged for 1.1 times, given the initial velocity $v_0 = 1.0m/s$ which is in this middle transitional region, it took a long time to converge the robot motion to the limit cycle pattern. Also, the steps in the transition did have similar initial and end velocity for each step. However, with the error being accumulated, walking eventually still converged to the two-step period limit walking cycle. This validates the above sketch about the two-step limit cycle pattern.

4.5 Derived GAOFSF Objective Functions for Basic Walking Patterns

Based on the results of the objective function study for basic walking pattern generation that allows low gain motion control for the achievement of a smooth, natural and stable locomotion, the following three motion behaviors have been found to be important.

(1) The accumulated gravitational momentum of the planned basic walking pattern is supposed to be close to zero over a step-period.

(2) Uniform walking velocity of the planned basic walking pattern.

(3) The stride-frequency should be selected to be close to the robot natural frequency.

The above three conditions combined together make it rather easy for the controller to maintain the desired average walking speed and the desired landing configuration without using critical and stiff control gains. Also, the target motion pattern (the least deviated motion pattern) can be identified as a step motion whose initial and end dynamics are rather equivalent.

Therefore, given the stride-frequency is already set to be closer to the robot natural frequency, to achieve the desired one-step period limit cycle behavior using the proposed soft motion controller, the objective functions that will be sent to the GAOFSF method are derived as:

(1) Make the walking velocity uniform as much as possible. (f_1 in the GAOFSF method, code Op_2)

(2) Make the motion as symmetrical as possible with respect to the body CG: $\Delta_X = |X^-| \doteq |X^+|$. ($f_2$ in the GAOFSF method, code Op_2)

From the perspective of motion robustness, the resulting ”U” shape walking velocity has already indicated the actual ZMP location displaced from the prescribed ZMP (computed based on the basic walking pattern) to the center line $X = 0$. The maximization of the ZMP displacement from the planned ZMP is related with the joint control gains applied to the robot. This can be always separately done by fine tuning the joint control gains. Then, the optimization of the ZMP of the actual motion is really about the optimization of the average ZMP location over a step period. In this case, if there is any correlation between the average ZMP position over a step of the generated basic walking pattern and that of the actual walking motion, the optimization of the average ZMP position in basic walking pattern will then indirectly lead to a safe ZMP location of the actual motion.

Given that the desired one-step period limit cycle walking pattern is achievable, the average ZMP position displacement from the prescribed one due to the position deviation will be just zero. Then the acceleration components play a major role for the ZMP displacement to the center line $X = 0$. The displacement can be computed as:

$$\Delta ZMP_1(t) = \frac{K(a_{1n} - a_{sn})}{m(g + \ddot{z}_G)} \tag{4.16}$$

$$\Delta ZMP_2(t) = \frac{K(a_{2m} - a_{sm})}{m(g + \ddot{z}_G)} \tag{4.17}$$

at any instant; where a_{1n} and a_{2m} are the accelerations at some moment before and after the V_{min}, respectively; a_{sn} and a_{sm} are the accelerations of the basic walking pattern before and after the V_{min}, respectively. Then the total displacement of the actual ZMP from the ZMP of the basic walking pattern towards the $X = 0$ position can be computed as:

$$\int_0^{t_m} \Delta ZMP_1(t)dt = \frac{K(\int_0^{t_m}(a_{1n} - a_{sn})dt)}{m(g + \ddot{z}_G)} \tag{4.18}$$

$$\int_{t_m}^{T-t_m} \Delta ZMP_2(t)dt = \frac{K(\int_{t_m}^{T-t_m}(a_{2m} - a_{sm})dt)}{m(g + \ddot{z}_G)} \tag{4.19}$$

Since the velocity in the basic walking pattern is rather uniform, as $V_{B1}(t) \doteq V_{B2}(t) \doteq const$,

$$\int_0^{t_m} a_{sn} dt = \int_{t_m}^{T-t_m} a_{sm} dt \to 0 \tag{4.20}$$

Note the one-step limit cycle pattern has the initial walking velocity of a step equal to the end velocity of this step, as: $V_{0i} = V_{ti}$, yielding

$$\int_0^{t_m} a_{1n} dt = \int_{t_m}^{T-t_m} a_{2m} dt \tag{4.21}$$

Apply the Equations (4.20) and (4.21) to Equations (4.18) and (4.19), it can be derived that the total displacement of the actual ZMP from the ZMP of the basic walking pattern towards the center line $X = 0$ in periods of $(0, t_m)$ and (t_m, T) are the same, as:

$$\int_0^{t_m} \Delta ZMP_1(t) dt = \int_{t_m}^{T-t_m} \Delta ZMP_2(t) dt \tag{4.22}$$

This indicates the average ZMP value of the actual robot motion will be the same as the average ZMP value of the basic walking pattern $B(t)$. This also explains why with the center of the planned ZMP of the motions x_{asym1} and x_{asym2} shifted forward, the center of the actual ZMP is observed to be also shifted in the same direction, as seen from the shift of the range of the initial walking velocities v_0. Therefore, to achieve a better motion robustness, the objective function (3) will be also added for performance trade-off.

(3) Make the location of the average of the ZMP close to the center of the stance footprint.(f_3 in the GAOFSF method, code Op_2)

Since the optimization of the average ZMP location is based on the condition that the one-step period limit cycle walking pattern is achievable, it leads to the logic that if performance (3) greatly conflicts with the performances (1) and (2), the optimization strategy will not work. However, if performance (3) does not much conflict with the performances (1) and (2), the optimization strategy will then be able to conduct a performance trade-off and give an overall ”optimum”.

As seen from the simulation results, a certain level of posture bias or inequivalent walking velocity can still result in the feasible limit cycle behavior with the desired step-magnitude being well maintained. Also, the ZMP of this planned pattern will be rather overlapped with the robot CG position if the walking velocity of the basic walking pattern is optimized to be uniform. Therefore, performance index (3) will not much conflict with (1) or (2). To testify the derived motion optimization strategy, the following section applies the objective functions (f_1, f_2 and f_3 code Op_2 in the GAOFSF method) to basic walking pattern generations for up- and down-slope walking motions. The applied motion controller is still soft.

4.6 Algorithm Generalized to Slope-terrain Walking

In this section, the sagittal motion control algorithm is generalized to slope-terrain walking control. The basic walking patterns generated are for walking on a 10^o up-slope and down-slope terrains.

4.6.1 Up-slope Walking

In the first slope walking example, the GAOFSF approach was used to generate a basic walking pattern, with walking velocity and the center of the ZMP optimized, for the simulated NUSBIP-I robot to walk up a 10^o slope. The desired step length is set to $0.26m$ and the average walking speed along the horizontal plane is set to $0.27m/s$ $(\omega_h = 3.2623)$. Note if this frequency or step-length is not very applicable for the actual dynamic walking control, the stride-frequency and step-length can be tuned directly through the parameter ω_h and R respectively without any other changes of the motion joint trajectories, as the parameter tuning will not affect the unform walking velocity, the

Table 4.5: GA Set-up for up-slope walking

Description	Remark
Objective components	Code Op_2: f_1, f_2, f_3
Constraint components	Code Op_2: $s_1 - s_5$
Chromosome representation	real-valued GA
Initial population M	150
Generation number T	250
Crossover operators	heuristic crossover simple crossover arthritic crossover
Mutation operators	multi-non-uniform mutation Uniform mutation Boundary mutation
Weights for objectives	$w_i = [50, 10, 40]$
Weights for constraints	$p_i = [15, 50, 20, 30, 500]$

average ZMP location or the posture correlation before and after the center line $X = 0$. The GA environment set-up and weighting parameters are shown in Table 4.5.

The best chromosome, $[A_i, B_i, C_i, c_h, c_k, t_1, t_2]$, was found by GA to be

$$\begin{aligned} x = [&0.222\ -0.041\ -0.011\ 0.014\ -0.005\ -0.5\ -0.035 \\ &0.097\ 0.039\ 0.004\ 0.650\ -0.096\ -0.087\ 0.085 \\ &0.024\ -0.144\ 0.267\ 0.001\ 0.540] \end{aligned} \tag{4.23}$$

Figure 4.25 shows the motion generation result in terms of the joint angle trajectories, ZMP trajectory, body position profile and pattern stick-diagram. It can be seen that the planned ZMP is still outside the supporting polygon for this step-length walking, but

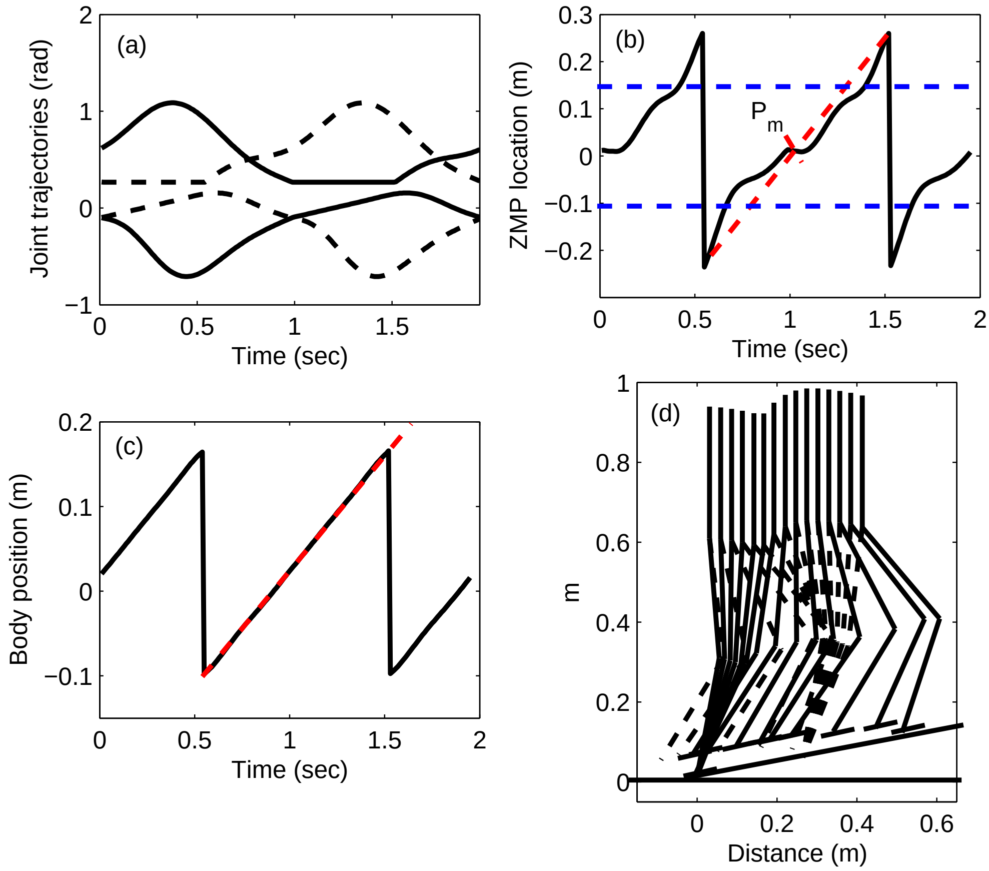


Figure 4.25: Motion generation result for 10^o up-slope walking.

the center of the ZMP, P_m, is near the center of the supporting polygon. The inclination of the body position profile, Figure 4.25(c), shows that the walking speed has been optimized to be uniform. Figure 4.25(d) shows the stick-diagram of the locomotion with the desired step-length well achieved.

When the control algorithm was applied using the basic pattern x_{up}, the resulting walking pattern converged to a stable steady-state. This is shown in the stick-diagram, Figure 4.26. Compared to Figure 4.25(d), the real walking pattern is also shrunk inside with the minimum walking velocity occurring roughly at the position $X = 0$, as shown in Figure 4.27. Figure 4.28 shows the walking velocity profile for the motion in transitions and the achieved limit cycle pattern. The limit cycle is shown to be just the one-step period type. The convergence behavior of the controller can be still observed the same as that

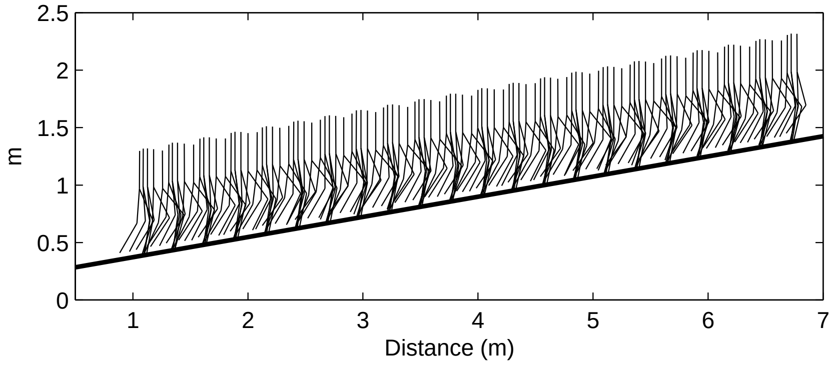


Figure 4.26: Stick-diagram of the dynamic 10^o up-slope walking motion.

of all the walking results presented above. Figure 4.29 provides the resulting dynamics of this up-slope walking.

4.6.2 Down-slope Walking

The generated down-slope walking pattern is specified as: the step length $0.35m$ and the average walking speed $0.6m/s$ $(\omega_h = 2\pi)$. The GA parameters used are shown in Table 4.6. The best chromosome obtained by GA, in the form of $[A_i, B_i, C_i, c_h, c_k, t_1, t_2]$, is

$$\begin{aligned} x = [&0.312\ -0.076\ 0.018\ -0.005\ -0.004\ -0.327\ -0.190 \\ &-0.096\ -0.031\ -0.009\ 0.144\ 0.118\ -0.007\ -0.114 \\ &-0.033\ -0.194\ 0.799\ 0.147\ 0.220] \end{aligned} \tag{4.24}$$

Figure 4.30 shows the motion generation result in terms of the joint trajectories, ZMP trajectory, body position profile and the pattern stick-diagram. Still, the ZMP is outside the supporting polygon for this step-length walking but the center of the ZMP trajectory

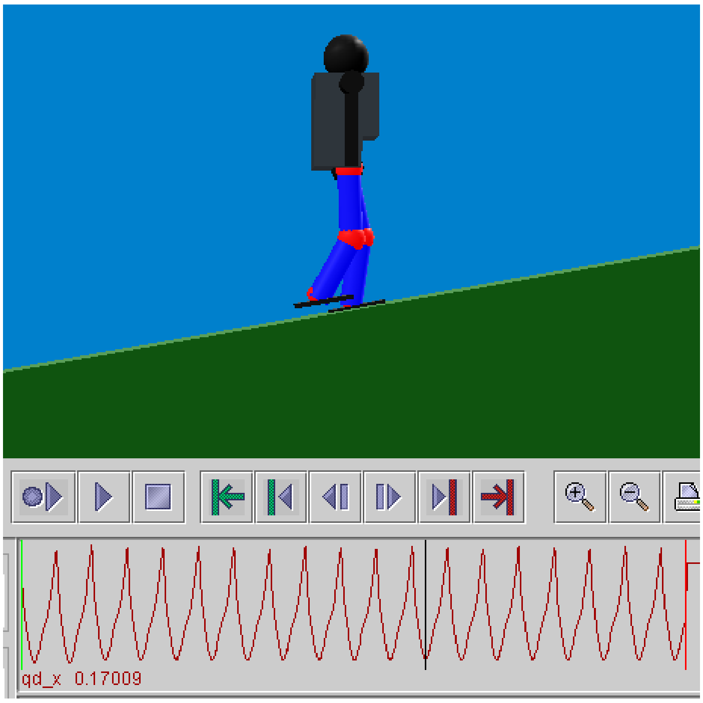


Figure 4.27: Posture having the minimum walking velocity V_{min} shown in the Yobotics! simulation.

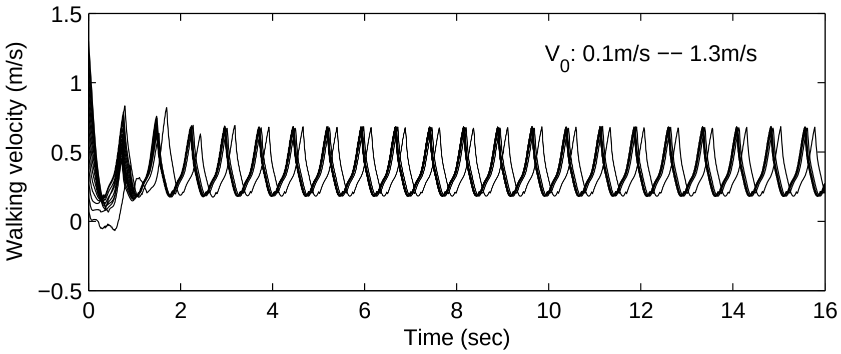


Figure 4.28: Walking velocity profile under different initial velocity v_0. 10^o up-slope.

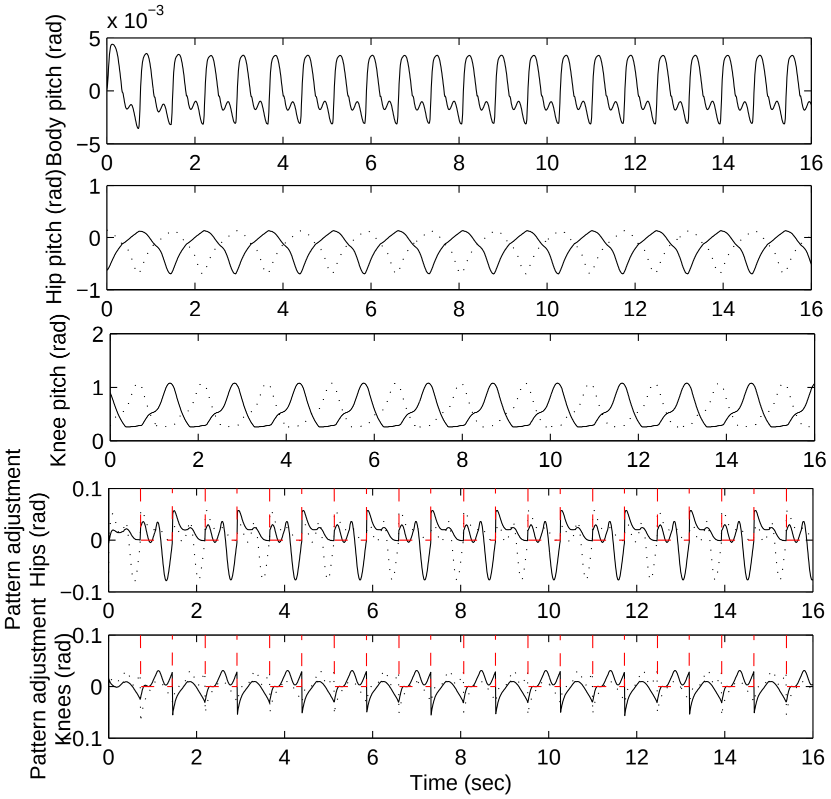


Figure 4.29: Resulting dynamics of the 10^o up-slope walking.

Table 4.6: GA Set-up for down-slope walking

Description	Remark
Objective components	Code Op_2: f_1, f_2, f_3
Constraint components	Code Op_2: $s_1 - s_5$
Chromosome representation	real-valued GA
Initial population M	150
Generation number T	250
Crossover operators	heuristic crossover simple crossover arthritic crossover
Mutation operators	multi-non-uniform mutation Uniform mutation Boundary mutation
Weights for objectives	$w_i = [50, 20, 30]$
Weights for constraints	$p_i = [15, 80, 20, 30, 800]$

is near the center of the supporting polygon. The inclination of the body position profile, Figure 4.30(c), shows the walking velocity has been optimized to be uniform. The desired walking speed and step length have been both achieved well, as shown in the position profile and the pattern stick-diagram, Figure 4.30(d).

The achieved dynamic walking pattern is shown in the stick-diagram, Figure 4.31. Comparing with Figure 4.32, in this case, the minimum walking velocity also occurs at about the vertical line $X = 0$. Figure 4.33 shows the walking velocity profile for motion in transitions and the achieved limit cycle pattern. The limit cycle is also of the one-step period. Figure 4.34 provides the actual dynamics of this down-slope walking, and the magnitude of a step is still shown to be maintainable.

From the results, it can be also noted that although the general shapes of the joint angle trajectories obtained are similar in form to that shown in Figure 3.3, there is a significant difference in the value of t_1, the instant when the stance knee starts to lock for the different slopes. For both flat terrains and for up-slope walking, t_1 is negligible but for

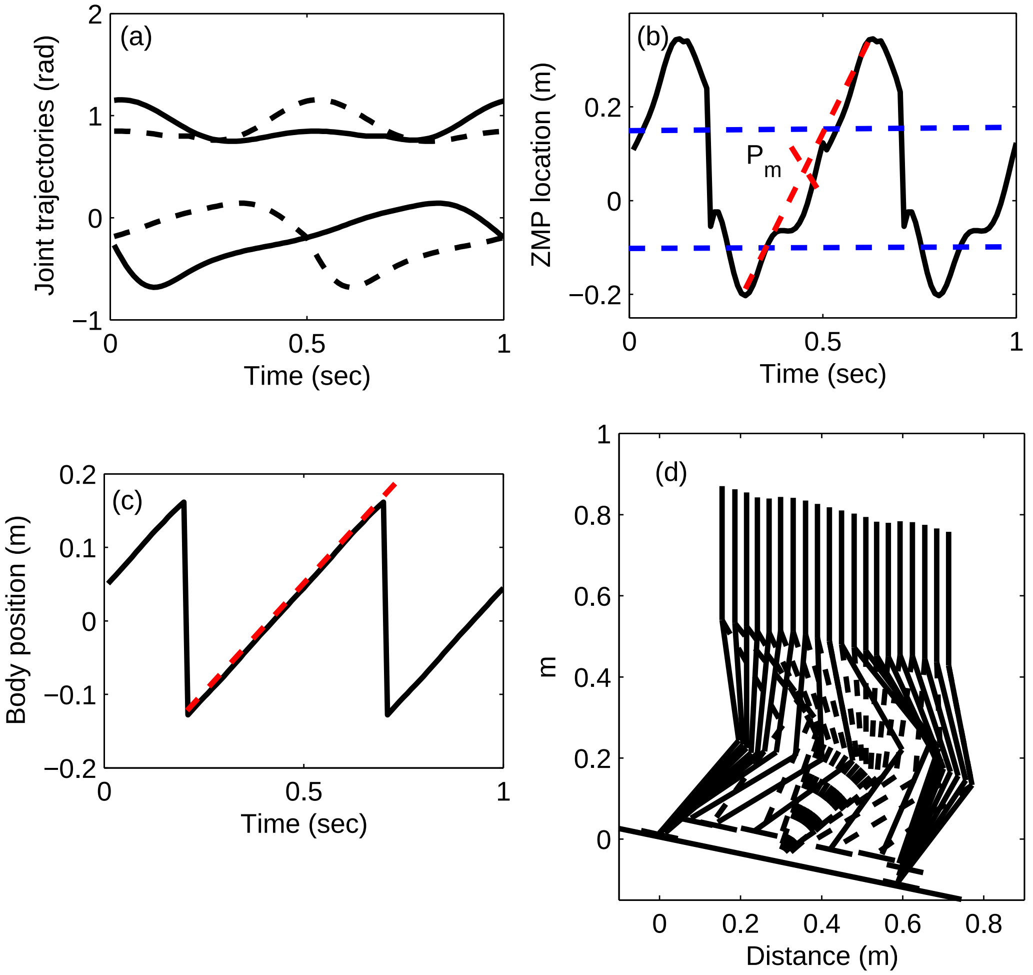


Figure 4.30: Motion generation result of the 10^o down-slope walking.

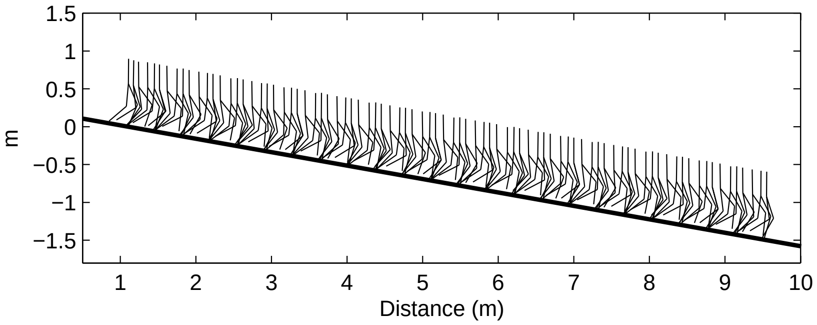


Figure 4.31: Stick-diagram of the actual 10^o down-slope walking motion.

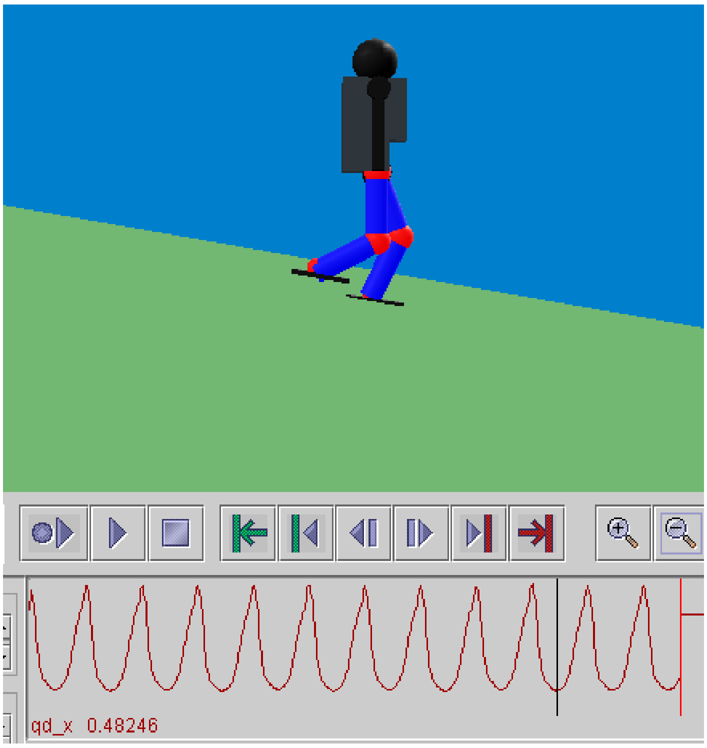


Figure 4.32: Posture having the minimum walking velocity V_{min} shown in the Yobotics! simulation.

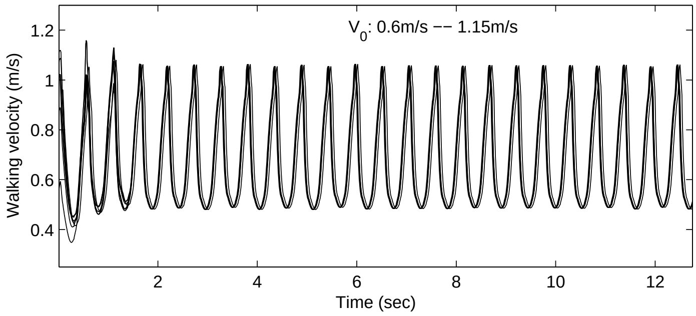


Figure 4.33: Walking velocity profile of motions given different initial velocity v_0.

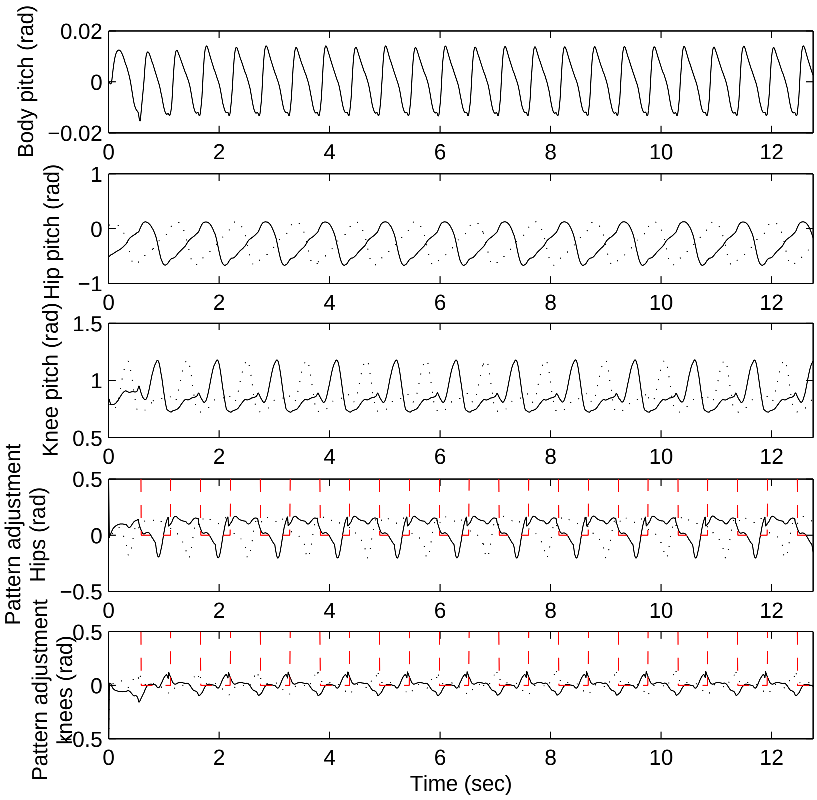


Figure 4.34: Resulting dynamics of the 10^o down-slope terrain walking.

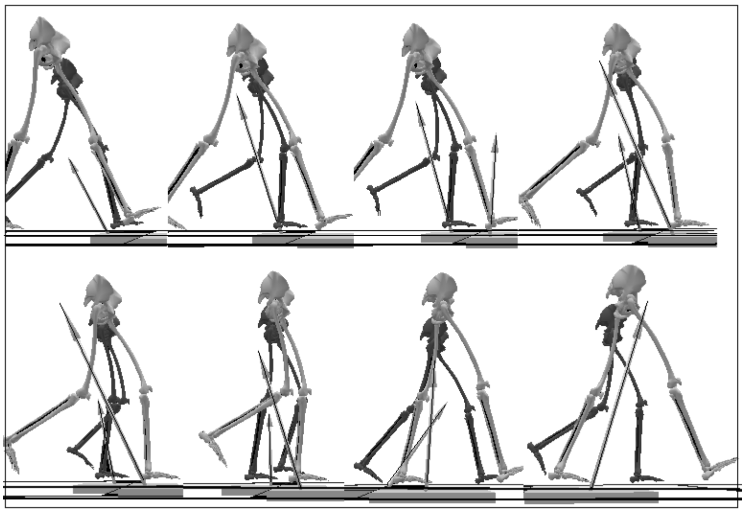


Figure 4.35: Orientation and the magnitude of reaction force in human walking recorded by VIOCON.

down-slope walking, the period from t_0 to t_1 is quite significant. This shows that, in the latter case, the stance knee joint is locked for only a very short duration and may also not be locked at all. Such behavior is also similar as what has been found in human gaits[86]. The following section particularly compares the generated walking gaits with human gaits.

4.7 Comparison With Human Gaits

As mentioned earlier, the target of the motion control is to be human-like. To show that the proposed motion generation method GAOFSF and the joint control altogether can result in the human-like walking behaviors, the ground reaction forces were investigated. Here the flat-terrain walking is used to give a comparison between the robot gaits and human gaits. The human gaits were recorded by the *VICON* system. Figure 4.35 shows the measured orientation and the magnitude of reaction force. For a better illustration of the orientation and magnitude changing of the ground reaction force, the human walking experiment is displayed at once with two different timings (dark gray and light gray), as shown in Figure 4.35.

Referring to Figure 4.36(a), the reaction forces (shear force and vertical reaction force) of the flat-terrain walking guided by B_{sym} is found to be quite acceptable based on the human-gaits analysis [86]. The length of vector is related to its force magnitude. Comparing the orientation and the magnitude of the reaction force in human walking, a good match between the generated gaits and the human counterpart can be noted. The force vectors' orientations rotate clockwise with the progress of the supporting phase. The value of vertical ground reaction force is about the same as the body weight in most of the time during the supporting phase. This confirms that the robot was stably supported.

Furthermore, the shear forces are negative at first and then becomes positive, as shown in Figure 4.36(a). This is exactly what is measured in human gaits. The negative part is the braking phase of touching down and the positive part is the motion propulsion phase when the supporting leg pulls the ground to drive the body. The peak value of the shear forces is about 15% of body weight, comparable to $16-20\%$ in human walking.

From Figure 4.36(b), it can be noted that the calculated friction coefficient that is required is small (< 0.25). It means that normal surfaces (PCV floor, wooden floor, brick path, asphalt) towards robot feet with shoe type sole (for those materials friction coefficient ranges from 0.5 to 0.9)[85] can easily satisfy the compensation of shear force without slip.

Similarly, the orientation and magnitude of the ground reaction force for up-slope and down-slope motions are shown in Figures 4.36(c) and (e), respectively. For going down a slope, it is quite natural to have a larger impact when the swing foot touches the ground comparing with up-slope or flat-terrain motions [86]. Figure 4.36(d) and Figure 4.36(f) show the required friction coefficient for the up-slope and down-slope motions, respectively. The small values in these figures indicate that walking can be realized by most ground situations.

From the above comparisons, it can be seen that the generated gaits have great similarities with human gaits and the required friction can be satisfied by most ground materials.

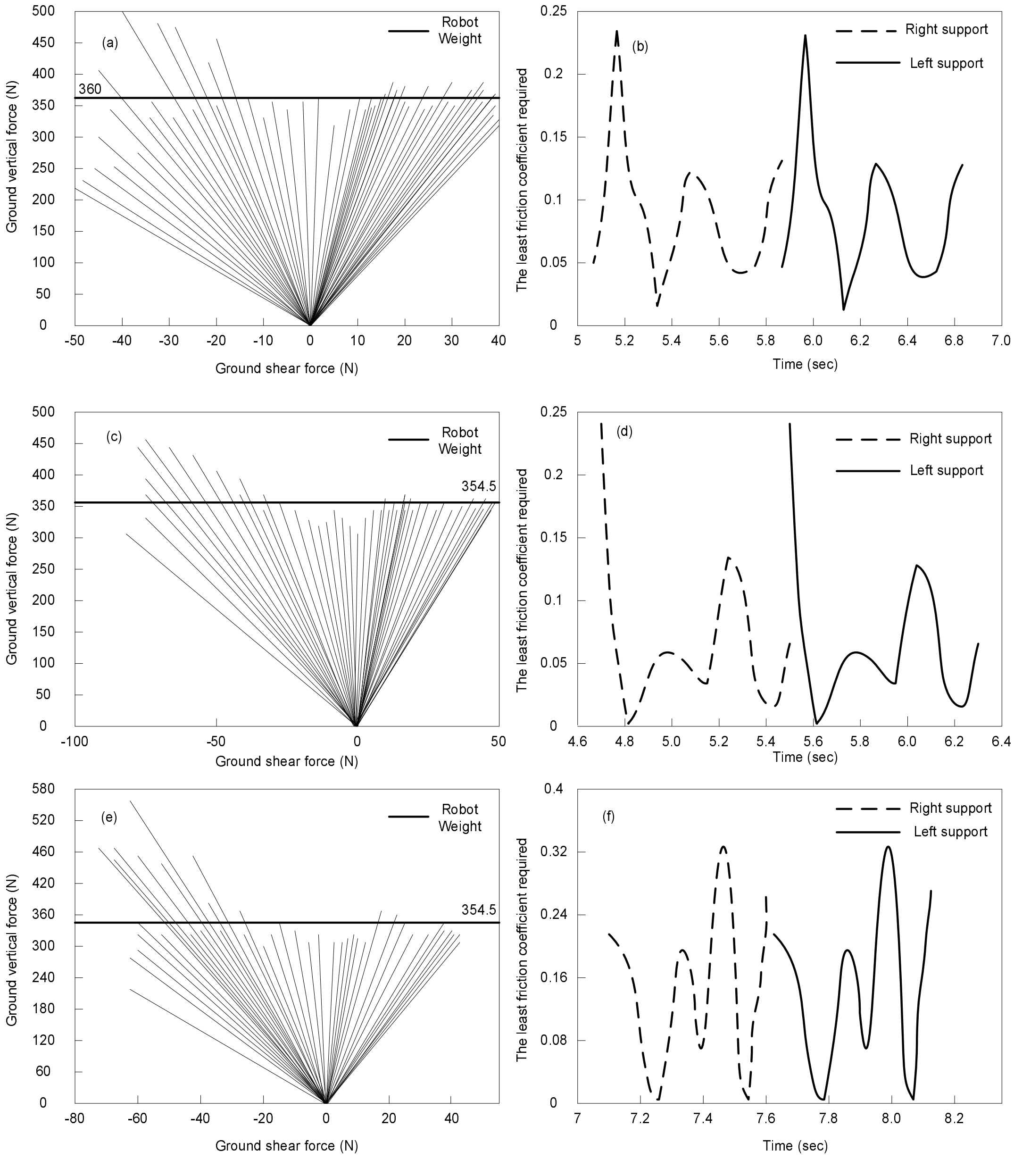


Figure 4.36: Ground reaction forces and required minimum friction coefficient for the generated walking on the flat-terrain, up-slope and down-slope.

4.8 Summary

In this chapter, the proposed motion control strategy has been shown effective and the limit cycle walking behavior is achievable. The objective functions for the basic walking pattern $B(t)$ have been studied and derived for the GAOFSF method. The generated walking patterns using the derived objective functions successfully guided the up-slope and down-slope walking in the simulation environment. The resulting walking behavior is stable, smooth and human-like.

Corresponding to the earlier mentioned motion control challenges, using the proposed motion strategy, the robot body motion has been controlled to be upright without any obvious pitching behavior. Also, low gain motion control has been achieved, resulting in the motion insensitive to the environment perturbations while converging to the desired walking behavior as well.

Besides the motion control strategy, the GAOFSF method using a Truncated Fourier Series (TFS) to model the joint trajectories for human-like walking patterns has been demonstrated. The generation method is shown to be general, with full dynamics incorporated. Therefore, it can be readily applied to different motions and robots with different geometrical and inertial properties.

The next Chapter will focus on the motion adjustment modes contained in the optimized TFS formulated basic walking patterns, and further show the advantages of using the TFS model and the robustness of the entrained limit cycle walking patterns.

Chapter 5

Sagittal Plane Motion Adjustment

Based on the knowledge obtained from human gaits research on CPG in rhythm generation and motion adaptation towards the environment, it can be summarized that walking is controlled by a basic function stored in human biological neural networks. During walking in different situations, this function is modified automatically according to human senses about the nature of the terrain and walking environment.

From the results obtained from Chapter 4, it can be seen that the TFS formulated motion pattern together with the proposed motion control law can automatically damp a range of motion perturbations and converge the robot walking into a limit cycle behavior. It thus provides the fundamental, which can also be compared to a CPG pattern stored in the low level, for further building up a walking network achieving the human-like rhythmic walking behavior and environment adaptation. Therefore, in this chapter, these human walking functions are explored based on the TFS formulated motion pattern using the proposed motion control law.

The key parameters that have been investigated in the TFS formulated motion pattern are: the fundamental frequency ω in Equations (3.10) and (3.11), the scaling parameter R of the Truncated Fourier series and the constant c attached at the end of the Fourier series. All these parameters have straightforward physical meanings related with walking behaviors such as walking stride-frequency, step-length and locomotion patterns. The following sections develop the motion adjustment modes for the stride-frequency, step-length and walking posture adjustments. The basic walking motion patterns used in this chapter are the symmetrical walking pattern on the flat-terrain x_{sym}, 10^o up- and down-slope walking patterns, x_{up} and x_{down}, generated in Chapter 4.

5.1 Stride-frequency Adjustment Mode

Referring to Equations (3.10) and (3.11), the fundamental frequency in the TFS formulation directly reflects the step frequency. Then, the basic pattern for a new frequency walking pattern can be directly derived from the basic walking pattern $B(t)$ generated by the GAOFSF method by varying the stride-frequency parameter ω_h. Simply adjusting this ω_h (ω_k has a fixed relationship with ω_h), the actual gait frequency will be simultaneously varied, suppose a stable limit walking cycle can be still entrained.

To explore the robustness of the limit cycle behavior, the adjustable range of the stride-frequency based on the GAOFSF generated basic walking patterns presented in Chapter 4 is first investigated. Here, the safe range of the stride-frequency adjustment is obtained from dynamic simulations:

(1) The achievable minimum and maximum stride-frequency for steady-state walking.

(2) The largest instantaneous stride-frequency transition without any external force acting on the robot.

Flat-terrain walking is used for robustness study. Figure 5.1 shows the walking velocity profile of a motion starting from the lowest permissible stride-frequency $\omega_h = 3$ to the highest $\omega_h = 5.1$, and then recovering back to the lowest. All the transitions are instantaneous.

It can be seen that the stride-frequency can be simply varied by tuning the fundamental frequency component in the TFS formulation. Also, it can be noted that the instan-

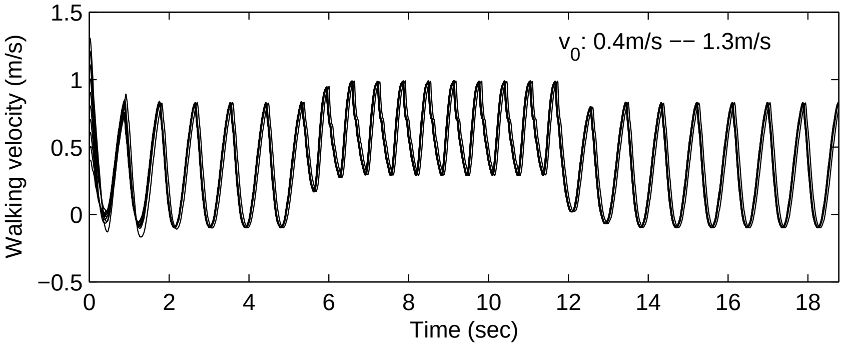


Figure 5.1: Walking velocity profile with instantaneous stride-frequency transitions between the highest and the lowest values.

taneous transitions do not lead to instability. Figure 5.2 shows the stick-diagram of the dynamic walking motion, from which it can be noted that the step magnitude is not very well maintained for the high-frequency walking using the same set of control gains. Due to the relatively softer control for this high frequency walking, the step-length is observed to be increased with the stance and swing phases both enlarged, but rather synchronized. However, a stable limit walking cycle can be still achieved and the steady-state is still of the desired one-step period type. As mentioned in Chapter 4, when walking under a higher frequency resulting in certain motion deviation from the planned step-magnitude, the least deviated pattern can be analogous to be a pattern maintained at a new step-magnitude level, have the neighboring position of the center line $X = 0$ responded. Thus, the end velocity might be slightly deviated from the initial velocity in the least deviated pattern, but after the mutual inhibition as illustrated in Figure 4.23(b), the resulting limit cycle may still appear to be of the one-step period. However, the location that the minimum walking velocity occurs could be a bit deviated from the center line $X = 0$ position. As seen from Figure 5.1, the minimum walking velocity during the high-frequency walking is shown to be slightly biased to the right side while that of the low frequency walking is almost in the middle of a step-period.

The direct adjustment of the stride-frequency has also been successfully applied to the 10^o up- and down-slope walking. The robustness of the basic walking pattern for slope terrains will be discussed in the later section. Table 5.1 shows the adjustable range of

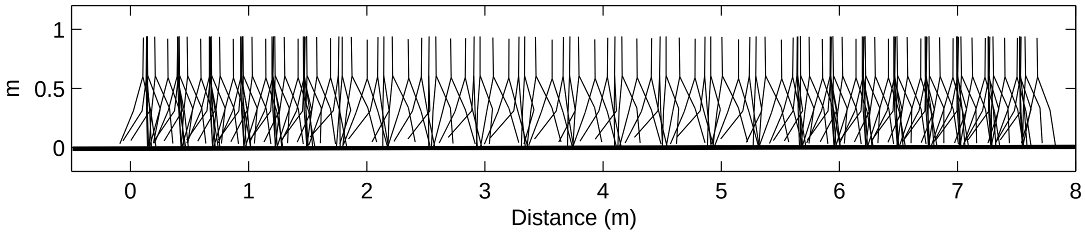


Figure 5.2: Stick diagram of the walking motion with instantaneous stride-frequency transitions between the lowest and the highest values.

Table 5.1: Adjustable range of the stride-frequency.

Terrain type	ω_{hmin}	ω_{hmax}	ω_h
Flat-terrain	3	5.1	4.36
10^o slope	2	5.5	4
-10^o slope	5.8	7	6.28

the stride-frequency for all the generated patterns $(x_{sym}, x_{up}, x_{down})$.

It can be seen that, taking the flat-terrain adjustment range as an example, these values are about 70% and 120% of the nominal value $\omega_h = 4.36$ obtained by the GAOFSF. This shows the robustness of the generated walking pattern. Furthermore, the nominal values show lower stride-frequency is more suitable for going up slopes while higher frequency is better for going down slope. Such behavior is also observed in human walking gaits [32].

The following subsection presents a learning agent for robot self-adjustment of the stride-frequency properly under external force perturbations.

5.1.1 Learning-Based Variable Stride-frequency Walking Under Perturbations

This subsection develops the application of reinforcement learning (RL) for online adjustment of the stride-frequency under external force perturbations. This is also a test of robustness for achieving the stable walking behavior using the generated basic motion pattern.

Affected by the external force perturbations, the robot stance leg will need to make some additional effort to maintain the current walking pattern if the stride-frequency is not adjusted. When the external perturbations become excessive, the robot stance foot may not be able to maintain the walking posture well enough because the torque that can be applied to the stance foot is limited. Even if the robot is able to successfully overcome the perturbation during the current step, the kinetic energy of the motion may not be enough for the subsequent walking when the external perturbation is suddenly removed. Consequently, the subsequent step may still fail the motion stability. Therefore, it is suggested to adjust the stride-frequency in response to external force perturbations.

From observations of human walking, the rhythmic walking behavior that adapts well to the environment seems to be a balanced performance between achieving the walking stability and minimizing the energy consumption. On one hand, for example the robot is being pushed, a walking pattern which consumes the least energy may just get to the maximum stride-frequency quickly. This means the motion is too compliant to the environment whereas the maximum stride-frequency is very limited by the robot physical system. On the other hand, a walking pattern staying the original frequency will also cost more energy to maintain balance under perturbations. Therefore, the desirable stride-frequency adjustment becomes a trade-off between the two motion indices: energy consumption and motion adaptation. This means the stride-frequency adjustment should not require the robot to walk completely compliant to the perturbation with less energy consumption or to stubbornly stick to the original walking pattern hereby requiring more energy consumption.

Since the feasible limit cycle behavior for various stride-frequency walking motions have already been entrained by the basic walking pattern in the low-level motion control, reinforcement learning method can then be applied in the high-level motion control to decide the appropriate frequency which balances the energy consumption and motion adaptation. The following presents the modules included in the Reinforcement Learning (RL) based agent.

State Variables

To have a better motion identification, the state variables are selected as:

(1) Standing phase $\theta_P = atan(X/H)$, where X is the body displacement referring to the center line $X = 0$ and H is the body height.

(2) Current velocity V_p of the body measured along walking slope.

(3) External disturbance force, f_t, applied at the CG of the robot body.

The action that the reinforcement learning agent will give is a proper stride-frequency ω_h that balances the energy consumption and the motion adaptation.

Based on the above state variables, whether the robot body lags behind or moves too forward in the current state can be identified by (1) and (2) together; and the subsequent robot motion trend due to the acceleration given by the external forces is also incorporated in deciding the stride-frequency ω_h.

Reward Function

The reward function gives rewards to motions which satisfy the motion objectives. At the same time, it also punishes a wrongly selected action. Equation (5.1) is the reward function used for this study:

$$reward = \begin{cases} k \cdot (C - |\Delta\omega_h|) & \text{for the successful actions} \\ -k_e|E| - k_\tau|\tau_q| & \text{for the failed actions} \end{cases} \tag{5.1}$$

where k, k_e, k_τ are the weighting parameters. C is a positive constant which stands for a fixed reward; E and τ_q are the stance phase error and the torque that is in excess of a preset torque limitation, respectively. In addition, the successful actions which cause the actual motion to be closer to the original walking pattern will be more encouraged.

Learning Tasks

In reinforcement learning (RL), the learning agent can only be more reliable after gaining more experience. The advantage of reinforcement learning is that once a suitable learning agent has been established, it can keep learning the different tasks that have been specified and reinforce the obtained experience without any manual intervention. However, it may not be easy to learn enough experience in a short period. To reduce the probability that the robot might encounter situations which have not been explored, six representative walking tasks have been assigned to the RL, as follows:

1) $\omega_0 = \omega_s$ and $f = random(f_s)$ with the objective: continuous walking for at least T_1 period with the frequency increased to ω_b.

2) $\omega_0 = \omega_s$ and $f = random(f_m)$ with the objective: continuous walking for at least T_2 period with the frequency increased to ω_b.

3) $\omega_0 = \omega_s$ and $f = random(f_b)$ with the objective: continuous walking for at least T_3 period with the frequency increased to ω_b.

4) $\omega_0 = \omega_b$ and $f = -random(f_s)$ with the objective: continuous walking for at least T_4 period with the frequency decreased to ω_s.

5) $\omega_0 = \omega_b$ and $f = -random(f_m)$ with the objective: continuous walking for at least

T_5 period with the frequency decreased to ω_s.

6) $\omega_0 = \omega_b$ and $f = -random(f_b)$ with the objective: continuous walking for at least T_6 period with the frequency decreased to ω_s.

$$\omega_s : \omega_h \in [2.9, 4.2] \Rightarrow \omega_t = 3.5 \quad \omega_m : \omega_h \in [4.3, 5.6] \Rightarrow \omega_t = 4.9$$

$$\omega_l : \omega_h \in [5.7, 7.0] \Rightarrow \omega_t = 6.3 \quad f_s : f \in [10, 30) \Rightarrow f_t = 20$$

$$f_m : f \in [30, 50) \Rightarrow f_t = 40 \quad f_l : f \in [50, 70) \Rightarrow f_t = 60$$

where,subscript 0 denotes the current walking step-frequency and subscript t denotes the dynamic information sent to the learning agent. ω_s, ω_m and ω_l denote the small, medium and large walking stride-frequencies, respectively. Similarly for the applied external forces f_s, f_m and f_l.

Based on simulations, the time limits that the robot can take the perturbation continuously without any stride-frequency adjustment are obtained for each of the walking tasks. The corresponding perturbation periods $T_1 \ldots T_6$ are chosen to be larger than the corresponding limits. Here, the assigned values for $T_1 \ldots T_6$ are 4.0s, 3.0s, 1.2s, 1.5s, 1.0s and 0.7s, respectively.

The common motion constraints are as follows: the absolute error of the trunk CG position from the basic walking pattern $|E| < 0.05m$, (height of the trunk CG is 0.925m); the sum of all the joint torques $\sum_i^6 \tau_i$ is less than $400Nm$; and the stance foot not to have any rotational angle according to the ground surface $\Delta\theta_f$ larger than 0.5^o. Given these specifications, with the torque constraint getting stricter, the energy consumption can be reduced and motion becomes more compliant to the perturbation. On the contrary, extending the desired walking time, the stride-frequency adjustment would be nearer to the original frequency in the basic walking pattern. Therefore the RL balances the energy consumption and motion adaptation.

Momentum Estimation

Together with the reinforcement learning agent, momentum estimation is another module for stride-frequency adjustment. This module is used to increase the confidence for the stride-frequency online adjustment, especially for states that have not been explored by the RL. Therefore, momentum estimation will be only used when the Q-value of the current state is lower than a preset threshold.

According to the sensed or the computed external force and the time-step for stride-frequency update, the input momentum given by the perturbation can be estimated by Equation (5.2) to (5.4).

$$M = f_t \cdot \triangle t = \Sigma_{i=1}^{7} m_i \cdot \triangle v_i \tag{5.2}$$

$$\triangle v_1 = 0; \triangle v_2 = l_2/L \cdot \triangle v; \triangle v_3 = l_1/L \cdot \triangle v \tag{5.3}$$

$$\triangle v_4 = \triangle v_5 = \triangle v_6 = \triangle v_7 = \triangle v \tag{5.4}$$

where l_1, l_2 and L are the effective lengths of the upper, lower and the whole leg respectively. i is the link number starting from the stance foot to the swing foot. The velocity change of the robot center of mass, Δv is estimated by substituting Equations (5.3) and (5.4) into Equation (5.2). Then, the change of stride-frequency is estimated using Equation (5.5):

$$\triangle \omega_h = \frac{\pi \cdot \triangle v}{S} \tag{5.5}$$

where S is the step-length.

R_I Update Function

As found in previous variable stride-frequency walking, the step-length given by the stance motion can be naturally increased when increasing the stride-frequency, without

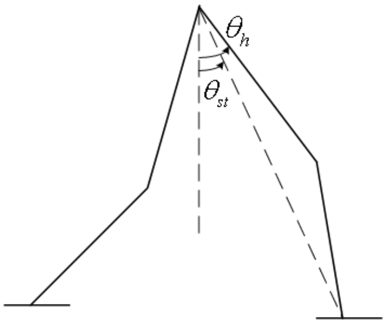


Figure 5.3: An illustration of θ_{st} θ_h in the standing phase.

scheduling the control gains. Since the stance motion is rather important for maintaining the stability of a dynamic robot walking pattern, the motion magnitude of the stance leg is better to be maintained, so as to reduce the perturbation affecting the motion convergence to the limit walking cycle pattern. Therefore, R_I will be adjusted at each touch-down moment to update the step-length. A stance phase P_t (Figure 5.3) is defined as Equation (5.6) for describing the standing posture.

$$P_t = \theta_{st} + 0.5(\theta_h - \theta_{st}) \tag{5.6}$$

where θ_{st} and θ_h are the angles formulated using the TFS model, as indicated in Figure 5.3 for describing the stance leg posture more precisely. Suppose only θ_{st} is concerned without taking the knee configuration into account, the identification of the stance phase may not be this representative when the stance leg is bent more obviously. On the other hand, if the hip joint angle θ_h is only considered, the posture identification may overlook the position relation to the stance foot. The update of R_I (scaling parameter of the stance leg) used in the work is the interpolated value in the adjustable range.

5.1.2 Training of the Reinforcement Learning Controller

The parameters of the reinforcement learning agent that was used in learning the assigned tasks is shown in Table 5.2. Figure 5.4 and Figure 5.5 show the learning results of the aforementioned walking tasks. Note that the training did not have many iterations

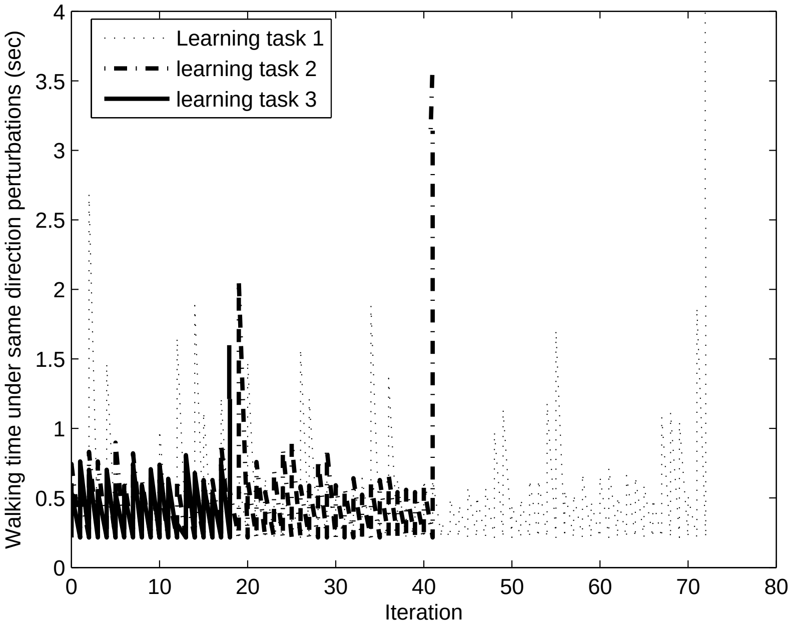


Figure 5.4: Learning performance of task 1 to 3.

to achieve the learning tasks. It can be noted that the training did not take much time to achieve the learning tasks and derive a policy for optimizing the trade-off between energy consumption and motion adaptation. This could be attributed to the regulation and optimization of the basic walking pattern.

5.1.3 Walking Results in Simulation

Incorporating the trained reinforcement learning agent and the momentum estimation module, the simulated NUSBIP-I robot was placed into a new environment. Then, some random force perturbations were applied. The environment is described as follows:

Forward external forces $20N$, $40N$, $60N$ were applied to the robot from $t = 2s$ to $t = 5s$; $t = 6s$ to $t = 6.8s$; and $t = 9s$ to $t = 9.25s$, respectively; and reversed external forces $-20N$,$-40N$,$-60N$ were applied from $t = 11.2s$ to $t = 11.6s$; $t = 13.7s$ to $t = 14.1s$; and $t = 16s$ to $t = 16.3s$, respectively. The robot has no prior knowledge of the disturbances. For motion recovery, whenever there is no perturbation sensed or the current

Table 5.2: Reinforcement Learning Set-up for stride-frequency ω_h adjustment

Description	Remark
Basic sagittal walking pattern	Flat-terrain walking example 1 in chapter 4
Learning output (action) u	parameter $\Delta\omega_h$
Reward function	Equation (5.1)
E_{max}	$0.01rad$
E_{min}	$-0.01rad$
$\theta_{f_{max}}$	$0.09rad/s$
$\theta_{f_{min}}$	$-0.09rad/s$
$\sum_i^6 \tau_i$	$< 400N \cdot m$
Discount factor γ	0.5
Action set U	$(0.001n \vert 2.9 \leq 0.001n \leq 7.0)$ and $n \in Z$
Policy	modified ε-greedy $(\varepsilon = 0)$
CMAC parameters	
Width: receptive field for θ_P Width: receptive field for V_P Width: receptive field for f_t Width: receptive field for u Receptive fields layers no. C Learning step-size α	$0.02m$ $0.1m/s$ $0.1N$ $0.1rad/s$ 128 0.25

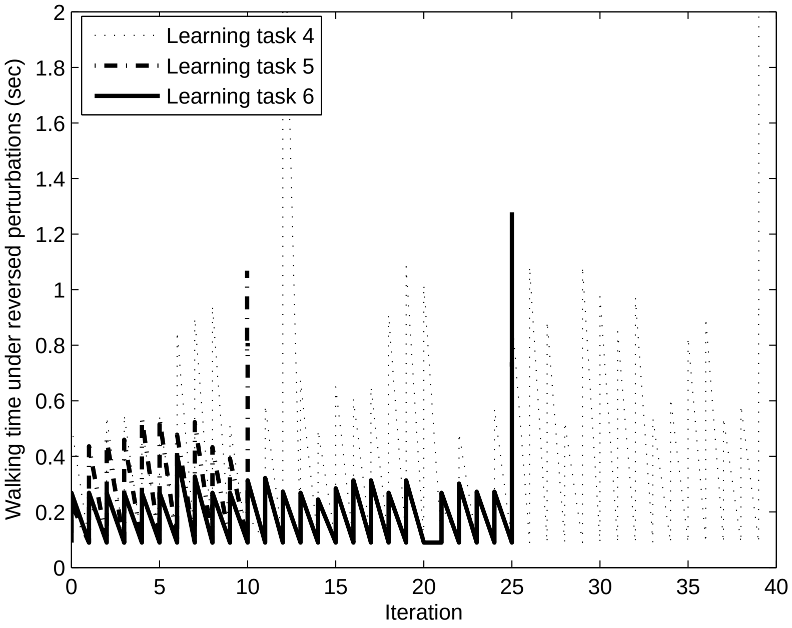


Figure 5.5: Learning performance of task 4 to 6.

motion state shows the walking satisfies all the requirements set in the learning agent, the recovery module will be called. The recovery module follows a linear way recovering to the desired walking pace.

Figure 5.6 shows the stick-diagram of the resulting walking motion in the prescribed walking environment without any step-frequency adjustment. It shows that the walking was finally disrupted due to the shortage of the kinetic energy after the external disturbances.

With the stride-frequency adjustment mode applied, the robot successfully went through all batches of the perturbations, as shown in Figure 5.7. The plots of the key data are provided in Figure 5.8, shows that the average walking velocity is always maintained at about $0.45m/s$. Also, the stride-frequency is observed to gradually increase for all the perturbations acting in the positive direction and reduce under the perturbations acting in the opposite direction. Furthermore, the simulation result shows the stride-frequency can be gradually recovered once the external force perturbation is not present. After all the external force perturbations were removed, the motion could converge to the

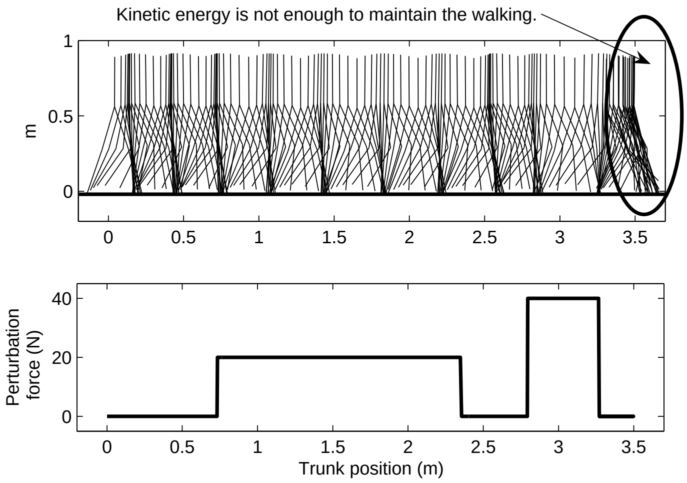


Figure 5.6: Stick-diagram of walking without the stride-frequency adjustment.

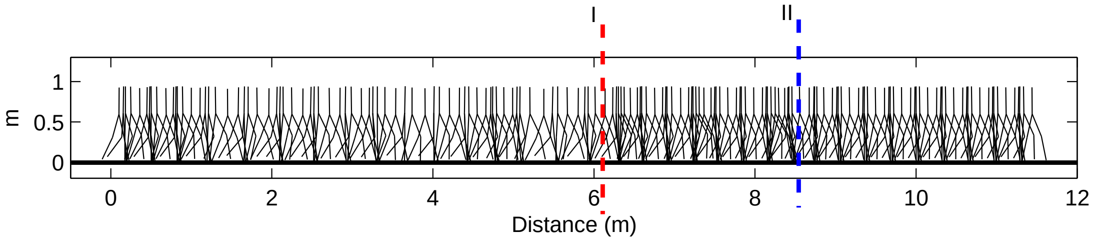


Figure 5.7: Stick-diagram of walking with online stride-frequency adjustment.Before dash line *I*, always perturbations in the positive direction. Between *I* and *II*: always perturbations in the negative direction. After dash line *II*: no perturbation.

original walking pattern. As indicated in Figure 5.8, the velocity profile of the initial several steps is the same as the velocity profile after the dash line *II*.

To further compare the performance of walking with and without the stride-frequency adjustment mode, the performances of walking under the 2*nd* batch of perturbations are compared. Figure 5.9(a) shows the resulting walking velocity and their desired average walking velocity for walking with and without the stride-frequency adjustment mode. Figure 5.9(b) shows the trunk CG position deviation from the basic walking pattern for both walking motions. It can be observed that the CG position errors of these two walking motions are quite comparable and both can be considered as small. This means the

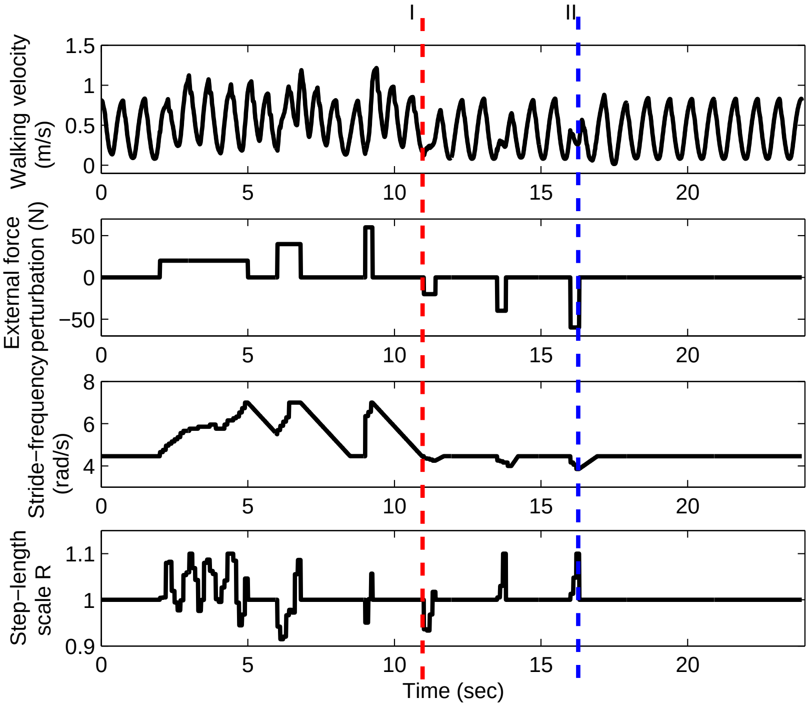


Figure 5.8: Resulting motion dynamics: walking velocity, external forces, stride-frequency, step-scale. Before dash line *I*, always perturbations in the positive direction. Between *I* and *II*: always perturbations in the negative direction. After dash line *II*: no perturbations.

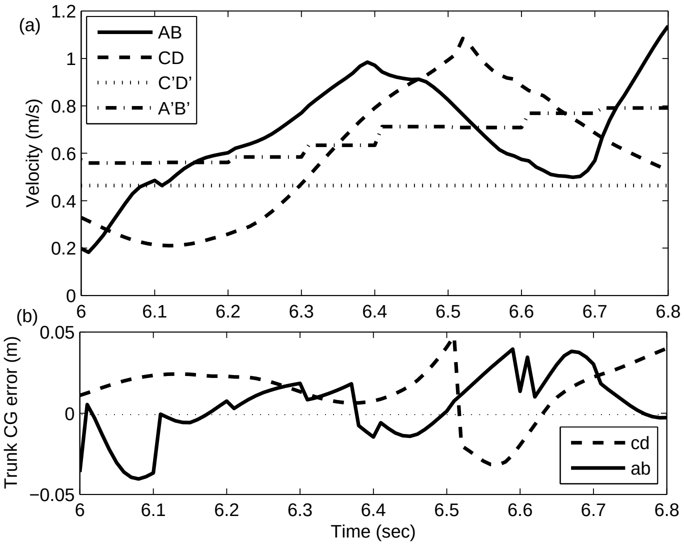


Figure 5.9: Walking velocity and trunk CG error of walking with and without stride-frequency adjustment (2nd batch of perturbation).
AB: Actual walking velocity with ω_h adjustment.
$A'B'$: Updated desired average walking velocity.
CD: Actual walking velocity without ω_h adjustment.
$C'D'$: Fixed average walking velocity.
ab: Resulting trunk CG error with ω_h adjustment.
cd: Resulting trunk CG error without ω_h adjustment.

applied joint control maintains the desired motion posture and step magnitude. However, referring to the area that curves AB and $A'B'$ enclose which shows the accumulation of the difference between the actual velocity and the desired average velocity for walking which has the stride-frequency adjustment. Similarly, the curves CD and $C'D'$ are the actual velocity and the prescribed velocity for walking without the frequency adjustment, respectively. It is observed that walking with the step-frequency adjustment has smaller area than the one which does not have the adjustment. Using the stance leg to maintain the walking posture, position error and velocity error both contribute to the final energy consumption. From the simulation, the total energy that the stance leg consumes is smaller for the case which has the step-frequency adjustment. This is also confirmed by the data as shown in Table 5.3.

It is observed that although walking without the stride-frequency adjustment passed the 1*st* and 2*nd* batches of the perturbations, the robot stance leg actually gave a relatively

Table 5.3: Stance leg energy consumption during a batch of perturbation

Perturbation No.	1*st*	2*nd*	3*rd*
Without ω_h adjustment	1344.25	422.72	*NIL*
With ω_h adjustment	1129.84	315.89	166.83

bigger effort to pull the robot back when it tended to speed up under the push forces. Then, the resulting kinetic energy is maintained by both external force perturbation and the joint torques during the motion. Therefore, once the external force was removed, the robot did not have enough actuation force to generate kinetic energy driving the motion forward. This is why after two batches of perturbation, the robot could no longer drive itself to walk forward. Nevertheless, the robustness of the control law has been shown by successfully overcoming the 1*st* and 2*nd* batches of perturbation.

However for robot walking with the stride-frequency adjustment mode, as long as the motion constraints can be satisfied, the robot will be allowed to vary its frequency in real-time to maintain an appropriate kinetic energy level for the next step. Besides, the simulation result also shows the advantages of the low gain motion control for walking in transitions and the convergence behavior of the proposed motion control strategy.

5.2 Step-length Adjustment Mode

In this section, the step-length adjustment mode is investigated. Under some circumstances, the step-length has to be adjusted in order to maintain a stable walking motion. For example, the ground may have ditches or the robot may land its swing foot earlier or later due to some environment perturbations. It can be observed from human walking gaits for walking over a ditch, as shown in Figure 5.10 recorded by *VICON* system, knee and hip joint trajectories are roughly scaled. To mimic such a behavior, the scaling parameter R of the TFS formulation (Equations (3.10) and (3.11)) is used for the adjustment.

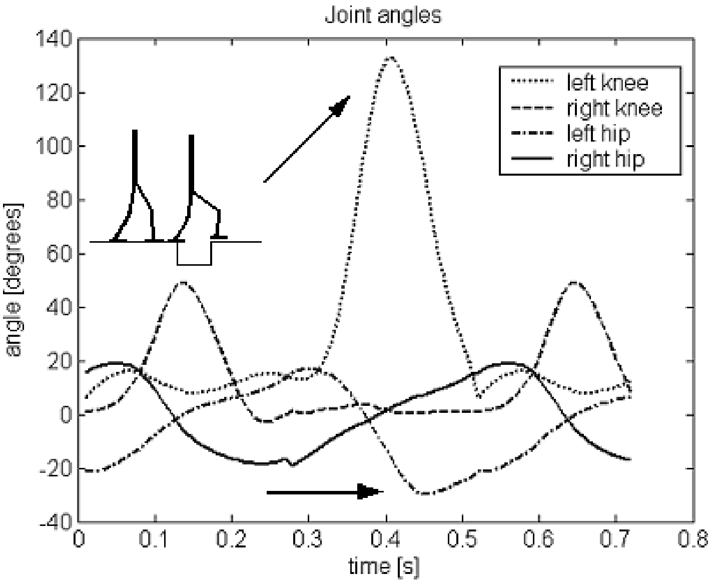


Figure 5.10: Human gaits for stepping over a ditch. Arrows indicate the motion of stepping over the ditch with both hip and knee joint trajectories roughly scaled.

Before discussing the step-length adjustment mode in detail, the stance phase-shift function is first introduced.

5.2.1 Phase-shift Function

The phase-shift function is used to connect different basic walking patterns to reduce the perturbation. The phase-shift function is described as follows:

Based upon the defined P_t, the whole period of a stance phase can be divided into n portions as $P_i(i = 1, 2 \cdots n)$. Then the phase-shift function is defined as:

$$if((P_t < P_{i+1})and(P_t > P_i)) \quad s_t = \frac{1}{n} i \cdot 0.5T$$

$$elseif(P_t < P_{min}) \quad s_t = 0;$$

$$elseif(P_t > P_{max}) \quad s_t = 0.5 \cdot T;$$

where T is the walking cycle period, P_{min} and P_{max} are the maximum and the minimum stance phase values, respectively, for the new updated basic walking pattern, s_t is the output of this phase-shift function in second.

5.2.2 Step-length Adjustment Methods

In the step-length adjustment mode, the adjustment is set at the following two time instants:

1. When the robot stance leg reaches the center line ($X = 0$).

2. When the swing foot lands on the ground.

The reason for choosing the first moment is: there is no obvious stance phase difference for different step-length walking motion at this instant. Also, the center line area has a larger stability region for transitions. Adjustment of R_I applied at this moment usually is for walking situations in which the robot is forced to change its step-length in the midst of the single support phase.

Updating the stance leg scaling parameter R_I at the second moment is aimed to reduce the stance phase error for the subsequent step, as mentioned in the above experiment for variable step-frequency walking under force perturbations.

In this subsection, two methods of step-length adjustment are discussed. These two step-length adjustment methods are respectively for the following two walking situations:

Situation 1: walking environment has no limit for the step-length.

Situation 2: walking environment has some limit for the step-length.

For situation 1, the scaling parameter of the swing leg motion R_{II} is adjusted to be the same value as the stance leg's scaling parameter R_I. This step-length adjustment method is named as step-length adjustment method 1. Since motions of the stance leg and the swing leg are equally scaled, the relative motion between the stance leg and the swing leg will not be changed with respect to the optimized basic walking motion. If the walking environment is not specified, this step-length adjustment method 1 will be the default step-length adjustment method.

Similar to the investigation of the boundary for the step-frequency adjustment, the safe range for the step-length is also obtained by the following results directly obtained from the dynamic simulations:

(1) The minimum and maximum step-length for steady-state walking motions.

(2) The largest immediate increase or decrease of the step-length without any help from the external force acting on the robot.

Figure 5.11 shows the velocity profile of a successful variable step-length flat-terrain walking with the immediate transitions between the step-length $(R_I = R_{II} = 0.5)$ to the step-length $(R_I = R_{II} = 1.2)$. It can be seen that the transitions are smooth and the resulting walking motions converge to the same steady-state motion. Furthermore, by simply varying the scaling parameter R of the basic walking pattern, the basic walking patterns of other step-length walking can be directly derived, achieving the corresponding steady-state behavior. In addition, the immediate transition is also found to be stable for up to 240% changes in step-length, showing the robustness of the basic walking patterns implemented by the proposed motion control law. Figure 5.12 shows the stick-diagram of this variable step-length walking.

The same procedure was also successfully applied to the 10^o up-slope and down-slope walking patterns obtained from the previous chapter. Based on the original step-frequency, all the obtained step-length adjustment boundaries for the flat-terrain, up-slope and down-slope walking are shown in Table 5.4. Then, based on the obtained step-length range, Table 5.4 also gives the corresponding range of the step-frequency adjustment. The wide range of the adjustable step-frequency reflects the robustness of the generated basic walking patterns.

The step-length adjustment method used for the situation 2 is named as step-length

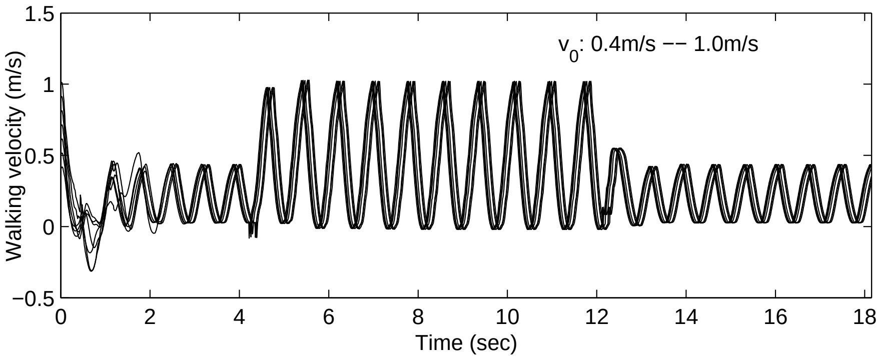


Figure 5.11: Walking velocity profile of the dynamic walking with immediate step-length transition between the largest and the smallest step-lengths.

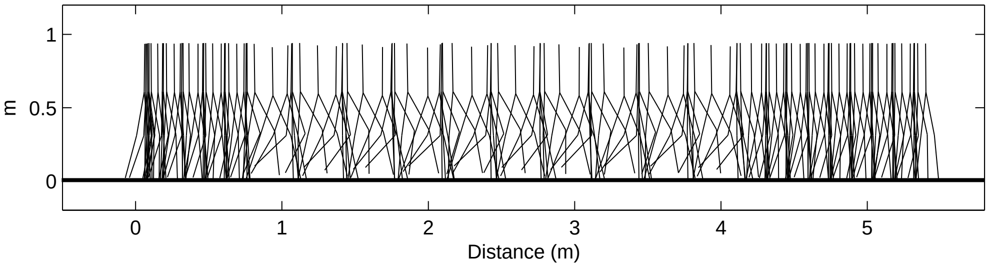


Figure 5.12: Stick-diagram of the dynamic walking with immediate step-length transition between the biggest and the smallest step-lengths.

Table 5.4: Adjustable step-length range and its min. and max. stride-frequency

	$R(min)$	$\omega_h(min)$	$\omega_h(max)$	$R(max)$	$\omega_h(min)$	$\omega_h(max)$
flat-terrain	0.5	1	8	1.2	4	5.5
10^o slope	0.5	1	7	1.3	2	5.8
-10^o slope	0.7	3	9	1.1	5.6	7

adjustment method 2. This method 2 is used when robot is suddenly forced to inhibit its motion and walk in a fixed smaller step-length. Under such a situation, the relative motion between the stance leg and swing leg should be adjusted gradually instead of scaling the bigger step motion to a smaller one immediately. This means R_I will be different from R_{II} during the adjustment, and the rule for this step-length adjustment (method 2) is given as follows:

In a one-step period, before the stance leg crosses the center line $X = 0$, the step-length adjustment method 1 will be used together with the above illustrated stance phase-shift function. After the stance leg crosses the center line, the swing leg scaling parameter R_{II} will then be adjusted by a look-up table. In this look-up table, the record is in an index of the desired step-length S_0, stance leg scaling parameter R_I, time-instant t_0 when R_I is updated, the landing time estimation $\hat{t}$ and the swing leg scaling parameter R_{II}. The following paragraph presents the generation of such a look-up table using numerical computations.

Look-up Table for Step-length Adjustment Method 2 A swing foot motion can be described by two constraints. They are the horizontal distance between of the swing foot with reference to the stance foot, $S(t)$, and the vertical distance from the bottom of the trunk to the swing foot, $H_{II}(t)$. From the kinematics $S(t)$ and $H_{II}(t)$ can be computed by Equations (5.7) and (5.8), respectively (Subscripts I and II denote the stance leg and swing leg respectively).

$$\begin{aligned} S(t) = {} & l_1 \cdot \sin\left(R_I \theta_{hI}(t)\right) \\ & + l_2 \cdot \sin\left(R_I(\theta_{hI}(t) + \theta_{kI}(t))\right) \\ & - l_1 \cdot \sin\left(R_{II}\theta_{hII}(t)\right) \\ & - l_2 \cdot \sin\left(R_{II}(\theta_{hII}(t) + \theta_{kII}(t))\right) \end{aligned} \tag{5.7}$$

$$\begin{aligned} H_{II}(t) = & \, l_1 \cdot \cos\left(R_{II}\theta_{hII}(t)\right) \\ & + l_2 \cdot \cos\left(R_{II}(\theta_{hII}(t) + \theta_{kII}(t))\right) \end{aligned} \tag{5.8}$$

θ_{hI} and θ_{hII} denote the stance hip and swing hip joint angle, respectively, and θ_{kI} and θ_{kII}, the stance knee and swing knee joint angles, respectively.

Suppose at $t = t_0$, the scaling parameter of the stance leg R_I is adjusted to R'_I as required. The scaling parameter of the swing leg R_{II} should be updated accordingly to R'_{II} to achieve a specified walking step-length. To calculate the value of R'_{II}, the estimated time $\hat{t}$ for the landing of the swing leg has to be considered. At the moment when the swing leg touches the ground, Equations (5.7) and (5.8) become (5.9) and (5.10), respectively:

$$\begin{aligned} S(t_0 + \hat{t}) = & \, l_1 \cdot \sin\left(R'_I\theta_{hI}(t_0 + \hat{t})\right) \\ & + l_2 \cdot \sin\left(R'_I(\theta_{hI}(t_0 + \hat{t}) + \theta_{kI}(t_0 + \hat{t}))\right) \\ & - l_1 \cdot \sin\left(R'_{II}\theta_{hII}(t_0 + \hat{t})\right) \\ & - l_2 \cdot \sin\left(R'_{II}(\theta_{hII}(t_0 + \hat{t}) + \theta_{kII}(t_0 + \hat{t}))\right) \end{aligned} \tag{5.9}$$

$$\begin{aligned} H_{II}(t_0 + \hat{t}) = & \, l_1 \cdot \cos\left(R'_{II}\theta_{hII}(t_0 + \hat{t})\right) \\ & + l_2 \cdot \cos\left(R'_{II}(\theta_{hII}(t_0 + \hat{t}) + \theta_{kII}(t_0 + \hat{t}))\right) \end{aligned} \tag{5.10}$$

Once the swing foot fully lands on the ground, the horizontal distance between the two legs should be approximately equal to the desired step-length, and the vertical distance from the bottom of the trunk to the swing foot should be equal to the stance height.

$$S(t_0 + \hat{t}) = S_0 \tag{5.11}$$

$$\begin{aligned} H_{II}(t_0 + \hat{t}) & = H_I(t_0 + \hat{t}) \\ & = l_1 \cdot \cos\left(R'_I\theta_{hI}(t_0 + \hat{t})\right) \\ & + l_2 \cdot \sin\left(R'_I(\theta_{hI}(t_0 + \hat{t}) + \theta_{kI}(t_0 + \hat{t}))\right) \end{aligned} \tag{5.12}$$

From Equation (5.9) to (5.12), the kinematic equations at the moment of swing foot landed on the ground are formulated as (5.13). S_0 is the desired step-length.

$$\begin{cases} l_1 \cdot \sin\left(R'_{II}\theta_{hII}(t_0+\hat{t})\right) \\ +l_2 \cdot \sin\left(R'_{II}(\theta_{hII}(t_0+\hat{t})+\theta_{kII}(t_0+\hat{t}))\right) \\ \qquad\qquad\qquad\qquad = S(R'_I,\hat{t})-S_0 \\ l_1 \cdot \cos\left(R'_{II}\theta_{hII}(t_0+\hat{t})\right) \\ +l_2 \cdot \cos\left(R'_{II}(\theta_{hII}(t_0+\hat{t})+\theta_{kII}(t_0+\hat{t}))\right) \\ \qquad\qquad\qquad\qquad = H_{II}(R'_I,\hat{t}) \end{cases} \tag{5.13}$$

where

$$\begin{cases} S(R'_I,\hat{t}) = l_1 \cdot \sin\left(R'_I\theta_{hI}(t_0+\hat{t})\right) \\ \qquad + l_2 \cdot \sin\left(R'_I(\theta_{hI}(t_0+\hat{t})+\theta_{kI}(t_0+\hat{t}))\right) \\ H_{II}(R'_I,\hat{t}) = l_1 \cdot \cos\left(R'_I\theta_{hI}(t_0+\hat{t})\right) \\ \qquad + l_2 \cdot \cos\left(R'_I(\theta_{hI}(t_0+\hat{t})+\theta_{kI}(t_0+\hat{t}))\right) \end{cases} \tag{5.14}$$

Given the values of S_0, R'_I and t_0, the solution of (5.13) will be:

$$\begin{cases} \hat{t}_1 : \begin{cases} R'_{II}(11), R'_{II}(12)\cdots \\ R'_{II}(21), R'_{II}(22)\cdots \end{cases} \\ \hat{t}_2 : \begin{cases} R'_{II}(11), R'_{II}(12)\cdots \\ R'_{II}(21), R'_{II}(22)\cdots \end{cases} \\ \vdots \\ \hat{t}_i : \begin{cases} R'_{II}(11), R'_{II}(12)\cdots \\ R'_{II}(21), R'_{II}(22)\cdots \end{cases} \end{cases} \tag{5.15}$$

where all the $\hat{t}_i$ the estimated landing period. The solutions $R'_{II}(1i), i = 1,2...$ are numerically computed by horizontal foot placement constraint and the solutions $R'_{II}(2i), i = 1,2...$ are numerically computed by the constraint of body height when both feet are on the ground. However there should be only one final solution selected from (5.13). Therefore, the next step will be the selection of the appropriate R'_{II} which gets the minimum value of $(R'_{II}(2j) - R'_{II}(1k))$, where $R'_{II}(1k)$ and $R'_{II}(2j)$ are the solutions of Equation (5.13) referring to $S(t)$ and $H_{II}(t)$, respectively. The final solution of R_{II} is in a pair with $\hat{t}$. Note that the selected value of R_{II} should also avoid any other time instant before $t = t_0 + \hat{t}$ that $H_I(R_I, t) = H_{II}(R_{II}, t)$. This motion will result in premature foot landing.

5.2.3 Variable Step-length Walking

Based on the above two step-length adjustment methods, a simulation has been conducted to demonstrate the step-length adjustment methods and the robustness of the basic walking patterns generated by the GAOFSF method. In the simulation: the robot is first required to reduce its step-length step-by-step following the sequence: $R_I = R_{II} = 1.0 \rightarrow R_I = R_{II} = 0.8 \rightarrow R_I = R_{II} = 0.65$ and then to $R_I = R_{II} = 0.5$. After this, a big external force perturbation $F = 30N$ is applied to the robot for $0.5s$. Then, the robot is forced to walk with the largest step-length $R_I = R_{II} = 1.3$. Right after this, the robot will be required to fix its step-length to $0.3m$ and walk until the moment that R_I is again equal to R_{II} and then adjust the step-length to be ($R_I = R_{II} = 1.1$) for the final steady-state walking. This inclusion of gradual and sudden changes is to test the robustness and adaptivity of the variable step-length walking motion.

In this simulation, the adjustment for the first 5 steps is based on the method 1. Thereafter method 2 is applied until the step with equivalent scaling parameter R_I and R_{II} occurs. From there, method 1 will again be used to achieve the final steady-state motion. Table 5.5 shows part of the look-up table that is applicable for this simulation.

Figure 5.13 shows the resulting data of the variable step-length walking motion. It is observed that the step-length adjustment method 1 successfully reduced the walking step-length through the initial four adjustments. Also, from the walking velocity profile, method 1 is observed to effectively reduced the kinetic energy with the reduction of the assigned step-length. Given the sudden kinetic energy injection by the external force ($30N$ for $0.5s$), the subsequent large step was found to successfully maintain the stable

Table 5.5: Part of the look-up table

S_0	0.3	0.3	0.3	0.3	0.3
t_0	0.01	0.01	0.01	0.01	0.01
R'_I	1.25	1.17	1.08	1.0	0.9
R'_{II}	0.85	0.8438	0.8563	0.865	0.9
$\hat{t}$	0.3	0.32	0.35	0.38	0.44

motion. The gradual transition for the following fixed step-length walking was also shown to be stable and smooth. After the big step, the step-length of the subsequent 5 steps were seen to be well constrained to be $0.3m$ using the step-length adjustment method 2 and the walking velocity is gradually reduced. Although method 1 may also be applicable for this step-length reduction to $0.3m$, the sudden motion inhibition given by equally adjusting R_I and R_{II} might result in motion instability. Also, the resulting motion may not be smooth.

After all the transitions, walking is observed to converge to the basic walking motion with $R_I = R_{II} = 1.1$. Figure 5.14 shows the stick-diagram of this simulated trial.

From this study, both step-length adjustment method 1 and method 2 have been shown to be effective for the sagittal plane walking adjustment. The initial few steps using method 1 shows the robustness of the limit cycle behavior excited by the GAOFSF generated basic walking pattern. Then, the gradual release of the kinetic energy shows the step-length adjustment method 2 is applicable for smooth and stable transitions.

5.3 Leg Pattern Adjustment Mode

The above sections described the real-time walking speed adjustment modes (walking speed can be varied either by the stride-frequency or the step-length). In this section, real-time pattern adjustment for lower extremities' motion, which can be used for walking on undulating terrain is discussed.

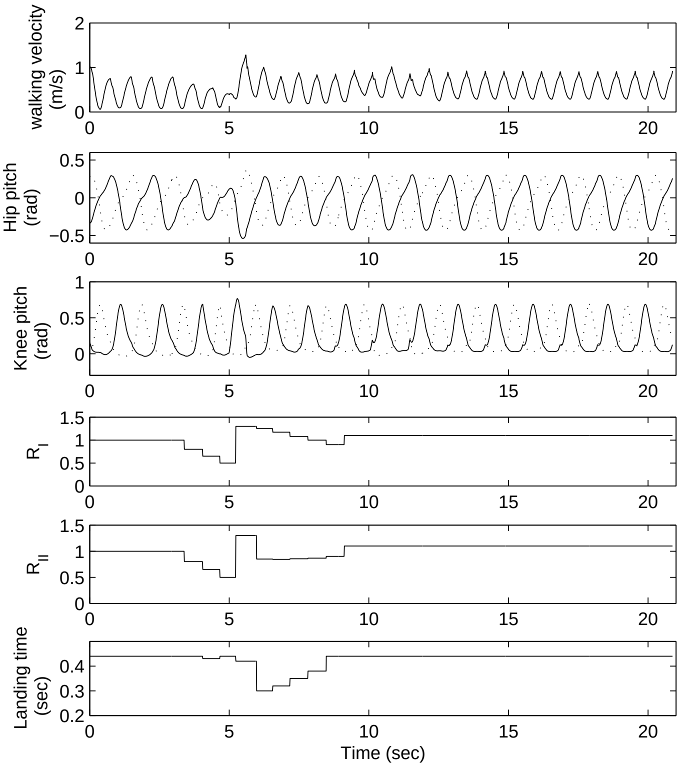


Figure 5.13: Results of the dynamic walking with the step-length online adjusted.

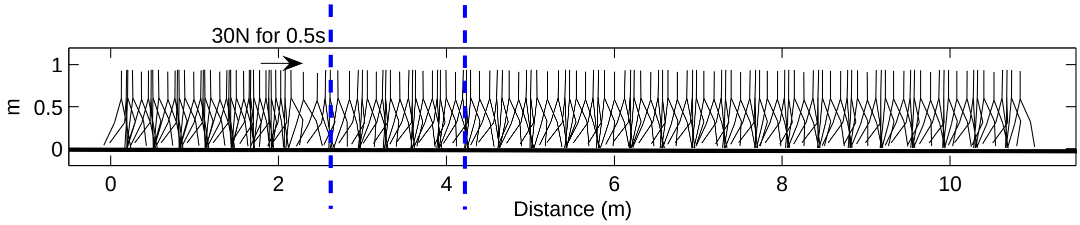


Figure 5.14: Stick diagram of the resulting motion pattern of the simulation trial.

In the TFS formulation, c_h indicates the point where the hip joints coincide with each other. This intersection point is dependent on the terrain profile. c_k is the knee joint angle during the stance lock phase. This value will also be dependent on the terrain profile, as highlighted by human gait analysis [84]. Therefore, c_h and c_k values can be considered as key parameters for walking on undulating terrains.

Inspired by human gaits on rough terrains, the relative motion between the two hips and that between the thigh-shank will also be maintained the same as that of the basic optimized walking motion. Therefore, $(A_i, B_i, C_i, t_1, t_2)$ is not adjusted in the TFS formulation for walking on undulating terrains.

Hence, tuning c_h and c_k values while keeping the values of $(A_i, B_i, C_i, t_1, t_2)$ fixed, is adopted as the strategy for adjusting the walking gaits on some irregular terrain. To optimize walking motions on slope terrains other than the 10^o and -10^o slopes by the TFS parameters (c_h, c_k), the soft motion constraint: step-length constraint, is released. (Note, step-length can be always adjusted by the scaling parameters individually). If necessary, to further improve the motion stability, a trunk pitch angle can be varied. Here, the body trunk's pitch is still set at zero.

For a target range of slopes, i.e., $[2^o, 15^o]$ and $[-15^o, -2^o]$ that the robot can walk stably, the optimized 10^o up-slope and down-slope motions are selected as the basic up- and down-slope walking motions. In the GAOFSF environment, (c_h, c_k) can be searched by the GA for slope range $[2^o, 15^o]$ or $[-15^o, -2^o]$. Figure 5.15 shows the relationship between the optimized values of $[c_h, c_k]$ versus the terrain slope (up-slope). It can be seen that with the increase of the gradient of the up-slope terrain, the motion's hip offset value c_h decreases monotonously. However, the knee joint angle in the lock phase c_k is increased. These behaviors resemble the human walking behaviors such as: 1) two hips cross each other in a more positive angle position when the up-slope gets steeper; 2) knee joint bends lesser on gentle slopes but more on steeper ones. Simple regressions using the cubic function can be adopted for the relation between (c_h or c_k) and the ground inclination, as shown in Equation (5.16), for real-time parameter adjustment:

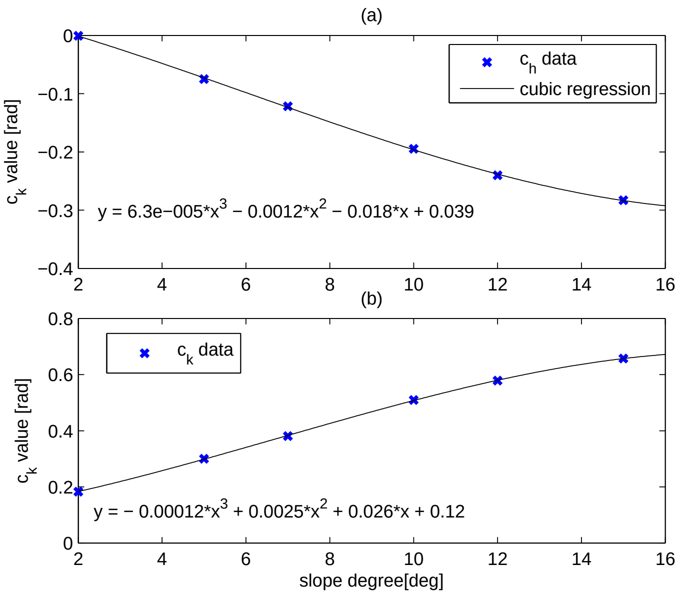


Figure 5.15: Function regression for climbing up: (a) c_h and (b) c_k.

$$\begin{cases} c_h = 6.3e^{-5}\alpha^3 - 1.2e^{-3}\alpha^2 - 1.8e^{-2}\alpha + 0.039 \\ c_k = -1.2e^{-4}\alpha^3 + 2.5e^{-3}\alpha^2 + 2.6e^{-2}\alpha + 0.12 \end{cases} \tag{5.16}$$

where α is the slope gradient. Residuals of these two regressions are 3.8803×10^{-3} and 3.8512×10^{-3}, respectively, which are considered to be very small.

For down-slopes, the hip offset value c_h is observed to increase when the slope becomes steeper. However, the change is just within a range of $[0, 0.035]rad$, as shown in Figure 5.16(a). Compared to the hip joint, the lock phase angle c_k of the knee joint decreases much more obviously for walking on uneven slopes, as shown in Figure 5.16(b). The change of c_k is in a range of $[0, 0.2]rad$ which is almost 70% more than the variation of the c_h value. Therefore, the knee joint motion is found to be very important for giving a stable locomotion on down-slope terrains. From the results obtained, walking patterns on various down-slopes are seen to resemble the human walking gaits, particularly the behavior that the knee will be bent more for going down a steeper slope but less on gentle slopes. The regressed functions of c_h and c_k versus the down-slope gradients are

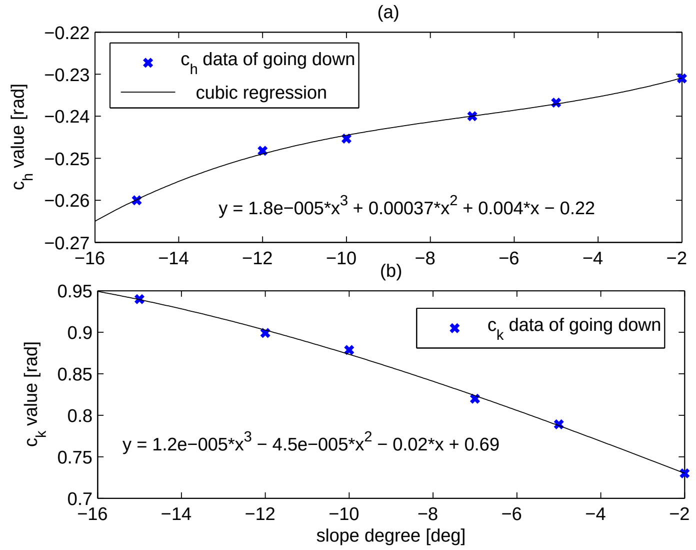


Figure 5.16: Function regression for going down: (a) c_h and (b) c_k.

given by Equation (5.17). Residuals of the c_h and c_k regressions are 1.0917×10^{-3} and 7.5059×10^{-3}, respectively.

$$\begin{cases} c_h = 1.8e^{-5}\alpha^3 + 3.7e^{-4}\alpha^2 + 4.0e^{-3}\alpha - 0.22 \\ c_k = 1.2e^{-5}\alpha^3 - 4.5e^{-5}\alpha^2 - 2e^{-2}\alpha + 0.69 \end{cases} \tag{5.17}$$

Based on the achieved basic walking patterns for different up-slopes and down-slopes, the following subsection presents the control results of walking on undulating terrains. The aforementioned stance phase-shift function is again incorporated for terrain adaptation.

5.3.1 Dynamic Simulations of Undulating-terrain Walking

This subsection gives a series of walking demonstrations on uneven terrains using the developed leg pattern adjustment mode. The followings are the uneven-terrain walking simulations that have been conducted:

1) Walking up from $0^o \rightarrow 10^o \rightarrow 5^o \rightarrow 15^o$.

2) Walking up from $15^o \rightarrow 10^o \rightarrow 15^o \rightarrow 0^o$.

3) Walking down from $0^o \rightarrow -10^o \rightarrow -15^o \rightarrow -5^o \rightarrow -10^o \rightarrow 0^o$.

In the implementation, the robot swing-foot ankle joint will be loosened so as to allow the foot to adjust to the new terrain surface and detect the new slope once the toe or heel of the swing foot touches the ground. Here, the slope sensed by the robot is computed by Equation (5.18):

$$\alpha = atan(\frac{z_{II} - z_I}{x_{II} - x_I}) \tag{5.18}$$

where x and z are the actual position of the ankle joint.

The walking velocity profile of the above three undulating-terrain walking motions are shown in Figure 5.17(a), (b) and (c), respectively. It can be seen that walking can always converge to a stable steady-state motion on different slopes. The transitions are observed to be quick and stable showing the robustness of the limit cycle behavior excited by the basic walking pattern.

Stick-diagrams of the rough terrain walking simulations (1), (2) and (3) are shown in Figure 5.18, Figure 5.19 and Figure 5.20 respectively. Figure 5.21 to Figure 5.23 show the key data of the achieved walking motions. The change of the joint trajectories can be clearly observed while the stable gaits are also achieved for walking on all the terrains. This demonstrates the robustness of the generated basic walking patterns, especially for the up-slope x_{up} and down-slope x_{down} motions generated in Chapter 4.

5.4 Summary

Based on the results obtained from the three developed basic motion adjustment modes, the following human-like behaviors have been achieved:

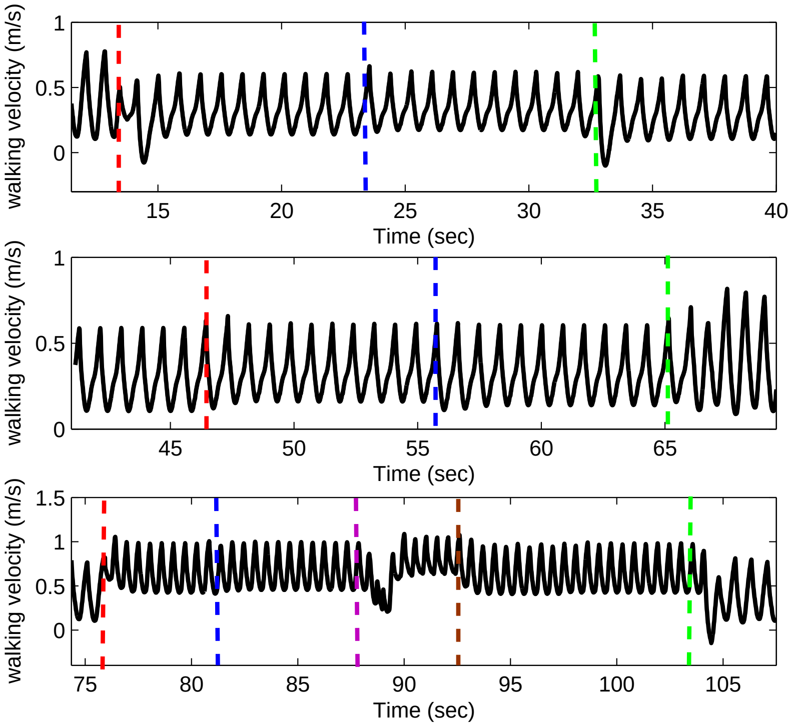


Figure 5.17: Walking velocity profile of the walking motions on the rough-terrain 1,2,3, respectively for the 1st, 2nd and 3rd plots. All the dash lines indicate when the terrain slope is changed.

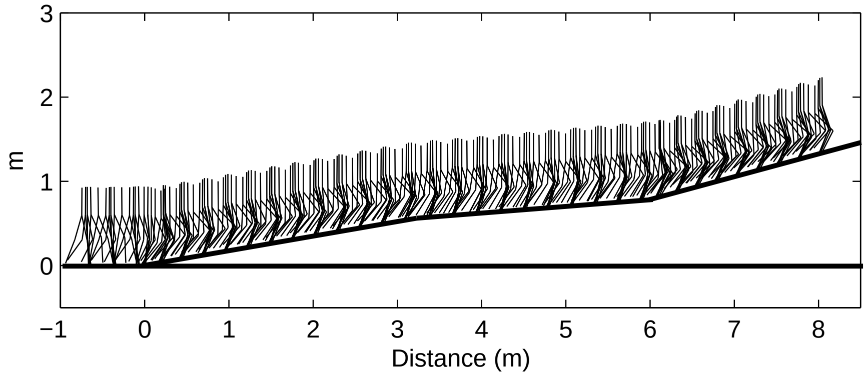


Figure 5.18: Stick diagram of walking on the rough-terrain 1.

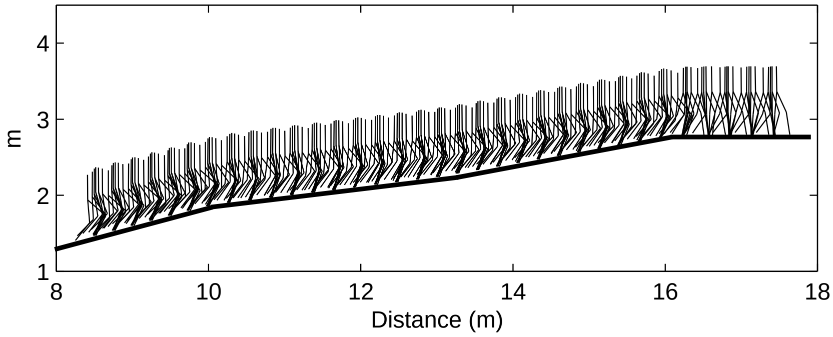


Figure 5.19: Stick diagram of walking on the rough-terrain 2.

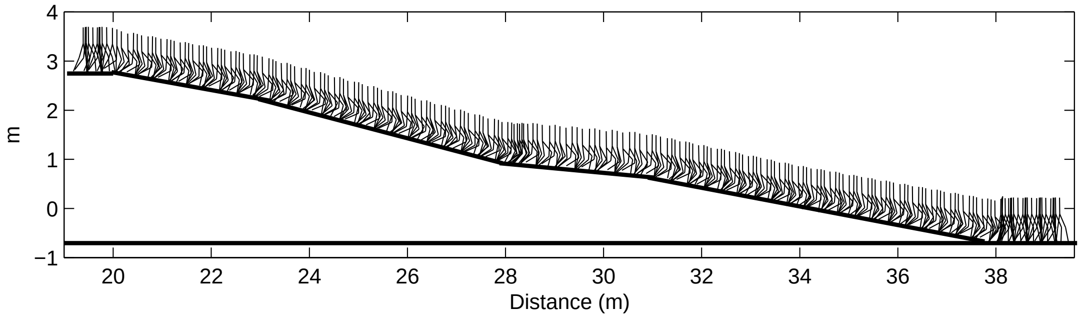


Figure 5.20: Stick diagram of walking on the rough-terrain 3.

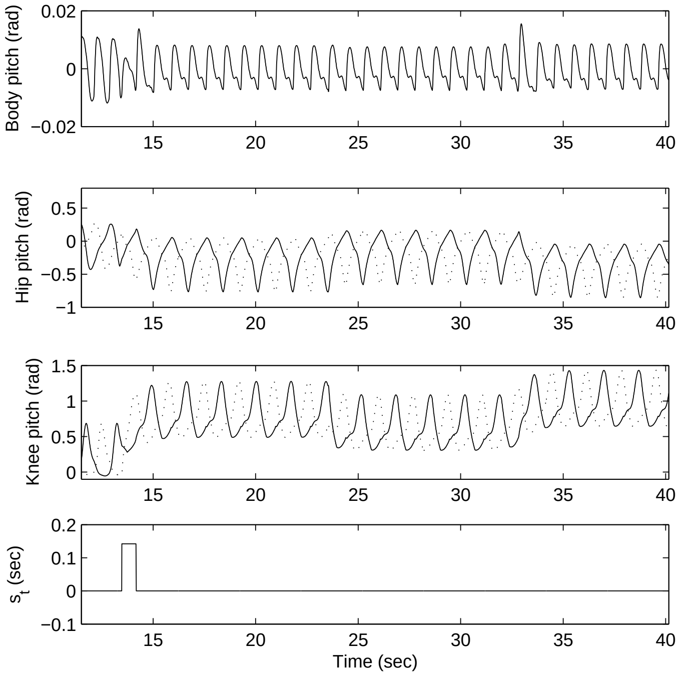


Figure 5.21: Dynamics of walking on the rough-terrain 1.

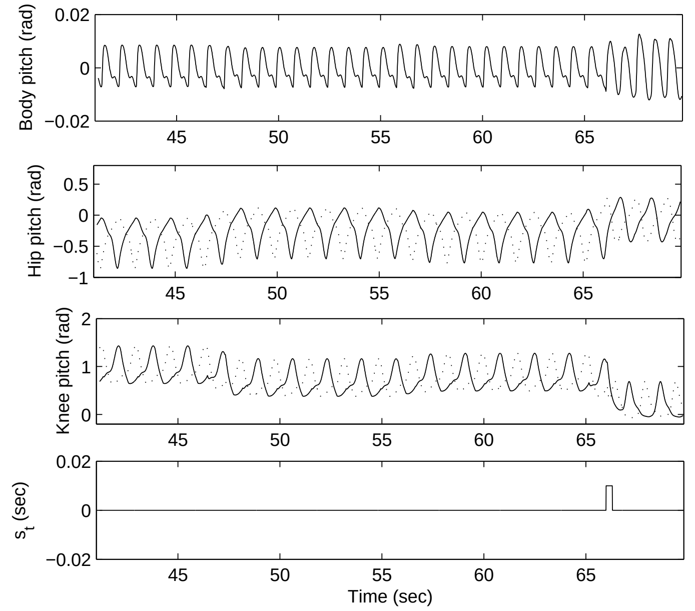


Figure 5.22: Dynamics of walking on the rough-terrain 2.

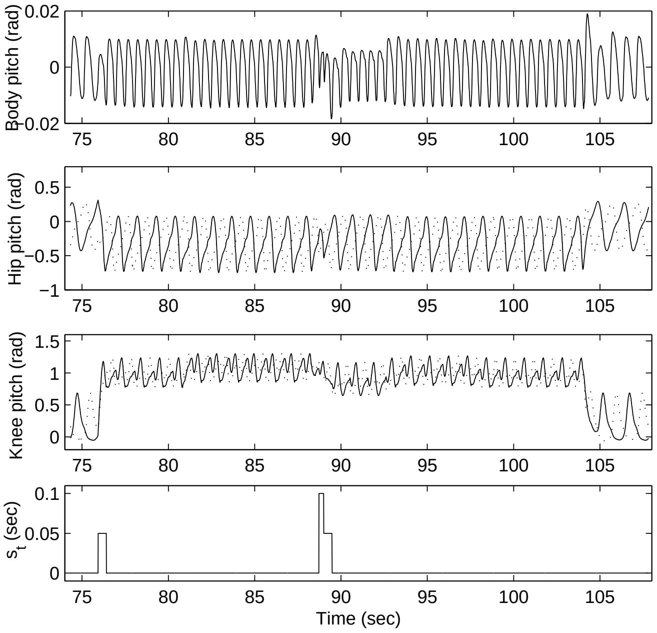


Figure 5.23: Dynamics of walking on the rough-terrain 3.

1) The human-like rhythmic walking motion with environment entrainment has been demonstrated. Figure 5.24 shows the overall control architecture.

It can be seen that the basic walking pattern $B(t)$ generated by the GAOFSF method is comparable to a central pattern stored in the spinal cord (low-level). For walking on any fixed sloped terrain, in order to get the motion into a stable walking steady-state, it may not be necessary to incorporate any high level modules such as step-frequency adjustment, step-length adjustment or terrain adaptation. It is because the derived basic walking pattern and motion controller presented in Chapter 4 can ensure stable walking provided the perturbations are not significant. The high-level modification for the basic walking pattern $B(t)$ is required only when the robot gets some sensory feedback that the external force perturbations, ground ditches, or terrain adjustment occurs. In a certain way, the high-level motion control which modifies the basic walking pattern through the TFS parameters $(\omega_h, R_I, c_h, c_k,)$ is analogous to how the brain moderates the walking motion.

2) The human-like up-slope and down-slope motions can be achieved by the proposed

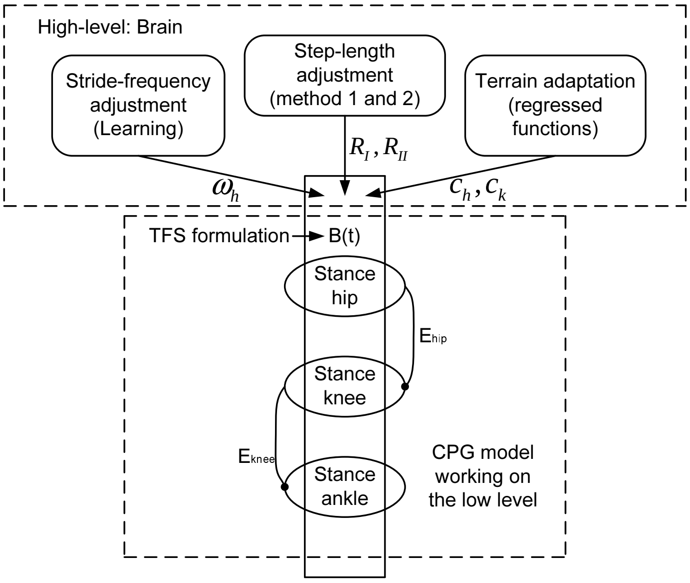

High-level: Brain
Stride-frequency adjustment (Learning)
Step-length adjustment (method 1 and 2)
Terrain adaptation (regressed functions)
R_I, R_{II}
ω_h
c_h, c_k
TFS formulation
B(t)
Stance hip
E_{hip}
Stance knee
E_{knee}
Stance ankle
CPG model working on the low level

Figure 5.24: Two-level walking network: low-level: CPG model, high-level: Brain.

Table 5.6: The resulting fastest and slowest walking

Waking speed	fastest	slowest
Flat-terrain	$0.95m/s$	$0.16m/s$
10^o slope	$0.78m/s$	$0.1m/s$
-10^o slope	$1.01m/s$	$0.3m/s$

approach: The stance leg tends to be straighter for gentle slopes but bents more on steeper slopes. Such a trend shows that the knee joint motion is correlated to the gradient of the terrain, and the motion stability is more associated with the knee-joint motion, especially for down-slope motions.

Furthermore, faster speed walking can be achieved by changing the stride-frequency and the step-length concurrently. Table 5.6 gives the fastest and slowest walking speeds achievable for stable locomotion on different terrains in the dynamic simulation environment. Since the walking down motion is the most challenging, Figure 5.25 and Figure 5.26 show the stick-diagrams of the walking-down motions on the -10^o slope with the minimum and maximum speeds, respectively. It can be seen that compared to the lowest

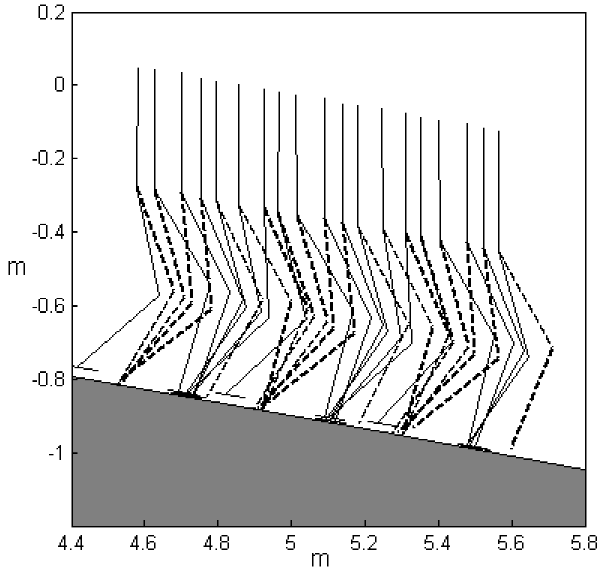


Figure 5.25: Smallest pace 10^o down-slope walking.

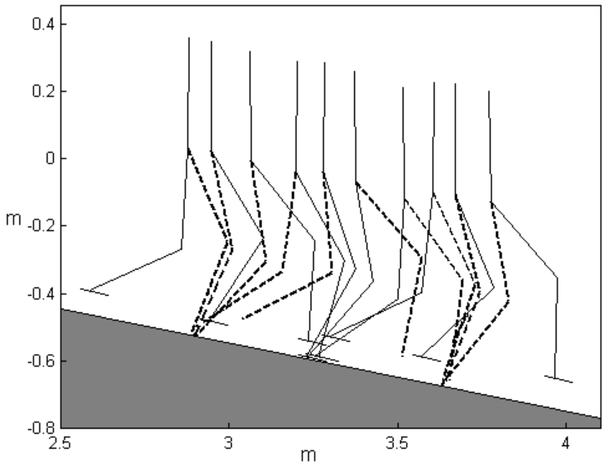


Figure 5.26: Biggest pace 10^o down-slope walking.

speed walking motion, the highest speed walking motion swings much higher using a bigger stride. Here, the sampling time of the slowest walking and the fastest walking motions are $0.15s$ and $0.1s$ respectively.

Based on all the results obtained in Chapter 4 and Chapter 5, the robustness of the developed sagittal plane motion control algorithm is demonstrated. The next Chapter will then explore the application of the TFS formulation for 3D bipedal walking control especially for the lateral motion balance behavior.

Chapter 6

Frontal Motion Balance Strategy 1

In comparison with those for sagittal plane motion control, much fewer strategies for frontal plane motion control have been proposed. Goddard et al.[81], and Hemami and Wyman [82] used a three-link planar model to study the frontal plane motion of a biped. Iqbal et al.[83] presented a model for studying the involvement of the central nervous system in the execution of voluntary movements in the frontal plane. In this chapter, the proposed motion control law based on the reaction torque and the TFS motion generation model have been further investigated for its application in lateral motion balance control for 3D walking locomotion. Here, the frontal plane motion is separately considered assuming that the motion in the three orthogonal planes (sagittal, frontal and transverse) are weakly coupled. The sagittal plane motion is still maintained human-like (stance leg is straightened during part of the stance phase), generated by the GAOFSF method presented in the previous chapters.

6.1 Joint Control Scheme for 3D Walking

Similar to the 2D motion control strategy, the 3D walking control algorithm is also aimed to maintain the robot body to be upright. In the meantime, the stance foot is not planned so as to naturally interact with the ground. The joint control method presented in Chapter 4 is generalized to the 3D space.

First, a notation for variables used in the joint controller formulations is given,

1) Subscripts 1 and 2 refer to the stance and swing legs, respectively.

2) Subscripts *h*, *k*, *a*, *b* denote the hip, knee, ankle and body. respectively.

3) X and Y represent the sagittal plane and frontal plane, respectively.

3) Subscripts *p*, *r*, *y* refer to the pitch, roll and yaw, respectively.

4) τ is the generated torque applied to the joints.

5) k_p and k_d are the proportional and derivative gains for the joint controllers, respectively.

Based on the joint control for 2D walking, the sagittal plane motion control for 3D walking is generalized as:

$$Body\,Control:\begin{cases}-R_p(T_{1_{hp}}+T_{2_{hp}})_X-R_r(T_{1_{hr}}+T_{2_{hr}})_X=PD_{bp}\\ PD_{bp}=k_{p_b}(\hat{\theta}_{bp}-\theta_{bp})+k_{d_b}(\dot{\hat{\theta}}_{bp}-\dot{\theta}_{bp})\\ R_p=R_{hz}R_{hx}R_{hy},\quad R_r=R_{hz}R_{hx}\\ T_{ihp}=[0\ \tau_{ihp}\ 0]^T,\quad T_{hr}=[\tau_{ihr}\ 0\ 0]^T,\quad i=1,2\end{cases}\tag{6.1}$$

$$Stance\,Leg\,Control:\begin{cases}\tau_{1_k}=-k_{p_{k1}}(\hat{\theta}_{1_h}-\theta_{1_h})-k_{d_{k1}}(\dot{\hat{\theta}}_{1_h}-\dot{\theta}_{1_h})\\ \tau_{1_{ap}}=-k_{p_{a1}}(\hat{\theta}_{1_k}-\theta_{1_k})-k_{d_{a1}}(\dot{\hat{\theta}}_{1_k}-\dot{\theta}_{1_k})\end{cases}\tag{6.2}$$

$$Swing\,Leg\,Control:\begin{cases}\tau_{2_{hp}}=k_{p_{h2}}(\hat{\theta}_{2_h}-\theta_{2_h})+k_{d_{h2}}(\dot{\hat{\theta}}_{2_h}-\dot{\theta}_{2_h})\\ \tau_{2_k}=k_{p_{k2}}(\hat{\theta}_{2_k}-\theta_{2_k})+k_{d_{k2}}(\dot{\hat{\theta}}_{2_k}-\dot{\theta}_{2_k})\\ \tau_{2_{ap}}=k_{p_{a2}}(\hat{\theta}_{2_a}-\theta_{2_a})+k_{d_{a2}}(\dot{\hat{\theta}}_{2_a}-\dot{\theta}_{2_a})\end{cases}\tag{6.3}$$

where R_{hz}, R_{hy}, R_{hx} are the rotational matrices along the *z*, *y* and *x* axis, respectively.

The body's pitch is controlled to the desired position through Equation (6.1). Here, the desired body-pitch is set at zero as a vertical position. The controllers of the other joints of the stance leg are formulated in Equation (6.2). The joints of the swing leg are controlled by the conventional PD position controller, formulated in Equation(6.3).

Similar to the joint control for the sagittal plane, the reaction torque is also used for the frontal plane motion control. The desired body's roll motion is set at zero using the controller formulated in Equation (6.4). The other controllers applied to the stance leg on the frontal plane are formulated as Equation (6.5). The swing leg control still uses the conventional PD position controller formulated as Equation (6.6). Here, since the two legs of the target robot are rather close to each other, the swing leg motion is maintained to be parallel with the stance leg motion avoiding any hit between the two legs during walking.

$$Body\ Control: \begin{cases} -R_p(T_{1_{hp}}+T_{2_{hp}})_Y - R_r(T_{1_{hr}}+T_{2_{hr}})_Y = PD_{br} \\ PD_{br} = k_{p_{br}}(\hat{\theta}_r - \theta_r) + k_d(\dot{\hat{\theta}}_r - \dot{\theta}_r) \\ R_p = R_{hz}R_{hx}R_{hy}, \quad R_r = R_{hz}R_{hx} \\ T_{ihp} = [0\ \tau_{ihp}\ 0]^T, \quad T_{hr} = [\tau_{ihr}\ 0\ 0]^T, \quad i = 1,2 \end{cases} \tag{6.4}$$

$$Stance\ Leg\ Control: \quad \tau_{1_{ar}} = -k_{p_{ar1}}(\hat{\theta}_{1_{h_r}} - \theta_{1_{h_r}}) - k_{d_{ar1}}(\dot{\hat{\theta}}_{1_{h_r}} - \dot{\theta}_{1_{h_r}}) \tag{6.5}$$

$$Swing\ Leg\ Control: \begin{cases} \tau_{2_{hr}} = k_{p_{h2_r}}(\hat{\theta}_{2_{h_r}} - \theta_{2_{h_r}}) + k_{d_{h2_r}}(\dot{\hat{\theta}}_{2_{h_r}} - \dot{\theta}_{2_{h_r}}) \\ \tau_{2_{ar}} = k_{p_{ar2}}(\hat{\theta}_{2_{a_r}} - \theta_{2_{a_r}}) + k_{d_{ar2}}(\dot{\hat{\theta}}_{2_{a_r}} - \dot{\theta}_{2_{a_r}}) \end{cases} \tag{6.6}$$

Note the torques involved in the stance leg control and the swing leg control can be directly computed, but the torques applied to the body control are coupled in 3D space. Aligning the Equations (6.1) and (6.4), torques $\tau_{1_{hp}}$ and $\tau_{1_{hr}}$ can then be computed.

For the transverse plane motion, the aim is just to maintain the two hip joints' yaw at the desired value which is also set at zero. Therefore, the joint control is formulated as

Equation (6.7):

$$\begin{cases} \tau_{1_{hy}} = k_{p_{hy}}(\hat{\theta}_{1_{hy}} - \theta_{1_{hy}}) + k_{d_{hy}}(\dot{\hat{\theta}}_{1_{hy}} - \dot{\theta}_{1_{hy}}) \\ \tau_{2_{hy}} = k_{p_{hy}}(\hat{\theta}_{2_{hy}} - \theta_{2_{hy}}) + k_{d_{hy}}(\dot{\hat{\theta}}_{2_{hy}} - \dot{\theta}_{2_{hy}}) \end{cases} \tag{6.7}$$

6.2 TFS Formulated Lateral Motion Optimization

Compared to the basic walking pattern planned for the sagittal plane motion, the major difference of the frontal plane motion are summarized as:

1) The overall angular momentum along the X-axis accumulated over one step-period can not be compensated. For example, it will be always negative for the step that the right leg is the stance leg.

2) Due to the change of direction of the lateral movement, the lateral walking velocity in the basic walking pattern cannot be planned to be uniform.

Due to the differences of the motion behavior, the low gain motion position control may not be applicable for the frontal motion to converge to the lateral steady-state pattern. Nevertheless, it can be seen that the formulated control law still aims to maintain the desired walking pattern through achieving the relative joint motions. Then, given higher and more stiff motion control gains, the response of the formulated motion control will be just similar to the reaction torque based PD position control tracking the absolute joint motion based on the world coordinate. Thus, the desirable lateral motion characters for the generation of the frontal plane basic walking pattern is directly based on the typical lateral motion behaviors listed as the follows:

1) The lateral velocity is supposed to be the highest as it passes through the middle body's position between the two legs.

2) The body's motion should be symmetrical about the middle position, and

3) To avoid undue force perturbations, the body's trajectory should be continuous in its

first and second derivations during the single leg supporting phase.

Based on the above basic motion characters, the lateral motion reference is thus formulated as (6.8) in the form of a TFS:

$$\theta_r = R\Sigma_{i=1}^{k} D_i sin(i\omega t) + c, \quad k = 1,2,3..., \quad \omega = \pi/T \tag{6.8}$$

where θ_r is the desired hip joint's roll trajectory, D_i the amplitude value of the series, R the scaling factor for the motion reference in the frontal plane, ω the walking stride-frequency, T the step-period or the time taken to complete a single step, and c a constant attached with the TFS formulation. For symmetrical motions, c is set to be zero assuming the motion is well tracked.

Again, the GAOFSF method [53] is used for 3D walking motion generation and optimization. The performance optimization index for the frontal plane motion is f_4 (Code: Op_2)given in Chapter 3, which minimizes the accumulation of the stance ankle's torque during the single support phase. The performance objective functions for the sagittal plane motion are exactly the same as for that for the previous 2D space walking motion. The detailed set-up of the GAOFSF method is shown in Table 6.1 using the optimization code: Op_2 which optimizes the sagittal plane and frontal plane motions concurrently.

To test the generality of the GAOFSF method and the proposed motion control law, the target robot for 3D walking algorithms is selected as the Fujitsu HOAP-I robot, whose mechanical properties are quite different from the NUSBIP robot. The GAOFSF generated walking solution, $x = [D_i, A_i, B_i, C_i, c_h, c_k, t_1, t_2]$, based on this HOAP-I robot is given as follows:

Table 6.1: GA Set-up for 3D walking

Description	Remark
Objective components	Code Op_2: f_1, f_2, f_3, f_4
Constraint components	Code Op_2: $s_1 - s_5$
Chromosome representation	real-valued GA
Initial population M	150
Generation number T	250
Crossover operators	heuristic crossover simple crossover arthritic crossover
Mutation operators	multi-non-uniform mutation Uniform mutation Boundary mutation
Weights for objectives	$w_i = [50, 5, 10, 100]$
Weights for constraints	$p_i = [15, 80, 20, 30, 800]$

$$\begin{aligned} x_{HOAP} = [&0.2298\ -0.0002\ -0.0042\ 0.0003\ 0.0001\ 0.2336\ -0.0639 \\ &0.0095\ 0.0035\ -0.0021\ -0.3338\ -0.0791\ 0.0159\ 0.0103 \\ &0.0053\ 0.4676\ 0.1932\ -0.006\ -0.0369\ -0.0412 \\ &0.000\ 0.000\ 0.001\ 0.430] \end{aligned} \tag{6.9}$$

It can be seen from the results obtained that the frontal motion can, without much loss of accuracy, be represented by a pure sinusoidal profile, as the amplitudes of the 2*nd* to the 5*th* order components were found to be relatively small compared with that of the fundamental. The sagittal walking posture is generated as shown in Figure 6.1 for the above 3D walking motion. It can be noted, the human-like motion pattern for the

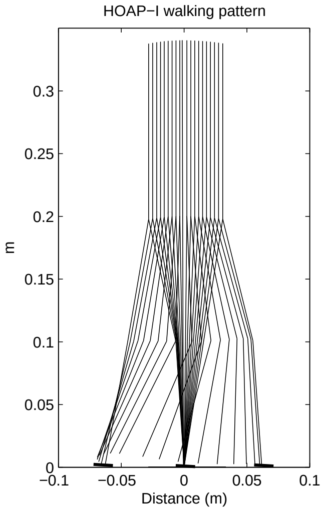


Figure 6.1: The generated HOAP-I's human-like basic walking pattern in the sagittal plane.

HOAP-I robot is shown to be similar to the 2D human-like walking posture for the NUSBIP-I robot, as shown in Figure 4.6(f).

6.2.1 3D Walking Control Results

Given suitable motion trajectories have been successfully generated using the GAOFSF method, the obtained motion trajectories were then applied to the HOAP-I robot directly in the dynamic simulation environment to check the dynamic behavior of the robot. The dynamic simulation takes into account dynamic behaviors such as the robot's inertia and ground contact forces.

The results of the dynamic simulation show that such a direct implementation of the walking motion can only achieve walking up to a certain distance (about 15 steps continuously for a human-like walking pattern), and the failure mainly comes from the frontal plane motion control. Figure 6.2 shows the resulting motion of the HOAP-I robot projected to the sagittal plane. Good walking posture was maintained, as compared with that in Figure 6.1, with the body properly controlled and maintained in an upright position, and the stance foot being stable on the ground. The resulting dynamics are shown

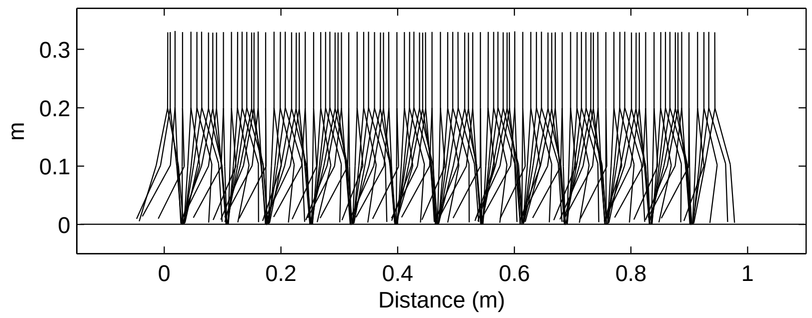


Figure 6.2: Stick-diagram of Hoap-I robot during 3D walking without reference adjustment.

in Figure 6.3. It can be noted from Figure 6.3 that, except for the hip roll, the rest of the joint trajectories are shown to be very periodical. This indicates that the generated motion pattern achieved a good motion coordination for the sagittal motion plane. For the hip roll in the frontal motion plane, while the shape of the resulting trajectory (solid line) follows closely the reference trajectory (dashed line), there is a certain difference in the amplitude of the motion.

In addition, particularly checking the forward walking velocity profile of the 3D walking motion, it is found that the velocity profile is no longer just shaped as a ”U” profile, first reduced and then increased, as shown in Figure 6.4. Instead, the velocity profile contains more terms of fluctuation, but the distribution is shown to be comparable before and after crossing the center line $X = 0$, as indicated at the marked step-motion. The solid line shows the velocity when robot body is located at the center line $X = 0$, and the dash lines indicate the walking velocity at the touch-down moment. Here, the main reason for the unnatural walking velocity profile because when the formulated control is applied to the frontal plane motion control, the applied joint control gains have to be higher and more stiff to maintain the desired lateral motion. Then, in order to match with the frontal plane motion, the sagittal motion should also be given higher and more stiff control gains. Since the sagittal plane motion is generated according to the same motion objective functions for the 2D walking motion, the generated sagittal pattern will be rather symmetrical referring to the center line $X = 0$ with the uniform walking velocity. Therefore, even if the higher control gains caused the motion velocity more fluctuated, the resulting velocity profile still appeared to be comparable for the swing-in and swing-out phases.

Because the velocity profile, although more fluctuated, is still comparable before and after the center line $X = 0$, the least deviated motion pattern can be still identified as a step motion whose end dynamics are close to the initial dynamics. Here, although the motion stability of the sagittal plane motion is still maintainable, the higher control gains already make the resulting motion rather close to the planned basic walking pattern. Note here the basic walking pattern is used to coordinate the real motion but itself is not a robust motion as the ZMP will be rather overlapped with the CG position when the walking velocity is optimized to be uniform. Due to the short step-length given to the above motion example, the ZMP of the planned motion can be still confined in the supporting polygon. Therefore, applying an initial condition close to the planned dynamics is still able to maintain the sagittal motion stability referring to the dynamics as shown in Figure 6.3.

However, compared to the sagittal motion, the frontal motion plane is more difficult to control as the angular momentum in the frontal motion plane cannot be compensated during a one step motion. Also, the acceleration of the basic lateral motion pattern is rather major. Therefore, motion failure eventually occurs to the frontal motion plane.

Based on the obtained 3D walking results, it can be seen that a fixed basic walking pattern given by the GAOFSF motion generation and optimization is applicable for the 3D walking control to some extent. The joint controllers are shown able to resemble the motion pattern given suitable control gains. However, the resulting frontal plane motion is not robust enough to achieve the long distance walking. Therefore, in order to achieve a prolong 3D walking motion, some kind of frontal motion adjustment strategy must be incorporated. The following sections present a strategy which online stabilizes the walking motion based on the TFS formulation.

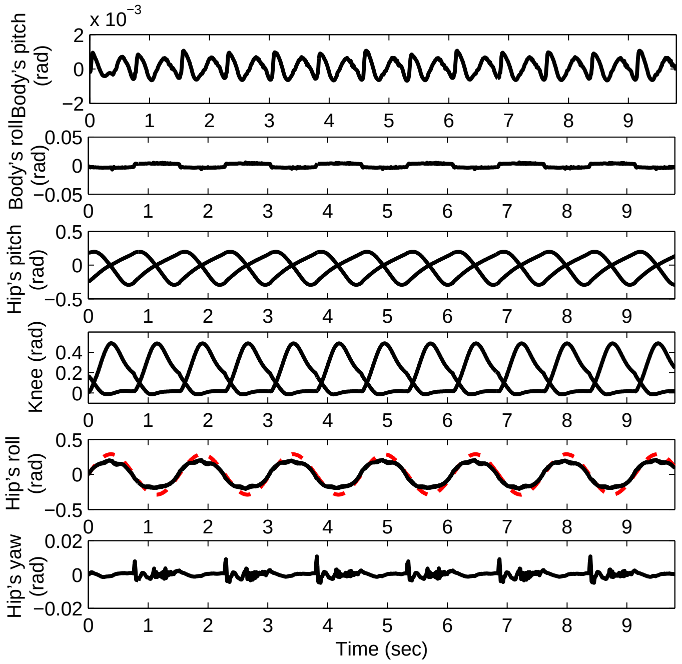


Figure 6.3: Hoap-I robot 3D walking dynamics without reference adjustment.

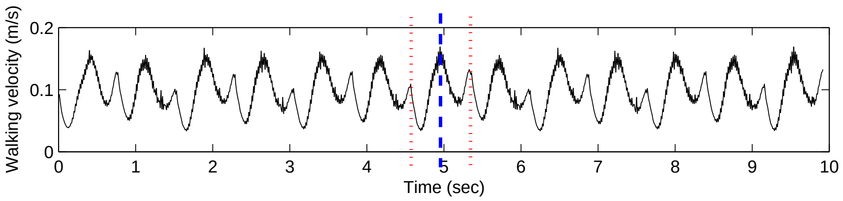


Figure 6.4: Walking velocity profile in the sagittal motion plane. The solid line indicates the velocity when robot body is located at the center line $X = 0$, and the dash lines indicate the touch-down moment walking velocity.

6.3 Frontal Plane Motion Balance Control

In this section, an approach which makes use of a method of learning to find appropriate parameters to close the motion control loop and achieve stable locomotion is investigated. Previous work have been done on the application of reinforcement learning for bipedal locomotion control [46][49]. The work presented here also adopted reinforcement learning to explore the potential of a simple optimized motion generator, the proposed Truncated Fourier Series Model (TFS), for 3D bipedal locomotion control. In the TFS model applications for 2D walking presented in Chapter 5, three key parameters determine the locomotion - the fundamental frequency which determines the pace of walking, the amplitude of the functions which determines the stride, and the constant terms used to adjust to different inclinations of the terrain. The following presents a motion balance strategy which is also based on the TFS model for the frontal plane motion.

In previous sections and chapters, the sagittal motion is shown stable without any parameter adjustment during walking. Because of assured stability of locomotion in the sagittal plane, using the GAOFSF generated trajectories, it only now remains to consider the frontal plane motion to achieve good stable 3D walking. For this purpose, the robot control model is simplified as a one-link model actuated by the stance ankle joint, as shown in Figure 6.5. The controller applied to the stance ankle joint in the frontal plane is considered as a pair of virtual spring k_p and damper k_d.

The walking results presented in the previous section without reference adjustment show that it could be difficult to achieve a long distance walking motion by using a symmetrically planned motion reference. Therefore, it is necessary to incorporate some real-time motion adjustment mode to online correct and balance the walking motion. In order to

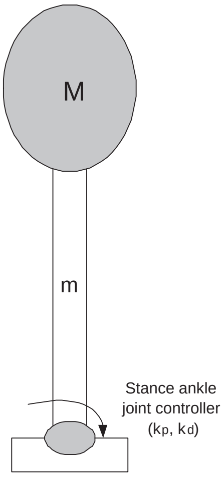


Figure 6.5: Robot Model for balancing control.

achieve that, the formulation of the lateral motion's reference is modified as:

$$\theta_r = \begin{cases} c_1 + R_1 \Sigma_{i=1}^{2k+1} D_{1i} sin(i\omega_1 t), \quad k = 1,2,3..., & t \in [0, T_m] \\ c_2 + R_2 \Sigma_{i=1}^{2k+1} D_{2i} sin(i\omega_2 (t - T_m)), & t \in [T_m, T] \\ \omega_1 = \frac{\pi}{2T_m}, \quad \omega_2 = \frac{\pi}{2(T - T_m)}, \\ c_1 + R_1(\Sigma_{i=1}^{2k+1} D_{1i}) = c_2 + R_2(\Sigma_{i=1}^{2k+1} D_{2i}) \end{cases} \tag{6.10}$$

This modified TFS formulation for frontal plane motion comprises two TFS functions joined and having the same values at $t = T_m$, or the point at which the lateral motion changes direction. The reference continuity is thus not affected and all solutions given by Equation (6.8) can also be approximated. In Equation (6.9), R_1 and R_2 are the scaling factors for the 1*st* and 2*nd* halves of the lateral motion respectively. ω_1 and ω_2 are their corresponding fundamental frequencies. c_1 and c_2 are the constants attached in the TFS formulation. Theoretically, through the adjustment of parameters $(c_1, c_2, T_m, D_{1i}, D_{2i})$, all the lateral motion patterns can be generated.

As discussed in the previous section when the motion trajectories were optimized by minimizing the time integral of the stance foot torque, the optimum lateral motion can be represented by just the fundamental component in the TFS without much loss of accuracy. To reduce computational cost, higher order terms in Equation (6.10) are neglected in this work leading to the simplified formulation given by (6.11):

$$\theta_r = \begin{cases} c_1 + R_1 D_1 sin(\omega_1 t) & t \in [0, T/2] \\ c_2 + R_2 D_1 sin(\omega_2 (t - T/2)) & t \in [T/2, T] \\ \omega_1 = \frac{\pi}{T}, \quad \omega_2 = \frac{\pi}{T} \\ c_1 + R_1 D_1 = c_2 + R_2 D_1 \end{cases} \tag{6.11}$$

The basic frontal plane motion is now completely defined by only the three parameters (c_1, R_1, c_2). Here, these three parameters are tuned by trial-and-error to define a basic frontal pattern. In adjusting the TFS generated lateral reference trajectory for stable long-distance 3D walking, the target of these adjustments will be the basic frontal plane motion, whose dynamics are very similar between the initial state and the end state of a step.

Based on the three parameters (c_1, R_1, c_2) which specifies the TFS motion, one motion balance strategy is investigated here. It is through the adjustment of c_1 for dynamics error regulation. Reinforcement Learning then is assisted to establish the robot self-learning mechanism for walking balance behavior.

6.3.1 TFS Motion Balance Strategy: c_1 adjustment

Dynamics perturbations from ground contact behavior occur at the instant that the swing foot lands on the ground. Compared to the sagittal motion, the frontal motion is more difficult to be controlled as the desired motion. Therefore, there is a need to adjust the reference trajectory to compensate for this disturbance so as to maintain stable locomotion.

Referring to the TFS formulation of (6.11), of the three parameters which specify the basic motion, the approach here is to adjust the first half of the basic step by adjusting c_1 while keeping R_2 and c_2, which together specify the $2nd$ half, constant. Referring to Figure 6.6 and Equation (6.12) to ensure continuity at the middle of the step at $t = T/2$, R_1 will be adjusted according to the relationship $c_1 + R_1 D_1 = c_2 + R_2 D_1$.

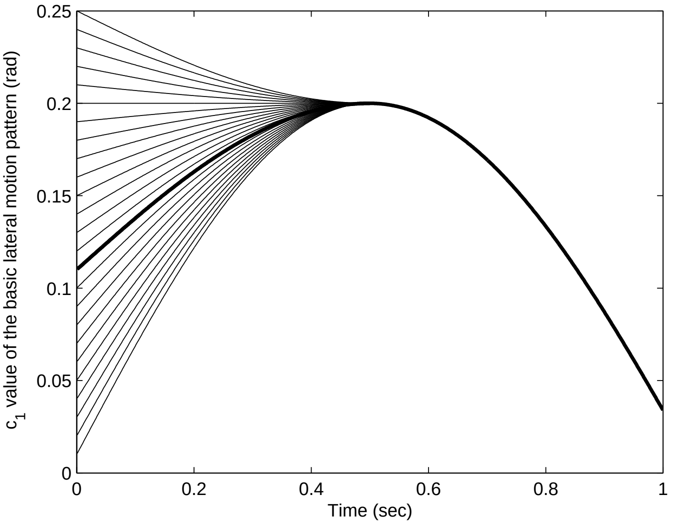


Figure 6.6: Illustration of the reference adjustment through c_1 adjustment.

Assuming the highlighted solid line is the selected basic motion's lateral reference, by varying c_1 the initial value of the reference position can be shifted. Therefore, if there is no under-actuation issue, there will always be a value of c_1 which will make the motion converge to the target frontal plane motion. Incorporating the foot-ground interactions, the learning agent is assigned the task to learn the appropriate value of c_1 based on the dynamics states without toppling over the stance foot. In the course of learning, the CMAC network always updates the state-action pair locations based on the walking experience so as to give a better function approximation. In this way, after a period of training, the trained CMAC network will be able to provide a better estimation of c_1 at the state of each step so as to achieve stable walking. This will be especially so in the region of the starting states of each step in which there had been more previous walking experience.

State variables Based on the assumption that the dynamics in the sagittal plane and the frontal plane are weakly coupled, the state variables chosen for the reinforcement learning are the deviation of the robot rolling position $\Delta\theta$ and velocity $\Delta\dot{\theta}$ from the basic pattern motion taken at the end of a step which also corresponds to the beginning of a new single support phase. The errors are defined as follows:

1. $\Delta\theta = \theta - \theta_b$, θ_b being the basic walking pattern's initial rolling position.

2. $\Delta\dot{\theta} = \dot{\theta} - \dot{\theta}_b$, $\dot{\theta}_b$ being the initial rolling velocity of the basic walking pattern.

Reward function and return computation For the frontal plane balance control, a ”myopic” learning agent ($\gamma = 0$) [63], which only tries to maximize the immediate reward, is used for reinforcement learning. The immediate reward need not be very precise. However, it must roughly represent how well the desired walking behavior is maintained after each step's balance control has been executed [49]. Such myopic learning applied to the bipedal locomotion control has also been successfully implemented in the work by Chew and Pratt [49]. Here, the performance of the walking behavior is evaluated by hip motion's dynamics error E formulated as Equation (6.12):

$$E = \Delta\theta^2 + r_1 \Delta\dot{\theta}^2 \tag{6.12}$$

where r_1 is the weight for the errors from different sources affecting the motion's balance behavior.

Using the ”myopic” learning agent, the reward function is therefore computed mainly in the form of the lateral motion's dynamics error as (6.13):

$$r = \begin{cases} -E & For: (\Delta\theta \subseteq U_{\Delta\theta})\ and\ (\Delta\dot{\theta} \subseteq U_{\Delta\dot{\theta}}) \\ R_f & otherwise\ (failure). \end{cases} \tag{6.13}$$

where R_f is a negative constant value assigned as a large penalty.

Learning implementations With the initial conditions set at $\theta_r = 0$, $\dot{\theta}_r = 1.32 rad/s$ and $D_1 = 0.22$ from the GAOFSF method, a suitable frontal basic pattern is found at: $c_1 = 0.15$, $c_2 = 0.003$, $\omega = 3.2$ and $R_1 = 0.675$ through trial-and-error. This basic 3D walking pattern results in similar initial and end dynamic conditions including both position and velocity of the robot. In that case, the basic motion becomes the motion target for the consequent steps trying to repeat.

Before reinforcement learning is executed, a pre-learning process is conducted to obtain some heuristics of the boundaries of the dynamics and weighting of the position error and velocity error in the reward function (6.13). This pre-learning is based on an exhaustive searching for action c_1 for each step under different position and velocity boundaries. Once the boundary is satisfied, the robot will just proceed to the next step. After obtaining a suitable number of successful and failed walking experience (say 50 states), the rolling position and velocity boundaries can be roughly sketched as shown in Table 6.2. Based on the selected boundaries, r_1 will be assigned a value that balances the weighting of position error and velocity error. The description of this implementation is summarized in Table 6.2. The width of the receptive field has to be tuned through trial-and-error for a better action estimation. The number, C, of receptive field is chosen based on some experience selection of previous works [49]. Usually, function approximation behaves better with a larger value for C but the computation cost is also accordingly increased. The learning step-size commonly is a small value [63].

Results Implementing the reinforcement learning algorithm as discussed in the previous section, Figure 6.7 to 6.12 show the walking results with the lateral motion balance agent. Figure 6.7 shows the length of time the robot was able to walk before falling versus the number of the trials. The learning experience is accumulative meaning that what is learned in all previous attempts is also applicable and used for each current attempts. Therefore, with the number of walking trials increased, the state-action function approximation behaves to be more reliable.

The experiment was terminated when the target of 50 steps of stable walking was

Table 6.2: Reinforcement Learning Set-up for balance control through c_1 adjustment

Description	Remark
Basic frontal walking pattern	$c_1 = 0.15$, $R_1 = 0.675$, $c_2 = 0.003$
Learning output (action) u	parameter Δc_1
Reward function	Equation (18)
$\Delta\theta_{max}$	$0.01rad$
$\Delta\theta_{min}$	$-0.01rad$
$\Delta\dot{\theta}_{max}$	$0.2rad/s$
$\Delta\dot{\theta}_{min}$	$-0.2rad/s$
r_1	2.5
Discount factor γ	0 (myopic)
Action set U	$(0.001n\| -0.02 \leq 0.001n \leq 0.02)$ and $n \in Z$
Policy	modified ε-greedy $(\varepsilon = 0)$
CMAC parameters	
Width: receptive field for $\Delta\dot{\theta}$ Width: receptive field for $\Delta\theta$ Width: receptive field for Δc_1 Receptive fields layers no. C Learning step-size α	$0.2rad/s$ $0.02m$ $0.01m$ 128 0.25

achieved. This tempt was achieved after 252 iterations as attempts. The walking pattern for the final 50-step walk is shown in the stick diagram in Figure 6.8, with the dynamics for the first period of the walk in Figure 6.9.

From the result, it can be seen that the walking pattern still maintains the human-like posture in the sagittal plane with the body held in an upright position with only small periodic variations. In Figure 6.9, it should be noted that without any parameter adjustment, the sagittal plane motion still can be maintained well, as seen from the very periodical actual joint trajectories (Figure 6.9 (d)). Also because the generated walking motion is optimized to be quite symmetrical according to the center line, the actual velocity profile will be also maintained rather equivalent before and after crossing the center line. Due to the higher joint control gains, the resulting velocity is still shown to be more fluctuated but the velocity profile in the positive and negative walking phases are rather equivalent according to the center line $X = 0$.

Figure 6.9(b) shows the actual hip joint roll angles for the stance and the swing legs. It can be seen that the two joint angles are almost overlapped showing the swing leg is well maintained to be parallel to the stance leg during walking. Furthermore, using reinforcement learning, data as shown in Figure 6.9 (f), the subsequent steps are controlled to be stable and very similar to the initial basic frontal plane motion. Figure 6.10 and Figure 6.11 show the frontal plane motion behavior in the Yobotics dynamic simulation environment. (Note: qd_y, lateral velocity; lstate, value of 1: left support, and value of 0: left swing, q_y, lateral motion displacement referring to the middle line $Y = 0$ in the frontal motion plane). It can be seen that the maximum lateral velocity occurs at the moment that the swing foot landed on the ground and the lateral velocity reduced to be zero about in the middle of a step, at $t = T/2$.

Relation of parameter c_1 and the frontal dynamics Referring to the learning states obtained, within a fixed speed walking motion, the general relation of parameter c_1 and the frontal plane motion dynamics is found to be: c_1 will be increased with the combined dynamics indicator $U = P_0 + \xi |v_y|$ decreased, where ξ is a weighting parameter balances

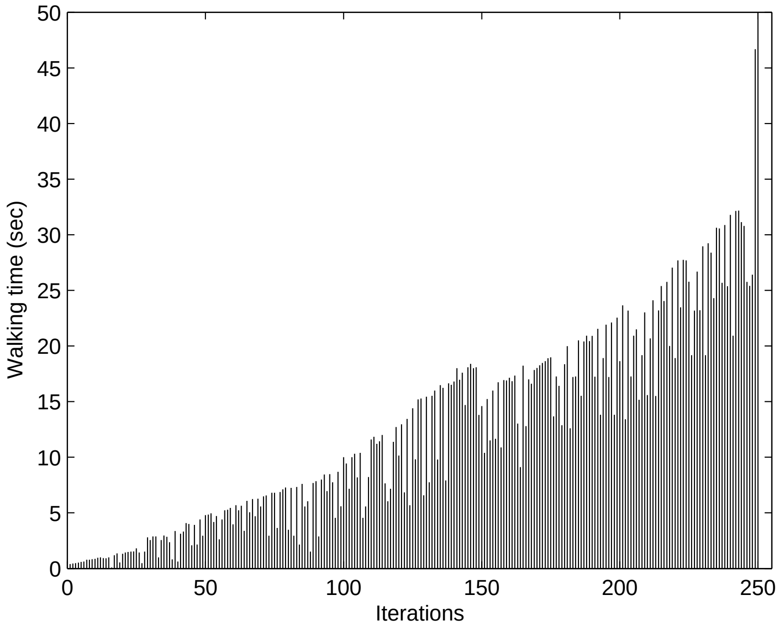


Figure 6.7: Learning profile for the simulated HOAP-I 3D walking.

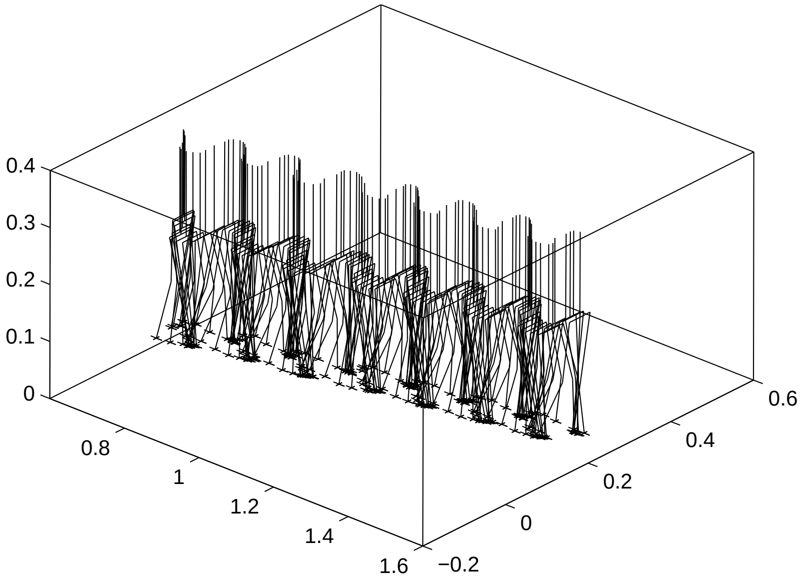


Figure 6.8: Stick-diagram of 3D dynamic walking under action c_1.

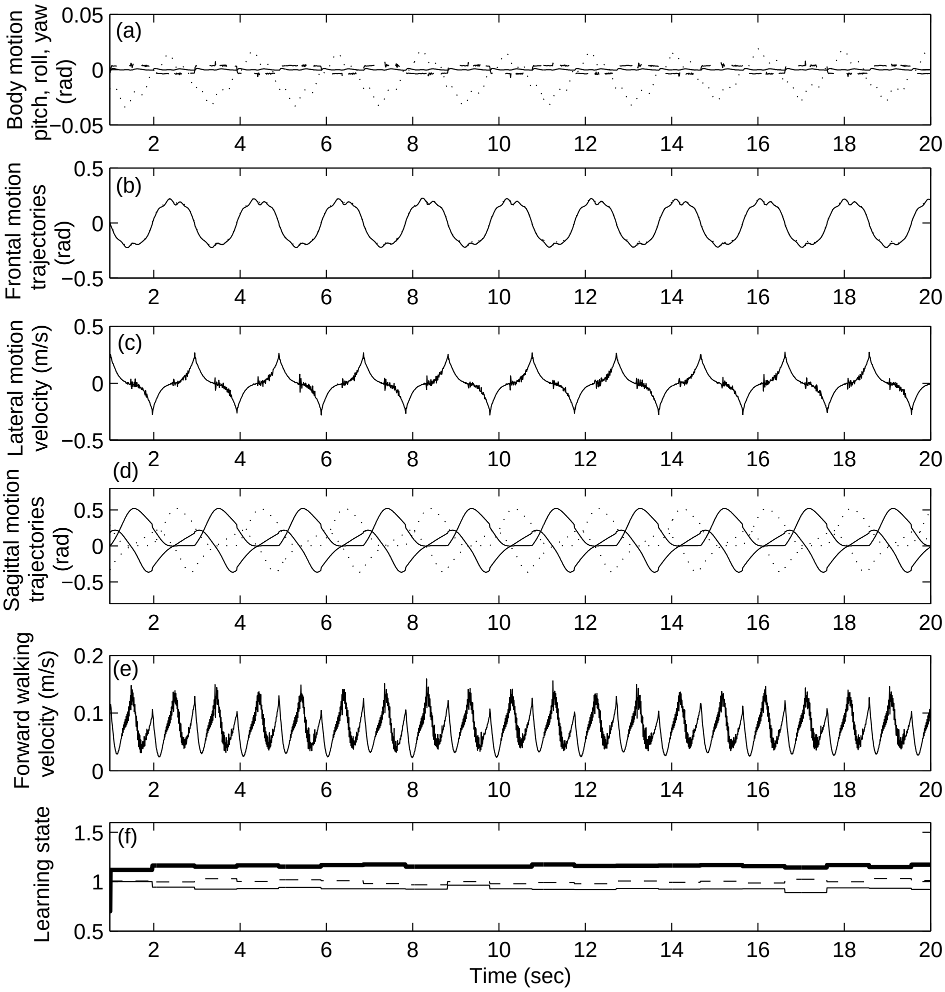
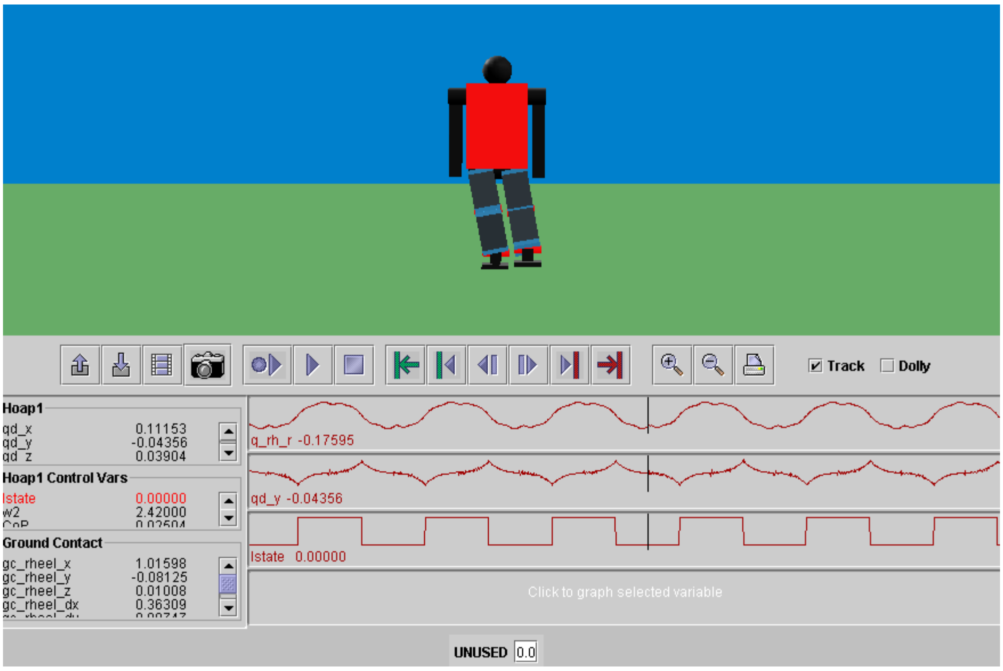


Figure 6.9: The resulting 3D motion dynamics under action c_1.



```
Hoap1
qd_x        0.11153
qd_y       -0.04356
qd_z        0.03904
Hoap1 Control Vars
lstate      0.00000
w2          2.42000
CoP         0.02504
Ground Contact
gc_rheel_x  1.01598
gc_rheel_y -0.08125
gc_rheel_z  0.01008
gc_rheel_dx 0.36309
```

q_rh_r -0.17595

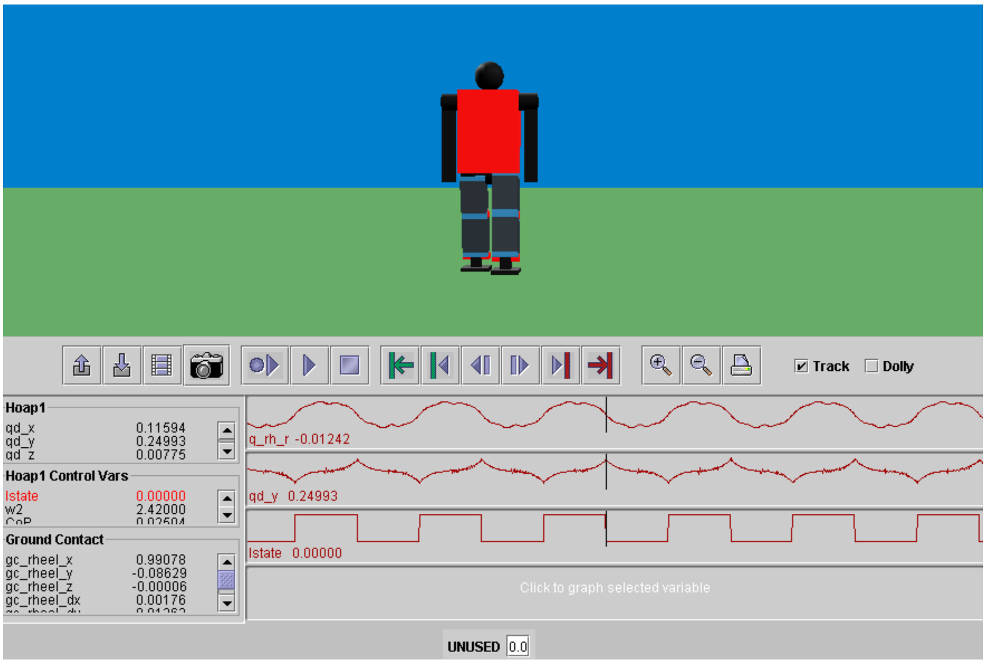
qd_y -0.04356

lstate 0.00000

Click to graph selected variable

UNUSED 0.0

Figure 6.10: Frontal motion balance behavior. $v_y(qd_y) - > 0$ at the middle of a step.



```
Hoap1
qd_x        0.11594
qd_y        0.24993
qd_z        0.00775
Hoap1 Control Vars
lstate      0.00000
w2          2.42000
CoP         0.02504
Ground Contact
gc_rheel_x  0.99078
gc_rheel_y -0.08629
gc_rheel_z -0.00006
gc_rheel_dx 0.00176
```

q_rh_r -0.01242

qd_y 0.24993

lstate 0.00000

Click to graph selected variable

UNUSED 0.0

Figure 6.11: Frontal motion control behavior: maximum lateral velocity occurs at the touch-down moment.

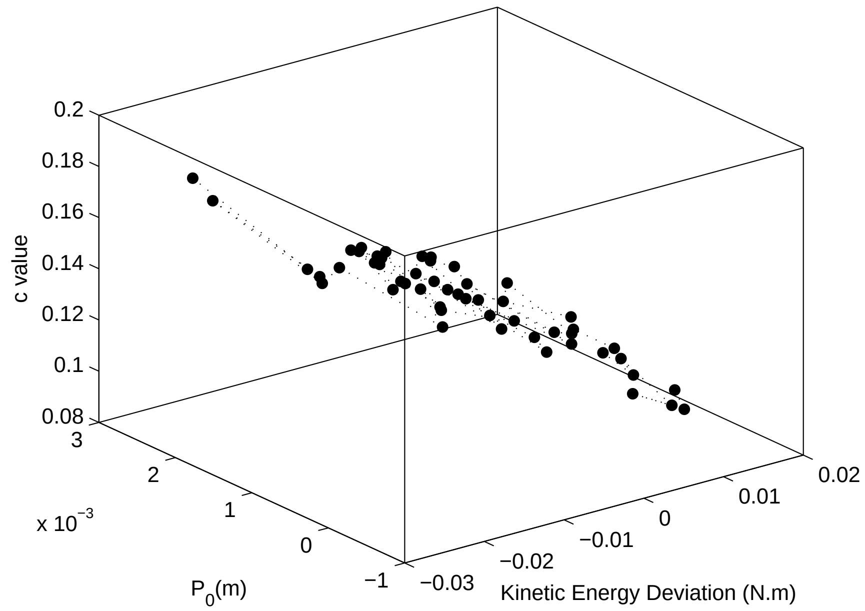


Figure 6.12: Relation between the action c_1 and dynamics at the touch-down moment.

the dynamics effect given by the position and velocity component, as shown in Figure 6.12.

The relation can be easily understood that when the lateral motion energy is not enough, c_1 needs to be properly increased to generate the driving force and vice versa for the situation when the motion energy is a bit excessive.

From the results obtained, it can be concluded that the parameter c_1 in the TFS model can be used for motion balance control and the application through reinforcement learning is effective. However, as seen from the lateral position and velocity trajectories, the lateral behavior is not of good motion smoothness. This is mainly because the resulting motion is controlled by stiff control gains instead of the incorporation of more natural dynamics using a soft motion controller.

6.4 Variable Speed 3D Walking Results

The application of the proposed balance control approach is further extended to the 3D variable speed walking to explore the potential or limit of this balance control strategy. The posture transition is mainly handled by the sagittal plane motion adjustment and the frontal plane motion is to maintain the motion balance.

Variable speed walking can be achieved through the adjustment of stride-frequency or step-length. To accurately identify a dynamic state for 3D variable speed walking, the learning agent should be further provided with information on the desired walking step-length R_d, the desired stride-frequency ω_d, the change of the step-length ΔR_d and the change of the stride-frequency $\Delta\omega_d$ in addition to the frontal plane dynamics state $(\Delta\theta, \Delta\dot{\theta})$. Therefore, the input state becomes $[R_d, \omega_d, \Delta R_d, \Delta\omega_d, \Delta\theta, \Delta\dot{\theta}]$ with subnetworks $c_1 = f_{c_1}(R_d, \omega_d, \Delta R_d, \Delta\omega_d), R_1 = f_{R_1}(R_d, \omega_d, \Delta R_d, \Delta\omega_d), c_2 = f_{c_2}(R_d, \omega_d, \Delta R_d, \Delta\omega_d)$. However, this will greatly enlarge the state-action space. Due to the current limitation of the state-action space for reinforcement learning, in the work here each walking pattern is just assigned a CMAC network still with the input state as $[\Delta\theta, \Delta\dot{\theta}]$.

Before conducting the action searching for variable speed walking, a series of basic walking patterns for different stable walking and speed transitions were firstly found by trial-and-error. To change from one basic walking pattern to another, a transition pattern is also needed. This transition pattern can be found through the optimization of energy consumption or the least change from the previous basic pattern. Let $F_i(\omega_d, R_d)$ be the desired frontal plane walking pattern i, and $f(\Delta\omega_d, \Delta R_d)$ be the basic transition pattern to be used when changing from pattern F_1 to F_2. The following are some demonstrations of variable speed walking achieved by this balance motion control strategy.

The first demonstration of walking speed transition is through the adjustment of the desired stride-frequency ω_d. The stride-frequency is required to be changed from $\omega_d = 3.2 rad/s$ to $\omega_d = 4.14 rad/s$ at the $6th$ step with the step-length fixed. The basic pattern F_1 is found at: $c_1 = 0.15$, $c_2 = 0.003$ and $R_1 = 0.675$. The transitional frontal plane pattern $f(0.9, 0)$ is found as: $c_1 = 0.159$, $R_1 = 0.625$ and $c_1 = 0.0036$ through minimizing the accumulation of the stance ankle joint torque during the stance phase. The basic pattern F_2 is found at: $c_1 = 0.16$, $R_1 = 0.6$ and $c_2 = 0.0035$. The learning conditions are the same as Table 6.2. Walking motion before transition has its reward based on the dynamics error formulation of the 1*st* basic pattern, and walking in transition and afterwards are both based on the dynamics formulation of the final motion after transition.

Figure 6.13 shows the resulting dynamics from which it can be seen that the robot body is still maintained to be upright referring to Figure 6.13(a). The walking velocity is successfully transited to a slightly higher level of walking as shown in Figure 6.13 (e). Still, the resulting walking velocity contains more fluctuations compared to the ”U” shaped velocity profile appeared in the 2D walking motions, but with the end velocity not very varied from the initial velocity. Therefore, the sagittal plane motion is still maintained stable for all the steps. Incorporating the developed balance strategy through c_1 adjustment, the frontal plane motion maintains the similarity of the dynamic behavior compared to their corresponding basic walking pattern in the frontal motion plane. The resulting walking posture in the sagittal plane is shown in the stick-diagram, Figure 6.14. The frequency change can be observed from the change of the stick-density.

The second demonstration of speed transition is through a step-length adjustment. The step-length is reduced from $R_d = 0.77$ to $R_d = 0.6$ with the stride-frequency fixed at $\omega_d = 4.14$. The basic pattern for F_1 is the F_2 used in the previous demonstration and $f(0, -0.17) = (c_1 = 0.124, R_1 = 1.27577, c_2 = 0.005)$. Similar to the previous demonstration of the stride-frequency adjustment, the transition pattern is selected by optimizing the integral of the stance ankle's torque, and then leading $F_2 = (c_1 = 0, R_1 = 1.5, c_2 = 0.0015)$. Figure 6.15(d) and (e) show that the sagittal plane motion is still well controlled for all the steps. Referring to the resulting walking velocity shown in Figure 6.15(e), it can be noted that the walking velocity was successfully transited to a lower level but the velocity profile is still not of a natural ”U” profile. Same as the previous demonstration, the frontal plane motion of this variable step-length walking also has achieved similar dynamic behavior for all the steps, however, the resulting motion is shown certain jerkiness especially in the frontal plane motion. The actual walking

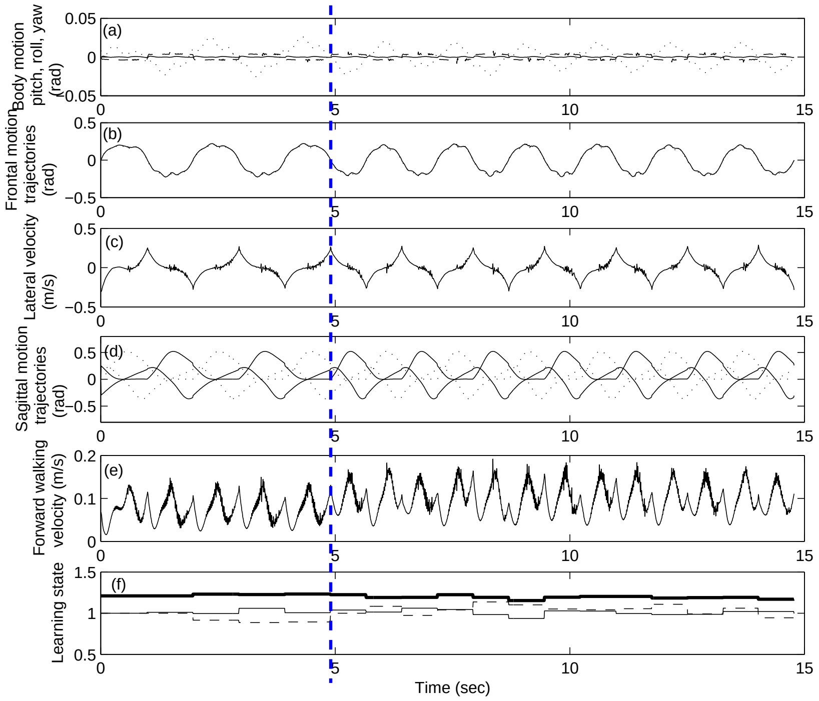


Figure 6.13: The resulting dynamics of walking with frequency changed from $\omega = 3.2rad/s$ to $\omega = 4.14rad/s$.

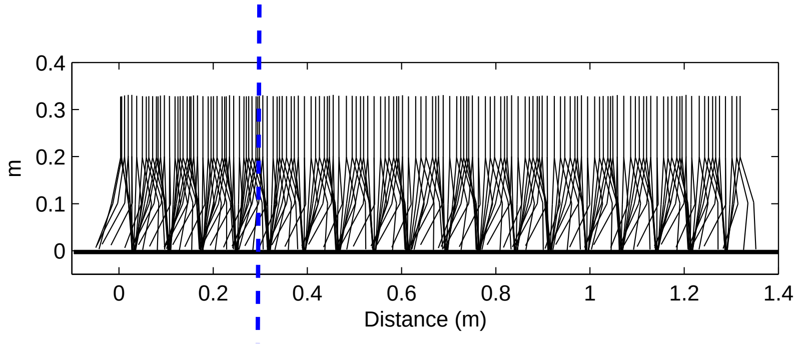


Figure 6.14: Stick-diagram of the sagittal motion: walking stride-frequency is varied from $\omega = 3.2rad/s$ to $\omega = 4.14rad/s$.

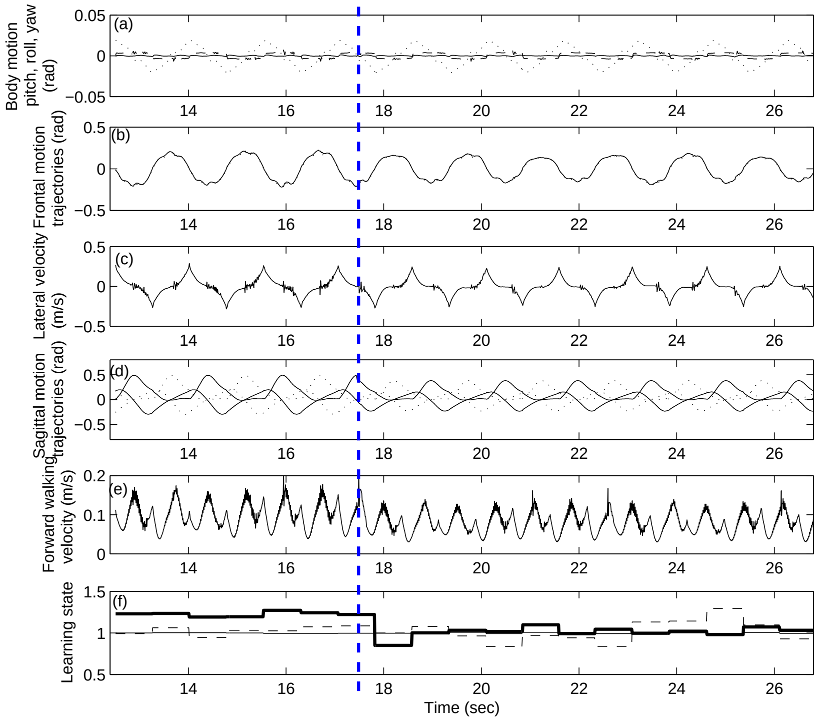


Figure 6.15: The resulting dynamics of walking: walking step-length is reduced.

posture in the sagittal plane is shown in the stick-diagram, Figure 6.16.

6.5 Summary

The application of the proposed TFS model for 3D bipedal walking control, especially for the lateral balance behavior has been demonstrated by several examples. Directly applying the optimized walking motion generated by the GAOFSF method to the simulated ”HOAP-I” robot without taking any dynamics feedback, stable 3D human-like walking can be maintained, but only for a certain distance. In order to achieve longer distance stable walking, a reference adjustment strategy, which is based on the TFS frontal plane model (c_1, R_2, c_2), has been investigated and shown to be effective for maintaining motion stability. The walking control results show that the parameters (c_1, R_2, c_2) are sufficient to determine the basic balance pattern and the parameter c_1 contained in

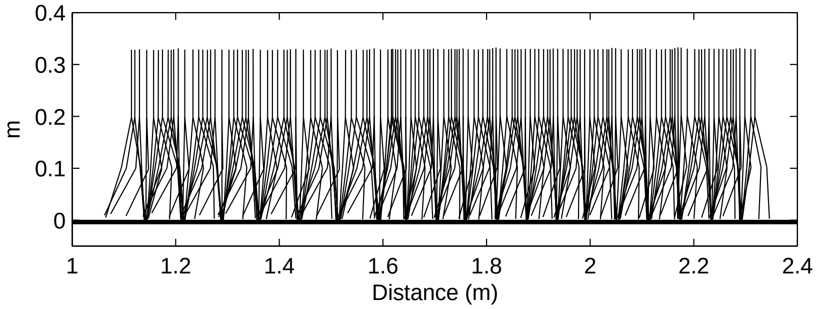


Figure 6.16: Stick-diagram of the sagittal motion: walking step-length is decreased.

the TFS lateral formulation can be adjusted in real-time to achieve stable long-distance fix-speed or rhythmic 3D walking. In addition, reinforcement learning has been successfully applied to the proposed balance control strategy. Therefore, it can be concluded that based on the TFS model, the biped robot can be given some intelligence to self-train the locomotion balance network and then achieve stable 3D dynamic walking.

Through the above examples of fixed or variable speed walking, the TFS model shows its use for balancing the 3D multi-speed bipedal walking. However, the major concern of using this balance strategy is the control gains have to be relatively high and critical so as to converge the motion to the planned motion pattern as much as possible. This may inhibit the excitation of natural dynamics, as noted from the change of the forward walking velocity profile. Since the planned basic motion is of a rather uniform walking velocity, in order to prevent the stance foot from toppling over, the achievable step-length will then be quite limited. Furthermore, although walking can be maintained stable as demonstrated above, the resulting motion is shown with certain jerkiness, especially for the frontal plane motion control.

In order to maintain the natural sagittal plane motion behavior as presented in the 2D walking experiments, the lateral motion control should be also more natural dynamics incorporated. Therefore, the subsequent chapter will focus on how damping behavior can be contributed to the frontal plane motion balance control.

Chapter 7

Frontal Motion Balance Strategy 2

In research into bipedal locomotion control, a major challenge is posed by the fact that the bipedal system is an under-actuated system with the stance-foot always interacting with the ground. Because of this under-actuation, motion trajectories generated without taking account the dynamic environment, i.e ground contact behavior, may not be feasible and the generated reference trajectories will not be properly tracked, especially for the frontal plane motion.

This is mainly because the frontal plane motion has a higher acceleration component for changing the lateral velocity's direction. Even if the generated motion trajectories can be followed to some extent through restricting the trajectory of the ZMP [75]-[79], it will be still difficult to achieve stable 3D walking for a long distance if some form of feedback is not incorporated to adjust to the disturbances caused by the robot-ground interaction or imperfect control gains. This actually has been shown by the results obtained in the previous chapter.

In the previous chapter, although three dimensional long distance dynamic walking has been achieved through the adjustment of the TFS parameter c_1, it is still shown difficult to achieve the natural and smooth sagittal plane walking motion as presented in the 2D walking control due to the stiff controls gain applied to the robot. Using the stiff control gains, the planned motion will be more approximated. However, when the velocity of the basic walking pattern is being uniform, it is not desirable to track the planned basic walking pattern very accurately because less acceleration can then be naturally generated by the formulated motion control law. Therefore, the result of the previous chapter shows even if walking motions can be maintained or adjusted, the adjustment range is quite limited and the resulting walking is of certain jerkiness, losing the good motion adaptation as shown in the 2D walking results.

In order to excite more natural dynamics which can smoothen the walking motion, a damping based frontal motion balance control strategy is investigated for achieving good motion stability and motion adaptation as well. In this chapter, a TFS formulation based damping model is proposed for the lateral velocity regulation. Note in the subsequent balance control strategy, only the damping component will be applied to the frontal motion plane. This means the spring factor $k_p = 0$ will be applied at the stance ankle joint torque.

7.1 Damping Based Frontal Plane Motion Control

Referring to previous research work [49], it can be known that damping behavior plays an important role for achieving the natural and smoother motion behavior. Since the previous chapter shows the direct open-loop joint trajectory tracking does not achieve a robustness locomotion. Also, a more active control based motion adjustment strategy failed to realize a smooth or natural locomotion, a TFS formulation based damping force generator, Equation (7.1) to (7.3), is therefore proposed for the real-time adjustment of the actuation force which balances the frontal plane motion during walking. This time, it can be noted that the TFS is eventually applied to the velocity planning instead of the position planning after the differentiation of Equation (7.1). Figure 7.1 illustrates the lateral position profiles when variable ϕ is zero, positive and negative.

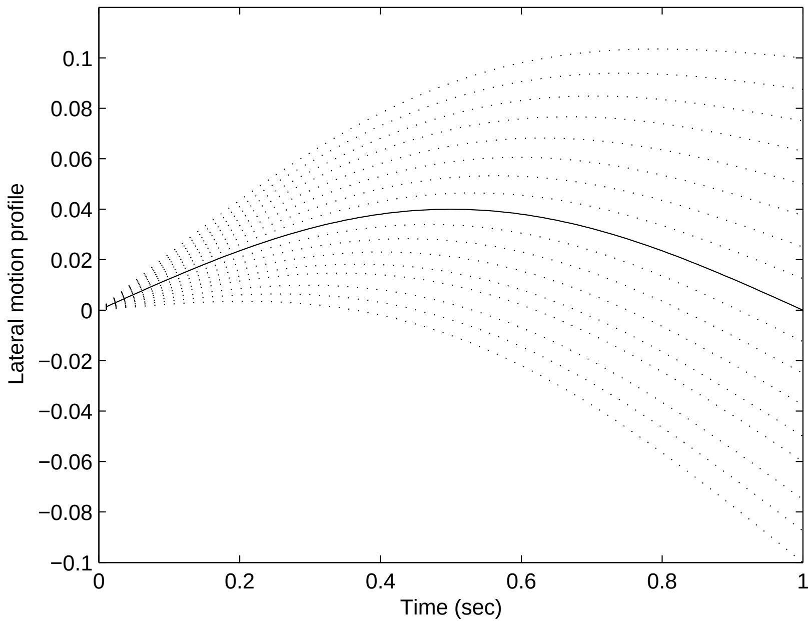


Figure 7.1: Lateral motion (position) profile for the balance control strategy 2.

$$y = R_y sin(\omega t) + \phi t \tag{7.1}$$

where ω is equal to the stride-frequency value for the sagittal motion. Differentiating the Equation (7.1), the desired velocity profile can then be obtained as shown in Equation (7.2).

$$\dot{\hat{y}} = R_y \omega cos(\omega t) + \phi \tag{7.2}$$

Referring to the reference velocity profile, the damper force generator which balances the lateral motion is formulated as Equation (7.3).

$$\tau'_{13} = k_d(\dot{\hat{y}} - \dot{y}) \tag{7.3}$$

where τ'_{13} is the reaction torque of the stance ankle joint torque.

From Equations (7.1), (7.2) and Figure 7.1, it can be seen that varying parameter ϕ, the correlation of the average lateral velocity of the 1*st* and the 2*nd* half lateral motion can be adjusted and achieve the time instant that the robot CG returns back to the middle line in the frontal motion plane to be the same as the moment that the swing foot lands on the ground. Since the frontal plane motion is barely controlled by the damping force, the velocity profile for the phases - swinging from the middle line to one side and swinging back from the side to the middle line - will be rather symmetrical and smooth, provided the body height is not much varied during walking. Then, once the end configuration of a step motion is about the same as the initial configuration in the frontal motion plane, the resulting lateral velocity will be also close to the initial velocity. Therefore, the end dynamics will not be much deviated from the initial dynamics if the end configuration of each step can be regulated well.

Since the adjustment of parameter ϕ is able to adjust the average velocity correlation between the swing-away and the swing-back phases, the strategy for this damping force based balance control can be then just based on the real-time adjustment of parameter ϕ by reinforcement learning algorithm. Similar to the selection of the basic frontal plane motion pattern presented in Chapter 6, the target motion pattern in the frontal motion plane in this chapter is allocated by parameters (R_y, ϕ) which can result in the end dynamics (position and velocity) similar to the initial dynamics, together with an assigned initial lateral velocity v_{y0} and a damping factor k_d.

Similar to the last chapter for motion balance learning through the TFS parameter c_1, the major modules contained in the established reinforcement learning agent for the real-time adjustment of the parameter ϕ are the same, briefly presented as follows:

State variables Based on the assumption that the dynamics in the sagittal plane and the frontal plane are weakly coupled, the state variables chosen for the reinforcement learning are still the deviation of the robot rolling position $\Delta\theta$ and velocity $\Delta\dot{\theta}$ from the basic pattern motion measured at the end of a step which also corresponds to the beginning of a new single support phase, based on the definitions in Chapter 6.

Reward function and return computation The performance of the walking behavior is also evaluated by hip motion's dynamics error E, formulated in Equation (6.12). Still, a ”myopic” learning agent is used with $\gamma = 0$. The reward function is formulated the same as that for the c_1 adjustment balance control strategy, using Equation (6.13).

Pre-learning Before reinforcement learning was executed, a pre-learning process was also conducted to obtain some heuristics of the boundaries of the dynamics and weighting relation between the position error and velocity error in the reward function. The boundaries were obtained in the same way as presented in Chapter 6.

7.2 Fixed Speed 3D Walking

The implemented reinforcement learning algorithm is set up as shown in Table 7.1 for a fixed-speed 3D walking motion control. The basic walking pattern in the sagittal motion plane is still selected as the solution x_{Hoap} given in Chapter 6. Figure 7.2 shows the length of time the robot was able to walk before falling versus the number of the trials. The learning experience is also accumulative meaning that what is learned in all previous attempts is also applicable and used for each current attempt. The experiment was terminated when the target of 50 steps of stable walking was achieved. This tempt was achieved after 128 iterations as attempts. The resulting walking motion mapped in the sagittal motion plane is shown in Figure 7.4 in the form of stick-diagram. Figure 7.3 shows the dynamics for the first period of the walk.

Results of the Frontal Plane Motion Referring to the actual dynamics of the frontal plane motion as shown in Figures 7.3(b) and (c), it can be clearly seen that the lateral motion displacement is naturally confined into a sinusoidal-like profile and it remains quite the same from step to step. Compared to Figures 6.9(b) and (c), the lateral velocity is obviously much smoother under this damping force based balance control. The resulting velocity also has the maximum value when the robot body gets back to the middle

Table 7.1: Reinforcement Learning Set-up for balance control through c_1 adjustment

Description	Remark
Basic frontal walking pattern	$R_y = 0.4$, $\phi = 0$
Learning output (action) u	parameter ϕ
Reward function	Equation (18)
$\Delta\theta_{max}$	$0.01rad$
$\Delta\theta_{min}$	$-0.01rad$
$\Delta\dot{\theta}_{max}$	$0.2rad/s$
$\Delta\dot{\theta}_{min}$	$-0.2rad/s$
r_1	2.5
Discount factor γ	0 (myopic)
Action set U	$(0.001n\vert -0.05 \leq 0.001n \leq 0.05)$ and $n \in Z$
Policy	modified ε-greedy $(\varepsilon = 0)$
CMAC parameters	
Width: receptive field for $\Delta\dot{\theta}$ Width: receptive field for $\Delta\theta$ Width: receptive field for ϕ Receptive fields layers no. C Learning step-size α	$0.2rad/s$ $0.02m$ $0.01rad$ 128 0.25

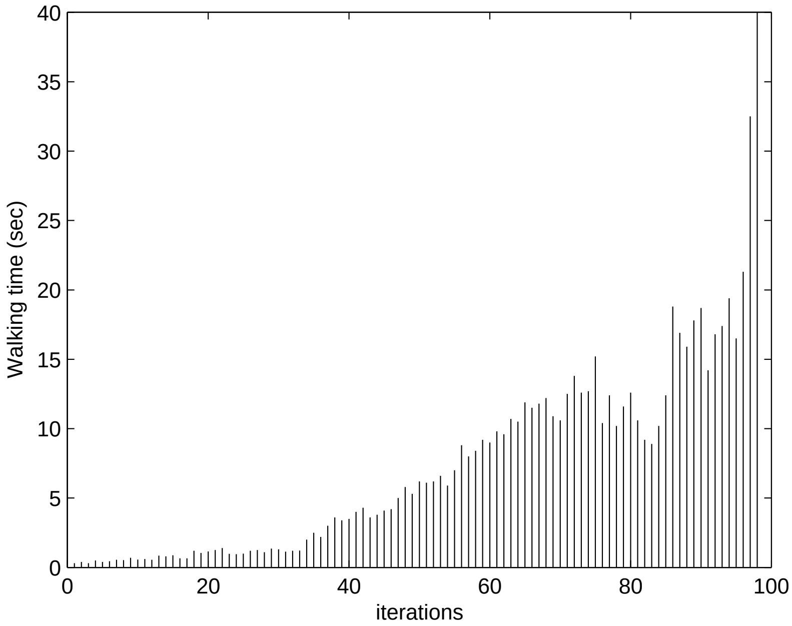


Figure 7.2: Learning profile for the simulated HOAP-I 3D walking.

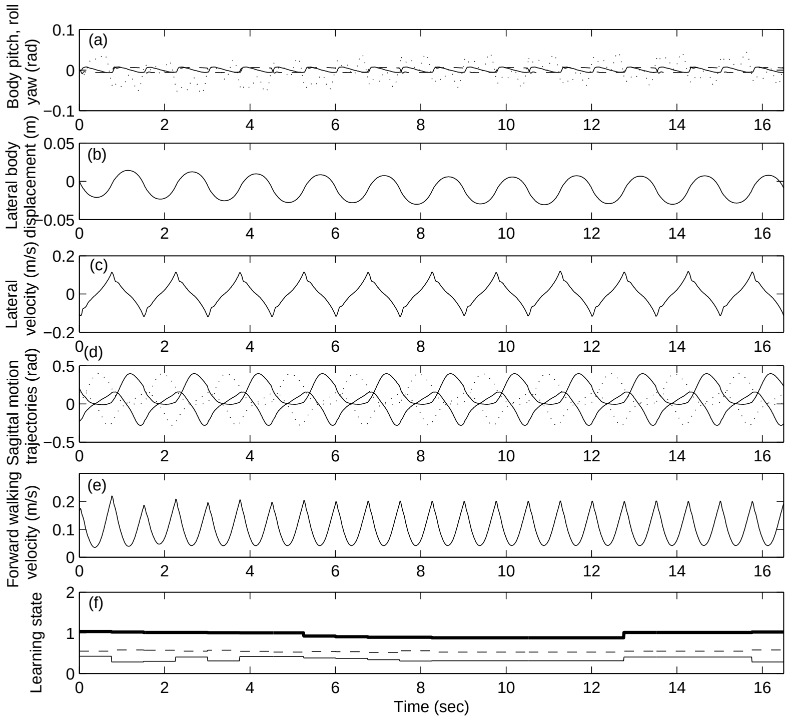


Figure 7.3: 3D walking dynamics under the proposed damping based force generator and sagittal motion control algorithm. (Hoap-I robot)

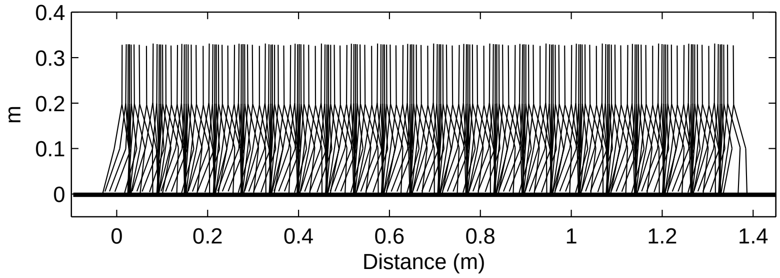
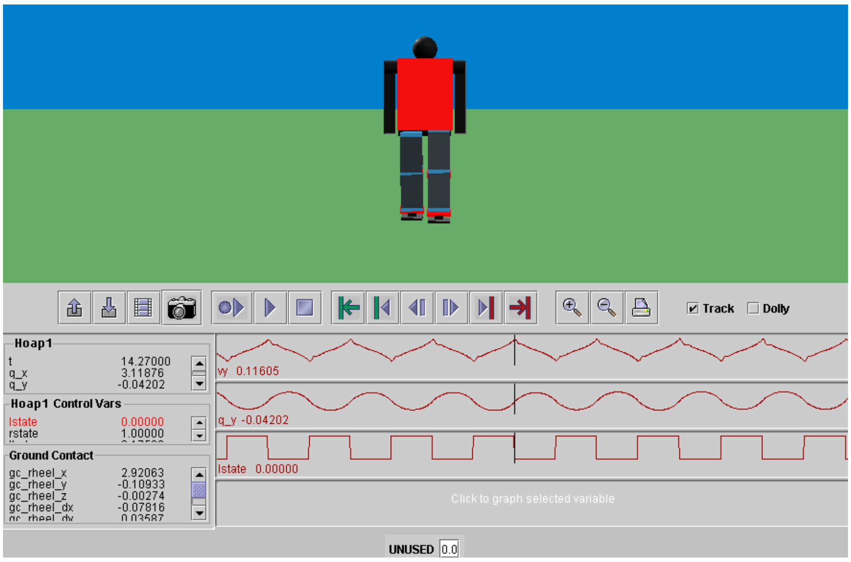


Figure 7.4: Stick-diagram of the resulting 3D walking pattern mapped into the sagittal motion plane.

line with the swing foot landed on the ground at the same time, as shown in Figure 7.5, and the zero lateral velocity occurs at about the instant $t = T/2$ as shown in Figure 7.6. Therefore, the average velocity of the 1st and the 2nd half of the lateral motion are about the same, but in different directions, and the overall average lateral velocity is about zero for each step motion. These behaviors, such as a smooth lateral motion displacement and overall average walking velocity $- > 0$ of each step-motion, are exactly as desired as good robot balancing behaviors.

Results of the Sagittal Plane Motion Based on the dynamics obtained, as shown in Figure 7.3, it can be seen that under this damping force based balance control, the sagittal motion of the 3D walking motion has the same features as those observed from the previous 2D walking motions. The robot body can be held in an upright position with only small periodic variations, referring to Figure 7.3(a), and the sagittal motion can successfully converge to the desired one-step period limit cycle pattern with the forward walking velocity shaped into a ”U” profile, referring to Figure 7.3(e). Also, the



☑ Track ☐ Dolly

Hoap1

```
t          14.27000
q_x         3.11876
q_y        -0.04202
```

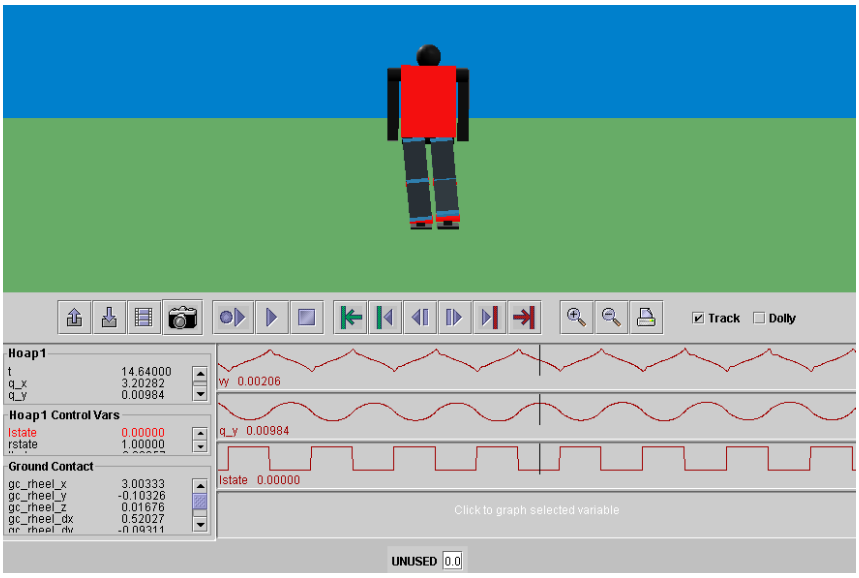
Hoap1 Control Vars

```
lstate      0.00000
rstate      1.00000
```

Ground Contact

```
gc_rheel_x     2.92063
gc_rheel_y    -0.10933
gc_rheel_z    -0.00274
gc_rheel_dx   -0.07816
gc_rheel_dy    0.03587
```

Figure 7.5: Frontal plane motion balancing behavior: maximum velocity occurs at the touch-down moment.



☑ Track ☐ Dolly

Hoap1

```
t          14.64000
q_x         3.20282
q_y         0.00984
```

Hoap1 Control Vars

```
lstate      0.00000
rstate      1.00000
```

Ground Contact

```
gc_rheel_x     3.00333
gc_rheel_y    -0.10326
gc_rheel_z     0.01676
gc_rheel_dx    0.52027
gc_rheel_dy   -0.09311
```

Figure 7.6: Frontal plane motion balancing behavior: maximum lateral swing range occurs at the middle of a step-motion.

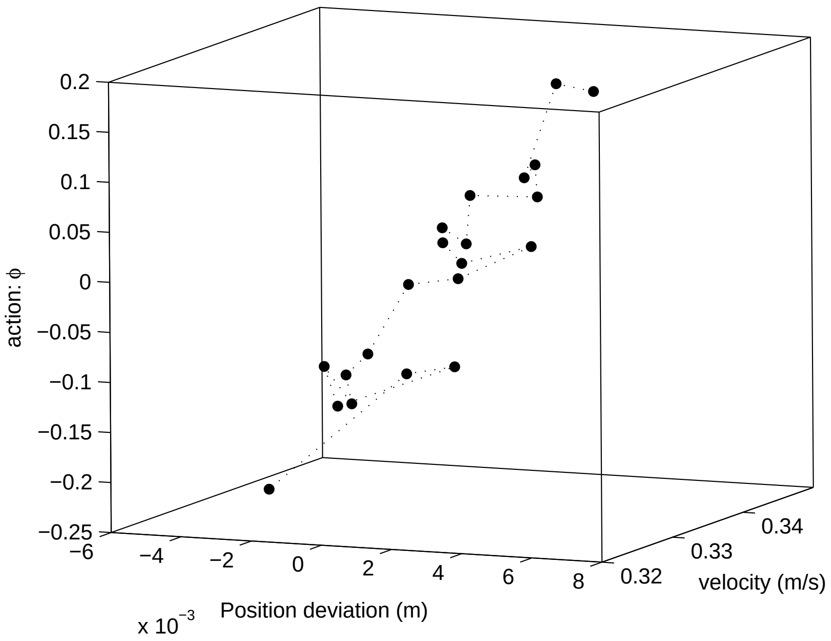


Figure 7.7: Relation of the action: parameter ϕ and robot dynamics at the touch-down moment.

maximum walking velocity occurs at the touch-down moment and the minimum walking velocity occurs close to the center line $X = 0$, as observed from the stick-density of the stick-diagram in Figure 7.4.

Relation of parameter ϕ and the frontal dynamics Referring to the learning states obtained, within a fixed speed walking motion, the general relation of parameter ϕ and the frontal plane motion dynamics is found to be: ϕ will be increased with the increment of the combined dynamics indicator $U = P_0 + r|v_y|$, where r is a weighting parameter, and vice versa, as shown in Figure 7.7.

The relation can be easily understood that when the lateral motion energy is not enough, ϕ should be reduced a bit to generate a larger driving force for swinging away and vice versa for the situation when motion energy is excessive. (Note, when the set point sent to the stance ankle joint controller is based on the cartesian space, y, instead of an angle in the joint space, the desired displacement must be inside the actual displacement to

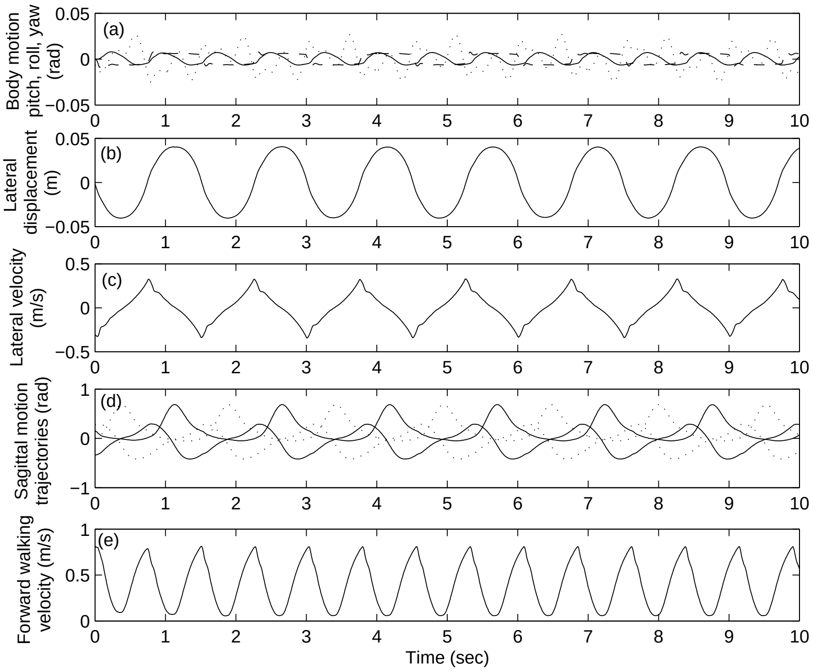
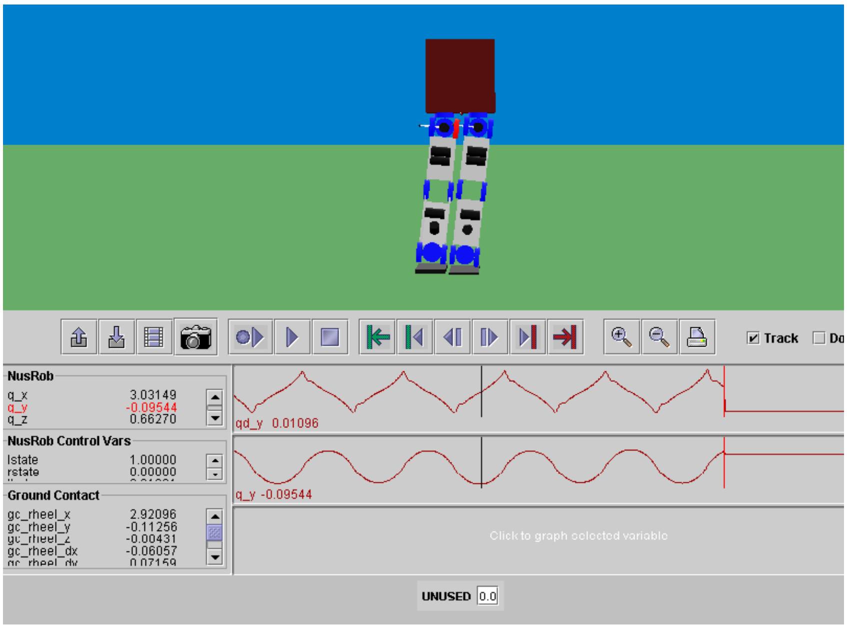


Figure 7.8: Resulting 3D walking dynamics under the proposed damping force based motion balance control and sagittal motion control algorithm (NUSBIP robot).

generate a torque to drive the motion to move in the desired direction. Therefore, here reducing ϕ means to generate more driving forces).

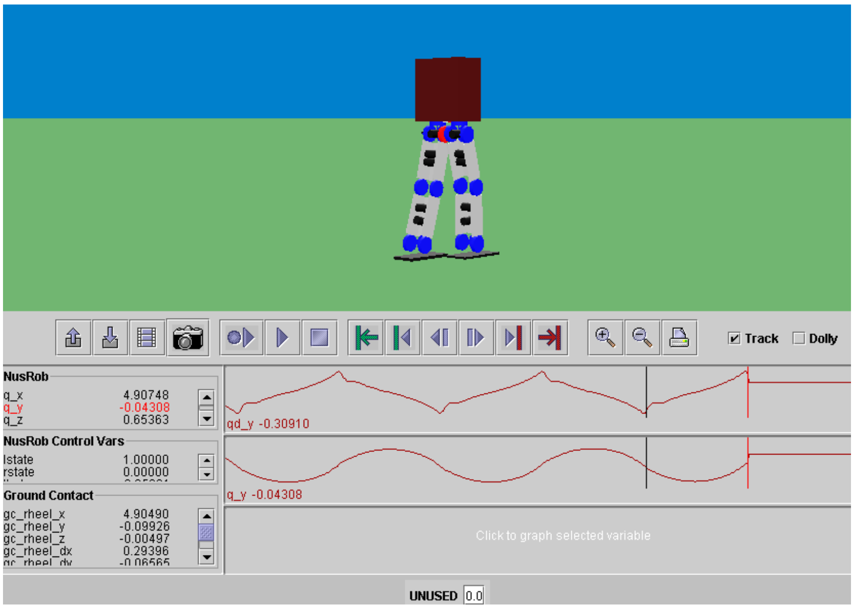
To test the algorithm generality for robots of different mechanical properties, this damping based balance control strategy has also been applied to the larger size robot, NUSBIP-I. Figure 7.8 shows the resulting dynamics of the NUSBIP-I robot's 3D walking motion. It can be seen that the forward walking velocity profile is still achieved as expected: smoothly first reduced and then increased with the minimum velocity occurring roughly at the position $X = 0$ and the maximum velocity occurring at the touch-down moment. Figure 7.9 shows the frontal motion behavior of the motion. The swing leg is shown to be parallel to the stance leg during walking. The maximum lateral velocity occurs at the touch-down moment and the zero lateral velocity is about at the instant $t = T/2$, as shown in Figure 7.9 and Figure 7.10, respectively. All these motion control behaviors confirm the generality of the sagittal motion control algorithm and further validate the frontal motion balance strategy as well.



```
NusRob
q_x            3.03149
q_y           -0.09544
q_z            0.66270
NusRob Control Vars
lstate         1.00000
rstate         0.00000
Ground Contact
gc_rheel_x     2.92096
gc_rheel_y    -0.11256
gc_rheel_z    -0.00431
gc_rheel_dx   -0.06057
gc_rheel_dy    0.07159
```

```
qd_y  0.01096
q_y -0.09544
```

Figure 7.9: NUSBIP robot frontal motion behavior: maximum lateral displacement occurs at the middle of a step-motion.



```
NusRob
q_x            4.90748
q_y           -0.04308
q_z            0.65363
NusRob Control Vars
lstate         1.00000
rstate         0.00000
Ground Contact
gc_rheel_x     4.90490
gc_rheel_y    -0.09926
gc_rheel_z    -0.00497
gc_rheel_dx    0.29396
gc_rheel_dy   -0.06565
```

```
qd_y -0.30910
q_y -0.04308
```

Figure 7.10: NUSBIP robot frontal motion balance behavior: maximum lateral velocity occurs at the touch-down moment.

7.3 Damping Based Variable Speed Walking Control

The proposed damping behavior balance control has also been applied to the variable speed 3D walking control. Since the full motion state for walking on a fixed terrain is supposed to be $[P_x, V_x, P_y, V_y, \Delta L, \Delta\omega_h]$, where $[P_x, V_x]$ and $[P_y, V_y]$ represent the dynamics in the sagittal and frontal motion plane, respectively, taking account of the cross coupling behavior during the 3D walking. Here, $[\Delta L, \Delta\omega_h]$ indicates the change of the desired walking pace. However, currently this big state is not very applicable for the used reinforcement learning algorithm. Therefore, each motion pattern is still separately given a CMAC network, assuming in each fixed speed motion, the cross coupling behavior is rather minor. The following shows the control results when the damping force based balance control is applied to the variable step-length and variable stride-frequency walking.

The first demonstration of speed transition is through a variable step-length robot walking motion. The step-length is first increased for 20% while the stride-frequency is maintained at $\omega_h = 4rad/s$. The basic frontal plane pattern of the first step-length walking is found at $F_1 = (R_y = 0.4, \phi = 0)$ and the basic frontal plane pattern of the new step-length walking is selected at $F_2 = (R_y = 0.4, \phi = 0.2)$ with the transition pattern $f(0, -0.2) = (R_y = 0.4, \phi = 0.11)$. Here, the basic frontal plane motion patterns are all obtained by trial-and-error when achieving the similar lateral motion dynamics at the initial and end instant of a step motion.

Observed from the walking results, the robot body is maintained to be upright at the desired posture during walking, as shown in Figure 7.11(a). Also, the resulting forward velocity profile is again confined into the graceful ”U” shape, as shown in Figure 7.11(e). The entrained limit cycle behavior is also shown of the desired one-step period type, as defined in Chapter 4. Therefore, the sagittal motion trajectories are concluded to be very periodical, referring to Figure 7.11(d).

The desirable sagittal motion behavior can be also observed from Figure 7.12 which shows the stick-diagram of the resulting 3D walking motion mapped into the sagittal

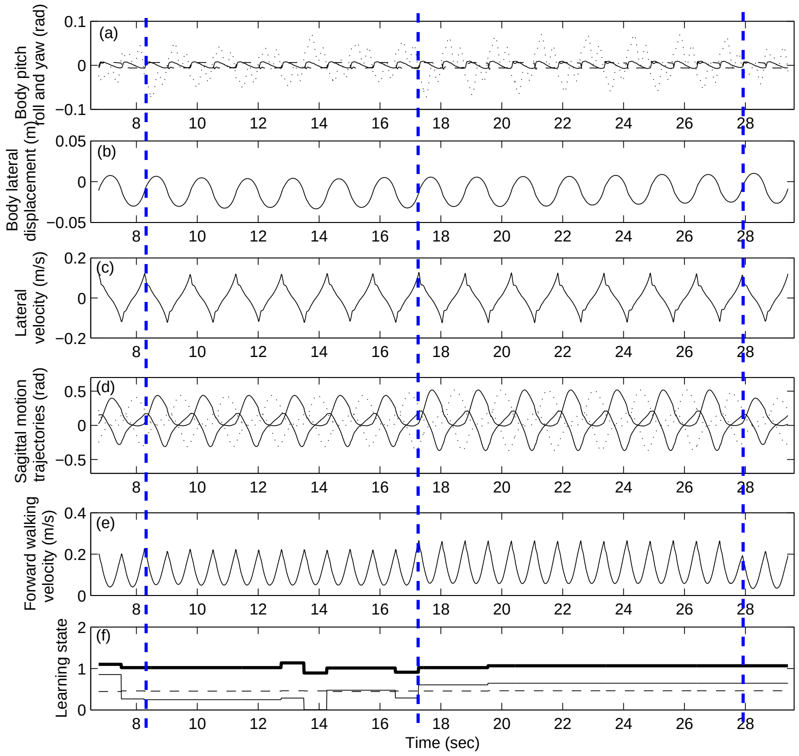


Figure 7.11: 3D variable-step length walking dynamics under the damping force based balance control. (Hoap-I robot).

motion plane. Compared to the stick-diagram, Figure 6.16, it can be clearly seen the differences that the highest stick-density here back to the moment that the robot CG reaches the center line $X = 0$ while the maximum walking velocity occurs at the touch-down moment for all the involved walking patterns. This indicates the resulting motion still remains the motion features presented in 2D walking part, and the resulting 3D motion is natural and smooth.

In addition, the frontal plane motion is also controlled to be periodical through adjusting the variable ϕ in real-time. The resulting frontal motion is much smoother compared to the frontal motion obtained in Chapter 6. There is no obvious jerkiness observed under this damping force based balance control, as shown in Figure 7.11(b) and (c).

The second demonstration of the variable speed walking is through the adjustment of the walking stride-frequency ω_d. The stride-frequency is required to be changed from $\omega_d = 4rad/s$ to $\omega_d = 4.8rad/s$ and then reduced to $\omega_d = 3.4rad/s$ as an example. The

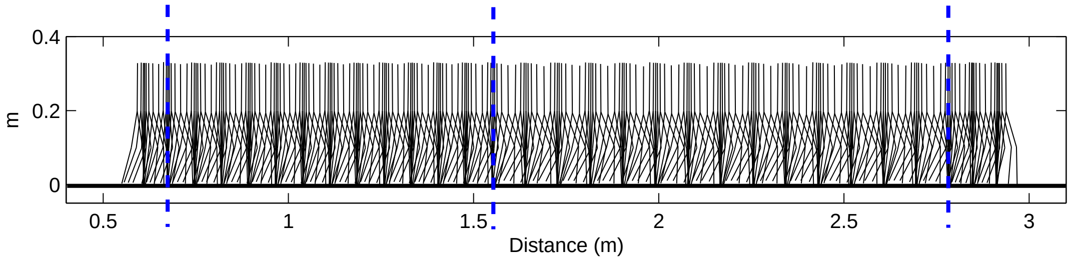


Figure 7.12: Stick-diagram of the conducted variable step-length 3D walking mapped into the sagittal motion plane.

learning conditions are set the same as provided in Table 7.1. Figure 7.13 shows the resulting dynamics from which it can be seen that the robot body is also maintained at the upright position. The forward walking velocity is also observed to be smoothly transited and then converge to the steady-state, as shown in Figure 7.13(e). Therefore, it shows that incorporating the damping based balance control, the dynamics in the sagittal motion plane can easily achieve the natural motion transitions as presented in the 2D walking part. Referring to Figure 7.13(b), the online adjustment of parameter ϕ also achieved the smooth lateral motion balance and naturally confined the lateral position into a sinusoidal-like profile.

7.4 Summary

From the results obtained, it can be concluded that parameter ϕ in the simple TFS formulated damping force generator is applicable for adjusting the motion actuation force so as to balance the walking motion stably. Furthermore, the application through reinforcement learning algorithm is again shown to be effective.

Compared to the previous balance control strategy through the adjustment of parameter c_1, the damping based balance control strategy can maintain the natural dynamic walking behavior in the sagittal motion plane. Furthermore, the resulting motion profile in the frontal plane motion is shown to be smooth, naturally of a sinusoidal profile.

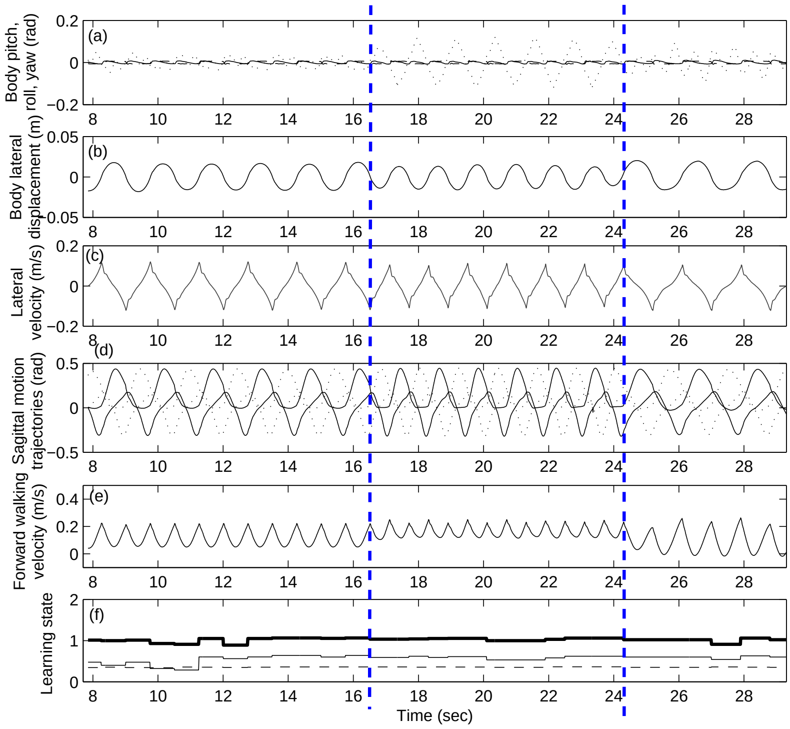


Figure 7.13: 3D walking dynamics of the variable-stride-frequency walking under the damping force based balance control. (Hoap-I robot).

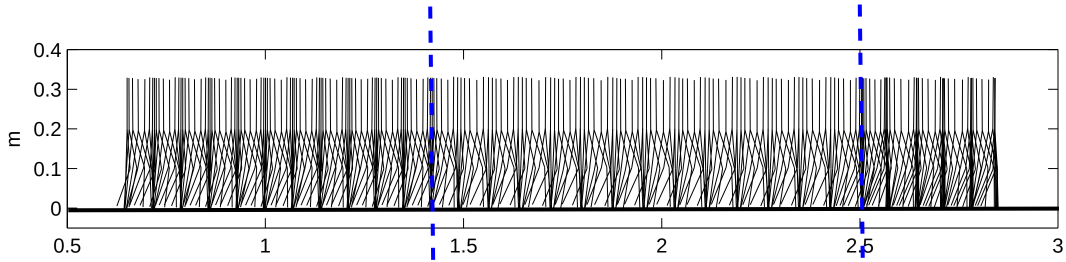


Figure 7.14: Stick-diagram of the variable stride-frequency 3D walking pattern, mapped into sagittal motion plane.

Furthermore, the variable speed 3D walking achieved by this damping based balance strategy is much less restricted by the step-length or stride-frequency. Compared to the walking speed achieved in Chapter 6, the achieved 3D walking speed has increased for about 70% in Chapter 7, as noted by the velocity plot in Figure 6.9 and Figure 7.3.

Chapter 8

Conclusion

Using a ”divide-and-conquer” approach, the 3D dynamic bipedal walking control task is partitioned into sub-tasks in the three orthogonal planes: sagittal, frontal and transverse. The combination of the sagittal motion algorithm and the frontal motion algorithm has been shown successful for achieving the stable 3D walking motion. The main strategy of the robot locomotion control presented in this thesis is to guide the actual walking motion to converge to a steady-state using low control gains based on a generated basic walking pattern. This is different from the typical motion control strategies: tracking an ”optimal” walking pattern precisely as much as possible. The advantage of this motion control strategy has been found as it can achieve more natural walking dynamics as well as the motion stability. Since the motion control is rather soft, the resulting motion appeared to be smooth and not very sensitive to the environment perturbations. Through the successful applications of walking in different situations, the derived objective functions for the generation of a basic walking pattern that allows low gain motion control appear to be robust. Furthermore, using the proposed joint control scheme, the robot body can be easily maintained at the desired upright position and thus the prescribed walking posture can be achieved well.

Besides the achievement of motion stability, motion versatility for walking on different terrains or walking under external force perturbations has been achieved to a good extent. Through the adjustment of the key parameters defined in the TFS motion formulation, the step-length and the stride-frequency can be easily adjusted and then result in a wide range of adjustment of walking pace in response to the perturbations. Also, the regressed functions for the constant component in the TFS motion formulation successfully guided the robot to walk on the undulating terrains compliantly.

Furthermore, human-like robot walking has been achieved to a good extent. The human-like behavior can be observed from the following aspects: 1) The human-like ground reaction force pattern and knee motions on terrains of different slopes; 2) The achieved sagittal steady-state walking with a smooth ”U” shape velocity pattern as that measured in human gait analysis; 3)The imitation of the CPG function to some extent. (The low level control can make itself converge to the steady-state walking under some range of perturbations automatically. The high level supervision is only required when the change of environment is rather obvious); 4)Two-level motion control (analogous to the relation between brain and spinal cord).

In addition, in the 3D walking space, reinforcement learning has also shown its potential applications for motion balance control. Two strategies of motion balance control have been successfully implemented on 3D long distance walking control. One is mainly based on an active motion control using the spring and damper components concurrently. Through the adjustment of c defined in the TFS formulation for the lateral motion balance, the motion energy can be well compensated at each touch-down moment. However, there is certain motion jerkiness incorporated in the achieved motion. It is because in this case as discussed in Chapter 6, the joint control gains for the 3D walking have to be rather stiff.

The second strategy completely based on the damping component not only successfully maintained the frontal motion energy through the online adjustment of the parameter ϕ formulated in the TFS, but also achieved motion smoothness in both the sagittal and frontal motion planes. The resulting forward walking velocity profile appeared to be graceful in a ”U” shape, and the lateral motion was naturally confined to be a sinusoidal-like profile.

Last but not least, the proposed motion generation method - GAOFSF method and the soft motion control strategy are shown to be effective and general for robots of different mechanical properties, as tested on two different bipedal robots, NUSBIP-I and HOAP-I in this thesis.

8.1 Future Work

Based on the work done in this thesis, the followings are some suggestions for the future research work.

Foot Placement Strategies: Currently in this thesis, for the sagittal plane motion, the stride-frequency, step-length and walking pattern adjustment modes have been developed. Some demonstrations for real-time adjustment of walking pace and postures in response to the perturbations have been demonstrated using reinforcement learning. Nevertheless, it will be also interesting to have a robust prediction for where and when the swing foot should land, i.e. what are the appropriate values for the step-length and stride-frequency parameters in the TFS motion model, to maintain stable and to continue with stable locomotion based on some nonlinear dynamics analysis. This is because learning based computation method is still tedious in implementations.

In addition, the foot placement strategies can be further extended in both the sagittal and frontal Planes concurrently. Currently in this thesis, the swing leg is always controlled to be parallel to the stance leg during walking. This definitely restricted the motion balancing behavior in response to the perturbations occur in the lateral direction. Therefore, strategies for giving suitable foot placement in the frontal motion plane will also be needed.

Strategies to Allow the Robot to Change Direction during Locomotion: In this thesis, there is no consideration for walking while changing motion directions. If the walking directions are changeable, the robot will be more flexible with the environment. This is also especially good for applications such as rescuing in a hazard environment.

Learning Algorithms: As discussed in the motion balance control strategies, the common assumption is made as assuming the cross coupling behavior is minor under a fixed-speed motion pattern. This is mainly to reduce the size of the state in the reinforcement learning algorithm. However, the disadvantage is the motion identification could be less accurate and the variable speed walking may need more CMAC networks to separately work for each walking pattern. Therefore, learning algorithms is needed to be further explored and compared for a better tolerance solving the larger state-space problems. Then, the learning agent will become more powerful in the sense of human brain's function imitation for walking adaptively in response to the environment changes.

Bibliography

[1] MIT Leg Laboratory 2000. Yobotics, Inc. Online introduction: http://yobotics.com/.

[2] M. Vukobratovic, B. Borovac and V. Potkonjak. ZMP: A Review of some basic misunderstandings, *International Journal of Humanoid Robotics (IJHR)*, 3(2), 153-175, 2006.

[3] J. S. Bay and H. Hemami. Modelling of a neural pattern generator with coupled nonlinear oscillators. *IEEE Transactions on Biomedical Engineering*, BME-34(4): 297-306, 1987.

[4] S. Kagami, A Fast Dynamically Equilibrated Walking Trajectory Generation Method of Humanoid Robot, *Autonomous Robots* 12 (2002) 71-82.

[5] J. A. Golden and Y. F. Zheng. Gait synthesis for the sd-2 biped robot to climb stairs. *International Journal of Robotics and Automation*, 5(4): 149-159, 1990.

[6] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. *IEEE International Conference on Robotics and Automation*, 1998.

[7] Y. Hurmuzlu. Dynamics of bipedal gait: Part i - objective functions and the contact event of a planar five-link biped. *Journal of Applied Mechanics*, 60: 331-336, 1993.

[8] Y. Hurmuzlu. Dynamics of bipedal gait: Part ii - stability analysis of a planar five-link biped. *Journal of Applied Mechanics*, 60: 337-343, 1993.

[9] S. Aoi and K. Tsuchiya, Stability analysis of a simple walking model driven by an oscillator with a phase reset using sensory feedback, *IEEE Transactions on robotics*, Vol.22(2), 391-397, 2006.

[10] K. Harada, S. Kajita, K. Kanekoan and H. Hirukawa, Analytical Method for Real-time gait planning for humanoid robots, *International Journal of Humanoid Robotics (IJHR)*, 3(1), 1-19, 2006.

[11] T. Mita, T. Yamaguchi, T. Kashiwase, and T. Kawase. Realization of a high speed biped using modern control theory. *International Journal of Control*, 40(1): 107-119, 1984.

[12] R. Katoh and M. Mori. Control method of biped locomotion giving asymptotic stability of trajectory. *Automatica*, 20(4): 405-414, 1984.

[13] Jerry E. Pratt and Gill A. Pratt. Exploiting natural dynamics in the control of a planar bipedal walking robot. *Proceedings of the Thirty-Sixth Annual Allerton Conference on Communication, Control and COmputing*, 739-748, 1998.

[14] Rowel O. Atienza and Marceloh. Ang Jr. A Flexible Control Architecture for Mobile Robots: An Application for a Walking Robot. *Journal of Intelligent and Robotic Systems*, Vol 30, 29-48, 2001.

[15] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal and M. Kawato. Learning from demonstration and adaptation of biped locomotion *Robotics and Autonomous Systems*, Vol. 47, 79-91, 2004.

[16] Y. Fukuoka, H. Kimura, A. H. Cohen. Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts. *The International Journal of Robotics Research*, Vol.22, No. 3-4, 187-202, 2003.

[17] A. Khoukhi, Neural based RSPN multi-agent strategy for biped motion control. *Robotica*, Vol. 19, 611-617, 2001.

[18] Y. Ogura, et al. A Novel Method of Biped Walking Pattern Generation with Predetermined Knee Joint Motion. *Proc. of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[19] P. Sardain and G. Bessonnet. Forces Acting on a Biped Robot. Center of Pressure-Zero Moment Point. *IEEE Trans. on Systems, Man and Cybernetics-part A:systems and humans,* Vol. 34, No.5, 2004.

[20] A. Goswami, Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point, *The International Journal of Robotics Research*, Vol. 18, No.6, 523-533, 1999.

[21] A. Takanish, H. Lim, and M. Tsuda, Realization of Dynamic Biped Walking Stabilized by trunk motion on a saggitally uneven surface, in *Proc. of the IEEE International Workshop on Intelligent Robots and Systems,* 323-330, 2000.

[22] Hun-ok Lim and A. Takanishi. Compensatory motion control for a biped walking robot. *Robotica,* Vol.23, 1-11, 2005.

[23] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. *IEEE International Conference on Robotics and Automation*, 1998.

[24] J. H. Park, Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy Sets and Systems,* Vol.134, 189-203, 2003.

[25] H. K. Lum, M. Zribi, Y.C. Soh. Planning and control of a biped robot. *International Journal of Engineering Science*, Vol.37, 1319 - 1349, 1999.

[26] S. Kajita, O. Matsumoto and M. Saigo, Real-time 3D walking pattern generation for a biped robot with telescopic legs, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea. May 21-26, 2001.

[27] Mark W. Spong. The swing up control problem for the acrobot. *IEEE Control Magazine*, 49-55, 1995.

[28] M. S. Dutra, A. C. de P. Filho and V. F. Romano, Modeling of a bipedal locomotor using coupled nonlinear oscillators of Van der Pol. *Biological Cybernetics*, Vol 88(4), 286-292, 2004.

[29] S. Kotosaka and S. Schaal. Synchronized robot drumming by neural oscillator, *Proc. of the International Symposium on Adaptive Motion of Animals and Machines*, 2000.

[30] T. Zielinska, Biological Inspirations in Robotics: Motion Planning, *Proc. of 4th Asian Conf. on Industrial Automation and Robotics*, Bangkok, Thailand, 2005.

[31] Matthew M. Williamson, Neural control of rhythmic arm movements, *Neural Networks*, Vol. 11, 1379-1394. 1998.

[32] T. Zielinska. Coupled Oscillators Utilised as Gait Rhythm Generators of Two Legged Walking Machine. *Biological Cybernetics*, Vol. 4 No. 3 263C273, 1996.

[33] G. Taga. A model of the neuro-musculo-skeletal system for human locomotion: I. emergence of basic gait. *Biological Cybernetics*, 73: 97-111, 1995.

[34] S. Grillner. Locomotion in vertebrates: Central mechanisms and reflex interaction. *Physiological Reviews*, (55):247-304, 1975.

[35] K. Matsuoka. Mechanism of frequency and pattern control in the Neural Rhythm Generators. *Biological Cybenetics*, 56:345-353, 1987.

[36] H. Kimura and Y. Fukuoka. Biologically Inspired Adaptive Dynamic Walking in Outdoor Environment Using a Self-contained Quadruped Robot: 'Tekken2'. *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1, 986 - 991, 2004.

[37] T. Zielinska. Biological Aspects of Locomotion. In Walking: Biological and Technological Aspects, *CISM Courses and Lectures n. 467. Ed. by F.Pfeiffer, T.Zielinska. Springer Verlag*, pp.1-30, 2004.

[38] T. Zielinska , C. M. Chew. Biologically Inspired Motion Planning in Robotics. *Robot Motion and Control, Lecture Notes in Control and Information Sciences no.335, Ed. Kozlowski K. Springer*, Chapter 13, pp.201-219, 2006, ISBN-13-9781846284045.

[39] M. Garcia, A. Chatterjee, and A. Ruina. Speed, efficiency, and stability of small-slope 2d passive dynamic bipedal walking. *IEEE International Conference on Robotics and Automation*, 2351-2356, 1998.

[40] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2): 62-82, 1990.

[41] Tad McGeer. Passive walking with knees. *Proceedings of the IEEE Conference on Robotics and Automation*, 1640-1645, 1990.

[42] A. Goswami, B. Thuilot and B. Espiau. Compass-like biped robot, Part 1: Stability and bifurcation of passive gaits. *INRIA Research Report*, No.2996, 1996.

[43] E. Borzova and Y. Hurmuzlu. Brief paper: Passively walking five-link robot. *Automatica*, Vol.40, 621-629, 2004.

[44] M. Garcia, A. Chatterjee, A. Ruina and M. Coleman. The simplest walking model: Stability, complexity and scaling. *ASME Journal of Biomechanical Engineering*, 120(2), 281-288. 1998.

[45] M. Coleman, M. Garcia, A. Ruina, J. Camp and A. Chatterjee. Stability and chaos in passive dynamic locomotion. *Proceedings of the IUTAM symposium on new applications of nonlinear dynamics and chaos in mechanics*, Cornell University, Ithaca, NY, 1997.

[46] H. Benbrahim and J. A. Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22: 283-302, 1997.

[47] R. Tedrake, T. Zhang and H. S. Seung. Stochastic Policy Gradient Reinforcement Learning on a Simple 3D Biped. *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2849 - 2854. 2004.

[48] C. M. Chew and G. A. Pratt, dynamic bipedal walking assisted by learning, *Robotica*, Vol.20, 477-491, 2002.

[49] C. M. Chew, and G. A. Pratt, Frontal plane algorithms for dynamic bipedal walking, *Robotica*, Vol. 22, 29-39, 2004.

[50] J. Pratt, P. Dilworth, and G. Pratt. Virtual model control of a bipedal walking robot. *IEEE International Conference on Robotics and Automation*, 193-198, 1997.

[51] Jerry E. Pratt. Virtual model control of a biped walking robot. *Master's thesis, Massachusetts Institute of Technology*, August 1995.

[52] Marc. H. Raibert. *Legged Robots That Balance*, MIET Press, Cambridge, MA., 1986.

[53] L. Yang, C. M. Chew, T. Zielinska, and A. N. Poo. A Uniform Biped Gait Generator with Off-line Optimization and On-line Adjustable Parameters. *Robotica* Vol. 25, No. 5, 549-565, 2007.

[54] L. Yang, C. M. Chew, A. N. Poo and T. Zielinska Adjustable Bipedal Gait Generation using Genetic Algorithm Optimized Fourier Series Formulation, *IEEE/RSJ Int.l Conf. on Intelligent Robots and Systems*, 4435 - 4440, 2006.

[55] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279-292, 1992.

[56] J. S. Albus. Brain, Behaivor and Robotics, (BYTE Books. McGraw-Hill, Peterborough, NH, 1981). Chapter 6, pp. 139-179.

[57] L. Yang, C. M. Chew, A. N. Poo and T. Zielinska. Autonomous Stride-Frequency and Step-Length Adjustment for Bipedal Walking Control. *Studies in Computational Intelligence, Autonomous Robots and Agents*, Spinger, Vol. 76, 189-198. 2007.

[58] L. Yang, C. M. Chew, A. N. Poo. Autonomous Bipedal Walking Pace Supervision under Perturbations. *IEEE Int. Conf. on Systems, Man and Cybernetics*. Montreal, Canada, 765 -770, Oct. 2007.

[59] L. Yang, C. M. Chew, A. N. Poo. Real-time Bipedal Walking Adjustment Modes using Truncated Fourier Series Formulation. *IEEE-RAS Int. Conf. on Humanoid Robots*, Pittsburgh, Pennsylvania, USA. Dec. 2007.

[60] L. Yang, C. M. Chew, A. N. Poo and T. Zielinska. Real-time Bipedal Walking Gait Adjustment Modes Based on a Truncated Fourier Series Model. To be appeared in *International Journal of Humanoid Robotics*.

[61] L. Yang, C. M. Chew, Y. Zheng and A. N. Poo. Truncated Fourier Series Formulation for Bipedal Walking Balance Control. To be appeared in *Robotica*.

[62] Z. Michalewicz. Genetic Algorithms+Data Structures = Evolution Programs. AI Series. *Springer-Verlag*, 1994.

[63] R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. *MIT Press*, Cambridge, MA, 1998.

[64] L. P. Kaelbling, M. L. Littman, and A. W. Moore. *Reinforcement learning, A survey Journal of Artificial Intelligence Research*, (4): 237-285, 1996.

[65] D. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. *Addison Wesley*, 1989.

[66] L. Davis. The Handbook of Genetic Algorithms. *Van Nostrand Reingold*, New York, 1991.

[67] J. Joines and C. Houck. On the use of non-stationary penalty functions to solve constrained optimization problems with genetic algorithms. *IEEE International Symposium Evolutionary Computation*, Orlando, 579-584, 1994.

[68] D. Bertsekas and J. Tsitsiklis. Neuro-Dynamic Programming. Athena Scientific, Belmont, MA, 1996.

[69] D. P. Bertsekas. Dynamic Programming and Optimal Control: Volume One. Athena Scientific, Belmont, MA, 1995.

[70] T. Jaakkola, S. P. Singh, and M. I. Jordan. On the convergence of stochastic iterative dynamics programming algorithms. *Neural Computation*, 6:1185-1201, 1994.

[71] John N. Tsitsiklis. Asynchoronous stochastic approximation and q-learning. Machine Learning, 16(3): 185-202, 1994.

[72] G. A. Rummery. Probelm Solving with Reinforcement Learning. PhD dissertation, Cambridge University, UK, 1995.

[73] B. Widrow and M. Hoff. Adaptive swing circuits. *1960 IRE WESCON Convention Record*, 96-104, 1960.

[74] H. K. Lum, M. Zribi and Y. C. Soh. Planning and control of a biped robot. *International Journal of Engineering Science,* Vol. 37, 1319-1349, 1999.

[75] C. Chevallereau and Y. Aoustin. Optimal reference trajectories for walking and running of a biped robot, *Robotica,* Vol. 19, 557-569, 2001.

[76] J. H. Park. Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motion of biped robots, *Fuzzy Sets and Systems,* Vol. 134, 189-203. 2003.

[77] X. P. Mu and Q. Wu. Synthesis of a complete sagittal gait cycle for a five-link biped robot, *Robotica*, Vol. 21, 581-587, 2003.

[78] Q. Huang, K. Yokoi, S.Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie. Planning Walking Patterns for a Biped Robot, *IEEE transactions on robotics and automation*, Vol.17, No. 3, 280-289, 2001.

[79] K. Harada, S. Kajita, K. Kanekoan, and H. Hirukawa. Analytical Method for Real-time gait planning for humanoid robots. *International Journal of Humanoid Robotics (IJHR),* Vol: 3 Issue: 1, 1 -19, 2006.

[80] Jerry E. Pratt and Gill A. Pratt. Exploiting natural dynamics in the control of a planar bipedal walking robot. *Proceedings of the Thirty-Sixth Annual Allerton Conference on Communication, Control and COmputing*, 739-748, 1998.

[81] R. E. Goddard Jr. and H. Hemami and F.C. Weimer. Biped side step in the frontal plane. *IEEE Transactions on Automatic Control*, Vol.28, No. 2, 179-186, 1983.

[82] H. Hemami and B. F. Wyman. Modeling and control of constrained dynamic systems with application to biped locomotion in the frontal plane, *IEEE Transactions on Automatic Control*, Vol.24, No. 4, 526-535, 1979.

[83] K. Iqbal, H. Hemami and S. Simon. Stability and control of a frontal four-link biped system. *IEEE Transactions on Biomedical Engineering*, BME-40, No. 10, 1007-1017, 1993.

[84] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. Lim, and A. Takanish, Development of a New Humanoid Robot to Realize Various Walking Pattern Using Waist Motions. *RoManSy*, 2006.

[85] T. Nagasaki, S. Kajita, K. Yokoi, K. Kaneko and K. Tanie, Running Pattern Generation and Its Evaluation Using a Realistic Humanoid Model, *Proc. of the 2003 IEEE International Conference on Robotics and Automation*, Taipet, Taiwan, 14-19, 2003.

[86] ”Biomechanics of motion”, CISM COURSES AND LECTURES, International centre for mechanical science. No. 263, 79-129. 1980.

[87] ”Fourier Series”, Wolfram MathWorld, http://mathworld.wolfram.com/FourierSeries.html.

Author’s Publications

Journal Papers

- L. Yang, C. M. Chew, Y. Zheng and A. N. Poo, ”Truncated Fourier Series Formulation for Bipedal Walking Balance Control”, *Robotica*, accepted for publication.

- L. Yang, C. M. Chew, A. N. Poo and T. Zielinska, ”Real-time Bipedal Walking Gait Adjustment Modes Based on a Truncated Fourier Series Model”, to be appeared in *International Journal of Humanoid Robotics*, accepted for publication.

- L. Yang, C. M. Chew, T. Zielinska and A. N. Poo, ”A Uniform Biped Gait Generator With Offline Optimization and Online Adjustable Parameters”, *Robotica*, Vol.25(05), pp549-565. 2007.

Book Chapters

- L. Yang, C. W. de Silva, A. N. Poo and C. M. Chew ”Kinematic Design Optimization of Acrobot”, *Mechatronic Systems : Devices, Design, Control, Operation and Monitoring.* CRC Press, ISBN-10: 0849307759, 2007.

- L. Yang, C. M. Chew, A. N. Poo and T. Zielinska, ”Autonomous Stride-Frequency and Step-Length Adjustment for Bipedal Walking Control”, *Studies in Computational Intelligence, Autonomous Robots and Agents*, Spinger, Vol. 76, pp. 189-198. 2007.

Conference Papers

- L. Yang, C. M. Chew, A. N. Poo and T. Zielinska ”Adjustable Bipedal Gait Generation using Genetic Algorithm Optimized Fourier Series Formulation”, *IEEE/RSJ Int.l Conf. on Intelligent Robots and Systems*. 2006, pp.4435 - 4440.

- L. Yang, C. M. Chew, T. Zielinska and A. Neow Poo, ”Reliable and Adjustable Biped Gait Generation for Slopes using a GA Optimized Fourier Series Formulation”, *Romansy 16: Robot design Dynamics and Control*, Springer, 2006, pp.187-194.

- L. Yang, C. M. Chew and A. N. Poo, ”Autonomous Bipedal Walking Pace Supervision under Perturbations”, *IEEE Int. Conf. on Systems, Man and Cybernetics*. 2007, pp.765 -770.

- L. Yang, C. M. Chew and A. N. Poo, ”Real-time Bipedal Walking Adjustment Modes using Truncated Fourier Series Formulation”, *IEEE-RAS Int. Conf. on Humanoid Robots*, 2007, pp.379 - 384.